

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

**ЭЛЕКТРОННЫЙ СБОРНИК  
ТРУДОВ МОЛОДЫХ СПЕЦИАЛИСТОВ  
ПОЛОЦКОГО ГОСУДАРСТВЕННОГО  
УНИВЕРСИТЕТА**

*Выпуск 30 (100)*

**ПРОМЫШЛЕННОСТЬ**

Информационные технологии

Машиностроение

Химические технологии

*Обновляется 1 раз в год*

Новополоцк

ПГУ

2019

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

*Главный редактор:*  
проф., д.т.н. Д.Н. Лазовский

*Заместитель главного редактора:*  
доц., к.т.н. Ю.П. Голубев

*Председатель ред. совета* проф., д.т.н. В.П. Иванов  
*Зам председателя ред. совета* проф., д.т.н. Н.Н. Попок

### *Информационные технологии:*

доц., к.т.н. В.Ф. Янушкевич  
доц., к.т.н. Р.П. Богуш  
доц., к.ф.-м.н. С.А. Вабищевич  
доц., к.ф.-м.н. А.А. Козлов

### *Машиностроение:*

доц., к.т.н. А.В. Дудан  
доц., к.т.н. Т.В. Вигерина  
доц., к.т.н. А.М. Долгих  
доц., к.т.н. В.Э. Завистовский

### *Химическая техника и технология:*

доц., к.т.н. А.В. Митинов  
доц., х.т.н. С.Ф. Якубовский  
доц., к.пед.н. И.В. Бурая  
доц., к.т.н. А.А. Ермак

**ЭЛЕКТРОННЫЙ СБОРНИК ТРУДОВ МОЛОДЫХ СПЕЦИАЛИСТОВ ПОЛОЦКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА. Выпуск 30 (100). Промышленность. – Новополоцк : ПГУ, 2019.**

*Издается с 2014 года (в печатном виде – с 2004 г.)*

*Сборник включен в Государственный регистр информационного ресурса. Регистрационное свидетельство № 3201404563 от 30.11.2014 г.*

*В сборнике представлены труды молодых специалистов Полоцкого государственного университета по направлениям прикладных исследований, проводимых в университете: информационные технологии, машиностроение и материаловедение, приборостроение, химическая технология, охрана труда.*

*Предназначен для работников науки, образования и производства, будет полезен студентам, магистрантам и аспирантам университетов.*

Учредитель: учреждение образования «Полоцкий государственный университет»  
211440, ул. Блохина, 29, г. Новополоцк, Беларусь  
тел. 8 (0214) 59 95 44, e-mail: [vestnik@psu.by](mailto:vestnik@psu.by)

Компьютерный дизайн *М.А. Мухоморовой*

Техническое редактирование и компьютерная верстка *С.А. Щиков, С.Е. Рясова, Т.А. Дарьянова*

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.223

### ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ИГРОВОГО ПРИЛОЖЕНИЯ «CUBE» ПОД УПРАВЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ IOS

**И.В. ЛАРИОНОВ**

*(Представлено: Ю.Н. КРАВЧЕНКО)*

*В данной статье рассматриваются способы хранения данных приложения, а также способы их защиты.*

**Введение.** Unity3D – кроссплатформенный игровой движок от компании Unity Technologies. Unity позволяет создавать приложения, работающие под более чем 20 различными операционными системами. Движок поддерживает множество популярных форматов моделей, звуков, материалов и текстур. Проект в Unity делится на сцены (уровни) — отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев, и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты — объекты, которые не имеют модели. Объекты, в свою очередь содержат наборы компонентов, с которыми и взаимодействуют скрипты. Все эти данные необходимо защищать от копирования, нелегального использования, пиратского распространения, профессионального анализа и взлома.

**Основной раздел.** При компиляции проекта Unity создается исполняемый файл игры, а в отдельной папке – данные игры (включая все игровые уровни и динамически подключаемые библиотеки).

**Способы хранения данных.** Информационные объекты, взаимосвязи между ними, а так же способы влияния их на систему должны быть представлены в конечном программном продукте с помощью модели данных. Отсюда следует, что модель данных - есть совокупность структур данных и операций по их обработке.

Разрабатываемое приложение будет включить большое количество заранее подготовленной для пользователя информации, и необходимо подобрать удобный и эффективный способ ее хранения и обработки.

Отметим тезисно основные категории данных, которые необходимо хранить для дальнейшей интерпретации игрой:

- информация об игровых уровнях;
- информация о ценах во внутри игровом магазине;
- заранее заданные настройки.

Для хранения приведенных выше данных подходят 2 варианта: хранение в формате XML и бинарная сериализация.

Идеальным вариантом будет не выбирать среди предложенных технологий, а использовать оба варианта, взяв лучшее из каждой и применив в идеально подходящей для текущей задачи ситуации.

Большим плюсом бинарной сериализации является небольшой размер выходных файлов. Данные очень компактно складываются в файлы и сжимаются самыми эффективными алгоритмами, но существует один недостаток: невозможность чтения данных без специального программного обеспечения.

Бинарная сериализация отлично подходит для хранения самых разнообразных настроек, цен и прочих конфигурационных параметров.

Способ хранения уровней требует прозрачности и простого доступа к данным. Данные должны с легкостью читаться при помощи выбранного языка программирования и при этом быть очевидными при простом просмотре документа. Именно поэтому для хранения игровых уровней было выбрано использование документов в формате XML.

**Защита данных.** Для Windows-сборок Unity компилирует и сохраняет исходный код всех игровых скриптов в директорию Managed. Находится код в библиотеках: Assembly-CSharp.dll, Assembly-CSharp-firstpass.dll и Assembly-UnityScript.dll. Для декомпиляции и просмотра managed-кода .NET библиотек существуют довольно удобные и при этом бесплатные утилиты: IISpy и dotPeek.

Данный подход особенно эффективен для наших целей: Unity очень скупко оптимизирует исходный код игровых скриптов, практически не изменяя его структуру, а также не скрывает названия переменных. Это позволяет с легкостью читать и понимать декомпилированный материал.

В таких случаях разработчикам приходится самим беспокоиться о безопасности своего кода. Для таких целей обычно используют обфускаторы.

Обфускация – это процесс, в результате которого код программы приобретает вид, трудный для анализа. Обфускация, осуществляется с целью защиты программного кода и алгоритмов, которые он реализует, от чужих глаз. Но в большинстве случаев обфускация имеет массу побочных эффектов. В особо неприятных обстоятельствах обфускатор может сделать программу совершенно непригодной к выполнению, в менее тяжёлых случаях в программе могут появиться новые ошибки. Поэтому обфускацию необходимо применять с максимальной осторожностью.

В AssetStore существует множество готовых решений, но большинство из них платные. Бесплатные версии, как правило, ограничены или имеют невысокую эффективность.

Большинство ресурсов Unity-проекта упаковываются в файлы проприетарного формата с расширениями .assets и .resources. Несмотря на закрытость форматов, существуют инструменты для распаковки таких файлов. Так, например, Unity Assets Explorer способен извлечь большинство текстур и шейдеров из игры. Полученные в результате текстуры будут иметь формат DDS, который можно открыть с помощью Windows Texture Viewer. Шейдеры извлекаются в уже скомпилированном виде и решений для их декомпиляции не существует. Тем не менее, это обстоятельство не мешает импортировать и использовать полученные шейдеры в другом Unity-проекте.

Трёхмерные модели в Unity-сборке расположены в различных ресурсах, а некоторые из них и вовсе могут генерироваться во время игры. Получить такие данные можно напрямую из памяти графического ускорителя. Когда игра запущена, вся информация о текстурах и моделях, видимых на экране, находится в памяти видеокарты. С помощью утилиты 3D Ripper DX можно извлечь всю эту информацию и сохранить в формате, понятном 3D-редакторам.

PlayerPrefs — это класс из стандартной библиотеки Unity, который позволяет сохранять данные в долговременную память устройства. Представляет собой пару ключ – значение. Он часто используется разработчиками для хранения различных настроек, достижений, прогресса игрока и другой информации о состоянии игры. На ОС Windows эти данные сохраняются в системном реестре. В других операционных системах данные сохраняются на устройстве в локальной папке приложения в специальном файле. В большинстве случаев к ним можно легко получить доступ и модифицировать с помощью текстового редактора. Например, в Windows достаточно использовать встроенную утилиту RegEdit чтобы модифицировать любые значения PlayerPrefs, изменяя тем самым конфигурацию и статус игры.

Простейшим способом противодействия является кодирование сохраняемых данных, например, base64. Данный способ не слишком эффективен, но может дать начальную защиту от просмотра. Проверить, изменились ли данные без ведома помогут хеш-функции: сравнив контрольные суммы хранимых данных, мы сможем убедиться, что никто и ничто, кроме нашего кода эти данные не изменяло.

Для более надёжной защиты необходимо использовать шифрование или различные комбинации из перечисленных методов.

Также можно реализовать свой формат сохранений. Благодаря Mono Unity поддерживает работу с файловой системой. Таким образом можно сериализовать все необходимые данные, применить надёжное шифрование и сохранить в надёжном месте.

**Заключение.** В данной статье были рассмотрены способы хранения данных для игрового приложения под управлением операционной системы iOS, а также способы их защиты.

К сожалению, существует не так уж много способов защитить игру от взлома. Будучи установленной на пользовательское устройство, она фактически раскрывает все текстуры, модели и исходный код. Если кто-то захочет декомпилировать приложение и украсть ресурсы — это лишь вопрос времени.

Невзирая на это, существуют действенные методы, которые позволят серьёзно усложнить жизнь злоумышленникам.

## ЛИТЕРАТУРА

1. Guide to Storage Encryption Technologies for End User Devices / Karen Scarfone, Murugiah Souppaya, Matt Sexton. – November 2007. – 40 p.
2. Базы данных и модели данных [Электронный ресурс]. Режим доступа: <http://edu.tltsu.ru>. Дата доступа: 22.09.2019.
3. Unity – Scripting API: PlayerPrefs [Электронный ресурс]. Режим доступа: <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>. Дата доступа: 22.09.2019.
4. A practical tutorial to hack (and protect) Unity games [Электронный ресурс]. Режим доступа: <https://www.alanzucconi.com/2015/09/02/a-practical-tutorial-to-hack-and-protect-unity-games/>. Дата доступа: 22.09.2019.

УДК 004.021

**РАЗРАБОТКА МОБИЛЬНОГО ИГРОВОГО ПРИЛОЖЕНИЯ «CUBE»  
ПОД УПРАВЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ IOS****И.В. ЛАРИОНОВ***(Представлено: Ю.Н. КРАВЧЕНКО)*

*В данной статье рассматриваются принципы построения графического интерфейса пользователя для мобильного игрового приложения «Cube» под управлением операционной системы iOS.*

**Введение.** Unity3D — кроссплатформенный игровой движок от компании Unity Technologies. Unity позволяет создавать приложения, работающие под более чем 20 различными операционными системами. Движок поддерживает множество популярных форматов моделей, звуков, материалов и текстур. Начиная с версии 4.6 была добавлена новая система интерфейса, позволяющая использовать встроенные средства движка для построения пользовательского интерфейса.

**Основной раздел.** В основе понимания новой концепции лежат три компонента: Canvas, Rect Transform и Event Trigger.

Canvas — является контейнером для всех элементов UI и определяет режим рендера. Таких контейнеров на сцене может быть более одного.

Rect Transform — этот компонент позволяет задать положение и размер игрового объекта, используя удобные визуальные средства управления. Он вводит понятия ширины и высоты, точки опоры, а также поддерживает якоря.

Задание якорей — простой и мощный способ контролировать положение и размер ваших элементов пользовательского интерфейса относительно их родителей. Это особенно удобно, когда вы изменяете размер родительских элементов. При изменении размера родителя наш элемент будет стараться сохранить равномерное расстояние к точкам якоря, заставляя его перемещаться или изменять размер вслед за родителем.

Точка опоры является точкой, вокруг которой производятся все преобразования. Другими словами, определяя положение элемента, вы определяете его точку опоры. Если вы станете вращать элемент пользовательского интерфейса, то он будет вращаться вокруг этой точки. Точка опоры обозначается нормализованными координатами. То есть, ее значение изменяется от 0 до 1 для ширины и высоты, где (0,0) — нижний левый угол, а (1,1) — правый верхний угол.

Event System — доработанная система событий, которая включает компоненты визуального управления вызовом событий. Отвечает за обработку ввода, рейкастинг и события в сцене Unity.

При помощи перечисленных инструментов стало возможно разрабатывать сложные интерфейсы на основе иерархии простых компонентов без особых сложностей. При этом интерфейс будет адаптивным и автоматически подстраиваться под различные разрешения экрана.

Интерфейс приложения сделан используя современную концепцию построения внешнего вида, так называемый, плоский дизайн. Основные принципы плоского стиля — простота, лаконичность и минимализм. Основные отличительные черты — это те вещи, на которых можно акцентировать внимание — яркие необычные цвета, нестандартная типографика, графические примитивы. Так достигается простота и универсальность интерфейса.

Основной принцип flat-дизайна — это создание двумерного изображения, которое уже само по себе является плоским. При этом нужно избавиться от всех элементов, которые могут добавлять хоть какую-то глубину изображению: градиенты, плавные переходы, тени, скосы, объем, текстуры и так далее. Все элементы рисунка имеют жесткие края и также не выделяются, и не имеют глубины относительно других элементов изображения.

Во flat-дизайне вы не встретите элементы, которые пытаются выглядеть реалистично (скевоморфизм), 3D-анимацию, реалистичные пиктограммы и т.д. Однако при этом сам дизайн пытается оставаться аналогичным традиционному, но теперь на передний план выходят надписи, кнопки и навигация.

На первый взгляд приложение с применением Flat дизайна может показаться чересчур простым, хотя на самом деле этому стилю присуща четкая иерархия элементов, удобное размещение всех инструментов взаимодействия пользователя, а, следовательно, имеет высокий показатель юзабилити.

Плоский дизайн по своей натуре прост, и он хорошо сочетается с минималистичным подходом. В общем дизайне нужно избегать слишком больших наворотов. Конечно, простых цветов и текста может быть недостаточно. Поэтому, чтобы добавить визуальных эффектов, нужно выбирать простые фотографии.

Для создания плавных анимаций в Unity можно использовать сторонние плагины, например, DOTween. DOTween — это быстрый, эффективный, объектно-ориентированный движок анимации, оптимизированный для C#.

При помощи DOTween можно перемещать, изменять размер, поворачивать каждый объект на сцене независимо от других, при этом имея возможность настроить каждый аспект анимации по отдельности. Обратные вызовы, петли, различные функции интерполяции значений и многие другие варианты анимации.

Кроме того, он чрезвычайно точный. Это означает, что 1000 циклов по 1 секунде будут играть ровно столько же, сколько один цикл на 1000 секунд. Логичный и простой в использовании API, созданный для повышения эффективности, интуитивности и простоты использования.

DOTween может анимировать каждое числовое значение, а также некоторые нечисловые.

Например, он может анимировать строки.

Плагин предоставляет полный контроль: Play, Pause, Rewind, Restart, Complete, Goto и множество других полезных методов для управления вашими анимациями. Анимации можно объединять в последовательности для создания сложных анимаций (которые не обязательно должны быть в последовательности, а также они могут перекрывать друг друга). Возможно изменение начальных / конечных значений и продолжительности воспроизведения движения в любой момент, даже во время воспроизведения.

DOTween построен с учетом расширяемой архитектуры, которая позволяет создавать собственные плагины для анимации в виде отдельных файлов. Поддерживаются вспомогательные виртуальные методы, чтобы делать такие вещи, как например вызов функции после определенной задержки.

Таким образом можно без труда настроить даже сложные последовательности анимаций без ущерба для производительности.

**Заключение.** В данной статье были рассмотрены основные принципы и способы построения графического интерфейса в Unity для мобильного игрового приложения «Cube» под управлением операционной системы iOS. Рассмотрены основные компоненты Unity UI system и способы построения анимации.

#### ЛИТЕРАТУРА

1. Unity – Scripting API: UI [Электронный ресурс]. Режим доступа: <https://docs.unity3d.com/Manual/UISystem.html>. Дата доступа: 24.09.2019.
2. Краткий обзор нового Unity UI с примерами организации интерфейса [Электронный ресурс]. Режим доступа: <https://habr.com/post/246605/>. Дата доступа: 24.09.2019.
3. Основные принципы плоского дизайна [Электронный ресурс]. Режим доступа: <http://x-site.by/info/flat-design>. Дата доступа: 24.09.2019.
4. DOTween [Электронный ресурс]. Режим доступа: <http://dotween.demigiant.com/index.php>. Дата доступа: 26.09.2019.

УДК 004.9

**ЭКОНОМИЧЕСКАЯ СОСТАВЛЯЮЩАЯ РАЗРАБОТКИ WEB-СЕРВИСА ДЛЯ ПОДАЧИ ОБЪЯВЛЕНИЙ, ОЦЕНКА ВЛИЯНИЯ ИНФОРМАЦИОННОЙ СРЕДЫ НА ЭКОНОМИКУ****В.С. ЛЕСНИЦКИЙ***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

В статье представлена информация об экономической части разработки web-приложения для подачи объявлений. Также представлена информация об основных стадиях разработки приложения, видах работ и их экономической составляющей.

Методология проектирования информационных систем описывает процесс создания и сопровождения программного обеспечения в виде жизненного цикла. Жизненный цикл разработки программного обеспечения – период времени, который начинается с момента принятия решения о необходимости разработки системы и заканчивается в момент ее полного выхода из употребления у всех пользователей. Программное обеспечение является конечным продуктом процесса разработки системы.

Традиционно выделяют следующие основные этапы жизненного цикла программного обеспечения:

- анализ требований и проектирование (постановка задачи, получение технических заданий, разработка спецификаций, разработка алгоритмов решения);
- реализация (кодирование спроектированной системы);
- тестирование и отладка (проверка программы на соответствие всем предъявляемым к ней требованиям, выявление дефектов);
- приемка заказчиком и доработка (заключительные приемо-сдаточные испытания, исправление выявленных ошибок).

На этапе анализа и проектирования, составление спецификации требований, предъявляемых к программному обеспечению, является одной из самых важных целей. Без этого невозможно переходить на следующие шаги. Компоненты должны быть описаны с такой точностью, которая позволила бы начать их реализацию. Из-за своей важности эта работа рассматривается как отдельная фаза и занимает около 20% отведенного на выполнение дипломной работы времени (2,5 недели).

На фазе реализации составляется код программного обеспечения, исходя из созданного на этапе проектирования программного проекта. Разделение на компоненты создает предпосылки по разделению труда, т.е. когда для кодирования системы привлекают к работе команду программистов. На его реализацию отводится 7 недель (58,33% отведенного времени).

Этап тестирования и отладки, по существу проверка достоверности системы, где главной целью является выяснение того, отвечает ли созданная система требованиям пользователей. Данный этап подразумевает под собой проведение ряда тестов, призванных выявить ошибки в реализации проекта. В случае если последние будут обнаружены, они должны быть исправлены. На проведение данного этапа отводится 12,5% времени, т.е. 1,5 недели.

На последнем этапе: этапе отправки приложения заказчику проводятся приёмочные испытания. По результатам данных испытаний заказчик может озвучить некоторые пожелания, внести незначительные корректировки, в соответствии с которыми система должна быть доработана. На проведении этого этапа отводится оставшаяся неделя рабочего времени.

Графическое отображение доли временных затрат для каждого из этапов разработки web-сервиса для подачи объявлений в общем объеме представлено на рисунке 1.

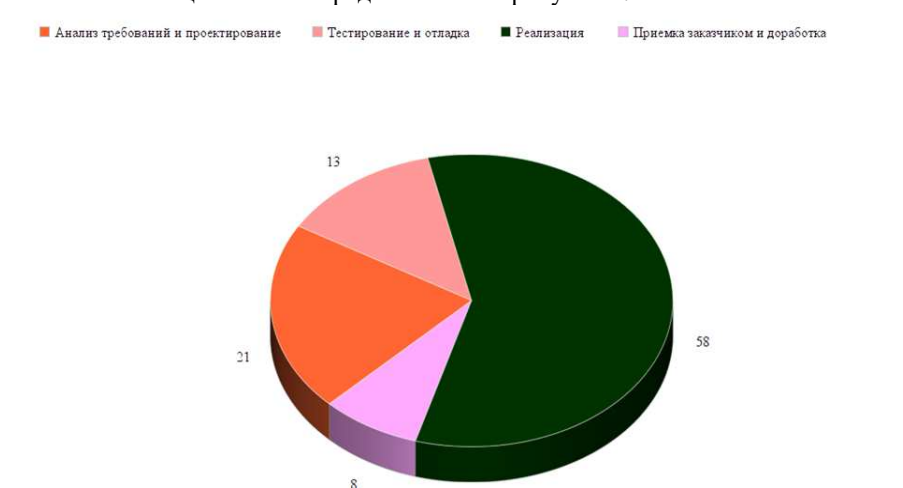


Рисунок 1. – Диаграмма распределения времени работы при разработке проекта

Виды работ. Кроме деления жизненного цикла на стадии и фазы, можно выделить восемь видов работ, которые выполняются в процессе создания программного продукта:

1. *Анализ требований и планирование.* Предусматривает разработку спецификаций, анализ и модификацию функциональных, технических, интерфейсных требований.

2. *Проектирование изделий.* Включает в себя определение, спецификацию, анализ и модификацию аппаратно-программной архитектуры проекта программы и базы данных.

3. *Программирование.* Детальное проектирование, кодирование, автономная отладка и комплексирование отдельных компонентов программы, а также планирование работ программистов, разработка базы данных, документирование отдельных компонентов и организация программирования на уровне компонентов.

4. Планирование отладки.

5. *Верификация.* Процесс проверки корректности требований, отладки изделий и приемные испытания.

6. *Управление проектом.* Планирование и контроль проекта, контроль и регулирование договоров, связь с пользователями.

7. *Контроль качества.* Разработка и контроль, стандартные и технические проверки программных средств и процессов разработки.

8. *Документирование.* Разработка и корректировка руководства для пользователя и операторов.

Деление жизненного цикла программных средств на стадии, фазы и этапы, а также более подробное описание работ, выполняемых в процессе их разработки, производства и эксплуатации, необходимо для детального определения затрат, требуемых на эти цели.

Расчет стоимости продукта. Оценка стоимости программного обеспечения и определение экономического эффекта у разработчика предполагает составление сметы затрат, которая, согласно Постановления Министерства экономики Республики Беларусь, Министерства финансов Республики Беларусь и Министерства труда и социальной защиты Республики Беларусь от 30.10.08 г. №210/161/151 «Об основных положениях по составу затрат, включаемых в себестоимость продукции (работ, услуг)», включает следующие статьи затрат:

1. *Электроэнергия* – так как продуктом является приложение, то при его разработке используются компьютеры, потребляющие электричество;

2. *Материалы и комплектующие* – расходные материалы, потраченные при создании продукта;

3. *Основная заработная плата исполнителей* – затраты на выдачу заработной платы сотрудникам, работающим над проектом;

4. *Дополнительная заработная плата* – затраты на оплату премий, рисков, отпусков и прочее;

5. *Отчисления на социальные нужды* – затраты на налоговые сборы, в фонд защиты населения, страхование, прочее;

6. *Амортизация* – затраты на непредвиденные обстоятельства;

7. *Расходы на спецоборудование* – затраты, связанные с приобретением специализированного оборудования;

8. *Прочие прямые расходы* – затраты на подготовку и приобретение различного рода информации.

Влияние статей затрат на окончательную себестоимость продукта можно увидеть на диаграмме, представленной на рисунке 2.

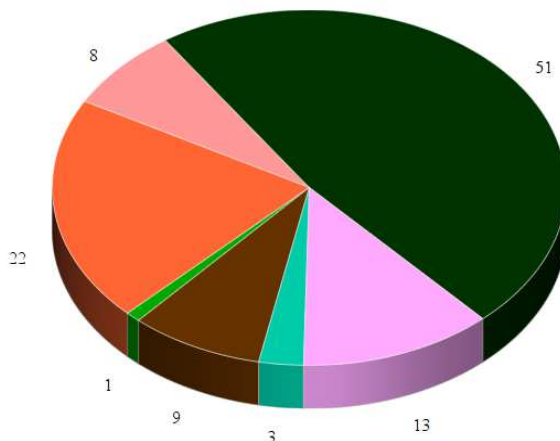
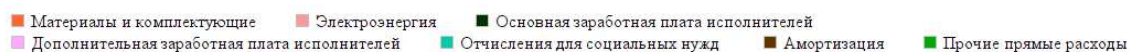


Рисунок 2 – Структура затрат на разработку обучающего приложения



Согласно представленной на рисунке 2 диаграммы, можно сделать вывод, что основные затраты на разрабатываемый продукт, вне зависимости от сроков разработки, приходятся на выдачу заработной платы сотрудникам.

Заказчик оплачивает разработчику всю сумму расходов по проекту, включая прибыль. После уплаты налогов из прибыли в распоряжении разработчика остается чистая прибыль от проекта. Ввиду того, что программное обеспечение разрабатывается для одного объекта, чистую прибыль можно считать в качестве экономического эффекта разработчика от реализованного программного обеспечения.

Отпускная цена продукции формируется исходя из плановой себестоимости производства продукции, всех видов установленных налогов и прибыли, а также качества, потребительских свойств продукции и конъюнктуры рынка.

**Заключение.** Таким образом затраты на электроэнергию составят 50,67 рублей, материалы и комплектующие – 32,6 рублей, основная заработная плата исполнителей – 3600 рублей, дополнительная заработная плата исполнителей – 360 рублей, отчисления на социальные нужды – 1370,16 рублей, амортизация – 39,68 рублей, прочие прямые расходы – 180 рублей, расходы на спец оборудование – 0 рублей. Таким образом себестоимость разработанного программного обеспечения поучилась 5633,11 рублей, а отпускная цена с учетом НДС 7773,69 рублей.

#### ЛИТЕРАТУРА

1. Habr.com [Электронный ресурс]. – Режим доступа: <https://www.habr.com>. – Дата доступа: 24.09.2019.
2. Richpro.ru – финансовый журнал [Электронный ресурс]. – Режим доступа: <http://richpro.ru>. – Дата доступа: 24.09.2019.

УДК 33; 339.1-051

**ВЫБОР СРЕДСТВ И МЕТОДОВ РЕШЕНИЯ ЗАДАЧ,  
ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ WEB-СЕРВИСА  
ДЛЯ ПОДАЧИ ОБЪЯВЛЕНИЙ****В.С. ЛЕСНИЦКИЙ***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

*В статье представлен объект разработки – web-сервис для подачи объявлений. Рассмотрены вопросы выбора и обоснования средств проектирования базы данных.*

**Выбор и обоснование средств и методов решения задач.**

Выбор системы управления базами данных (СУБД) представляет собой сложную многопараметрическую задачу и является одним из важных этапов при разработке клиент-серверных приложений. Определяющим здесь являются вид программного продукта и категории пользователей (или профессиональные программисты, или конечные пользователи, или и то, и другое).

Другими показателями, влияющими на выбор СУБД являются:

- удобство и простота использования;
- качество средств разработки, защиты и контроля базы данных;
- уровень коммуникационных средств в случае применения ее в сетях;
- фирма-разработчик;
- стоимость.

Языком программирования для реализации серверной части приложения был выбран PHP (Personal Home Page). Это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения web-разработок и его код может встраиваться непосредственно в HTML (HyperText Markup Language). Данный язык нацелен именно на web-разработку, следовательно в нашем случае выбор является обоснованным.

В качестве фреймворка для разработки серверной части выбран Laravel. Laravel – бесплатный web-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC. Laravel выпущен под лицензией MIT. Исходный код проекта размещается на GitHub. Он намного упрощает разработку приложения и несет в себе массу возможностей. Очень много современных компаний, таких как Яндекс, CarPrice, BelkaCar, LinkedIn используют его при создании собственных продуктов.

Для проектирования базы данных был выбран MySQL (версия 5.6) – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Для реализации приложения была выбрана среда разработки WebStorm – язык программирования JavaScript. Был выбран именно этот язык по причине следующих его достоинств:

- ни один современный браузер не обходится без поддержки JavaScript;
- с использованием написанных на JavaScript-плагинов и скриптов справится даже не специалист;
- полезные функциональные настройки;
- постоянно совершенствующийся язык – сейчас разрабатывается бета-вариация проекта;
- взаимодействие с приложением может осуществляться даже через текстовые редакторы – Microsoft Office и Open Office;
- перспектива использования языка в процессе обучения программированию и информатике.

Для реализации приложения была выбрана клиент-серверная архитектура. Основными преимуществами работы в клиент-серверном варианте являются:

- высокая производительность при обработке больших объемов данных информационной базы;
- удобство администрирования и упорядочивание доступа пользователей к информационной базе.

Учитывая выше сказанное, MySQL и JavaScript (с использованием среды разработки WebStorm и PHPStorm) подходят для реализации web-сервиса для подачи объявлений.

**Проектирование базы данных.** Главной целью проектирования базы данных является выбор оптимального количества таблиц для хранения данных, полей, которые должны войти в ту или иную таблицу, а также планирование отношений между таблицами. Эту и многие другие проблемы при проектировании баз данных можно решить с помощью нормализации [4].

*Нормализация* – процесс преобразования отношений базы данных к виду, отвечающему нормальным формам.

*Нормальная форма* – свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

При проектировании баз данных могут применяться различные подходы. Основная задача проектирования баз данных – это сокращение избыточности хранимых данных, а, следовательно, экономия используемых ресурсов (оперативной и дисковой памяти), уменьшение затрат на обновление избыточности копий данных, а также снижение вероятности нарушения целостности данных.

Реляционная база данных – это совокупность отношений, в которых хранится вся информация баз данных. Для пользователя такая база данных представляется набором двумерных таблиц, что облегчает понимание структуры данных и управления ими. Таблицы реляционной базы данных связаны между собой отношениями.

Схема базы данных web-сервиса для подачи объявлений представлена на рисунке.

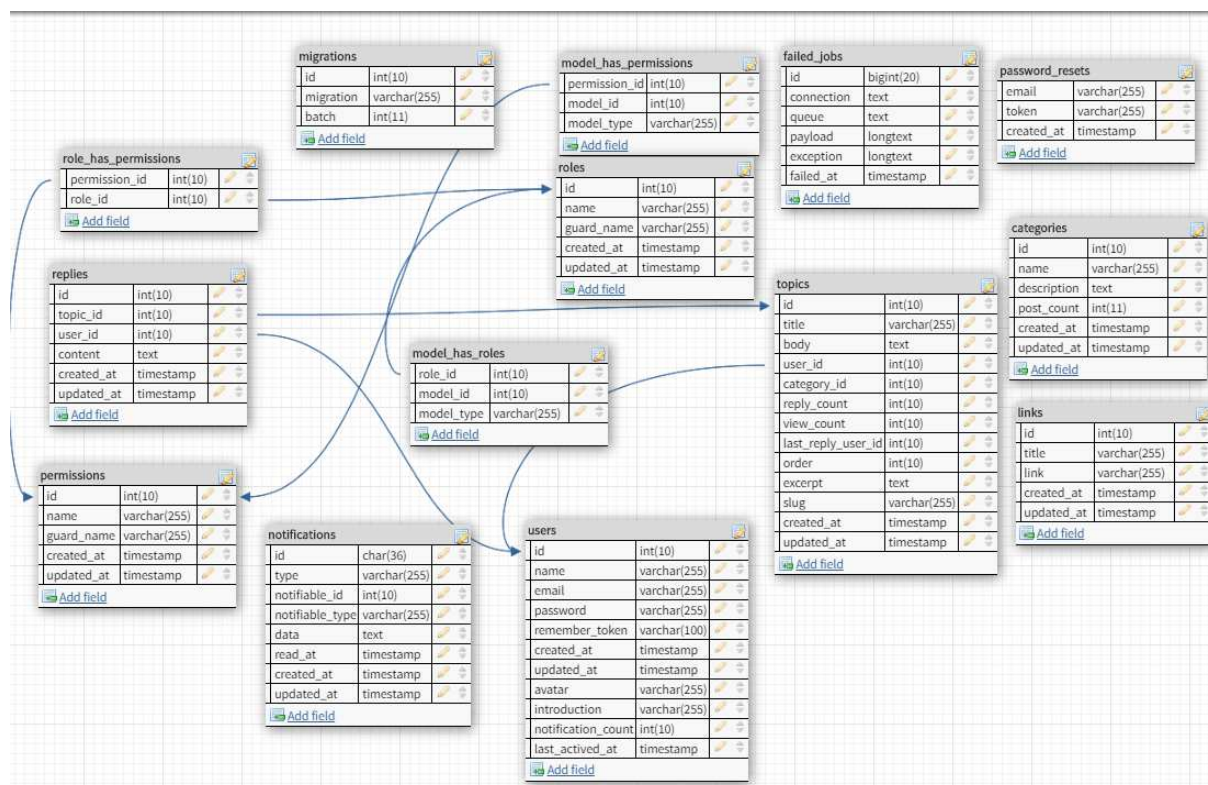


Рисунок. – Схема базы данных web-сервиса для подачи объявлений

Таким образом разработанная база данных позволяет быстро и эффективно работать с данными. Удобный интерфейс программы позволяет легко ориентироваться в программе и имеет следующие сущности:

- Пользователь;
- Категория;
- Тема;
- Ответы;
- Оповещение;
- Аватар;

- Последняя активность пользователя;
- Сообщение пользователю;
- Репост комментария;
- Проект;
- Администратор.

**Заключение.** В результате проделанной работы был выбрана среда разработки приложения и спроектирована база данных для реализации web-сервиса для подачи объявлений.

#### ЛИТЕРАТУРА

1. Habr.com [Электронный ресурс]. – Режим доступа: <https://www.habr.com>. – Дата доступа: 24.09.2019.
2. Малыгина, М. Базы данных: основы, проектирование, использование / М. Малыгина. – М.: ВНУ, 2004. – 512 с.
3. Лучшие JS редакторы [Электронный ресурс]. – Режим доступа: <http://uchitel-program.ru/web/redaktery/javascript>. – Дата доступа: 24.09.2019.
4. Нормализация [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Нормализация>. – Дата доступа: 24.09.2019.

УДК 004.021; 004.946(036); 364.042

## ЭЛЕКТРОННЫЙ ПУТЕВОДИТЕЛЬ ПО ГОРОДУ НОВОПОЛОЦКУ ДЛЯ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

А.Ю. СОЛОЦКИЙ

(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЁНОК)

В статье рассматривается разработка веб-ресурса с размещенной на нем картой и полезной информацией об объектах городской инфраструктуры для людей с ограниченными возможностями. Рассмотрена актуальность проблемы. Проводится анализ используемых для разработки технологий.

**Введение.** Люди с ограниченными возможностями нуждаются в постоянной поддержке других членов общества и государства. По Республике Беларусь на сегодняшний день насчитывается более полумиллиона инвалидов [1]. Современные информационные технологии могут облегчить жизнь таким людям, а повсеместное распространение портативных, весьма производительных, мобильных устройств и удешевление услуг связи и доступа к сети Интернет, для поддержания информации в актуальном состоянии, делают их доступными им.

В Новополоцке проживает более двухсот инвалидов-колясочников, многие из которых не ведут активный образ жизни в связи с отсутствием информации о беспрепятственном доступе к объектам социальной инфраструктуры города. Поэтому разработка электронного путеводителя по городу Новополоцку для людей с ограниченными возможностями является актуальной задачей.

**Основной раздел.** Проектирование программного продукта такого рода можно разделить на несколько этапов. Необходимо разработать подсистему хранения информации об объектах города, административный интерфейс для управления этой информацией, пользовательский интерфейс для конечного отображения данной информации конечным пользователям в удобном виде, а также промежуточную инфраструктуру, позволяющую взаимодействовать остальным компонентам системы.

В качестве базы данных была выбрана реляционная система управления базами данных (СУБД) MySQL 8.0. Данная СУБД отлично подходит для работы с не очень большими объемами данных, не имеющих сложную структуру (например, структура разработанной базы данных см. рисунок 1), предоставляя высокую производительность в сочетании с приемлемым потреблением ресурсов, достаточную надёжность и гибкость. Немаловажным фактом является её зрелость (разработка ведётся более 20 лет) и бесплатное распространение (распространяется по лицензии GNU Public License).

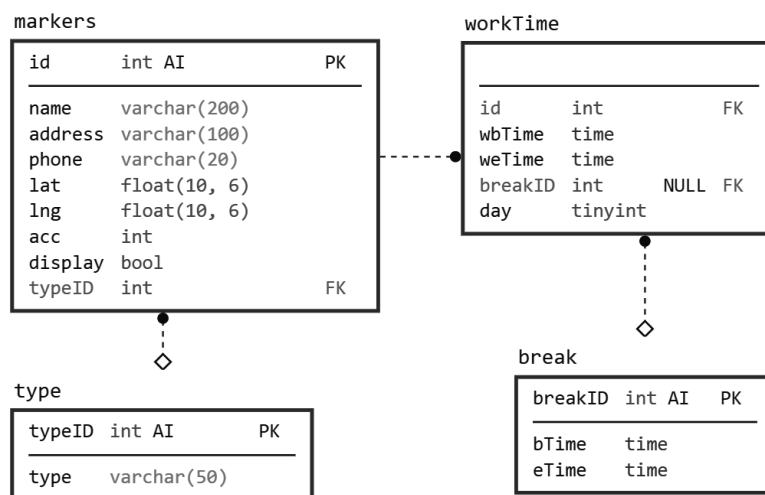


Рисунок 1. – Схема базы данных объектов

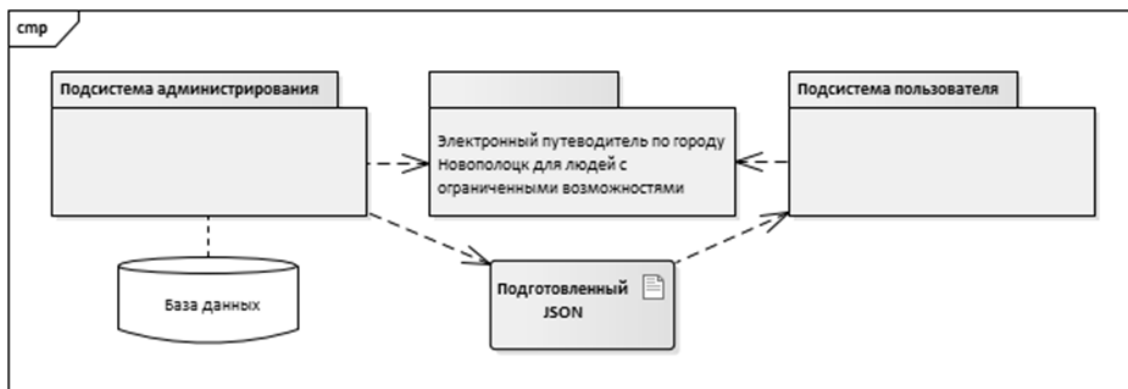
С СУБД MySQL отлично взаимодействует другая открытая технология – PHP (англ. PHP: Hypertext Preprocessor – PHP: препроцессор гипертекста). Данный язык был выбран для выполнения следующих функций: предоставление защищённого и безопасного интерфейса к для изменения информации в базе данных, генерации промежуточного представления пользовательских данных. PHP весьма практичен для быстрого и эффективного решения задач веб-разработки.

Для общей архитектуры приложения был выбран подход REST (сокращение от англ. Representational State Transfer – «передача состояния представления»). Данный стиль взаимодействия компонентов приложения предусматривает некоторые ограничения, накладываемые на систему, которые обеспечива-

ют повышение производительности, надёжности, прозрачности взаимодействий, упрощение архитектуры и обеспечение лёгкости внесения изменений и эволюции, а именно:

- клиент-серверная модель, в том числе отделение базы данных от клиента и самостоятельное определение клиентом представления данных, полученных от сервера;
- отсутствие состояния на сервере – все условия, необходимые для однозначного решения запроса клиента предоставляются им самим; кэширование, позволяющее во многих случаях не выполнять повторную обработку запроса, даже если он пришёл от другого клиента, и сразу вернуть результат;
- единообразие интерфейса, позволяющее улучшать внутреннюю реализацию компонентов независимо друг от друга; а также система слоёв взаимодействия клиента с сервером, что позволяет упростить кэширование и распределение запросов [2].

В разработанной системе можно выделить четыре компонента (рис. 2).



**Рисунок 2. – Функциональная структура электронного путеводителя по городу Новополоцку для людей с ограниченными возможностями**

Подсистема администрирования отвечает за операции над маркерами. Она и только она взаимодействует с базой данных, и она же генерирует промежуточный JSON-файл, доступный любому пользователю сети. В данном случае его можно рассматривать как своего рода кэш. Подсистема пользователя используя данные этого файла отображает маркеры на карте и информацию о них, позволяет фильтровать данные. При этом, после получения этой информации, нет никакой необходимости во взаимодействии с сервером, а после загрузки необходимой области на карте и вовсе в доступе к сети, и страница может работать полностью автономно, экономя трафик пользователя и ресурсы сервера.

В качестве веб-сервера можно использовать Nginx или Apache HTTP Server, однако второй предпочтительнее, так как не требует дополнительной прослойки для генерации динамического контента, а целевая аудитория разработанного приложения не настолько обширна, чтобы вызвать необходимость разделения серверной системы на независимые модули, предусмотренные первым программным обеспечением.

Таким образом, все упомянутые технологии широко распространены, проверены временем и хорошо изучены, что упрощает развёртывание системы на любом окружении. При необходимости адаптация или масштабирование системы могут быть проведены в короткие сроки.

Для разработки приложения использовалась среда JetBrains PhpStorm. Это обусловлено поддержкой всех используемых современных технологий: SQL, PHP, HTML, CSS, JavaScript для разработки пользовательской части; наличием мощных инструментов автогенерации, рефакторинга и автодополнения; глубокой интеграцией с системами управления версиями в одном инструменте.

**Заключение.** Разработанное приложение соответствует задачам, определяемым предметной областью, таблицы созданной базы данных отвечают требованиям нормализации, что позволяет обеспечить целостность и непротиворечивость хранимых данных. Для создания приложения были использованы современные языки программирования и среды разработки.

#### ЛИТЕРАТУРА

1. Численность инвалидов, получающих пенсию в органах по труду и социальной защите | Министерство труда и социальной защиты Республики Беларусь. Официальный сайт [Электронный ресурс]. – Режим доступа: <http://www.mintrud.gov.by/ru/chislinv>. – Дата доступа: 09.09.2019.
2. Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST) [Электронный ресурс]. – Режим доступа: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm). – Дата доступа: 11.09.2019.

УДК 004.021; 004.514; 004.55

**РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ВЕБ-САЙТА, С УЧЕТОМ ТРЕБОВАНИЙ ОТЗЫВЧИВОСТИ, АДАПТИВНОСТИ И ПОВЫШЕНИЯ ДОСТУПНОСТИ ДЛЯ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ****А.Ю. СОЛОЦКИЙ***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЁНОК)*

*В статье рассматриваются способы разработки адаптивной вёрстки веб-сайта с учетом использования его людьми с ограниченными возможностями.*

**Введение.** Развитие технологий привнесло множество изменений в обычный образ жизни человека. Люди ежедневно взаимодействуют с множеством устройств различных форм-факторов с помощью графического интерфейса. В некоторых случаях, например, при работе с веб-сайтами, возникает необходимость отображать одно и то же содержимое на разных устройствах комфортным для пользователя образом. Что включает в себя не только конфигурацию блоков на экране, но и подстройку размера текста, расположения и размера изображений и иногда цветовой схемы. Это отражается на желании заказчика получить результат в максимально сжатые сроки и с минимальными затратами, а также желании разработчика провести максимальную унификацию для упрощения поддержки, особенно в случае внутренних проектов. В связи с этим, адаптивная вёрстка является важным подходом, решающим сразу множество проблем.

**Основной раздел.** Адаптивный веб-дизайн – подход к разработке вёрстки страницы, при котором интерфейс корректно отображается на устройствах с различной ориентацией, размером и чёткостью экрана, размером окна браузера и, обычно, способен динамически подстраиваться в случае изменений.

Текущий подход к разработке адаптивных интерфейсов появился как результат взаимодействия нескольких стратегий разработки. Далее подробнее остановимся на указанных стратегиях.

– *Прогрессивное улучшение.* Стратегия предусматривает постепенное улучшение и расширение пользовательского интерфейса посредством всё более сложных и современных CSS-стилей и JavaScript сценариев для устройств, поддерживающих их, но продолжая предоставлять базовую функциональность менее современным клиентам. Эта стратегия была представлена Стивеном Чампионом на конференции SXSW в 2003 году.

– *Отзывчивый дизайн.* Подразумевает «резиновые» блоки, растягивающиеся изображения и широкое использование медиа-запросов из CSS. Впервые понятие было введено Итаном Маркоттом в 2010 году [1]. В дальнейшем Джеффри Зельдман предложил расширить данное понятие для любых техник, позволяющих достигнуть корректного отображения интерфейсов вне зависимости от характеристик экрана [2].

– *Mobile First.* Стратегия, предусматривающая изначально разработку мобильной версии ресурса с последующей адаптацией к другим устройствам. Эта стратегия описана Люком Вроблевски в 2011 году [3].

Также важно отметить Аарона Густавсона, который описал стратегию совместного использования описанных выше подходов [4].

Существуют различные методы адаптации. Возможно разделить устройства на подгруппы и разрабатывать вёрстку для каждой отдельно, с нуля. Однако, это очень усложняет разработку с единым стилем, внесение изменений и в принципе очень трудозатратно, а также негативно воспринимается поисковыми системами в случае размещения различных версий на разных поддоменах.

Реализация согласно изначальной задумке Итана Маркотта также имеет свои недостатки. Часто возникают сложности с изображениями и появляется свободное место в районе боковых панелей по мере прокрутки страницы.

Разработаны техники (например, Responsive-Images от filamentgroup или использование UserAgent для определения типа устройства) позволяющие загружать другие наборы ресурсов в зависимости от характеристик клиента, однако это нельзя рассматривать как самостоятельное решение.

Резиновая вёрстка предусматривает изменение расположения лент разметки (например, фреймворк Bootstrap). Перенос блоков использует изменение структуры макета при изменении ширины экрана. Скрытие и панели предусматривает возможность отображения только части интерфейса по умолчанию для некоторых устройств, но с сохранением к ним доступа, например, по нажатию кнопок.

Помимо уже упомянутого варианта с генерацией HTML-разметки на сервере с учётом пользователя, возможно использование дополнительных файлов CSS-стилей и медиа-запросов, а также JavaScript.

Медиа-запросы (@media), а также проверки поддержки (@supports), позволяют применять стили только для конкретных типов устройств с определёнными характеристиками и поддержкой этих стилей. Например, можно легко разработать версию для печати страницы или с некоторым соотношением сторон (рис.).



Рисунок. – Адаптивный интерфейс на мобильном устройстве и персональном компьютере

Возможно изменение и создание различных схем верстки с использованием таблиц, блоков и более современных гибких блоков (Flexbox) и сеток (Grid layout). Два последних подхода позволяют создавать довольно сложные одно- и двухмерные конструкции соответственно, с использованием широкого спектра ограничений и настроек разного типа одновременно.

JavaScript предоставляет полный контроль над страницей посредством объектной модели документа. Это позволяет подгружать дополнительные файлы стилей, применять стили к элементам напрямую, реализовать сложные реакции, не предусмотренные HTML и CSS по умолчанию, например, нажатия на объекты. С его помощью возможно реализовать также реактивность, например, на постепенный ввод или прокрутку страницы, различные связанные взаимодействия.

Важным параметром при реализации адаптивности, в том числе и для людей с ограниченными возможностями, является готовность страницы работать с нестандартным размером шрифта, который можно установить через настройки браузера. Для этого важно, чтобы подавляющее большинство элементов страницы было установлено не в абсолютных единицах (пикселях и их производных), а в относительных – процентах и процентах относительно окна, em и rem (размерах шрифта). Также, последние единицы позволяют легко централизованно менять размер некоторых внешних элементов прямо на странице, меняя размер шрифта корневого элемента.

Комбинация JavaScript и CSS позволяет реализовать разные цветовые схемы для людей с нарушениями зрения. Также, важно указывать атрибут alt у изображений, предоставляя текстовое описание для них. Это также может быть полезно для пользователей с низкой скоростью доступа или отключёнными изображениями. Описание должно быть осмысленным, чётким, ёмким, при наличии результата клика – описывать его. Важно использовать и верные HTML-теги, например, иерархию заголовков, так как программы для незрячих в значительной степени полагаются на них и ускоряют навигацию по странице.

Оптимизация верстки для мобильных устройств также включает предоставление корректных типов полей ввода, отключение для некоторых из них автокоррекции, изменение размеров активных элементов интерфейса и поддержка взаимодействия с ними посредством касаний, повсеместное использование автозаполнения, отказ от отслеживания наведения (замена на касания).

**Заключение.** В данной статье рассмотрены основные технологии и подходы создания адаптивных интерфейсов, в том числе подготовленных для использования людьми с ограниченными возможностями. Результаты данного исследования были использованы в разработке электронного путеводителя по городу Новополюцку для людей с ограниченными возможностями.

#### ЛИТЕРАТУРА

1. Responsive Web Design – A List Apart [Электронный ресурс]. – Режим доступа: <https://alistapart.com/article/responsive-web-design>. – Дата доступа: 02.09.2019.
2. Responsive Design. I don't think that word means what you think it means. | Zeldman on Web & Interaction Design [Электронный ресурс]. – Режим доступа: <http://www.zeldman.com/2011/07/06/responsive-design-i-dont-think-that-word-means-what-you-think-it-means>. – Дата доступа: 02.09.2019.
3. Вроблевски Л. Сначала мобильные! – М.: Манн, Иванов и Фербер, 2012. – 176 с.
4. Gustafson A. Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement. – Easy Readers, 2011. – 114 с.



УДК 004.5; 004.921

**ВЕБ-САЙТ «ВЫПУСКНИКИ» В РАБОТЕ УНИВЕРСИТЕТА****В.А. ЦЫБУЛЬСКИЙ***(Представлено: канд. техн. наук., доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассматривается создание веб-сайта для работы в университете на примере ресурса СПбГУ «Ассоциация выпускников СПбГУ».*

**Введение.** Сейчас век информационных технологий и большинство людей общаются друг с другом посредством интернета. Раньше, окончив университет, многие одноклассники старались поддерживать связь друг с другом, но контактные данные или адреса могут легко потеряться, после чего, общение может прекратиться. Люди общались при помощи писем, редко могли поговорить по телефону. В настоящее время, на смену письмам в конвертах, а также минутным звонкам пришли различные мессенджеры, онлайн-чаты, социальные сети, а также появилась возможность круглосуточно разговаривать по телефону с кем угодно.

Для выпускников необходимо создать сайт, на котором они смогут найти своих одноклассников, людей, с которыми дружили во время учебы в университете, связаться с ними по предоставленным данным на сайте, предложить собраться всем вместе, как в старые добрые времена.

По результатам поиска, было обнаружено, что у таких сайтов вовсе нет и это огромный минус, ведь многие, кто окончил университет, хотят найти своих друзей и хоть какие-то контактные данные. Именно это послужило причиной для создания ресурса.

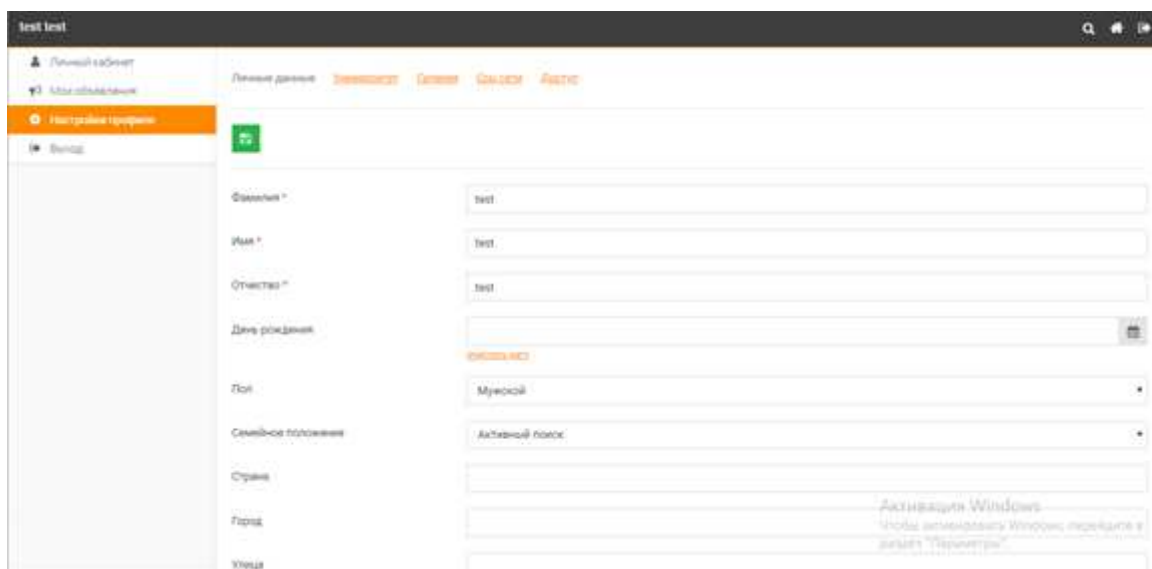
Далее рассмотрим основные функции предусмотренные в представленной реализации сайта:

1. Зарегистрироваться могут только выпускники университета.
2. Сайт является адаптивным.
3. Приятные дизайн.
4. Сайт является универсальным.
5. Используется база данных.
6. Личный кабинет у каждого выпускника.
7. Регистрация по номеру диплома.
8. Поиск выпускников университета.
9. Объявления выпускников

Почему сайт универсальный? На сайте имеется панель администратора, где администратор может настроить все так, как ему необходимо, таким образом, сайт можно внедрить в различные университеты, и даже в школы. Используя такой ресурс, никак не обойтись без базы данных. Так как проект делался на языке PHP (Hypertext Preprocessor), принято решение для организации хранения данных использовать систему управления базами данных MySQL 8.0. Она также является удобной, предусмотрена возможность конвертации данных и администратору нет необходимости ее заполнять вручную, ему достаточно только нажать кнопку, выбрать файл Excel и все информация о выпускниках университета будет загружена в базу данных и отображена на сайте. Но есть небольшой нюанс, чтобы данные правильно загрузились, необходимо их в правильном порядке внести в файл Excel. Кроме внесения информации через файл Excel, можно внести вручную по одной записи, а также отредактировать или удалить ее.

Если пользователю интересен какой-либо вопрос, он может в любой время обратиться в техническую поддержку, где ему помогут его решить.

Одной из главных особенностей данного сайта является наличие личного кабинета у каждого выпускника. Чтобы зайти в свой личный кабинет, пользователю необходимо зарегистрироваться. Для этого нужно ввести свое ФИО, почту и номер диплома. Если номер диплома в базе данных отсутствует, пользователю не удастся зарегистрироваться. Если регистрация прошла успешно, то, по указанному адресу email, выпускник может получить пароль для входа в свой личный кабинет. Также предусмотрена возможность смены пароля. В своем личном кабинете можно редактировать информацию о себе, менять фотографию, а также скрывать некоторые данные, от пользователей, которые не зарегистрированы, кроме перечисленного, выпускники могут указать свои социальные сети, где они зарегистрированы. При открытии личного кабинета, у пользователя будет отображаться две информации, одна из университета, а вторая заполненная им лично. Пример страницы редактирования личной информации показан на рисунке 1.

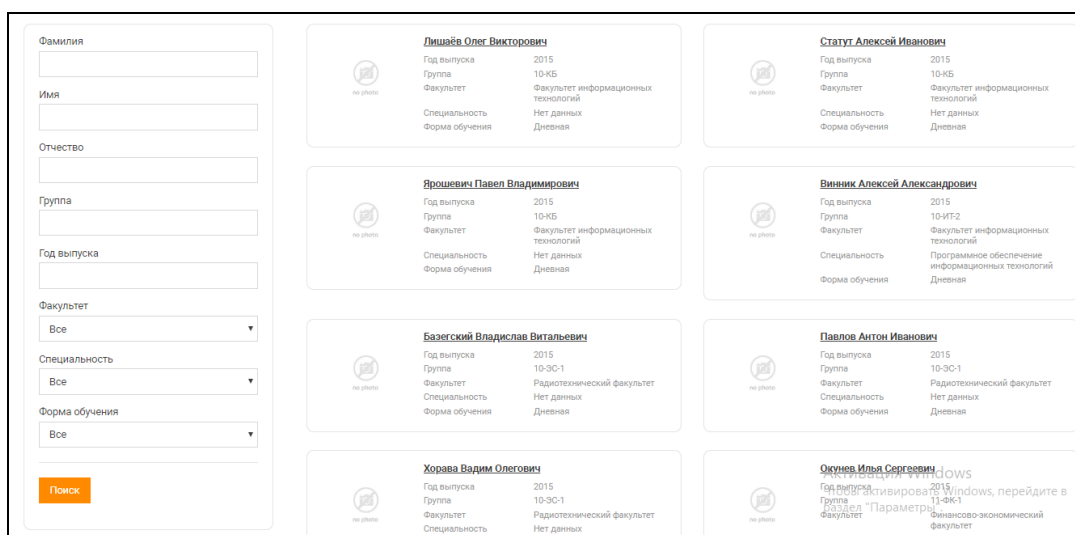


**Рисунок 1. – Страница редактирования личной информации**

Еще одной из главных особенностей данного сайта, является возможность поиска выпускников по следующим данным полям ввода:

1. Фамилия.
2. Имя.
3. Отчество.
4. Группа.
5. Год выпуска.
6. Факультет.
7. Специальность.
8. Форма обучения.

Все указанные поля необязательны для ввода. Если пользователь ничего не введет, то на экран выведутся все выпускники университета. Пример страницы поиска выпускника показан на рисунке 2. Данные отфильтрованы по полям факультет, специальность, форма обучения пользователь может получить из списка, который также загружается из базы данных.



**Рисунок 2. – Страница поиска выпускника**

В своих личных кабинетах, выпускники могут создавать объявления, которые отображаются на главной странице. Считаю, что данная функция очень удобная, так проще собрать своих одноклассников на какую-нибудь встречу, ведь не нужно обзванивать каждого и приглашать, тем более, что у некоторых

может отсутствовать информация о контактных телефонах. Чтобы удостовериться, что выпускник создал объявление с приемлемым контентом, оно проходит модерацию, и как только администратор удостовериться, что содержимое объявления действительно соответствует требованиям, он делает его видимым для всех выпускников, однако если объявление имеет непристойное содержимое, администратор имеет возможность его редакции.

Таким образом разработанный программный продукт способствует воспитанию в студенческой среде чувства гордости за родной университет, своей причастности к одному из лучших университетов страны и мира, к элите современного интеллектуального общества, а также позволит осуществлять пропаганду выдающихся достижений его выпускников.

**Заключение.** Данный сайт уже находится на стадии внедрения в учреждение образования «Полочский Государственный университет». Разработанный продукт позволит облегчить труд сотрудникам университета. С легкостью можно будет сообщить всем выпускникам о планируемом мероприятии в честь них. Кроме того, выпускники имеют возможность найти своих одногруппников по поиску, предоставленному на сайте, внести информацию о себе в личный кабинет. В данный момент сайт находится на этапе тестирования и доработки.

#### ЛИТЕРАТУРА

1. Unity User Manual [Электронный ресурс]. – Режим доступа: <http://www.php-s.ru/self-teacher>. – Дата доступа: 16.09.2018.

УДК 004.5; 004.921

**ВЕБ-САЙТ «ВЫПУСКНИКИ» В СФЕРЕ ОБРАЗОВАНИЯ****В.А. ЦЫБУЛЬСКИЙ***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассказывается о предназначении веб-сайта «Выпускники» в сфере образования.*

В современном мире без информационных технологий не обойтись. Сейчас они используются абсолютно во всех сферах жизни человека, хотя раньше, человек даже понятия не имел, что это такое.

Когда информационные технологии не были так популярны, выпускникам учебных заведений приходилось связываться друг с другом при помощи писем, сарафанного радио, а также через преподавателей. В настоящее время, выпускникам легче связываться между собой, ведь появились различные гаджеты для связи, но не все могут помнить номера и адреса своих одноклассников, одногруппников, однопочечников, ведь они могут измениться. Также, чтобы собрать выпускников на встречу, руководителям приходилось либо обзванивать каждого, либо вешать объявления, что не всегда помогало. Все это помогло принять решение для создания сайта для выпускников, на котором пользователи могут посмотреть фотографии и другие личные данные других выпускников, а также заполнить свои.

При создании сайта удалось найти совсем немного таких ресурсов, которые помогут выпускникам собраться, или найти своих друзей. На основе аналогов был создан сайт «Выпускники».

Для того, чтобы он хорошо функционировал, пришлось поднять все архивы с начала открытия учреждения образования и получить все данные, хотя бы от самого первого выпуска. Все собранные данные о выпускнике загружаются на сайт, на котором он может зарегистрировать при помощи своей ФИО, действительной на момент обучение, номером диплома, а также электронной почтой, на которую, при регистрации, придет пароль для входа. После регистрации, в своем личном кабинете, пользователь увидит информацию о себе, которая хранилась в университете, но, он может вписать свои личные данные. Кроме этого, каждому пользователю доступна загрузка фотографии на сайт, так как это поможет облегчить поиск своих друзей на сайте.

На сайте имеется возможность создания объявлений. Это облегчает труд работникам учреждения образования, так как им не нужно обзванивать тысячи выпускников, чтобы предупредить их о приближающейся встрече выпускников. Они могут создать объявление на сайте, где есть возможность указать, какой год выпуска приглашается. Данное объявление видно всем зарегистрированным пользователям. При помощи данного функционала, у работников учреждения образования уменьшится количество монотонной работы.

Еще одной особенностью данного сайта является то, что объявления могут оставлять не только администраторы сайта, но также и зарегистрированные пользователи. Пользователь, в своем личном кабинете, может создать объявление какое он захочет и отправить на опубликование на главной странице сайта, но оно не опубликуется, так как сразу оно отправляется на проверку и редактирование администратору, чтобы предотвратить возможность опубликования объявлений не по теме.

Сайт «Выпускники» можно считать универсальным и интересным по дизайну и содержанию. Его можно внедрить абсолютно в любое учреждение образования, ведь за настройку наполнения сайта отвечает администратор, он может настраивать сайт так как ему хочется, в различных тонах. Единственным минусом данного сайта, является загрузка информации об выпускниках на сайт. Ведь, для того, чтобы все корректно отображалось, необходимо правильно заполнить таблицу в Excel, откуда будет браться вся информация.

Данный сайт поможет всем выпускникам учебных заведений находить своих одногруппников и друзей, контактные данные которых могли потеряться. Ведь на сайте присутствует поиск по 8 параметрам:

1. Фамилия выпускника;
2. Имя выпускника;
3. Отчество выпускника;
4. Группа, в который он обучался;
5. Год его выпуска;
6. Факультет, на котором он обучался, если сайт будет использоваться для университета;
7. Специальность, на которой он обучался, если сайт будет использоваться для университета;
8. Форма обучения, на которой он обучался, если сайт будет использоваться для университета;

В процессе разработки программного продукта были соблюдены все ограничения, накладываемые на базу данных предметной областью, таким образом обеспечено поддержание целостности системы. Ограничение целостности заключается в отсутствии в отношениях записей с одинаковыми значениями

первичного ключа, а также в отсутствии записей, в которых значение первичного ключа не задано. Данное ограничение работает из-за соблюдения базовых свойств первичного ключа.

Важным отличием современных СУБД, является стандартное средство соблюдения целостности баз данных при изменении и удалении записей в отношениях.

Для внешних ключей ограничение целостности заключается в том, что значение первичного ключа должно быть равным значению внешнего ключа, т.е. каждое значение атрибута должно быть неопределенным.

Хранение необходимых данных реализовано в свободной реляционной системе управления базами данных MySQL 8.0, которая имеет преимущества по сравнению с аналогами:

1. большое количество плагинов и вспомогательных приложений, которые помогают упростить работу с базами данных;
2. широкий выбор инструмента, который позволяет реализовывать практически любой проект;
3. имеется множество встроенных функций безопасности, которые работают по умолчанию;
4. в равной степени легко может быть использована для работы с малыми и большими объемами данных;
5. за счет некоторых используемых стандартов имеется высокая производительность системы.

Университет состоит из нескольких факультетов. Каждый факультет имеет несколько кафедр, также может присутствовать несколько специальностей и форм обучения. Таким образом, каждый студент числится на факультете, на определенной кафедре, специальности и форме обучения и имеет уникальные личные данные. Таким образом, включая все варианты объединения выпускника с университетом, была разработана следующая схема, представленная на рисунке.

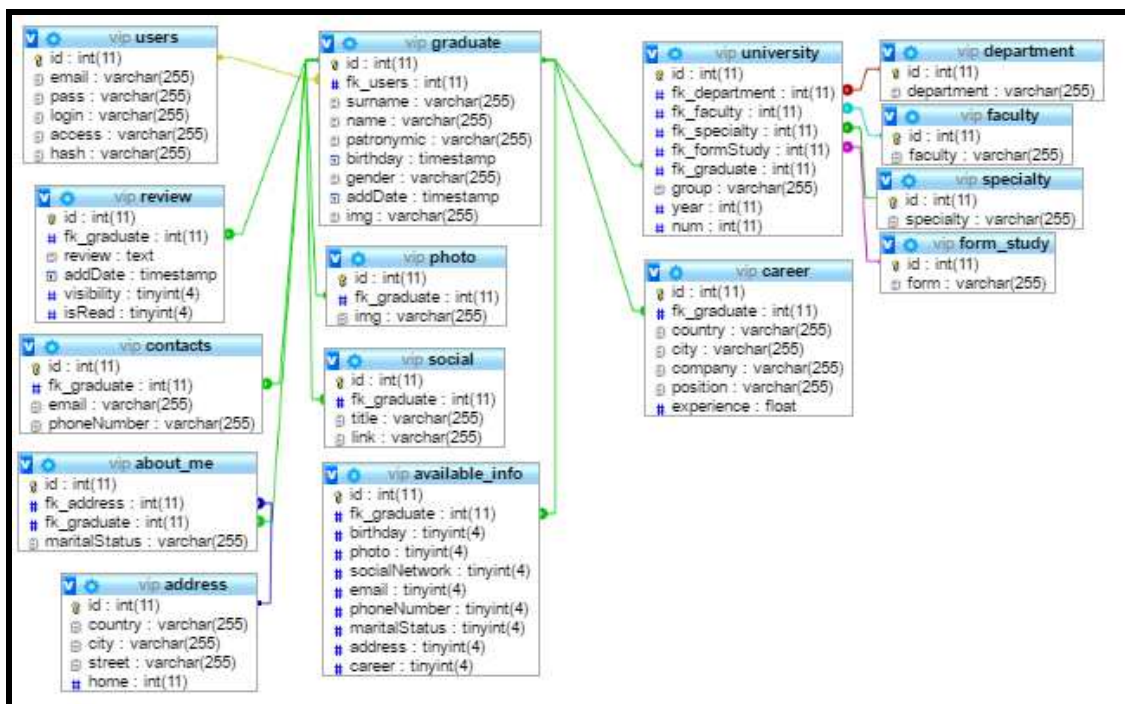


Рисунок. – Схема реляционной базы данных

При реализации данной реляционной схемы базы данных максимально использовались возможности языка SQL. SQL (англ. Structured Query Language – язык структурированных запросов) – универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

На основании разработанной базы данных, а также пользовательского интерфейса, был реализован программный продукт, который можно использовать в любых учреждениях образования.

#### ЛИТЕРАТУРА

1. Роль информационных технологий [Электронный ресурс]. – Режим доступа: [http://informatika.ru/Lectures/informacionnyh\\_tehnologii\\_v\\_sovremennom\\_mire.html](http://informatika.ru/Lectures/informacionnyh_tehnologii_v_sovremennom_mire.html). – Дата доступа: 21.09.2019.
2. Разработка веб-дизайна [Электронный ресурс]. – Режим доступа: <https://webformyself.com/razrabotka-krasivogo-veb-sajta-s-nulya>. – Дата доступа: 21.09.2019.

УДК 004.453.2

## РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АНАЛИЗА КОМПОНЕНТОВ ТЕНЗОРА ДИЭЛЕКТРИЧЕСКОЙ ПРОНИЦАЕМОСТИ УГЛЕВОДОРОДНОЙ ЗАЛЕЖИ

**В.Б. ТАЛДЫКИН**

(Представлено: С.В. КАЛИНЦЕВ)

Полноценное проведение анализа процесса взаимодействия электромагнитных волн (ЭМВ) с залежью углеводородов (УВЗ) невозможно без использования современных вычислительных средств, и в качестве основной цели данной статьи, помимо теоретического изучения процесса взаимодействия модулированных ЭМВ с УВЗ, рассматривается разработка структуры программного продукта с удобным пользовательским интерфейсом для моделирования процесса взаимодействия модулированных ЭМВ с УВЗ.

**Введение.** В настоящее время существуют эффективные методы для анализа частотных характеристик компонентов тензора диэлектрической проницаемости среды, а также воздействующего и отражённого сигнала для различных режимов модулированных ЭМВ с использованием вейвлет-преобразования. Подобный анализ позволит выявить зависимость характеристик среды и отражённого сигнала от параметров воздействующего сигнала, а также определить условия наступления резонанса, что в конечном счете позволит повысить качество и эффективность поиска и обнаружения УВЗ на основе использования модулированных ЭМВ.

Полноценное проведение описанного выше анализа невозможно без использования современных вычислительных средств, и в качестве основной цели данной статьи разрабатывается структура мощного программного продукта для моделирования процесса взаимодействия модулированных ЭМВ с УВЗ и анализа частотных характеристик компонентов тензоров диэлектрической проницаемости среды и последующее проведение численных экспериментов на широком диапазоне соотношений частот, амплитуд и параметров сигналов.

**Методика исследования.** В данной статье будет рассмотрен двумерный случай и для обоих случаев АМ и ЧМ сигналов проведем анализ величин:

$$\begin{aligned} \varepsilon_r(\omega) &= \varepsilon_1(\omega) + \varepsilon_2(\omega) \\ \varepsilon_L(\omega) &= \varepsilon_1(\omega) - \varepsilon_2(\omega) \end{aligned} \quad (1)$$

где  $\varepsilon_r(\omega)$  и  $\varepsilon_L(\omega)$  – сумма и разность компонентов тензора.

Отметим, что для физического моделирования зависимости тензора диэлектрической проницаемости от параметров ЭМВ, оказывается удобным проводить анализ именно величин (1). Будет произведена дискретизация этих величин на равномерной сетке, затем будет осуществлено обратное преобразование Фурье для перехода от частотного к временному пространству, а затем будет проведено вейвлет-преобразование полученных величин, что позволит эффективно вычислить отражательные характеристики среды. Наконец, с использованием полученных отражательных характеристик среды, окажется возможным вычисление отражённого сигнала по заданному падающему сигналу. [1]

В начале проектирования необходимо, исходя из характеристик программного продукта разработать функциональную структуру разрабатываемой системы. Соответствующий результат представлен на рисунке.

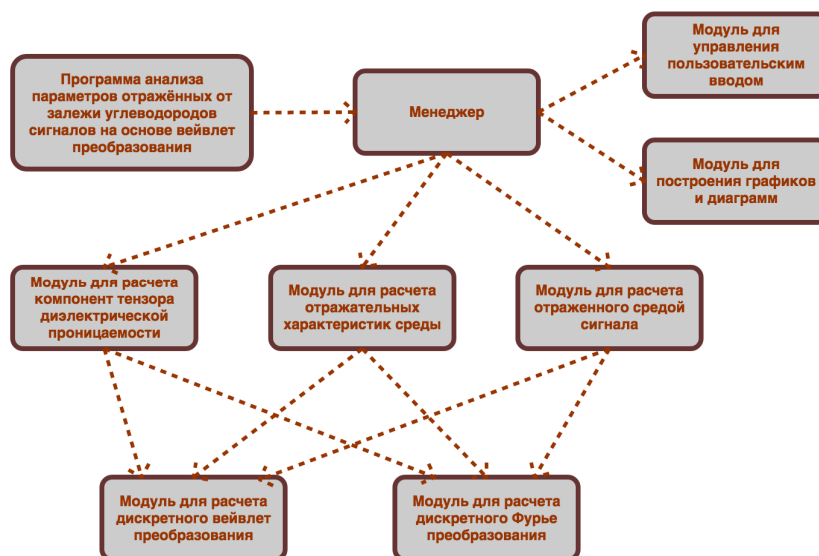


Рисунок. – Функциональная структура разрабатываемого программного обеспечения

В соответствии с описанием разрабатываемой системы, сформулированными ранее, программа разбивается на ряд самостоятельных функциональных блоков (или модулей), которые в рамках модульной структуры, поддерживаемой языком программирования Python [2] и стандартной практики объектно-ориентированного программирования организованы как следующие отдельные компоненты:

- 1) Менеджер – осуществляет отрисовку графического интерфейса, отслеживание пользовательских событий, запуск расчетов и процедур отображения результатов в графической и текстовой форме.
- 2) Модуль для управления пользовательским вводом – осуществляет непосредственный контроль вводимой пользователем информации.
- 3) Модуль для построения графиков и диаграмм – осуществляет непосредственное построение графиков и диаграмм.
- 4) Модуль для расчета компонент тензора диэлектрической проницаемости – осуществляет вычисление компонент тензора диэлектрической проницаемости, их Фурье и вейвлет преобразований.
- 5) Модуль для расчета отраженного средой сигнала – осуществляет вычисление отраженного средой сигнала с использованием Фурье и вейвлет преобразований.
- 6) Модуль для расчета дискретного вейвлет преобразования – осуществляет расчет соответствующего преобразования в дискретной форме.
- 7) Модуль для расчета дискретного Фурье преобразования – осуществляет расчет соответствующего преобразования в дискретной форме.

Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Реализация компонента осуществляется с использованием классов и их взаимодействий, которые отображаются в логическом представлении. В логическом представлении каждый класс реализует, как минимум, один компонент.

На рисунке 2 представлена соответствующая диаграмма компонентов создаваемой программы анализа параметров отражённых от залежи углеводородов сигналов на основе вейвлет-преобразования.

В соответствии с представленной на рисунке 2 диаграммой, выделяются следующие основные компоненты:

- 1) Менеджер программы анализа параметров отражённых от залежи углеводородов сигналов на основе вейвлет-преобразования – осуществляет общий контроль процесса функционирования программы, в том числе, осуществляет отрисовку графического интерфейса, отслеживание пользовательских событий, запуск расчетов и процедур отображения результатов в графической и текстовой форме.
- 2) Python библиотека NumPy – подключаемая внешняя библиотека, которая используется для эффективного проведения научных вычислительных процедур. Отметим, что существенным преимуществом данной библиотеки является оптимизация под векторные и матричные вычисления.
- 3) Python библиотека Matplotlib – подключаемая внешняя библиотека, которая используется для отрисовки графиков и диаграмм.
- 4) Python библиотека PyQt – подключаемая внешняя библиотека, которая используется для создания графического пользовательского интерфейса.
- 5) Python библиотека fbs – подключаемая внешняя библиотека, которая используется для сборки исполняемого exe файла из отдельных python модулей проекта. Отметим, что данная библиотека предполагает использование специальной структуры файлов и папок проекта, которая будет описана в следующей части работы.

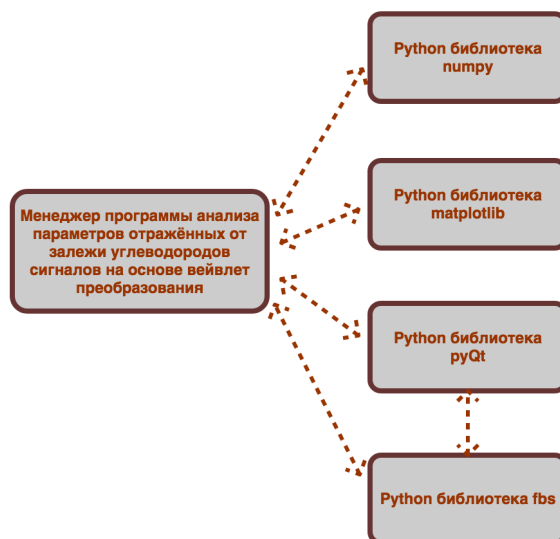


Рисунок 2. – Диаграмма компонентов программы

Для организации эффективных векторных и матричных операций мы используем библиотеку с открытым исходным кодом NumPy, обладающую поддержкой многомерных массивов (в том числе, векторов и матриц) и высокоуровневых математических функций, предназначенных для работы с многомерными массивами.

**Заключение.** В ходе проведенных исследований была разработана структура программного продукта с удобным пользовательским интерфейсом для моделирования процесса взаимодействия модулированных ЭМВ с УВЗ. Для отображения данных в форме графиков и диаграмм был применён пакет Matplotlib, являющийся библиотекой на языке программирования Python для визуализации данных в виде двумерной и трехмерной графики. Библиотека Matplotlib построена на принципах объектно-ориентированного программирования, но имеет процедурный интерфейс PyLab [3], который предоставляет практически полные аналоги команд Matlab. Matplotlib является гибким, легко конфигурируемым пакетом.

#### ЛИТЕРАТУРА

1. Гололобов, Д.В. Взаимодействие электромагнитных волн и углеводородных залежей. / Д.В. Гололобов. – Минск: Бестпринт, 2009. – 185 с.
2. Язык программирования python 3.7 [Электронный ресурс]. – Режим доступа: <https://www.python.org/downloads/release/python-370/>. – Дата доступа: 12.08.2019
3. Py2exe, Python библиотека [Электронный ресурс]. – Режим доступа: <http://www.py2exe.org> – Дата доступа: 12.08.2019



УДК 004.453.4

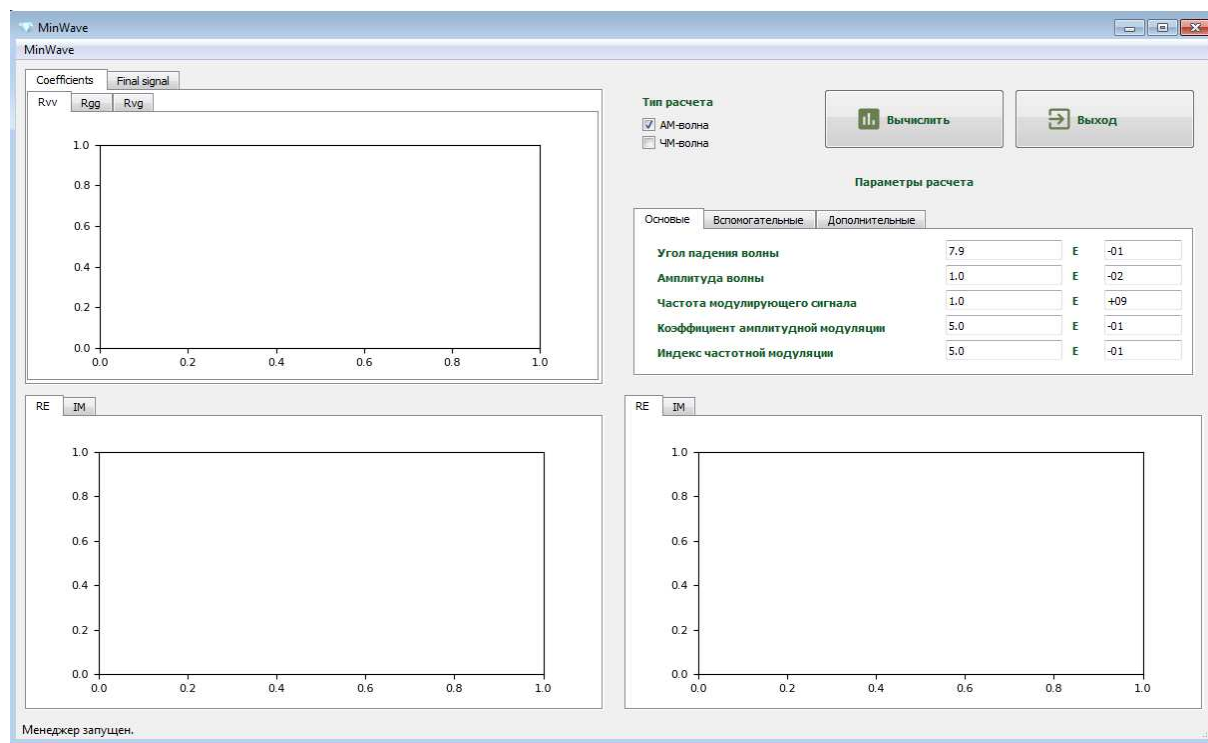
## РАЗРАБОТКА ИНТЕРФЕЙСА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АНАЛИЗА КОМПОНЕНТОВ ТЕНЗОРА ДИЭЛЕКТРИЧЕСКОЙ ПРОНИЦАЕМОСТИ УГЛЕВОДОРОДНОЙ ЗАЛЕЖИ

**В.Б. ТАЛДЫКИН***(Представлено: С.В. КАЛИНЦЕВ)*

В данной статье будут рассмотрены основные практические аспекты реализации интерфейса программного продукта для моделирования процесса взаимодействия модулированных электромагнитных волн (ЭМВ) с залежью углеводородов (УВЗ) и анализа частотных характеристик компонентов тензоров диэлектрической проницаемости среды с последующим проведением численных экспериментов для различных соотношений параметров сигналов.

**Введение.** Разрабатываемый интерфейс программы MinWave на языке программирования Python позволит эффективно производить моделирование процесса взаимодействия модулированных ЭМВ с УВЗ и анализ частотных характеристик компонентов тензоров диэлектрической проницаемости среды.

**Методика исследования.** Пользовательский интерфейс данной программы имеет вид, представленный на рисунке 1.

**Рисунок 1. – Интерфейс программы**

Рабочее окно разбито на сетку размера 2 на 2. В верхней правой ячейке находится основная панель управления программой. Данная панель позволяет выбрать тип расчета (АМ-волна или ЧМ-волна), а также задать значения параметров расчета. Все расчетные параметры разнесены условно по трем вкладкам: «Основные», «Вспомогательные» и «Дополнительные» (рис. 1). Задание параметров в программе производится в удобной математической нотации: в первой ячейке указывается значение, а во второй – соответствующий показатель экспоненты. Так, например, указанное на рисунке 1 значение для частоты модулирующего сигнала соответствует значению 1 ГГц и т.д.

В программе осуществляется контроль и проверка корректности значений параметров, вводимых пользователем. При попытке запуска расчета с введенным отрицательным значением для числа точек дискретизации, программа выдает соответствующее сообщение об ошибке.

Запуск расчета в программе осуществляется по нажатию на кнопку «Вычислить». Выход из программы может быть осуществлен при нажатии на кнопку «Выход». Данный функционал дублируется в выпадающем меню MinWave в верхней левой части окна программы.

После нажатия на кнопку «Вычислить» и окончания вычислений, в трех ячейках программы отображаются соответствующие результаты в графической форме (рис. 1).

В левой нижней ячейке представлены значения обратного преобразования Фурье  $\hat{\epsilon}_r$  компоненты  $\epsilon_r$  тензора диэлектрической проницаемости среды над УВЗ, а в правой части преобразование  $\hat{\epsilon}_i$  компоненты  $\epsilon_i$ . Для обеих компонент возможно переключение между режимами отображения действительной («RE») и мнимой («IM») частей (см. соответствующее переключение на мнимую часть на рисунке 14).

В левой верхней ячейке возможно переключение между отображением результата для обратного преобразования Фурье  $\hat{R}_{vv}$ ,  $\hat{R}_{gg}$ ,  $\hat{R}_{vg}$ , величин отражательных характеристик среды  $R_{vv}$ ,  $R_{gg}$ ,  $R_{vg}$  (вкладка «Coefficients») и итоговым отраженным сигналом  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  (вкладка «Final signal»). [1]

На рисунке 2 приводятся полученные результаты для величин  $\hat{R}_{vv}$ ,  $\hat{R}_{gg}$  и  $\hat{R}_{vg}$  соответственно и результаты полученных значений для величины  $x_2(t)$ .

Таким образом, созданный интерфейс программы функционирует корректно, что позволяет осуществлять моделирование процесса взаимодействия излучения с УВЗ и отображает в графической форме все необходимые результаты

Специалист геофизического профиля может осуществить тонкую настройку параметров расчета в графическом интерфейсе и провести полноценный анализ интересующих его физических процессов, связанных с отражением излучения от УВЗ.

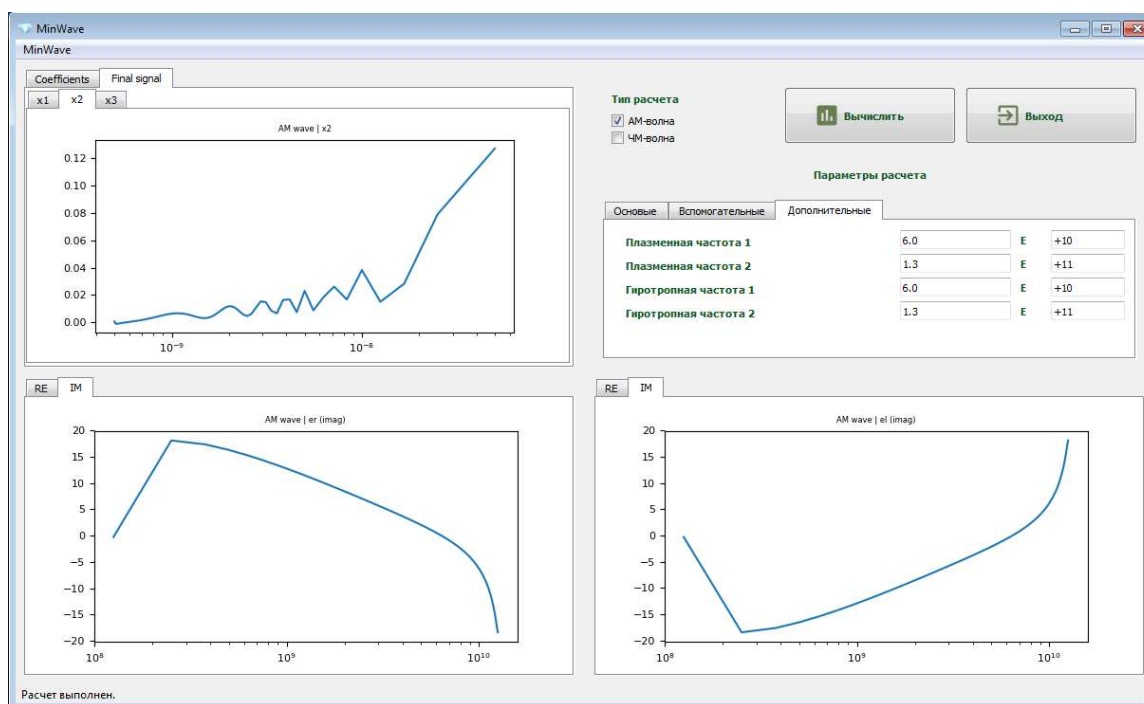


Рисунок 2. – Результат расчета, переключение вкладок с графиками

**Заключение.** В ходе проведенных исследований была разработана программа для моделирования процесса взаимодействия модулированных ЭМВ с УВЗ. Для отображения данных в форме графиков и диаграмм был применен пакет Matplotlib, являющийся библиотекой на языке программирования Python для визуализации данных в виде двумерной и трехмерной графики. [2] При разработке графического интерфейса программы на языке программирования Python также был использован модуль PyQt5, который является оберткой к мощной C++ библиотеке графических интерфейсов Qt5.

#### ЛИТЕРАТУРА

1. PyInstaller, python библиотека [Электронный ресурс]. – Режим доступа: <https://www.pyinstaller.org> – Дата доступа: 12.09.2019
2. Язык программирования python 3.7 [Электронный ресурс]. – Режим доступа: <https://www.python.org/downloads/release/python-370/>. – Дата доступа: 12.08.2019

УДК 004.021

## РАЗРАБОТКА ИГРОВЫХ ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ FACEBOOK

М.А. ТРАЩЕНКО

(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)

*В данной статье рассматриваются принципы разработки моментальных игр для платформы Facebook Instant Games.*

Facebook Instant Games является довольно молодой платформой, количество выпущенных на данной платформе приложений - порядка 3000, что в 600 и 1300 раз меньше, чем на платформах AppStore и Google Play соответственно. Так же стоит заметить, что приложения опубликованные на платформе Facebook Instant Games являются мультиплатформенными: приложения запускаются через мессенджер, который доступен на платформах iOS, Android, Mac OS, Windows. Приложения Facebook набирают аудиторию на манер «сарафанного радио», подобно цепной реакции. Если у вас есть действительно стоящая идея, то реализовав ее, вы получаете потенциальную аудиторию в 20 млн. пользователей Facebook, при гораздо меньшей конкуренции со стороны других разработчиков. При таком размахе не удивительно, что некоторые любительские приложения для Facebook продаются за десятки тысяч долларов.

Платформа Facebook Instant Games дает возможность пользователям Facebook играть в небольшие бесплатные игры без скачивания. На данной платформе Приложения можно запускать из ленты новостей или во время разговора в Messenger. Базовая технология для разработки приложения для платформы Facebook Instant Games – HTML5 и WebGL.

Любое приложения создается не только, для развлечения пользователей, а так же для получения прибыли. Платформа Facebook Instant Games предоставляет классические способы монетизации приложений: можно реализовать IAP (in-app purchase, внутриигровые покупки) и рекламу (операторами выступают аффилированные Facebook сети). Можно отметить, что данная платформа была открыта для разработчиков в марте 2018 года, сейчас на ней опубликовано около 3000 игр.

Особенности становления платформы на разных этапах:

- 14 марта 2018: стало доступно API Facebook Instant Games для разработчиков по всему миру;
- 21 марта 2018: из-за скандала вокруг утечки данных, Facebook останавливает публикацию игр почти на месяц и прекращает проверку новых приложений;
- 18 апреля 2018: площадка снова открывается, но, как показывает опыт GBN, ревью игр приходится ждать 5 недель вместо заявленных 3-5 дней. К тому же на этапе проверки выясняется, что разработчики ряда стран, в том числе России, не имеют доступа к рекламным возможностям Instant Games, поскольку, по мнению Facebook, из этих регионов велика вероятность фрода (fraud, вид мошенничества в области информационных технологий) и некачественного контента [2].

Требования и рекомендации для тех, кто хочет выпустить игру для Instant Games.

Требования:

- размер игры не должен превышать 200 Мб;
- скорость загрузки должна быть не больше 5 секунд;
- адаптация игры как под веб-браузеры на компьютерах, так и для мобильных устройств.[1]

Обязательные элементы: регистрация через аккаунт Facebook, лидерборды (leaderboard, рейтинг участников какого-либо соревнования) и страница приложения.

Рекомендации:

- встраивание виральных механик в игру;
- использование ботов для возвращения игроков в приложение (есть ограничения на отправку сообщений: не больше 5 сообщений за 10 дней с момента последней игровой сессии).

Подход к разработке игр для мессенджера отличается от того, что используется при дизайне нативных мобильных или социальных игр. Это не останавливает многих мобильных разработчиков от прямого портирования мобильных игр в мессенджер. В исключительных случаях это работает (Cut the Rope и Adventure Capitalist), но большинство портов проваливаются [1].

Это значит, что новая платформа означает смену не только технической парадигмы, но и подхода к дизайну. Иными словами, переход на Instant Games влечёт за собой глобальные метаморфозы для продуктов, как это было при восхождении мобильных игр в 2010–2011 годах.

Да, то, что касается взаимодействия пользователя с игровыми объектами и в целом UX (user experience, пользовательский опыт) по-прежнему применимо и Instant Games (это всё-таки платформа с тач-управлением). Но чтобы достичь нужных показателей удержания, обычного портирования недостаточно. Основной цикл, игровой прогресс, взаимодействие с игроками – всё это нужно при переносе мейнтейн. В ином случае реализовать потенциал игры на новой платформе не удастся.

Мессенджер в первую очередь всё-таки чат-приложение. Это значит, что места под игры на такой платформе не так-то много. То есть игра должна быть сделана так, чтобы оставаться актуальной в рамках чат-приложения.

В интерфейсе Facebook-мессенджера есть вкладка с играми (Games) и поисковиком новых игр (Discover), но одних этих элементов интерфейса вам будет недостаточно, чтобы удерживать аудиторию. Вам нужно формировать для игрока «социальное окружение» и правильно использовать чат-ботов. Социальное окружение – это чаты с друзьями и групповые обсуждения, в рамках которых ваша игра будет релевантна в отношении игр-конкурентов.

Учтите, что чат-боты при неправильном использовании могут восприниматься как спам, а Facebook очень строг в том, что касается частоты оповещений. Вскоре после появления в Facebook игр соцсеть научилась делать так, чтобы разработчики не становились помехой для пользователей платформы. Если игроки не будут заходить в чат с ботом, он быстро пропадёт с домашней страницы мессенджера. Кроме того, если бот будет раздражать игроков, те мгновенно отключат оповещения.

Таким образом, перечислим главные испытания для разработчика игр под мессенджеры.

– Как сделать игру, которая будет актуально и для персональных чатов, и для групповых?

– Как сделать так, чтобы боты не засоряли чат пользователей, а были им полезны?

– Как настроить ботов и социальное окружение так, чтобы игроку хотелось возвращаться к игре?

Как видите, это не те вопросы, которые стоят перед дизайнерами нативных мобильных игр. Лучшие игры для мессенджеров – это те, что лучше всех умеют наладить социальное взаимодействие игроков. Это повышает как виральность (способность контента распространяться самостоятельно) проекта, так и удержание. Новизна игр для мессенджеров в динамике групповых чатов. Хорошие проекты подталкивают игроков к тому, чтобы вовлекаться в групповое обсуждение: совместно решать какую-то задачу или соперничать друг с другом.

Лучший пример подобного социального взаимодействия – система «групповых рейдов». Групповой рейд – это когда пользователи запускают сложный уровень, который можно пройти только сотрудничая с другими игроками в групповом чате. Чем сильнее участники группы, тем сложнее будет сам уровень, и тем выше окажется награда. Такая система позволяет разноуровневым игрокам играть вместе. Она вознаграждает пользователя за совместную игру с друзьями и мотивирует на то, чтобы играть больше. Однако подобная механика может применяться только в достаточно глубоких играх типа RPG, не все проекты могут использовать этот подход. Некоторые игры для налаживания социального взаимодействия игроков используют пошаговый геймплей. После каждого хода игрок отправляет сообщение о своих действиях оппоненту/союзнику. [1]

У игр для Instant Games всё ещё период детского роста, но этот рынок развивается очень быстро. За один год здесь уже произошли большие сдвиги, выявившие, какие игры подходят для платформы лучше. Многие тайтлы, которые при появлении платформы оказались очень популярны (Galaga, Space Invaders, PAC-Man), фактически пропали, а немало новинок, появившихся в последние полгода, наоборот, стали топовыми (Snake Mania, Cookie Crush). Игры для Instant Games в ближайшее время будут «Диким Западом» для разработчиков. Чем больше разработчиков пойдёт на платформу и чем острее встанет проблема поиска игр, тем сильнее дизайн этих игр будут двигаться в сторону социального взаимодействия.

#### ЛИТЕРАТУРА

1. App2Top [Электронный ресурс] Facebook Instant Games: как разрабатывать игры для мессенджеров. – Режим доступа: [https://app2top.ru/game\\_development/facebook-instant-analysis-127048](https://app2top.ru/game_development/facebook-instant-analysis-127048) – Дата доступа: 13.09.19.
2. AppTractor [Электронный ресурс] Разработка Facebook Instant Games / – Режим доступа: <https://apptractor.ru/info/articles/razrabotka-facebook-instant-games-opyit-cat-grows-fat-ot-gbn.html>. – Дата доступа: 14.09.19.

УДК 004.021

**ПРИНЦИПЫ РАЗРАБОТКИ ИГРАБЕЛЬНОЙ РЕКЛАМЫ****М.А. ТРАЩЕНКО***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)**В данной статье рассматриваются основные принципы разработки играбельной рекламы*

Играбельная реклама (playables) – «тест-драйв» мобильного приложения до его загрузки. «Тестирование» происходит в рамках рекламы, показываемой в приложении. Играбельная реклама длится от 15 секунд до минуты. Обычно playables содержит в себе три компонента: обучение, геймплей (gameplay, компонент игры, отвечающий за интерактивное взаимодействие игры и игрока) и так называемую «финальную карту» (End card).

Обучение, которое должно занимать по времени меньше трех секунд, знакомит игрока с процессом и объясняет, как играть.

Игроку не надо искать, как скачать версию игры на пробу. Вместо этого рекламная платформа предоставляет ему кусочек реального геймплея. Платформа надеется, что пользователь будет заинтригован и захочет продолжить игру, скачав её полную версию. Геймплей длится от 10 до 20 секунд. За это время у игрока должно сформироваться отчетливое понимание, на что похожа полная версия игры.

Финальная карта (End card) – это призыв игрока к действию. Реклама просит установить приложение или совершить какой-либо иное действие.

Важно: на протяжении всей рекламы у пользователя всегда должна быть опция закрыть ее, чтобы игрок чувствовал контроль над ситуацией, а его опыт от взаимодействия с продуктом оставался положительным.

Для большинства проектов главным показателем успешности является retention (удержание пользователя), что весьма логично, ведь чем дольше пользователь остается в игре, тем больше денег он принесет проекту. Экосистема мобильных приложений сегодня очень конкурентна, поэтому получить загрузки не так уж просто. При этом ситуация с удержанием пользователей еще более ужасающая.

У лучших игр под операционную систему iOS retention 30-го дня составляет 15,8%. У таких проектов, как Candy Crush Saga, Sudoku и Angry Birds 2 показатель удержания пользователя составляет порядка 11–15%. На операционной системе Android – все показатели еще хуже. Вот почему играбельная реклама столь важна для получения качественных пользователей. Когда у аудитории есть возможность сначала попробовать игру, то та ее часть, которая после тестирования геймплея в рекламе игру все-таки скачает, превратится для продукта в высококачественных пользователей, которые не сотрут и не забудут продукт в ближайшее время.

Играбельная реклама также положительно влияет на конверсию. Поскольку играбельная реклама интерактивна, ее параметры вовлечения легко отслеживать. Благодаря этому маркетологи видят, насколько пользователи заинтересованы в проекте. Но самое важное здесь то, что пользователи совершают, когда реклама подходит к концу. У играбельной рекламы высокая конверсия – в некоторых случаях переход в магазин приложения увеличивается в 3–4 раза.

Поскольку играбельная реклама чуть более сложный формат, она требует больше времени (и большего бюджета) на создание. Тут уже неважно, работает ли над ней издатель игры или его партнер по маркетингу. Затраты все равно окажутся больше, чем при работе с более распространенными форматами.

Как и любая другая реклама, играбельная со временем надоедает, особенно если показатель frequency (частота показов рекламного объявления одному и тому же пользователю) больше 2. Так что при использовании формата следует сосредоточить свои усилия на первой половине кампании, когда реклама только-только вышла и воспринимается как новый опыт. Играбельная реклама присутствует на рынке уже несколько лет, но до сих пор только небольшой процент рекламных бюджетов расходуется на этот формат. При этом к нему очень лояльно относятся пользователи. Он является вторым по популярности полноэкранным форматом среди рекламодателей (22%), сразу после вознаграждаемого видео (rewarded video) (40%) [1].

Аудитория менее терпима по отношению к полноэкранным статичным баннерам (interstitials), показываемым прямо посреди геймплея видео (pre/mid-roll video) и рекламе, выдающей себя за часть нативного контента (in-feed ads, размещение контента в новостной ленте онлайн-издания). Чтобы подсластить опыт взаимодействия с играбельной рекламой никто не мешает сделать ее также вознаграждаемой. Тем самым возможно повысить лояльность к рекламируемому продукту со стороны пользователей. В данный момент семь из десяти топовых мобильных рекламных компаний уже протестировали играбельный формат. Год назад эксперименты с этим форматом проводили только три из десяти компаний, однако 71% среди тех, кто прибежал к подобной рекламе, нашли ее эффективной [1].

IT-гиганты также следуют за трендом. Google не только представила мгновенные игры (instant games) в Play Store, но и продвигает данный формат в социальных медиа и рекламе. Facebook делает что-то очень похожее в своей рекламной ленте.

Играбельная реклама по рассмотренным выше причинам пользуется наибольшим интересом со стороны тех мобильных издателей, которые занимаются издательством разного рода игр. Особенно это касается компаний, специализирующихся на казуальных и стратегических играх. Причина в том, что их геймплей более-менее стандартен (простой концепт: не требует обучения и сводится к простым взаимодействиям, вроде свайпа (swipe – проводить не отрывая палец от экрана) или нажатия). Аркады, настольные, карточные и азартные игры также идеально подходят, как и головоломки, викторины или словесные продукты.

#### ЛИТЕРАТУРА

1. App2Top [Электронный ресурс] Особенности работы с рекламным форматом Playable Ads. – Режим доступа: <https://app2top.ru/marketing/osobennosti-raboty-s-reklamny-m-formatom-playable-ads-opy-t-adcolony-123777>. – Дата доступа: 13.09.19.
2. AppTractor [Электронный ресурс] Путеводитель IAB по играбельной рекламе. – Режим доступа: <https://apptractor.ru/marketing-monetization/ad-networks/iab-playable-ad.html>. – Дата доступа: 14.09.19.

УДК 004.432.2

## МОДУЛЬНОЕ ТЕСТИРОВАНИЕ КАК АЛЬТЕРНАТИВНЫЙ ИНСТРУМЕНТ ДЛЯ ПРОВЕРКИ ТИПОВ

**К.И. ЛАПКОВСКИЙ**

(Представлено: Д.В. ПЯТКИН)

В статье рассматривается использование модульного тестирования для проверки интерфейсов объекта или класса. Сначала познакомимся с проблемой проверки интерфейсов, которая стоит перед языками с динамической типизацией. Реализуем проверку интерфейса для произвольного объекта на javascript. Проанализируем плюсы и минусы использования такого подхода, сравним со статической типизацией.

**Введение.** Динамическая типизация - приём, широко используемый в языках программирования и языках спецификации, при котором переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной. Таким образом, в различных участках программы одна и та же переменная может принимать значения разных типов. Примеры языков с динамической типизацией – Smalltalk, Python, Objective-C, Ruby, PHP, Perl, JavaScript, Lisp, xBase, Erlang, Visual Basic [1].

Основная проблема языков с динамической типизации - их ненадежность. В таких языках нельзя описать абстрактный метод или интерфейс. Хорошее решение в такой ситуации - выбрасывать ошибку при обращении к методу базового класса, однако это не заставит программиста переопределить всех потомков абстрактный метод, скорее запретит вызов метода базового класса. Сам факт вызова такого метода может обнаружиться только во время выполнения программы, поэтому ошибка может долго оставаться незамеченной.

**Основной раздел.** Реализуем проверку типов на языке программирования JavaScript. Для написания модульных тестов используются библиотеки mocha и chai.

### Проверка наличия свойств объекта

В листинге 1 создается функция, которая на вход принимает объект. В этой функции с помощью утилит describe и it соответственно описывается и создается модульный тест. В модульном тесте производится проверка наличия свойства в объекте.

Создавая такую функции для каждого интерфейса, который необходимо проверить, можно существенно сократить повторяемость кода, по сравнению с подходом, где для каждой реализации интерфейса создаются свои тесты.

```
const expect = require('chai').expect

const interfaceChecker = instance => {
  describe('#propA', () =>{
    it('should has propA', () => {
      expect(instance).to.have.property('propA')
    })
  })
}

interfaceChecker({
  propA:1
})
```

Листинг 1. – Проверка наличия свойств объекта

### Проверка наличия методов объекта

Методы соответствуют определенному интерфейсу если сигнатуры этих методов совпадают. Сигнатура метод – это совокупность названия метода и количества принимаемых аргументов. Предыдущие два утверждения справедливы только для языков с динамической типизацией.

```
const expect = require('chai').expect

const interfaceChecker = instance => {
  describe('#methodA', () =>{
    it('should has methodA', () => {
```

```
        expect(instance).respondTo('methodA')
    })

    it('methodA should have 2 arguments', () =>{
        expect(instance.methodA.length).equals(2)
    })
})
}

interfaceChecker({
    methodA(a,b){}
})
```

#### Листинг 2. – Проверка наличия метода объекта

В листинге 2 проверяется наличие метода `methodA` у объекта, переданного в функцию `interfaceChecker`.

Основные плюсы использования тестов:

- выявление ошибок реализации интерфейса во время выполнения модульных тестов
- возможность реализовать проверку типов вне зависимости от языка программирования

минусы использования тестов:

- затраты на написание тестов
- необходимость наличия этапа выполнения модульных тестов во время сборки или деплоя программы.

**Заключение.** Использование модульных тестов для проверки типов отлично подходит для повышения надежности критически важных частей программы, не влияя на часто изменяющиеся части программы не имеющие стабильного интерфейса. Однако модульные тесты необходимо поддерживать, поэтому если программа объемна, то лучше задуматься о использовании языка программирования со статической типизацией.

#### ЛИТЕРАТУРА

1. Wikipedia [Электронный ресурс] Динамическая типизация. Режим доступа: [https://ru.wikipedia.org/wiki/ Динамическая\\_типизация](https://ru.wikipedia.org/wiki/Динамическая_типизация). Дата доступа: 26.09.2019.



УДК 004.432.2

## УНИВЕРСАЛЬНЫЙ NULL ОБЪЕКТ НА JAVASCRIPT

К.И. ЛАПКОВСКИЙ

(Представлено: Д.В. ПЯТКИН)

Статья посвящена использованию паттерна *null object*. Рассмотрим цели использования этого паттерна в приложениях. Познакомимся со средствами, необходимыми для реализации поставленной задачи. В результате будет реализован универсальный *null object* на javascript.

**Введение.** В объектно-ориентированном программировании *null object* – это объект с определенным нейтральным («null») поведением [1]. Есть несколько причин использования *null object*:

1. Защита от *null pointer exception*. Своя версия *null pointer exception* есть во многих современных языках программирования: Java, JavaScript, C#. Такая ошибка появляется при обращении к свойствам или методам переменной, значение которой равно *null*. *null object* реализует интерфейс объекта, тем самым подстраиваясь под полноценный инстанс класса.

2. Предоставление дефолтного (нейтрального) поведения. Часто под нейтральным поведением понимают его отсутствие, то есть метод который необходимо реализовать просто оставляют пустым.

Основная проблема *null object* – его реализация. Под каждый интерфейс необходимо написать свой *null object*. В результате количество классов в системе вырастает пропорционально используемых абстракций.

Универсальный *null object* разработан для решения первой задачи. Основное преимущество универсального *null object* – способность подстраиваться под любой интерфейс.

**Основной раздел.** Для реализации универсального *null object* необходимо познакомиться с классом, который входит в стандартное окружения браузера – *Proxy*. Объект *Proxy* «оборачивается» вокруг другого объекта и может перехватывать (и, при желании, самостоятельно обрабатывать) разные действия с ним, например чтение/запись свойств и другие [2]. Работа с проксируемым объектом построена на концепции ловушек. Если необходимо контролировать все обращения к полям проксируемого объекта, необходимо реализовать *get*-ловушку, также есть ловушки для вызова методов, удаления/записи свойств и т.д.

Реализация перехвата получения значения свойства объекта

```
const proxy = new Proxy({}, {
  get(target, prop, receiver) {
    return receiver
  }
});
```

**Листинг 1. – Реализация перехвата получения значения свойства объекта**

Ловушка получения значения свойства объекта принимает три параметра: *target* – проксируемый объект, в нашем случае это пустой объект (`{}`), *prop* – название свойства, значение которого необходимо получить, *receiver* – это либо сам *proxy* объект, либо его наследник. [MDN] Как видно из листинга 1, любое обращение к любому свойству объекта возвращает *proxy* объект.

Для проверки работоспособности попробуем обратиться к вложенным полям *proxy* объекта.

Листинг 2 – проверка реализации перехвата получения значения свойства объекта

```
proxy.a.b.c
```

Ошибка `TypeError: Cannot read property`, аналог *null pointer exception* в javascript, не выбросилась, значит универсальный *null object* реализован.

**Заключение.** В статье рассмотрен пример реализации универсального *null object* на javascript. Такой инструмент может заменить множество уже существующих *null object*, что упростит структуру программы уменьшив количество повторяющегося кода.

## ЛИТЕРАТУРА

1. Wikipedia [Электронный ресурс] Null Object (шаблон проектирования). Режим доступа: [https://ru.wikipedia.org/wiki/Null\\_object\\_\(шаблон\\_проектирования\)](https://ru.wikipedia.org/wiki/Null_object_(шаблон_проектирования)). Дата доступа: 26.09.2019.
2. The Modern JavaScript tutorial [Электронный ресурс] Proxy и Reflect. Режим доступа: <https://learn.javascript.ru/proxy>. Дата доступа: 26.09.2019.
3. MDN web docs [Электронный ресурс] handler.get. Режим доступа: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Proxy/handler/get](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Proxy/handler/get). Дата доступа: 26.09.2019

УДК 004.421

**ОПРЕДЕЛЕНИЕ ТОЧНОГО МЕСТОПОЛОЖЕНИЯ ПОЛЬЗОВАТЕЛЯ САЙТА  
С ИСПОЛЬЗОВАНИЕМ JAVASCRIPT****Н.И. ЛАПКОВСКИЙ**  
(Представлено: Д.В. ПЯТКИН)

Большинство современных устройств способны определять свое местоположение с помощью GPS, WiFi или IP-геолокации. Разработчики могут использовать эту информацию, чтобы предоставить лучшие варианты поиска, расположенные поблизости магазины и реализовать все виды полезных взаимодействий с картами в своих приложениях и на веб-сайтах.

В статье представлен простой способ, основанный на чистом JavaScript, для доступа к местонахождению устройства, не полагаясь на какие-либо внешние зависимости или сторонние сервисы.

**Источники информации о местоположении.** JavaScript предлагает простой, но мощный инструмент для определения местоположения устройств в виде API геолокации. Он состоит из небольшого набора простых в использовании методов, которые могут получить положение устройства с помощью всех трех из упомянутых выше служб:

- GPS – в первую очередь на мобильных устройствах, очень точный (до 10 метров);
- WiFi – доступен на большинстве подключенных к интернету устройств, также очень точный;
- IP-геолокация – ограничена регионом и часто ненадежна, используется как сценарий наихудшего случая, когда два других способа терпят неудачу.

Когда запрашиваются геоданные, браузер пытается использовать все три вышеупомянутых варианта в зависимости от того, что доступно. Результаты из источника WiFi используются чаще всего, поскольку он быстрее, чем GPS, и намного точнее, чем IP-геолокация [1].

**Использование API геолокации.** Geolocation API имеет почти полную межбраузерную поддержку, но чтобы убедиться, что он доступен для наших пользователей, лучше перед тем, как что-либо делать, проверить, существует ли объект геолокации в интерфейсе Window.navigator. Пример представлен в листинге 1.

```
if (navigator.geolocation) {  
    // геолокация возможна  
}  
else {  
    //геолокация не поддерживается  
}
```

Внутри объекта navigator.geolocation находятся все методы для API:

- Geolocation.getCurrentPosition () – определяет текущее местоположение устройства;
- Geolocation.watchPosition () – прослушивает изменения в местоположении и запускает обратный вызов при каждом движении;
- Geolocation.clearWatch () – удаляет обработчик события watchPosition.

**Листинг 1 – Проверка существования объекта геолокации в интерфейсе Window.navigator**

Методы getCurrentPosition () и watchPosition () используются практически одинаково. Они оба работают асинхронно, пытаясь получить положение устройства и в зависимости от результат попытки запустить обратный вызов успешного завершения или обратный вызов ошибки, если она имеется [2]. Реализация метода getCurrentPosition () представлена в листинге 2.

```
navigator.geolocation.getCurrentPosition(  
  
    // Success callback  
    function(position) {  
  
        /* Объект position содержит различную информацию о местоположении устройства:  
        position = {  
            coords: {  
                latitude – Географическая широта в десятичных градусах.  
                longitude – Географическая долгота в десятичных градусах.            }  
        }  
    }  
);
```

```
altitude - Высота в метрах относительно уровня моря.  
accuracy - Возможная погрешность для координат в метрах.  
altitudeAccuracy - Возможная погрешность для высоты в  
метрах.  
heading - Направление устройства в градусах относи-  
тельно севера.  
speed - Скорость устройства в метрах в секунду.  
    }  
timestamp - Время, когда местоположение было получено.  
    }  
*/  
  
},  
  
// Optional error callback  
function(error) {  
  
    /*  
    В объекте error хранится причина неудачной попытки:  
  
    error = {  
        code - Код ошибки, представляющий её тип  
            1 - PERMISSION_DENIED  
            2 - POSITION_UNAVAILABLE  
            3 - TIMEOUT  
  
        message - Детали об ошибке в читаемом для человека формате.  
    }  
    */  
  
}  
  
);
```

#### Листинг 2. – Реализация метода `getCurrentPosition()`

Использовать Geolocation API довольно просто — нам просто нужно вызвать правильный метод, подождать, пока он вернет координаты, и затем сделать с ними все, что мы захотим.

**Разрешение пользователя.** Поскольку Geolocation API предоставляет глубоко личную информацию, когда приложение пытается получить к ней доступ в первый раз, появляется диалоговое окно с запросом разрешения. Это гарантирует, что пользователи не будут раскрывать свои личные данные, если они явно не разрешат это (рис. 1).

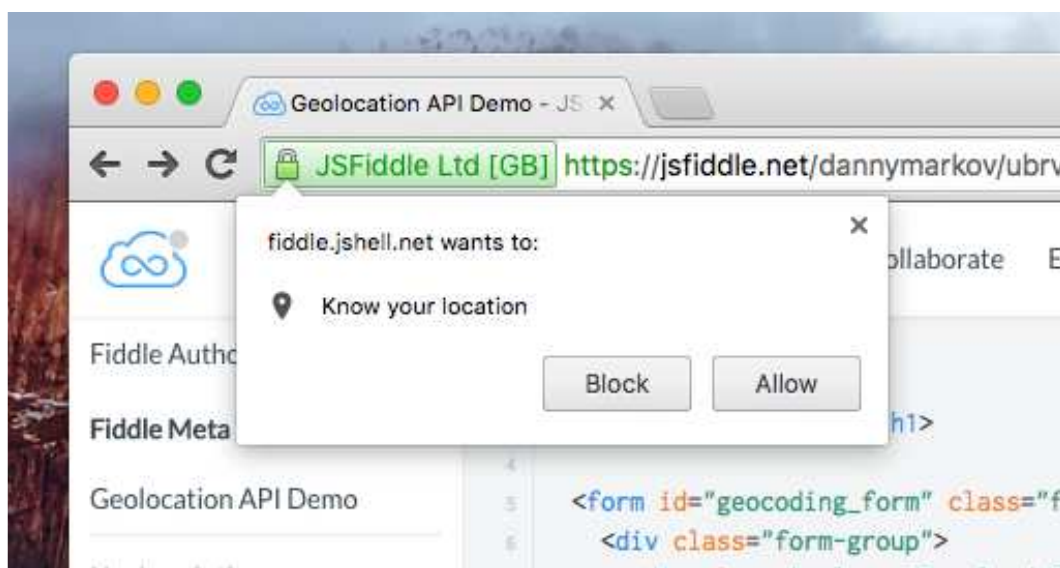


Рисунок 1. – Диалоговое окно с запросом разрешения использования данных геолокации

Обычно браузер сам заботится об отображении диалога, но разрешение также может быть запрошено разработчиком программно. Это иногда необходимо, так как однажды запрещенный исходный диалог, сгенерированный браузером, не появляется во второй раз.

Другая защитная мера - использование HTTPS-соединения. Из-за новой политики веб-безопасности Google Chrome (как для настольных, так и для мобильных версий) больше не позволяет незащищенным хостам запускать API геолокации. Вместо этого разработчики, которые хотят использовать эту функцию, должны обслуживать свои приложения по протоколу HTTPS, таким образом сводя к минимуму риски кражи или злоупотребления данными людей [3].

**Демо приложение.** В листинге 3 показан исходный код простого приложения, демонстрирующего некоторые возможности API. Оно состоит из кнопки, которая при нажатии захватывает координаты устройства и передает их в плагин GMaps, указывая точное местоположение на карте.

```
findMeButton.on('click', function(){

    navigator.geolocation.getCurrentPosition(function(position) {

        // Получаем координаты текущей позиции.
        var lat = position.coords.latitude;
        var lng = position.coords.longitude;

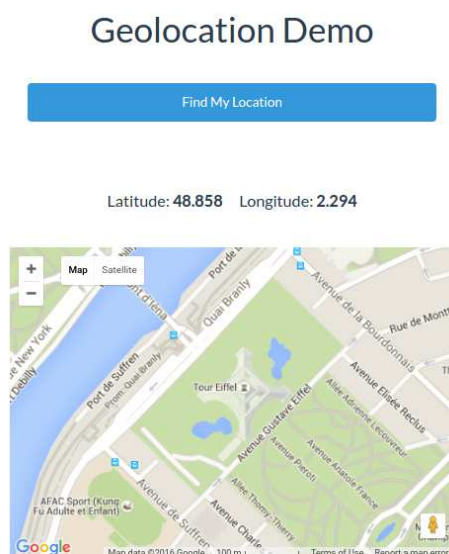
        // Создаём новую карту и устанавливаем маркер на местоположе-
ние устройства.
        var map = new GMaps({
            el: '#map',
            lat: lat,
            lng: lng
        });

        map.addMarker({
            lat: lat,
            lng: lng
        });

    });

});
```

**Листинг 3. – Исходный код простого приложения,  
демонстрирующего некоторые возможности Geolocation API**



**Рисунок 2. – Приложение, демонстрирующее возможности Geolocation API**

**Заключение.** В ходе данного исследования был рассмотрен способ, основанный на чистом JavaScript, для доступа к местонахождению устройства, не полагаясь на какие-либо внешние зависимости или сторонние сервисы. Так же было написано простое веб-приложение, демонстрирующее возможности Geolocation API.

#### ЛИТЕРАТУРА

1. HTML5 Geolocation [Электронный ресурс] – Режим доступа: [https://www.w3schools.com/html/html5\\_geolocation.asp](https://www.w3schools.com/html/html5_geolocation.asp) - Дата доступа: 22.09.2019.
2. Geolocation: Displaying User or Device Position on Maps [Электронный ресурс] – Режим доступа: <https://developers.google.com/maps/documentation/javascript/geolocation> - Дата доступа: 23.09.2019.
3. Geolocation API Removed from Unsecured Origins in Chrome 50 [Электронный ресурс] – Режим доступа: <https://developers.google.com/web/updates/2016/04/geolocation-on-secure-contexts-only> – Дата доступа: 23.09.2019.
4. Introduction: callbacks [Электронный ресурс] – Режим доступа: <https://javascript.info/callbacks> – Дата доступа: 23.09.2019.
5. MDN web docks [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/> - Дата доступа: 23.09.2019.

УДК 004.421

**ПРЕОБРАЗОВАНИЕ РЕЧИ В ТЕКСТ С ПОМОЩЬЮ JAVASCRIPT****Н.И. ЛАПКОВСКИЙ***(Представлено: Д.В. ПЯТКИН)*

*Статья посвящена работе с API Web Speech. Это очень мощный интерфейс браузера, который позволяет записывать человеческую речь и преобразовывать ее в текст. Также будет рассмотрен способ использовать его для обратного – считывания строк человеческим голосом.*

**Конвертирование речи в текст.** Web Speech API фактически разделен на два совершенно независимых интерфейса: SpeechRecognition для понимания человеческого голоса и превращения его в текст и SpeechSynthesis для чтения строк вслух сгенерированным компьютером голосом.

API распознавания речи удивительно точен для бесплатной функции браузера. Он правильно распознал почти все мои разговоры и знал, какие слова объединяются в фразы, которые имеют смысл. Это также позволяет диктовать специальные символы, такие как точки останова, вопросительные знаки и новые строки. Пример представлен в листинге 1 [1].

Первое, что нужно сделать, это проверить, есть ли у пользователя доступ к API и показать соответствующее сообщение об ошибке. К сожалению, API преобразования речи в текст поддерживается только в Chrome и Firefox (с флажком), поэтому многие пользователи, вероятно, увидят это сообщение.

```
try {
  var SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
  var recognition = new SpeechRecognition();
}
catch(e) {
  console.error(e);
  $('#no-browser-support').show();
  $('#app').hide();
}
```

**Листинг 1. – Проверка наличия у пользователя доступа к API**

Переменная recognition даст нам доступ ко всем методам и свойствам API. Существуют различные доступные варианты, но мы только установим для параметра recognition.continuous значение true. Это позволит пользователям говорить с более длинными паузами между словами и фразами.

Прежде чем мы сможем использовать распознавание голоса, мы также должны настроить несколько обработчиков событий. Большинство из них просто слушают изменения в статусе распознавания. Пример представлен в листинге 2.

```
try {
  var SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
  var recognition = new SpeechRecognition();
}
catch(e) {
  console.error(e);
  $('#no-browser-support').show();
  $('#app').hide();
}
```

**Листинг 2 – Настройка обработчиков событий перед использованием распознавания голоса**

Однако есть специальное событие onresult, которое очень важно. Оно выполняется каждый раз, когда пользователь произносит слово или несколько слов в быстрой последовательности, давая нам доступ к текстовой транскрипции сказанного.

Когда мы записываем что-то с помощью обработчика onresult, мы сохраняем его в глобальной переменной и отображаем в текстовой области. Пример представлен в листинге 3.

```
recognition.onresult = function(event) {  
  
    // event это SpeechRecognitionEvent объект.  
    // Он хранит весь записанный диалог  
    // Нам нужен только текущий диалог/фраза  
    var current = event.resultIndex;  
  
    // Получаем текст сказанного.  
    var transcript = event.results[current][0].transcript;  
  
    // Добавляем текст к контенту нашего текстового документа.  
    noteContent += transcript;  
    noteTextarea.val(noteContent);  
}
```

#### Листинг 3. – обработчик события recognition.onresult

Приведенный выше код немного упрощен. Существует очень странная ошибка на устройствах Android, из-за которой все повторяется дважды. Официального решения пока нет, но мне удалось решить проблему без каких-либо явных побочных эффектов. Имея в виду эту ошибку, код выглядит так [2]:

```
var mobileRepeatBug = (current == 1 && transcript ==  
event.results[0][0].transcript);  
  
if(!mobileRepeatBug) {  
    noteContent += transcript;  
    noteTextarea.val(noteContent);  
}
```

#### Листинг 4. – Упрощенный код предыдущего листинга, решающий проблему с повторением выходного текста дважды на устройствах с ОС Android

Всё настроено, теперь можно начать использовать функцию распознавания голоса браузера. Чтобы начать, нужно просто вызвать метод start ():

```
$('#start-record-btn').on('click', function(e) {  
    recognition.start();  
});
```

#### Листинг 5. – Вызов метода start () для начала распознавания голоса

При этом браузер попросит пользователей дать разрешение. Если разрешение будет дано, микрофон устройства будет активирован.

Браузер будет прослушивать некоторое время, и каждая распознанная фраза или слово будут транслироваться в текст. API прекратит прослушивание автоматически через пару секунд тишины или при остановке вручную.

```
$('#pause-record-btn').on('click', function(e) {  
    recognition.stop();  
});
```

#### Листинг 6. – Обработка события нажатия кнопки остановки трансляции речи в текст

**Конвертирование текста в речь.** Синтез речи на самом деле очень прост. API доступен через объект speechSynthesis, и есть несколько методов для воспроизведения, приостановки и других вещей, связанных со звуком. Также есть несколько интересных функций, которые меняют высоту, скорость и даже голос читателя.

Все, что понадобится для демонстрации, это метод speak(). Он ожидает один аргумент — экземпляр класса SpeechSynthesisUtterance.

В листинге 7 представлен весь код, необходимый для считывания строки.

```
function readOutLoud(message) {
    var speech = new SpeechSynthesisUtterance();

    // Устанавливаем атрибуты текста и голоса.
    speech.text = message;
    speech.volume = 1;
    speech.rate = 1;
    speech.pitch = 1;

    window.speechSynthesis.speak(speech);
}
```

**Листинг 7. – Чтение текста вслух голосом, синтезированным компьютером**

**Заключение.** В статье был рассмотрен способ конвертирования человеческой речи в текст средствами API Web Speech. Так же был исследован способ трансляции текста в человеческую речь синтезированным компьютером голосом.

В эпоху, когда голосовые помощники стали популярнее, чем когда-либо, подобный API дает быстрый способ создания ботов, которые понимают и говорят на человеческом языке.

Добавление голосового управления к приложениям также может быть отличной формой улучшения доступности. Пользователи с нарушениями зрения могут пользоваться как пользовательскими интерфейсами «речь в текст», так и «текст в речь».

API синтеза речи и распознавания речи работают довольно хорошо и легко обрабатывают различные языки и акценты. К сожалению, на данный момент они имеют ограниченную поддержку браузеров, что ограничивает их использование в производстве.

#### ЛИТЕРАТУРА

1. Web Speech API – Web APIs | MDN [Электронный ресурс] – Режим доступа: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Speech\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API) – Дата доступа: 24.09.2019.
2. HTML5 Speech Recognition API – codeburst [Электронный ресурс] – Режим доступа: <https://codeburst.io/html5-speech-recognition-api-670846a50e92> - Дата доступа: 24.09.2019.
3. Интерфейсы веб API | MDN [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/API> – Дата доступа: 25.09.2019.
4. Интерфейсные события [Электронный ресурс] – Режим доступа: <https://learn.javascript.ru/event-details> – Дата доступа: 25.09.2019.



УДК 004.221

## РАЗРАБОТКА СИСТЕМЫ ДЛЯ СКРЫТИЯ ИНФОРМАЦИИ В ЗВУКОВОЙ СРЕДЕ

И.С. АНДРЕЕВ

*(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)*

*В данной статье рассматриваются различные алгоритмы сокрытия информации в звуковой среде и их преимущества. В ходе исследований был выбран оптимальный вариант алгоритма.*

**Ключевые слова:** стеганография, сокрытие информации, сигнал, музыкальная стегосистема.

**Введение.** Задача надежной защиты и сокрытия информации идет из далекой глубины веков. Еще в античные времена применяли различные методы защиты важной информации от “посторонних глаз”. Это были скиталы, квадрат Полибия и первые сдвиговые шифры. Позже возникли шифры простой замены (рассказ Артура Конан Дойла “Пляшущие человечки”) и перестановочные шифры (решетка Кардано). В XVIII веке возник шифр “по книге”, который можно рассматривать как развитие шифра Цезаря (таким шифром пользуется герой Ю. Семенова - Штирлиц в романе “17 мгновений весны”).

Но все шифры обладают конечной стойкостью и могут быть расшифрованы за конечный промежуток времени. Решить эту проблему помогает стеганография, которая скрывает сам факт передачи секретного сообщения.

**Основной раздел.** В статье будут рассмотрены стеганографические методы сокрытия информации в звуковых файлах, как с точки зрения устойчивости к атакам, так и с точки зрения сохранения приемлемого качества звукового сигнала.

**Метод наименьших значащих битов** применяется при цифровом представлении аудиосигнала и пригоден для использования при любых скоростях связи. При преобразовании звукового сигнала в цифровую форму всегда присутствует шум дискретизации, который не вносит существенных искажений. “Шумовым” битам соответствуют младшие биты цифрового представления сигнала, которые можно заменить скрываемыми данными. Данный метод обладает чрезвычайно низкой стегостойкостью и простой реализацией. Изменение звукового сигнала можно обнаружить.

**Методы широкополосного кодирования** используют те же принципы, что методы сокрытия данных в изображениях. Их суть заключается в незначительной одновременной модификации целого ряда определенных битов контейнера при сокрытии одного бита информации. Данный метод имеет среднюю устойчивость к атакам и искажениям и сложную в реализации. Звуковой сигнал практически не изменяется.

**Метод сокрытия в эхо-сигнале** скрывает данные путем внедрения эха в звуковой сигнал. Известно, что при небольших временных сдвигах эхо-сигнал практически неразличим на слух. Поэтому, если ввести определенные временные задержки, величина которых не превышает порог обнаружения, то, разбивая исходный звуковой сигнал на сегменты, в каждый из них можно ввести соответствующий эхо-сигнал, в зависимости от скрываемого бита. Данный метод стегоустойчив и сложен в реализации. Звуковой сигнал практически не изменяется.

**Фазовые методы сокрытия** применяются как для аналогового, так и для цифрового сигнала. Они используют тот факт, что плавное изменение фазы на слух определить нельзя. В таких методах защищаемые данные кодируются либо определенным значением фазы, либо изменением фаз в спектре. Изменения в звуковом файле при использовании данного метода невозможно обнаружить с помощью человеческого слуха, но чрезвычайно сложно извлечь информацию при малейшем повреждении сигнала.

**Музыкальные стегосистемы.** Музыкальная форма звуковой среды занимает большую часть информационного пространства Internet. Помимо этого, она широко используется в радиосетях общего назначения и распространяется на электронных носителях информации, которые, в связи с развитием компьютерной техники, получили широкое распространение. В связи с этим использование музыкальной среды для сокрытия информационных сообщений представляется достаточно перспективным. Для сокрытия данных можно применять методы, основанные на модификации тех параметров музыкальной среды, которые в теории музыки можно описать качественно. Музыкальная среда имеет свое текстовое отображение в виде нот и других знаков, которые позволяют достаточно адекватно отображать музыкальное произведение и его внутреннюю структуру такими элементами, как ноты, гаммы, периоды, такты, каденции, аккорды, мотивы, модуляции, тональности, различные виды развития, секвенции и пр. Построения музыкальных фрагментов подчиняются синтаксическим правилам, которые можно описать, что

позволяет строить логические взаимоотношения и, соответственно, описание структур музыкальных произведений.

Музыкальные стегосистемы обеспечивают сокрытие информации в музыкальной среде по аналогии с импровизацией музыкальных произведений. По существу, импровизация представляет собой такое изменение музыкального произведения или его фрагментов, которое сохраняет основные темы первоначального произведения в виде мелодий, но при этом расширяет образ музыкальной темы другими, дополняющими основной образ чертами, которых не было в основном музыкальном произведении. Основное отличие музыкальной стеганографии от импровизации состоит в том, что целью является не расширение образов базового музыкального произведения, а внесение изменений, которые сохраняют мелодию основного произведения, соответствуют всем правилам построения данного произведения и при этом кодируют скрываемое сообщение, не искажая главной темы произведения.

Фрагмент музыкального произведения может быть описан в виде некоторой логической структуры. Аналогом слова текстового предложения в музыкальном произведении будет один такт мелодии, а аналогом предложения в музыке будем считать фрагменты, разделяемые цензурами. Как правило, музыкальное произведение состоит из ряда фраз, которые состоят из тактов. Внедрение текста в музыкальное произведение осуществляется отдельными предложениями, каждое из которых может сопоставляться с отдельной мелодией. Далее формируется нотное отображение расширенного музыкального произведения с внедренным в него скрываемым сообщением. На основании нотного отображения расширения осуществляется его музыкальная реализация с помощью современных компьютерных систем, представляющих собой программно-аппаратные синтезаторы звука. Музыкальная стегосистема обладает высокой стойкостью к атакам и практически невозможностью обнаружения, однако для реализации этой системы необходимо потратить огромное количество ресурсов.

В ходе исследований было обнаружено преимущество фазового метода сокрытия перед другими методами.

**Заключение.** В данной статье были рассмотрены алгоритмы сокрытия данных в звуковых файлах, описана работа музыкальной стегосистемы и сделан вывод об оптимальном методе для реализации системы.

#### ЛИТЕРАТУРА

1. Мельников В.В. Защита информации в компьютерных системах. Электронинформ, 1997. – 368 с.
2. Барабаш А.В., Шанкин Г.П. История криптографии. Ч. 1. – М.: Гелиос АРВ, 2002. – 240 с.
3. Домарев В.В. Защита информации и безопасность компьютерных систем. — К.: ДиаСофт, 1999. – 480 с.
4. Ярочкин В.И. Безопасность информационных систем. – М.: Ось-86, 1996. – 320с.
5. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях / под ред. В.Ф. Шаньгина. – 2-е изд., перераб. и доп. – М.: Радио и связь, 2001. – 376 с.
6. Герасименко В.А., Малюк А.А. Основы защиты информации. – М.: МГИФИ, 1997. – 538 с.

УДК 004.021

**ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА СИСТЕМЫ  
ДЛЯ СКРЫТИЯ ИНФОРМАЦИИ В ЗВУКОВОЙ СРЕДЕ****И.С. АНДРЕЕВ***(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)*

В данной статье рассматривается способ создания графического интерфейса системы для скрытия информации в звуковой среде. Проведён анализ методов проектирования графического интерфейса, наиболее подходящих для разработки приложения. Задача: реализация прототипа интерфейса на основе веб-технологий.

**Ключевые слова:** информационные технологии, скрытие информации, стеганография, алгоритм, Java.

**Введение.** На сегодняшний день, любое приложение должно быть основано на поддерживаемых стандартах разработки и обладать понятным для пользователя интерфейсом. Разрабатываемое приложение должно позволять производить работу с данными пользователя, а также удовлетворять требованиям к предоставляемым данным, по видам информации и её представлению.

**Выбор технологий разработки интерфейса.** Любое приложение должно быть грамотно спроектировано и разделено на отдельные модули, которые должны быть относительно независимыми друг от друга. Подобное разделение значительно облегчает не только реализацию приложения, но и возможную его модификацию. Данное приложение планируется разрабатывать с помощью языка программирования Java и графического фреймворка Swing.

Swing предоставляет более гибкие интерфейсные компоненты, которые разработаны для одинаковой кросс-платформенной работы. Компоненты Swing поддерживают специфические динамически подключаемые виды и поведения) благодаря которому возможна адаптация к графическому интерфейсу платформы (то есть к компоненту можно динамически подключить другой, специфический для операционной системы, в том числе и созданный программистом вид и поведение). Таким образом, приложения, использующие Swing, могут выглядеть как родные приложения для данной операционной системы. Основным минусом таких «легковесных» компонентов является относительно медленная работа. Положительная сторона — универсальность интерфейса созданных приложений на всех платформах.

**Проектирование интерфейса.** Интерфейс приложения состоит из следующих экранов: основной экран, экран выбора файла. Сами элементы графического интерфейса подобраны так, чтобы быть максимально интуитивно понятными и знакомыми большинству пользователей. экран добавления друга. Сами элементы графического интерфейса подобраны так, чтобы быть максимально интуитивно понятными и знакомыми большинству пользователей.

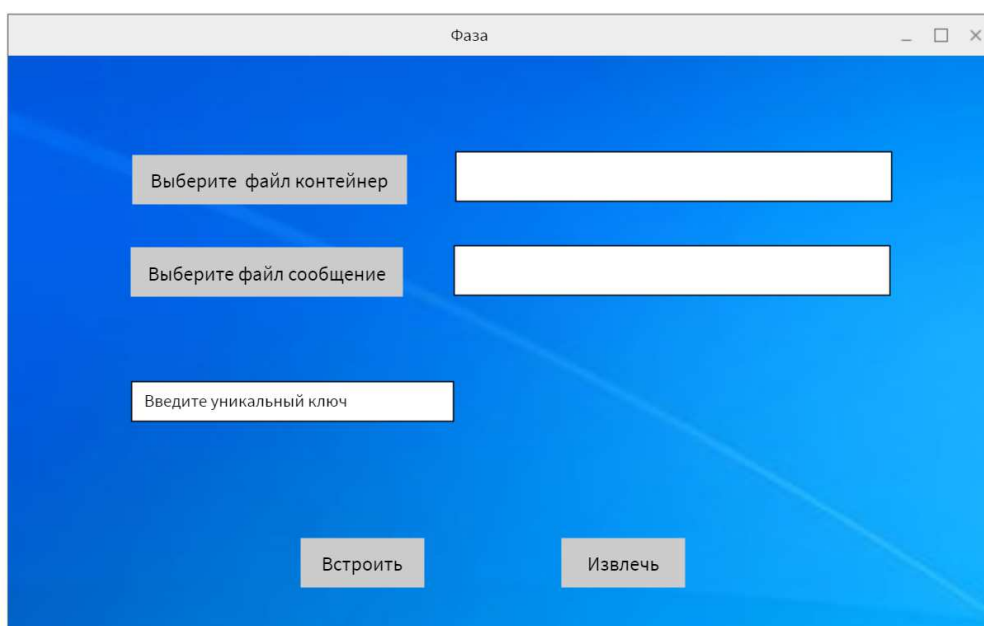


Рисунок. – Основная страница приложения

При запуске приложения, пользователь изначально направляется на основную страницу. На этой странице пользователю предлагается выбрать файл контейнер и файл сообщение из системных папок. Далее необходимо ввести уникальный ключ, без которого невозможно вставка и извлечение информации в файл. Так же на главной странице расположены кнопки “Вставить” и “Извлечь”, которые инициируют процедуры встраивания и извлечения информации из контейнера.

**Заключение.** В данной статье рассмотрен способ построения графического интерфейса системы скрытия информации в звуковой среде. Рассмотрена технология Swing, которая позволяет создать адаптивный интерфейс, любого уровня сложности, при этом сохраняя его функциональные возможности и привлекательный внешний вид.

#### ЛИТЕРАТУРА

1. Стеганографические системы. Критерии и методическое обеспечение: учебно-методическое пособие / В. Г. Грибунин [и др.]. – Саров : РФЯЦ-ВНИИЭФ, 2016. – 323 с.
2. Коржик, В. И. Лекции по основам стеганографии.
3. Шилдт Г. Swing руководство для начинающих.

УДК 004.056.52

## АНАЛИЗ МЕТОДОВ ЗАЩИТЫ ПРИЛОЖЕНИЙ ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ

А.Г. АНДРЕЙЧИКОВ

(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)

*В статье проведён анализ методов защиты программного обеспечения от несанкционированного использования. Рассмотрена задача реализации надёжной системы защиты критически важного программного продукта.*

На сегодняшний день, любое программное обеспечение, работающее с пользовательскими данными должно быть защищено от несанкционированного использования. Приложение, а также информация, обрабатываемая в приложении должны быть защищены. Существуют программные продукты, получение возможного доступа к которым может нанести существенные финансовые убытки или допустить утечку данных пользователей. Утечка данных пользователей может дать предполагаемому злоумышленнику данные для последующей кражи финансовых средств данного пользователя.

Защита программного обеспечения — комплекс мер, направленных на защиту программного обеспечения от несанкционированного приобретения, использования, распространения, модифицирования, изучения и воссоздания аналогов.

Защита от несанкционированного использования программ — система мер, направленных на противодействие нелегальному использованию программного обеспечения. При защите могут применяться организационные, юридические, программные и программно-аппаратные средства.

Защита от копирования к программному обеспечению применяется редко, в связи с необходимостью его распространения и установки на компьютеры пользователей. Однако, от копирования может защищаться лицензия на приложение (при распространении на физическом носителе) или его отдельные алгоритмы. [1]

Методы можно классифицировать по способу распространения защищаемого программного обеспечения и типу носителя лицензии.

**Локальная программная защита.** Требование ввода серийного номера (ключа) при установке/запуске. История этого метода началась тогда, когда приложения распространялись только на физических носителях (к примеру, компакт-дисках). На коробке с диском был напечатан серийный номер, подходящий только к данной копии программы. Недостаток данного метода в простоте обхода защиты - можно беспрепятственно передавать серийный номер, а компакт-диск эмулировать. Как метод защиты в текущих реалиях практически бесполезен, но может использоваться в совокупности с другими методами защиты.

**Сетевая программная защита.** Локальная разновидность защиты предполагает сканирование сети что исключает одновременный запуск двух программ с одним регистрационным ключом на двух компьютерах в пределах одной локальной сети. Данный метод можно обойти, имея небольшие знания настройки брандмауэра. Если заблокировать программе доступ к локальной сети или отфильтровать исходящие пакеты можно избежать блокировки. Глобальная разновидность защиты использует централизованный сервер для контроля доступа проверяя на этапе соединения регистрационный ключ или другую информацию исключающую неавторизованный доступ. Метод обхода данного метода защиты заключается в развертывании поддельного сервера, исключающего авторизацию или авторизирующей по любым пользователям. Таким образом, получив копию оригинального сервера можно полностью проигнорировать защиту и получить доступ к программному продукту.

**Защита при помощи электронных ключей.** Электронный ключ — вставленный в один из портов компьютера (с интерфейсом USB, LPT или COM) содержит ключевые данные, называемые также лицензией, записанные в него разработчиком. В настоящее время из-за сложности похищения данного ключа является хорошим методом защиты при условии, что используется самый современный вариант ключа, и он защищён от посторонних лиц.

**Защита программ от копирования путём переноса их в сеть Интернет.** Стремительно набирающий популярность метод защиты, который заключается в предоставлении функционала программ (всего или части), как сервиса онлайн, в сети Интернет. При этом код программы расположен и исполняется на сервере, доступном в глобальной сети

**Привязка к параметрам компьютера и активация.** Механизм защиты прост, при установке программа генерирует ключ и при запуске считывает оборудование и опять генерирует ключ, проверяя соответствии с первоначальным ключом при каждом запуске. При смене конфигурации компьютера такой вид защиты приводит к неработоспособности ПО.

**Защита кода от анализа.** Отдельный вид защиты предполагает использования методов запутывания кода программы для противодействия копированию или модификации. Метод затрудняет, но не исключает возможности разобраться в коде, защищенном таким методом.

**Защита с помощью простой аутентификации.** Классическая защита доступа к приложению посредством ввода логина и пароля пользователя которые предоставляются пользователю по закрытым каналам связи при условии зашифрованного хранения в самом приложении может достаточно хорошо защитить приложения от несанкционированного использования. Устойчивость данного метода защиты зависит от устойчивости логина и пароля к подбору, а также от безопасности каналов передачи. При соблюдении указанных данных метод может быть использован для реализации эффективной защиты.

**Защита посредством сеансовых ключей.** Метод используется совместно с защитой простой аутентификацией. После ввода логина и пароля пользователю необходимо ввести сеансовый ключ. Сеансовый ключ может находиться на пластиковой карте или предоставляться пользователю посредством sms. Такой метод защиты традиционно используется в банковской сфере при совершении платёжных операций. Однако такой метод может использоваться и для других программных решений. Достаточно новый метод доставки сеансовых ключей заключается в обращении к сторонним сервисам, которые предоставляют сеансовые ключи меняя их через определённый интервал времени.

**Защита с помощью биометрических систем.** Биометрические системы аутентификации — системы аутентификации, использующие для удостоверения личности людей их биометрические данные.

Биометрическая аутентификация — процесс доказательства и проверки подлинности заявленного пользователем имени, через предъявление пользователем своего биометрического образа и путём преобразования этого образа в соответствии с заранее определённым протоколом аутентификации.

Не следует путать данные системы с системами биометрической идентификации, каковыми являются, к примеру системы распознавания лиц водителей и биометрические средства учёта рабочего времени. Биометрические системы аутентификации работают в активном, а не пассивном режиме и почти всегда подразумевают авторизацию. Хотя данные системы не идентичны системам авторизации, они часто используются совместно. [2]

Для защиты доступа к приложениям можно использовать аутентификацию по лицу, голосу поскольку практически все современные ноутбуки и телефоны оснащены камерами и микрофонами высокого качества. К сожалению такую защиту можно взломать, зачастую не приложив никаких усилий. Многие системы распознавания ошибочно могут разрешить доступ по фотографии. Для обхода аутентификации по голосу можно использовать высококачественную запись. Несмотря на это биометрические системы осуществляют хороший уровень защиты совместно с другими методами защиты.

**Эффективная защита.** Основной принцип эффективной защиты современного программного обеспечения основан комбинировании методов защиты. Чем больше методов защиты используется, тем сложнее получить доступ. Однако использование большого количества методов защиты сказывается на удобстве пользователей и зачастую приходится искать компромисс между защитой и удобством.

**Проблемы современных методов защиты.** Из-за сильной зависимости от финансовой составляющей многие разработчики программного обеспечения не могут разработать собственную систему защиты и прибегают к сторонним разработкам что в свою очередь приводит к шаблонности систем защиты. Следствием такого является упрощения взлома.

В зависимости от сферы применения программного обеспечения. В игровой индустрии в последнее время чаще всего используется связка привязки аккаунта с ключом к компьютеру. Таким образом при изменении конфигурации компьютера пользователю нужно будет только подтвердить конфигурации временным ключом, присланным на email или телефон.

В сфере финансов чаще всего распространена защита посредством электронных ключей совместно с проверкой компьютера пользователя по базе. Принцип работы прост в начале работы с программой пользователь авторизуется с помощью электронного ключа далее проверяется конфигурация его компьютера и сверяется с базой если компьютер не найден соединения блокируется.

В данной статье представлен обзор основных методов защиты от несанкционированного использования. Рассмотрены преимущества и недостатки также представлены комбинации методов, которые используются в современном программном обеспечении. На основании представленной информации в качестве системы защиты была выбрана комбинация серийного номера и привязки к параметрам компьютера.

## ЛИТЕРАТУРА

1. Защита программного обеспечения [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Защита\\_программного\\_обеспечения](https://ru.wikipedia.org/wiki/Защита_программного_обеспечения). – Дата доступа: 23.09.2019.
2. Биометрические системы аутентификации [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Биометрические\\_системы\\_аутентификации](https://ru.wikipedia.org/wiki/Биометрические_системы_аутентификации). – Дата доступа: 23.09.2019.
3. Защита ПО от копирования и взлома: основные методы и стратегии [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/skillbox/blog/440836/>. – Дата доступа: 23.09.2019.
4. Защита от несанкционированного копирования [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Защита\\_от\\_несанкционированного\\_копирования](https://ru.wikipedia.org/wiki/Защита_от_несанкционированного_копирования). – Дата доступа: 23.09.2019.

УДК 004.89

**НЕЙРОННЫЕ СЕТИ. СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ****А.Г. АНДРЕЙЧИКОВ***(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)*

*Разобрано понятие искусственной нейронной сети и принципы ее работы. Рассмотрены некоторые задачи использования нейронных сетей, а также нейронные сети, используемые при решении этих задач. Рассмотрены сверточные нейронные сети.*

Человеческий мозг является сложным и интеллектуальным “компьютером”. Взяв за основу принцип образования нейронных связей в мозге человека, учеными была создана новая модель программирования – искусственная нейронная сеть, имеющая способность к самообучению. С помощью искусственных нейронных сетей удалось решить задачи, которые невозможно решить с помощью обычных алгоритмов. В последнее несколько лет нейронные сети достигли потрясающих результатов. Стало возможно создавать картины копируя оригинальный стиль художника.

Нейронные сети помогают решать задачи распознавания образов, сложные прогнозы, а также определение закономерностей на основании анализа большого количества данных. В решении такого рода задач даже безграничные вычислительные мощности компьютеров не будут эффективны человек же справляется с такого рода задачами очень легко. Именно этот факт и повлиял на появление нейронных сетей.

Нейронные сети (искусственная нейронная сеть) — это система соединенных и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, эти процессоры вместе способны выполнять довольно сложные задачи, поскольку нейронные сети обучаются в процессе работы. [1]

Все задачи, которые могут решать нейронные сети, так или иначе связаны с обучением. Среди основных областей применения нейронных сетей — прогнозирование, принятие решений, распознавание, оптимизация, анализ данных.

Задача прогнозирования актуальна в сфере финансов, разработке роботов, систем управления машинами. Точность прогноза нейронной сети зависит от распределения данных для обучения и от их количества. Для решения задач, основанных на прогнозировании, используются рекуррентные нейронные сети.

Если остановиться на рекуррентных ИНС немного подробнее, то выяснится, что наиболее современные из них берут начало из структуры, называемой многослойным перцептроном. При этом со времен своего появления они претерпели значительные изменения – и ИНС «нового поколения» устроены гораздо проще своих предшественников, при том, что они позволяют с успехом решать задачи запоминания последовательностей. Так, к примеру, наиболее популярная на сегодняшний день сеть Элмана устроена таким образом, что обратный сигнал с внутреннего слоя поступает не на «главные» входные нейроны, а на дополнительные входы – так называемый контекст. Эти нейроны хранят информацию о предыдущем входном векторе (стимуле); получается, что выходной сигнал (реакция сети) зависит не только от текущего стимула, но и от предыдущего. [2]

Задача оптимизация чаще всего возникает в промышленности. Классический пример задачи минимизации расходов. Под оптимизацией понимают процесс выбора наилучшего варианта из всех возможных. Больше количество задач сводятся к задачам оптимизации: безусловная оптимизация нелинейных функций; метод наименьших квадратов; решение нелинейных уравнений; линейное программирование; квадратичное программирование; условная минимизация нелинейных функций; многокритериальная оптимизация. Задач оптимизации решается с помощью нейронной сети Хопфилда.

Нейронная сеть Хопфилда — полносвязная нейронная сеть с симметричной матрицей связей. В процессе работы динамика таких сетей сходится к одному из положений равновесия. Эти положения равновесия определяются заранее в процессе обучения, они являются локальными минимумами функционала, называемого энергией сети (в простейшем случае — локальными минимумами отрицательно определённой квадратичной формы на  $n$ -мерном кубе). Такая сеть может быть использована как ассоциативная память, как фильтр, а также для решения некоторых задач оптимизации. В отличие от многих нейронных сетей, работающих до получения ответа через определённое количество тактов, сети



Хопфилда работают до достижения равновесия, когда следующее состояние сети в точности равно предыдущему: начальное состояние является входным образом, а при равновесии получают выходной образ [3].

Задача распознавания включает в себя множество разновидностей более локальных задач таких как распознавание объектов, эмоций, лица, образов. Основной нейронной сетью, которая отвечает за решение подобных задач является сверточная нейронная сеть. Распознавание лиц используется в поиске людей, безопасности. Распознавание эмоций используется при разработке роботов, которые способны опознавая эмоции человека, соответствующие реагировать. Что касается распознавания объектов его сфера применения промышленность, поиск явлений, в системе перемещения Дронов и роботов. Система распознавания используется, например, в обнаружении пожаров, поиске животных, распознавании текста для последующей обработки.

Свёрточная нейронная сеть — специальная архитектура искусственных нейронных сетей, предложенная Яном Лекунем в 1988 году и нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения. Использует некоторые особенности зрительной коры, в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток. Таким образом, идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв и субдискретизирующих слоёв. Структура сети — однонаправленная (без обратных связей), принципиально многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов — любая, по выбору исследователя.

Работа свёрточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям, и далее к ещё более абстрактным деталям вплоть до выделения понятий высокого уровня. При этом сеть самонастраивается и вырабатывает сама необходимую иерархию абстрактных признаков, фильтруя маловажные детали и выделяя существенное.

Подобная интерпретация носит скорее метафорический или иллюстративный характер. Фактически «признаки», вырабатываемые сложной сетью, малопонятны и трудны для интерпретации настолько, что в практических системах не особенно рекомендуется пытаться понять содержания этих признаков или пытаться их «подправить», вместо этого рекомендуется усовершенствовать саму структуру и архитектуру сети, чтобы получить лучшие результаты. Так, игнорирование системой каких-то существенных явлений может говорить о том, что-либо не хватает данных для обучения, либо структура сети обладает недостатками, и система не может выработать эффективных признаков для данных явлений.

В обычном перцептроне, который представляет собой полносвязную нейронную сеть, каждый нейрон связан со всеми нейронами предыдущего слоя, причём каждая связь имеет свой персональный весовой коэффициент. В свёрточной нейронной сети в операции свёртки используется лишь ограниченная матрица весов небольшого размера, которую «двигают» по всему обрабатываемому слою (в самом начале — непосредственно по входному изображению), формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. То есть для различных нейронов выходного слоя используются одна и та же матрица весов, которую также называют ядром свёртки. Её интерпретируют как графическое кодирование какого-либо признака, например, наличие наклонной линии под определённым углом. Тогда следующий слой, получившийся в результате операции свёртки такой матрицей весов, показывает наличие данного признака в обрабатываемом слое и её координаты, формируя так называемую карту признаков. Естественно, в свёрточной нейронной сети набор весов не один, а целая гамма, кодирующая элементы изображения. При этом такие ядра свёртки не закладываются исследователем заранее, а формируются самостоятельно путём обучения сети классическим методом обратного распространения ошибки. Проход каждым набором весов формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной (много независимых карт признаков на одном слое). Также следует отметить, что при переборе слоя матрицей весов её передвигают обычно не на полный шаг (размер этой матрицы), а на небольшое расстояние. Так, например, при размерности матрицы весов  $5 \times 5$  её сдвигают на один или два нейрона вместо пяти, чтобы не «перешагнуть» искомый признак.

Операция субдискретизации, выполняет уменьшение размерности сформированных карт признаков. В данной архитектуре сети считается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотнённой карты признаков меньшей размерности. За счёт данной операции, помимо ускорения дальнейших вычислений, сеть становится более инвариантной к масштабу входного изображения.

Рассмотрим типовую структуру свёрточной нейронной сети более подробно. Сеть состоит из большого количества слоёв. После начального слоя (входного изображения) сигнал проходит серию свёрточных слоёв, в которых чередуется собственно свёртка и субдискретизация. Чередование слоёв

позволяет составлять «карты признаков» из карт признаков, на каждом следующем слое карта уменьшается в размере, но увеличивается количество каналов. На практике это означает способность распознавания сложных иерархий признаков. Обычно после прохождения нескольких слоёв карта признаков выродается в вектор или даже скаляр, но таких карт признаков становятся сотни. На выходе свёрточных слоёв сети дополнительно устанавливают несколько слоёв полносвязной нейронной сети, на вход которому подаются оконечные карты признаков. [4] Преимущества сверточных нейронных сетей перед полносвязными нейронными сетями заключается в гораздо меньшее количество настраиваемых весов. Высокая скорость работы за счёт возможности реализации работы и обучения на графических процессорах. Однако такая модель содержит такая нейронная сеть сложна в обучении и требует точной настройки и добавления слоёв субдискретизации для повышения быстродействия.

В ходе данного исследования были рассмотрены нейронные сети, а также сферы и их применения. Представлены примеры реальных задач, которые ставятся перед нейронными сетями. Подробно рассмотрена задача распознавания и используемая в этой задаче свёрточная нейронная сеть.

#### ЛИТЕРАТУРА

1. Нейронные сети [Электронный ресурс]. – Режим доступа: <https://hi-news.ru/tag/nejronnye-seti>. – Дата доступа: 25.09.2019.
2. Прогнозирование при помощи рекуррентных нейронных сетей [Электронный ресурс]. – Режим доступа: <http://datareview.info/article/prognozirovanie-pri-pomoshhi-rekurrentnyih-neyronnyih-setey/>. – Дата доступа: 25.09.2019.
3. Нейронная сеть Хопфилда [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Нейронная\\_сеть\\_Хопфилда](https://ru.wikipedia.org/wiki/Нейронная_сеть_Хопфилда). Дата доступа: 25.09.2019.
4. Нейронная сеть Хопфилда [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Свёрточная\\_нейронная\\_сеть](https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть). Дата доступа: 25.09.2019.
5. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление / Б.А. Алпатов [и др.] – М. : Радиотехника, 2008. – 176 с.
6. Ngan, K. Video Segmentation and Its Applications / K. Ngan, H. Li // Springer, 2011. – 165 p.
7. Библиотека компьютерного зрения OpenCV [Электронный ресурс]. – Режим доступа: <http://opencv.org/>. – Дата доступа: 25.09.2019.

УДК 004.4'234

## КАК УСТРОЕН HEADLESS CHROME

М.А. БАЛАБАШ

(Представлено: канд. физ.-мат. наук, доц. А.Ф. ОСЬКИН)

В статье дается определение headless веб-браузерам и способы их применения. Рассматриваются основные компоненты Headless Chrome веб-браузера и протокол взаимодействия.

**Введение.** Headless браузер это незаменимый инструмент front-end разработчика, с помощью которого можно тестировать код, проверять соответствие верстке и множество других возможных применений.

Headless браузеры обеспечивают автоматизированное управление веб-страницей в среде, аналогичной популярным веб-браузерам. Они предоставляют специальные программные интерфейсы для взаимодействия со средой выполнения и позволяют управлять веб-браузером, исполнять код в его среде и получать любую доступную информацию из контекста веб-браузера.

Chromium — это открытая реализация браузера на основе WebKit. Google Chrome — это брендированная версия Chromium, у которой есть проприетарные кодеки, интеграции с сервисами Google и др. Headless Chrome - это headless версия браузера Google Chrome. Именно её мы и будем исследовать.

В качестве интерфейса для управления (API) используется Chrome DevTools protocol. Chrome DevTools - это набор инструментов для веб-разработчиков, встроенный непосредственно в браузер Google Chrome.

**Компоненты Chromium.** Самый низкий уровень Platform layer.

- Ozone — абстрактный менеджер окон в Chrome — то, с чем взаимодействует оконный менеджер операционной системы. На Linux это либо X-server, либо Wayland. На Windows это менеджер окон Windows;

- Scheduler — тот самый планировщик, который занимается синхронизацией потоков и процессов, т.к. Chrome — это многопроцессное приложение;

- Net — компонент для работы с сетью, который парсинг HTTP, создает заголовки и т.д.

Уровень Content layer — самый большой компонент, который есть в Chrome.

- Blink — веб-движок на основе WebCore из WebKit для работы с HTML и CSS; V8 (JavaScript движок); API для всех расширений, которые мы используем в Chrome. Туда же входит DevTools protocol;

- Content API — это интерфейс, при помощи которого можно очень легко использовать все возможности веб-движка. Поскольку внутри Blink много всего (несколько сотен тысяч интерфейсов), то, чтобы не потеряться во всех этих методах и функциях, нужен Content API.

Уровень безголового браузера (Headless layer).

- Headless library;

- Embedder API интерфейс для встраивания Headless library в приложение;

- Client API — интерфейс, который использует Puppeteer.

Уровень приложения (Application layer).

- Ваше приложение (Embedding app);

- Мини-приложения, например, Headless shell.

**Chrome DevTools protocol**

Все front-end разработчики (и не только) сталкивались с Chrome DevTools protocol, поскольку пользовались панелью разработчиков в Chrome или удаленным отладчиком — теми же самыми средствами разработки. Если вы запускаете средства разработчика в удаленном режиме, общение с браузером происходит при помощи именно DevTools protocol. Когда вы ставите debugger, смотрите code coverage, используете геолокацию или еще что-нибудь — все это управляется при помощи DevTools.

В протоколе есть 2 компонента:

- DevTools target — вкладка, которую вы инспектируете;

- DevTools client — например, это панель разработчика, которая запущена в удаленном режиме.

Они общаются при помощи простого JSON:

- Есть идентификатор команды, название метода, который нужно выполнить, и некоторые параметры;

- Ответ тоже выглядит очень просто: идентификатор, который нужен потому, что все команды, которые выполняются при помощи протокола, асинхронные. Для того, чтобы мы всегда могли сопоставить, какой ответ на какую команду нам пришел, нам и требуется идентификатор;

- Результат.

**Puppeteer.** Это библиотека (доступная для нескольких языков программирования), которая предоставляет API высокого уровня для управления Chrome или Chromium по протоколу DevTools. На текущий момент это лучшее решение для работы с Headless Chrome браузером. Она активно развивается и поддерживается командой из Google.

Большинство вещей, которые вы можете сделать вручную в браузере, можно сделать с помощью Puppeteer.

Несколько примеров использования:

- Генерация скриншотов и PDF-файлов страниц;
- Сканирование SPA (одностраничное приложение) и генерация предварительно визуализированного контента (рендеринг на стороне сервера);
- Автоматизация отправки форм, тестирование пользовательского интерфейса, ввод с клавиатуры и т.д.;
- Замерять и диагностировать показатели производительности;
- Перехват сетевых запросов / ответов;
- Тестирование расширений Chrome.

При первой установке Puppeteer скачивает версию Chromium и устанавливает ее именно под вашу архитектуру и ОС. Вам не нужно ничего скачивать дополнительно, он делает это автоматически. Вы также можете использовать и вашу любимую версию Chrome, которая установлена у вас в системе.

**Заключение.** Headless Chrome — инструмент, позволяющий тестировать наши веб-приложения и автоматизировать взаимодействия с ними. DevTools protocol имеет богатый набор возможностей, который позволяет разрабатывать полноценные и самостоятельные программные решения для широкого спектра задач (мониторинг веб-приложений, среда разработки VS Code). Возможность получать информацию, исполнять код и автоматизировать действия браузера — открывают много возможностей.

#### ЛИТЕРАТУРА

1. GitHub [Электронный ресурс] Headless Chrome Node.js API. Режим доступа: <https://github.com/GoogleChrome/puppeteer>. Дата доступа: 27.09.19.
2. ChromeDevtools [Электронный ресурс] Chrome DevTools Protocol Viewer. Режим доступа: <https://chromedevtools.github.io/devtools-protocol/>. Дата доступа: 27.09.19.
3. Chromium [Электронный ресурс] The Chromium Projects. Режим доступа: <https://www.chromium.org/>. Дата доступа: 27.09.19.

УДК 004.75

## SOCKJS ДЛЯ АСИНХРОННОГО ОБМЕНА ДАННЫМИ В РЕАЛЬНОМ ВРЕМЕНИ

М.А. БАЛАБАШ

(Представлено: канд. физ.-мат. наук, доц. А.Ф. ОСЬКИН)

В данной статье рассматривается библиотека с открытым исходным *sockjs*. Библиотека предоставляет согласованный кросс-браузерный *Javascript* API, который создает полнодуплексный междоменный канал связи с малой задержкой между браузером и веб-сервером для обмена данными в реальном времени.

**Введение.** Интернет создавался главным образом на основе так называемой парадигмы запросов и ответов, реализованной с помощью протокола HTTP. Она заключается в том, что пока клиент не отправит запрос на открытие следующей страницы, она не загрузится. Примерно с 2005 г. с появлением технологии AJAX работа в Интернете стала более динамичной. При этом обмен данными по протоколу HTTP все равно инициировался клиентом, и это требовало от пользователя выполнения определенных действий или периодического опроса сервера для загрузки обновленной информации.

Технологии, позволяющие серверу отправлять клиенту новые данные в момент их появления, существуют довольно давно. Они известны как *Push* или *Comet*. Один из распространенных способов создания иллюзии соединения, инициируемого сервером, – это так называемый метод длинного опроса. Его суть в том, что клиент создает подключение к HTTP-серверу и оно не закрывается до отправки ответа. Если на сервере действительно есть обновленные данные, он отправляет ответ (в других технологиях для этого используются *Flash*, *multipart*-запросы *XHR* и особые *HTML*-файлы). Технология длинного опроса, как и другие подобные приемы, прекрасно работает. Вы пользуетесь ими каждый день, например, в чате *Gmail*.

Однако у этого метода есть один недостаток: он использует протокол HTTP, что плохо подходит для приложений с низким временем реакции. В частности, я имею в виду финансовые приложения с непрерывным потоком данных (биржи) и другие сетевые приложения в реальном времени.

**Основной раздел.** *WebSocket* — протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени.

*WebSocket* разработан для реализации в веб-браузерах и веб-серверах, но он может быть использован для любого клиентского или серверного приложения. Протокол *WebSocket* — это независимый протокол, основанный на протоколе TCP. Он делает возможным более тесное взаимодействие между браузером и веб-сайтом, способствуя распространению интерактивного содержимого и созданию приложений реального времени.

Под капотом *SockJS* всегда пытается использовать нативные *WebSockets*. Если это не удастся, он может использовать различные транспортные протоколы для конкретного браузера и представлять их в виде абстракций, подобных *WebSocket*.

Основными сферами применения технологии являются:

- Быстрый обмен данными для онлайн игр, чатов и пр.;
- Приложения с интенсивным обменом данными, требовательные к скорости обмена и каналу;
- *Push* уведомления;
- IoT-приложения;
- Удалённый доступ.

Преимущества данного подхода:

1. Высокую скорость и эффективность передачи обеспечивает малый размер передаваемых данных, который иногда даже будет помещаться в один TCP-пакет — здесь, конечно, же все зависит от вашей бизнес-логики.

2. В отличие от HTTP веб-сокеты не имеют ограничений на время жизни в неактивном состоянии. Это значит, что больше не надо периодически обновлять соединение, т.к. его не вправе закрывать всякие прокси-серверы. Значит, соединение может висеть в неактивном виде и не требовать ресурсов. Конечно, можно возразить, что на сервере будут забиваться TCP-сокеты. Для этого достаточно использовать хороший мультиплексор, и сервер легко потянет до миллиона открытых соединений.

3. Как известно в HTTP предусмотрено ограничение на число одновременных открытых сессий к одному серверу. Из-за этого если у вас много различных асинхронных блоков на странице, то вам приходилось делать не только серверный, но и клиентский мультиплексор. К счастью, это ограничение не распространяется на веб-сокеты. Открываете столько, сколько вам нужно. А сколько использовать — одно (и через него все мультиплексировать) или же наоборот — на каждый блок свое соединение — решать вам. Исходите из удобства разработки, нагрузки на сервер и клиент.

4. Возможность разрабатывать кросс-доменные приложения. AJAX имеет ряд ограничений на создание кросс-доменных запросов. WebSockets не имеет таких ограничений.

Sockjs подходит для использования на клиенте и сервере. Для работы с технологией WebSockets на сервере существует множество реализаций socks на других языках программирования (node.js, python, erlang, java, scala, rust, go, ruby).

Протоколом проводной связи (установления соединения и обмена данными между клиентом и сервером) для технологии WebSocket является RFC6455. Последние версии Chrome и Chrome для Android полностью соответствуют спецификации RFC6455, в том числе и в части отправки двоичных сообщений. Этот протокол также поддерживают браузеры Firefox 11+ и Internet Explorer 10+.

Работа с библиотекой не представляет большой сложности.

Пример создания сервера на node.js:

```
const http = require('http');
const sockjs = require('sockjs');
const echo = sockjs.createServer({ prefix: '/echo' });

echo.on('connection', function(conn) {
  conn.on('data', function(message) {
    conn.write(message);
  });
  conn.on('close', function() {});
});

const server = http.createServer();
echo.attach(server);
server.listen(9999, '0.0.0.0');
```

На клиенте, после загрузки библиотеки, необходимо создать экземпляр SockJS объекта с которым мы будем работать. Доступные параметры:

- server — строка URL для фактического подключения к данным;
- transports — позволяет задать реализации транспорта, которые будут использованы если WebSockets не доступен;
- sessionId — идентификатор сеанса для клиента и сервера;
- timeout — минимальный таймаут в миллисекундах для использования в реализациях транспорта.

После создания объекта нам доступны следующие события для обработки:

- onopen — при установке соединения с сервером;
- onmessage — при получении сообщения от сервера;
- onclose — после закрытия соединения с сервером.

Дополнительно можно отметить, что соединение может устанавливаться между клиентами (которые поддерживают WebSockets) без участия сервера.

**Заключение.** WebSockets — технология, которая позволяет создавать интерактивное соединение между клиентом и сервером для обмена сообщениями в режиме реального времени. Веб-сокеты, в отличие от HTTP, позволяют работать с двунаправленным потоком данных, что делает эту технологию совершенно уникальной.

#### ЛИТЕРАТУРА

1. Wikipedia [Электронный ресурс] WebSocket. – Режим доступа: <https://en.wikipedia.org/wiki/WebSocket>. – Дата доступа: 27.09.19.
2. IETF Documents [Электронный ресурс] The WebSocket Protocol. – Режим доступа: <https://tools.ietf.org/html/rfc6455>. – Дата доступа: 27.09.19.
3. Github [Электронный ресурс] sockjs-client. – Режим доступа: <https://github.com/sockjs/sockjs-client>. – Дата доступа: 27.09.19.

УДК 004.5; 004.514

**РОЛЬ ПОРТФОЛИО В ПОВЫШЕНИИ  
КОНКУРЕНТОСПОСОБНОСТИ ВЫПУСКНИКА ВУЗА****Г.Е. БЕЛЬСКИЙ***(Представлено: канд. физ.-мат. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассматривается роль составления личного портфолио и его влияние на повышение конкурентоспособности выпускника вуза при будущем устройстве на работу.*

Республика Беларусь 15 мая 2015 года на Конференции министров образования государств участников Европейского пространства высшего образования (г. Ереван) стала 48-м членом Болонского процесса – процесса сближения и гармонизации систем образования стран Европы с целью создания единого европейского пространства высшего образования.

Приоритетными задачами болонского процесса являются создание европейского пространства высшего образования, призванного стимулировать конкурентоспособность системы высшего образования Европы и ее привлекательность, а также создавать возможности для мобильности студентов и их дальнейшего трудоустройства [4].

Данный процесс ставит перед любым вузом (в том числе и учреждением образования «Полоцкий государственный университет») важнейшую задачу, подготовки конкурентоспособного и востребованного специалиста на рынке труда.

Как показывает практика, в ряде стран отмечается снижение качества образования на первой ступени в связи с введением сокращенных сроков обучения. Многие выпускники не соответствуют в полной мере требованиям, которые предъявляются работодателями: вузы выпускают специалистов с общими представлениями об профессиональной деятельности, в большинстве случаев не готовых к оперативному включению в данную область деятельности [1].

В Республике Беларусь массовый переход на дифференцированные сроки подготовки специалистов с высшим образованием осуществлен с 1 сентября 2008 г., однако реализован он одновременно с оптимизацией учебно-программной документации, регламентирующей образовательный процесс, и введением образовательных стандартов нового поколения, реализующих компетентный подход в подготовке специалистов, что позволило избежать снижения качества подготовки специалистов [5].

Сегодня в условиях реализации новых образовательных стандартов вуз должен обеспечить не только процесс развития компетенций специалиста, но и отслеживание и предъявление результатов и достижений его деятельности будущему работодателю. Одной из современных технологий подготовки студента к будущей профессиональной деятельности, позволяющей ему эффективно планировать и оценивать процесс и результаты своего обучения, является *технология составления портфолио*, стремительно развивающаяся в зарубежном высшем образовании.

«Портфолио» в широком смысле слова – это способ фиксирования, накопления и оценки индивидуальных достижений студента в определенный период его обучения. Портфолио активно применяется в зарубежных системах образования, относящих его к разряду «аутентичных» индивидуализированных оценок, ориентированных на новые формы оценивания. Оно позволяет учитывать результаты, достигнутые студентом в разнообразных видах деятельности: учебной, творческой, социальной, коммуникативной и другой, а также является важным элементом практико-ориентированного подхода к образованию. В зарубежной традиции портфолио определяется как «коллекция работ и результатов учащегося, которая демонстрирует его усилия, прогресс и достижения в различных областях» [3].

Портфолио является одним из условий повышения мотивации студента, формирования навыков рефлексии. По мнению Е. С. Полат, портфолио – это инструмент самооценки собственного познавательного, творческого труда обучающегося, рефлексии его собственной деятельности [2].

Портфолио может быть в бумажном или электронном варианте. Электронная версия может быть представлена в виде презентации на сайте (веб-портфолио). Таким образом, в системе подготовки будущих высококвалифицированных кадров, разработка веб-сервиса для размещения онлайн портфолио студентов вуза обуславливает актуальность представленной работы.

Именно поэтому разработанный веб сервис ставит важный акцент на его соответствии всем текущим стандартам качества. К примеру, можно привести разработанную базу данных, представленную на рисунке 1.

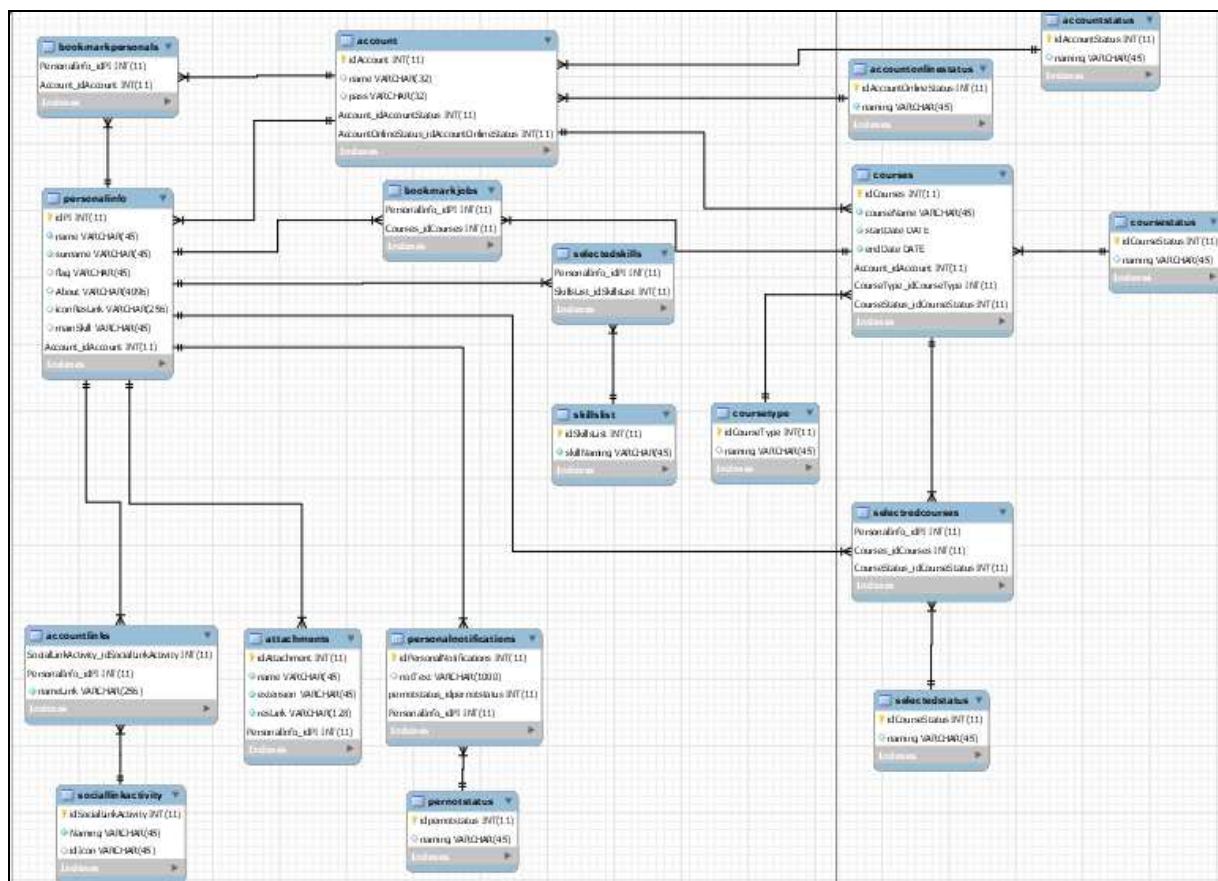


Рисунок 1. – Разработанная база данных проекта

Соответствие стандартам качества очень важно при разработке правильной базы данных, потому что именно она является сердцем приложения. Разработанная в рамках проекта база данных проста в обновлении и поддержке. Языком базы данных был выбран язык СУБД MySQL который обеспечивает высокое быстродействие при большом накоплении данных, что полностью подходит концепции разработанного веб сервиса, так как он будет оперировать большим количеством пользовательской информации. Все данные максимально независимы друг от друга, что помогает поддерживать быстродействие работы с многими одновременно работающими с продуктом пользователями. Все данные защищены на уровне базы данных и программно, гарантируя защиту от непреднамеренного нарушения секретности, искажения или разрушения пользовательской информации.

Сам по себе веб сервис составления портфолио несет в себе огромное количество положительных функций, среди которых следует выделить обеспечение более глубокой вовлеченности студента в научную и самообразовательную среду, более детальное знакомство с которой будет являться первым шагом студента на пути к его будущему успеху. Более детально основные функции использования веб сервиса представлены на рисунке 2.

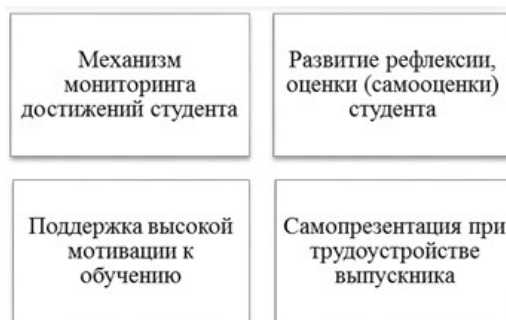


Рисунок 2. – Основные функции разработанного приложения



Структура портфолио студента вуза строится на основе нормативных документов конкретного вуза (Миссии, политики) и в соответствии с требованиями Министерства образования.

Анализ нормативных документов учреждения образования «Полоцкий государственный университет» позволило определить следующую структуру для организации электронного портфолио студента.

Портфолио разработанного веб сервиса поддерживает разделы, отражающие формирование общекультурных и профессиональных компетенций, храня достижения и результаты по овладению студентом профессиональных знаний, умений и навыков; умения проектировать свой профессиональный рост. Содержание разделов отражает достижения в учебно-профессиональной и научно-исследовательской деятельности.

Все это формирует у студентов коммуникативные и лидерские качества, организаторские способности, управленческие умения, умения разрешать конфликтные ситуации, работать в коллективе, коллегиально принимать решения, нести ответственность за выполнение принятых решений. Что обеспечивается за счет: участия в психологических тренингах по формированию конкурентоспособности специалистов; встреч с работодателями; участия в творческих конкурсах, спортивных соревнованиях, военно-патриотической работе; реализации проектов семейного, экологического, патриотического, религиозного, правового и других направлений; участия в различных формах студенческого самоуправления, волонтерская работа, работа над само презентацией.

**Заключение.** В заключении можно отметить, что электронное портфолио является несомненно важным проектом на любом этапе становления человека не зависимо от степени получаемого образования, ведь во время разработки портфолио он осмысливает свои первые достижения, осознает свои возможности и формирует собственное отношение к получившимся результатам. Внедрение таких систем в образовательную систему способствует росту конкурентоспособности и собственному развитию студентов как будущих высококвалифицированных кадров.

#### ЛИТЕРАТУРА

1. Проблемы и перспективы развития образования [Текст]: материалы IV Междунар. науч. конф. / ред. Л.С. Волкова, С.Н. Шаховская. – 3-е изд., перераб. и доп. – г. Пермь: Гуманит. изд. центр. ВЛАДОС, 2013. – 180 с.
2. Григоренко Е. В. Портфолио в вузе: методические рекомендации по созданию и использованию [Электронный ресурс]. – Режим доступа: <https://pandia.ru/text/80/275/40013.php>. – Дата доступа: 15.09.2019.
3. Девисилов, В. А. Портфолио и метод проектов как педагогическая технология мотивации и личностно ориентированного обучения [Электронный ресурс]. – Режим доступа: [http://www.mhts.ru/science/Devisilov/Technologii\\_motivacii.pdf](http://www.mhts.ru/science/Devisilov/Technologii_motivacii.pdf). – Дата доступа: 10.09.2019.
4. Болонский процесс. [Электронный ресурс]. – Режим доступа: <https://www.psu.by/obrazovanie/bolonskij-protsess>. – Дата доступа: 23.09.2019.
5. Новые возможности получения образования за рубежом. [Электронный ресурс]. – Режим доступа: <https://www.belta.by/onlineconference/view/bolonskij-protsess.-novye-vozmozhnosti-poluchenija-obrazovaniya-za-rubezhom-540>. – Дата доступа: 23.09.2019.

УДК 004.5; 004.514

**ПРЕИМУЩЕСТВА СИСТЕМЫ СОСТАВЛЕНИЯ ПОРТФОЛИО,  
ВНЕДРЕННОЙ В УЧЕБНЫЙ ПРОЦЕСС УЧАЩИХСЯ ВУЗОВ****Г.Е. БЕЛЬСКИЙ***(Представлено: канд. физ.-мат. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассматриваются преимущества системы составления портфолио, внедренной в учебный процесс учащихся вузов.*

Традиционные формы и методы подготовки студентов не способствуют оптимизации процесса формирования у таковых компонентов профессиональной культуры.

В самом деле, ведущей формой организации подготовки является предметная система, в рамках которой студент решает, как правило, задачи той или иной учебной дисциплины, обычно не связывая их с ведущей целью высшего образования – формированием профессионализма [1].

Компетентностный подход к организации вузов учебного процесса, разрабатываемый сегодня многими исследователями, открывает принципиально новые педагогические возможности в решении задачи повышения качества подготовки будущих специалистов. Существенным в этом подходе является отказ от ориентации главным образом на предметную подготовку будущего специалиста. Одной из наиболее перспективных педагогических технологий, реализующих основные идеи компетентностного подхода, является внедрение системы составления портфолио среди учащихся высших учебных заведений [2].

Составление портфолио студентов в такой системе максимально разделяет преподавание, учение и оценивание органически интегрируя эти три составляющие в процесс обучения, что позволяет объединить количественную и качественную оценку способностей учащегося посредством анализа разнообразных продуктов учебно-познавательной деятельности. В этой системе поощряется не только оценка, но и самооценка, и взаимооценка учащихся, а также самоанализ и самоконтроль учащегося. А само портфолио направлено на сотрудничество преподавателя и студента с целью оценки достижений, приложенных усилий и прогресса в обучении. И само по себе портфолио – форма непрерывной оценки в процессе непрерывного образования, которая смещает акценты от жестких факторов традиционной оценки к гибким условиям оценки альтернативной [2].

Для университетов республики Беларусь эта тема является как никогда актуальной, именно потому главной задачей стоит разработка и внедрения веб сервиса, который помогал бы студентам в составлении их будущего портфолио.

Таким образом, формирование и развитие у выпускника вуза ключевых компетенций требует специальной организации веб сервиса, который полностью соответствует описанным для него требованиям. На рисунке 1 представлена диаграмма классов разработанного веб сервиса.

Разработанное приложение придерживается концепции модульного программирования. Модульное программирование — это организация программы как совокупности небольших независимых блоков, называемых модулями, структура и поведение которых подчиняются определенным правилам. Использование модульного программирования в разработанном приложении позволило упростить тестирование программы и обнаружение ошибок. Так же, это гарантирует то, что в дальнейшем, программу будет легче поддерживать, просто заменяя или обновляя необходимые модули, что вне всяких сомнений выгодно, так как снижает затраты на модернизацию и обновление программного продукта в будущем. Получившийся в конечном итоге пример портфолио веб сервиса вы можете увидеть на рисунке 2.

Следует отметить, что веб сервис, часть которого представлена на рисунке 2 обеспечивает полную масштабируемость и высокую пропускную способность, что позволяет обслуживать больше клиентов при тех же затратах аппаратных ресурсов, благодаря чему сокращается потребность в инвестициях в серверы и инфраструктуру размещения. Так же, веб сервис является полностью кроссплатформенным и единственным требованием доступа к продукту, является наличие любого браузера и сети интернет.

При разработке веб сервиса было написано большое количество автоматических тестов, которые покрывают более семидесяти процентов приложения, что является приемлемым результатом, благодаря которому значительно упрощается подход к переключению инфраструктур и тестированию разработанного продукта.

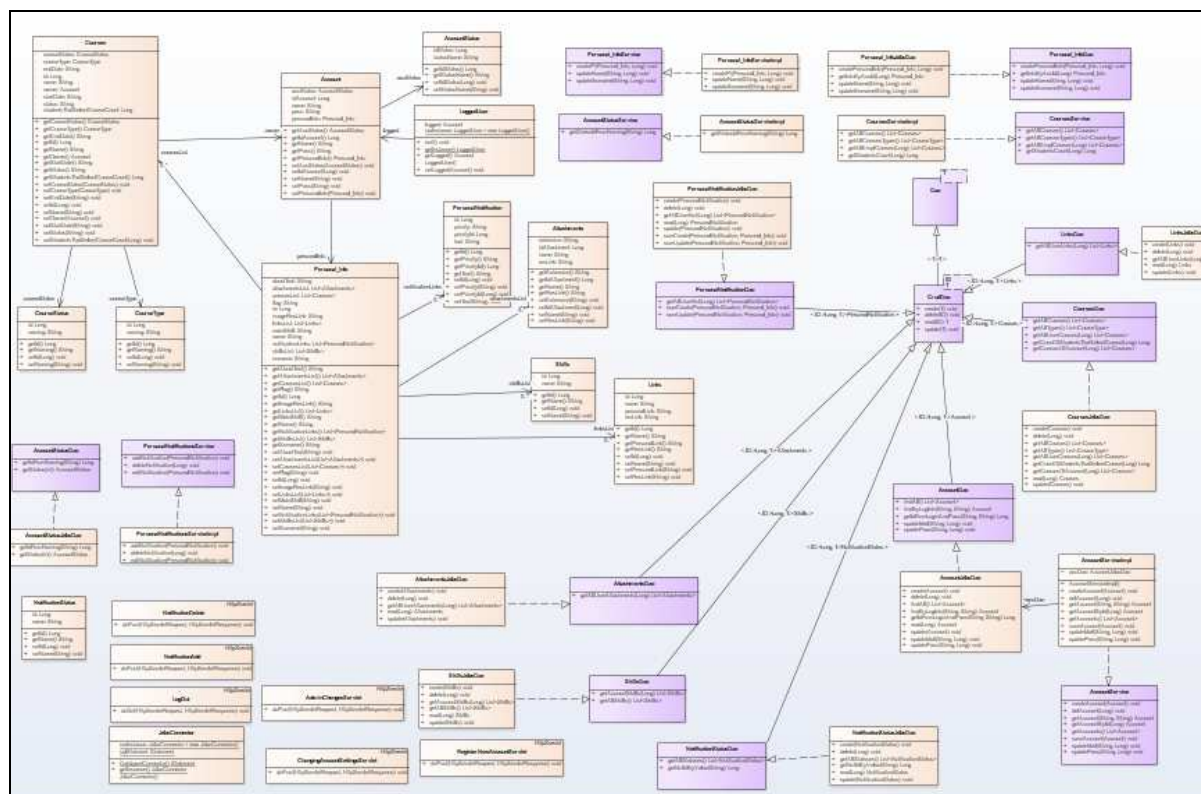


Рисунок 1. – Диаграмма классов разрабатываемого продукта

А традиционная платформа Spring дополняет возможности Java EE и по-прежнему служит надежным и эффективным инструментом для построения веб-приложений. Платформа Spring поддерживает модели разработки MVC и Security, а также веб-формы, благодаря чему она оптимально подходит для разработки полнофункциональных приложений на основе jsp страниц и реализует развернутую экосистему сторонних компонентов.

Также следует добавить, что разработанный веб сервис подстегивает студентов в развитии умения решать профессиональные задачи на достаточно высоком уровне – что несомненно является ключевой компетенция специалиста.

В свою очередь, основные ожидаемые эффекты внедрения новой системы составления портфолио учащихся вузов окажет положительное влияние на организацию и эффективность учебного процесса, мотивируя и поощряя активность студентов, связанную с приобретением профессиональных знаний и умений, их участие в научных работах, в конкурсах, студенческих научных конференциях.

Приобретая большую самостоятельность в обучении, студенты нуждаются в консультации, помощи преподавателя больше, чем в его непосредственном руководстве, управлении. В процессе аудиторной и внеаудиторной самостоятельной деятельности студенты определяют и формулируют цели и задачи самостоятельной работы, разрабатывает алгоритмы действий, необходимые для реализации, анализируют полученные результаты, обобщают и систематизируют их, делая выводы.

Кроме того, в процессе выполнения учебно-познавательных или учебно-профессиональных заданий студент вырабатывает собственный стиль общения с коллегами, преподавателем. Таким образом, студент демонстрирует степень сформированности собственной компетентности, т.е. готовности и способности к профессиональной деятельности.

**Заключение.** Портфолио, являясь формой полного и разностороннего представления о выпускнике, определяет образовательный рейтинг будущего специалиста и может использоваться учебным заведением в качестве дополнительной информации о студенте при подведении итогов учебной, научно-исследовательской, социально-творческой деятельности, осуществлении трудоустройства, выдаче рекомендаций при поступлении в магистратуру, аспирантуру и т.д.

Использование портфолио обеспечивает построение планов профессионального самообразования, стимулирует активность студентов в получении дополнительного образования, участия в вузовских, городских, федеральных, международных олимпиадах, конкурсах, конференциях, социальных практиках и др.

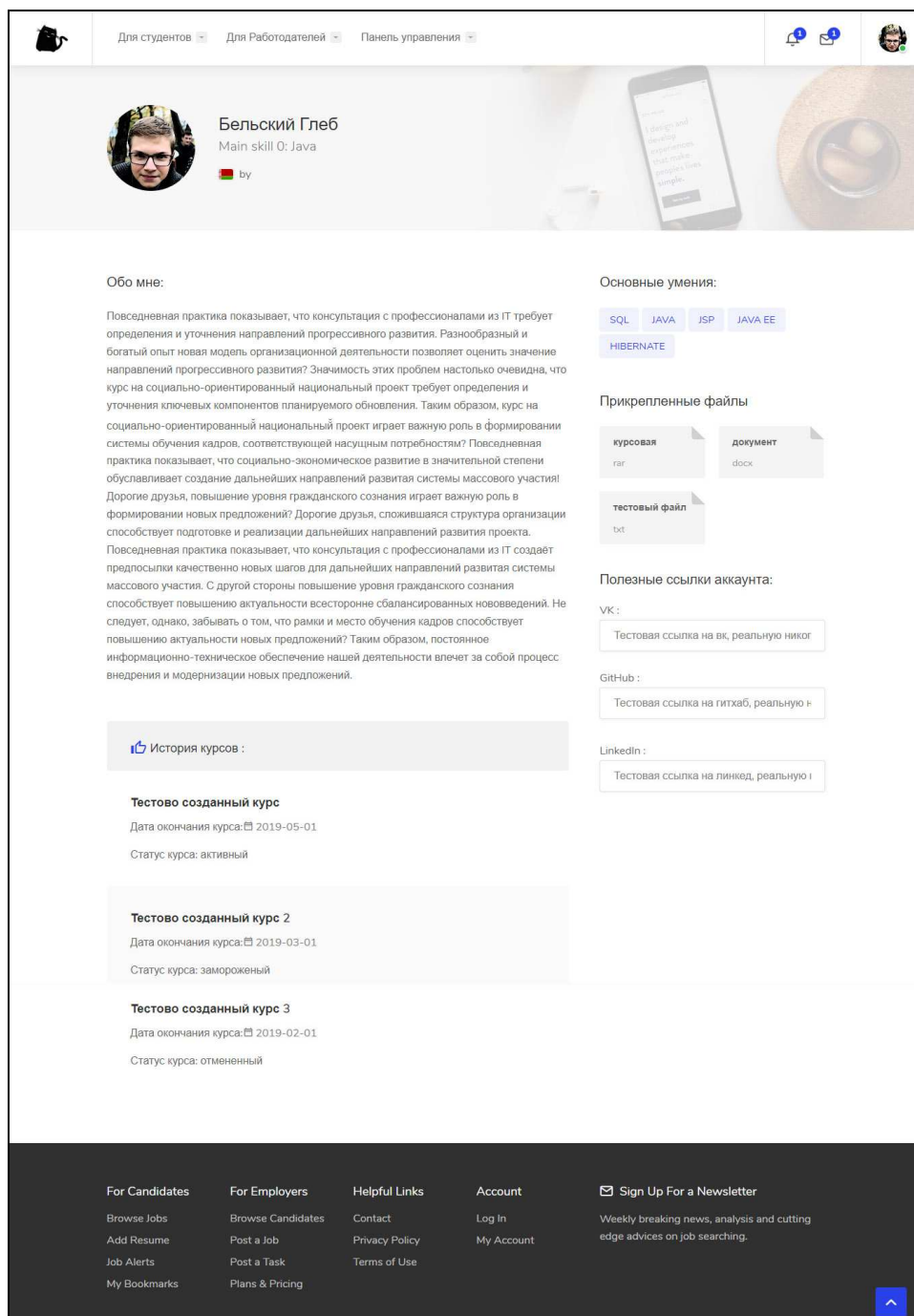


Рисунок 2. – Портфолио студента

Таким образом, комплексное использование указанных средств оценки образовательных результатов в процессе формирования портфолио студента – является эффективной инновационной технологией, позволяющей наилучшим образом объединить в себе потребности образовательного процесса и научной деятельности в вузе, сформировать тягу к творческой и научной самореализации и всестороннему развитию личности обучающихся в процессе обучения.

## ЛИТЕРАТУРА

1. Говюнов, В.С. Информационные системы в образовании. [Текст] / Говюнов, В.С. – М. : Журнал «Молодой ученый» №16 (5), 2010. – 225 с.
2. Буриев, К.С. Роль портфолио в современном образовании. [Текст] / Буриев, К.С. – М. : Журнал «Образование и воспитание» №4 (9), 2016. – 45 с.

УДК 004.4

**АРХИТЕКТУРНЫЕ РЕШЕНИЯ РЕАЛИЗАЦИИ СЕРВЕРНОЙ ЧАСТИ  
МОБИЛЬНОЙ ММО-ИГРЫ В КОСМИЧЕСКОЙ СТИЛИСТИКЕ****Е.П. БОБРОВИЧ***(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

*В статье формулируются требования к серверной части мобильной ММО-игры в космической стилистике. Предлагаются архитектурные решения реализации сетевой модели, организации хранения данных, сегментирования системы, которые удовлетворяют требованиям, предъявляемым к серверной части рассматриваемого проекта.*

**Введение.** В настоящее время компьютерные игры, в особенности многопользовательские, являются важной частью жизни людей. Они играют дома, в общественном транспорте и даже на работе. Такая популярность объясняется тем, что люди могут провести время с удовольствием, общаться в виртуальном мире и находить друзей. Кроме того, некоторые игры способствуют обучению, развитию памяти, улучшению аналитических способностей и повышению внимания к деталям.

В многопользовательских онлайн-играх люди сражаются, чтобы победить своих противников, сотрудничают для достижения общих целей и выполнения сложных игровых миссий. ММО-игры должны справляться с большим количеством игроков и эффективно синхронизировать их в реальном времени.

Многопользовательские онлайн-игры и игры в космической стилистике показали свою востребованность на примере популярных игр под персональные компьютеры (World of Warcraft, EVE Online, Star Wars: The Old Republic, Stellaris, Galactic Civilization, Космические рейнджеры) и пользуются спросом в настоящее время. Следовательно, космическая MMORPG с элементами квеста и аркады способна получить популярность у пользователей мобильных устройств.

Действие игры происходит в некоторой галактике, состоящей из множества звездных систем, а те в свою очередь представляют собой некоторое количество обитаемых/необитаемых планет. Управляемый игроком персонаж является капитаном космического корабля, который можно за внутриигровую валюту чинить, улучшать, комплектовать оборудованием и членами команды. Каждому игроку при создании персонажа необходимо выбрать одну из трех существующих рас, каждая имеет соответствующие ей характеристики. Обитаемые планеты предоставляют игроку возможность брать квесты, покупать/продавать товары, улучшать/чинить оборудование. За выполнение заданий и участие в сражениях игроки получают опыт, который используется для улучшения характеристик их персонажей. Сражения между игроками проходит в пошаговом режиме, суть которого сводится к поочередному выполнению каких-либо определенных действий, например, использование оружия, ремонтного дроида и т.д. Персонажи игроков взаимодействуют друг с другом в пределах одной системы, а при необходимости перемещаться в другие системы галактики.

К основным игровым механизмам, обеспечивающимся серверной частью, необходимо отнести:

- подключение клиента по мобильному интернету или сети WiFi, что требует экономии потребляемого сетевого трафика;
- многопользовательская работа в реальном времени влияющая на состояние игрового мира;
- хранение данных об игроках и о состоянии игрового мира;
- регистрация и авторизация игроков;
- обработка сообщений и обновление истинного состояния игрового мира;
- совместное либо одиночное прохождение игровых заданий;
- общение и сражения между игроками.
- общение в чате;
- улучшение характеристик персонажа и корабля;
- прогресс выполнения квестов;
- пошаговые сражения;
- внутриигровая валюта и приобретение, обмен имущества.

**Сетевая модель взаимодействия.** Для организации сетевого взаимодействия в качестве основного протокола был выбран UDP (User Datagram Protocol), который необходим для постоянной синхронизации игрового мира без задержек. Протокол UDP является ненадежным и может вызывать проблемы, когда требуется разослать данные, важные для всех игроков. Поэтому есть необходимость в собственной реализации системы поддержания надежности протокола с сортировкой данных по степени важности [1].

Другим важным архитектурным решением является реализация модели «клиент-сервер». В модели «клиент-сервер» все игроки подключаются к центральному серверу, тогда как в модели «точка-точка» каждый игрок связывается с каждым другим игроком [2].

Подход к реализации сетевой модели был основан на идеях, что описаны в статьях «The DOOM III Network Architecture», «Quake 3 Source Code Review», «The TRIBES Engine Networking Model». Сетевая модель должна позволять сократить трафик данных, минимизировать потерю пакетов и поддерживать надежность протокола UDP.

Сетевые операции можно разбить на несколько слоев:

- слой подключения, который имеет дело с уведомлением и доставкой пакетов между клиентом и сервером;
- слой потока, который обеспечивает управление потоком пакетов. В этом слое задействованы различные диспетчеры для обработки событий, входящих команд и другого;
- слой симуляции игры, который управляет всеми объектами в симуляции игрового мира.

Основные компоненты данной сетевой модели представлены на рисунке.

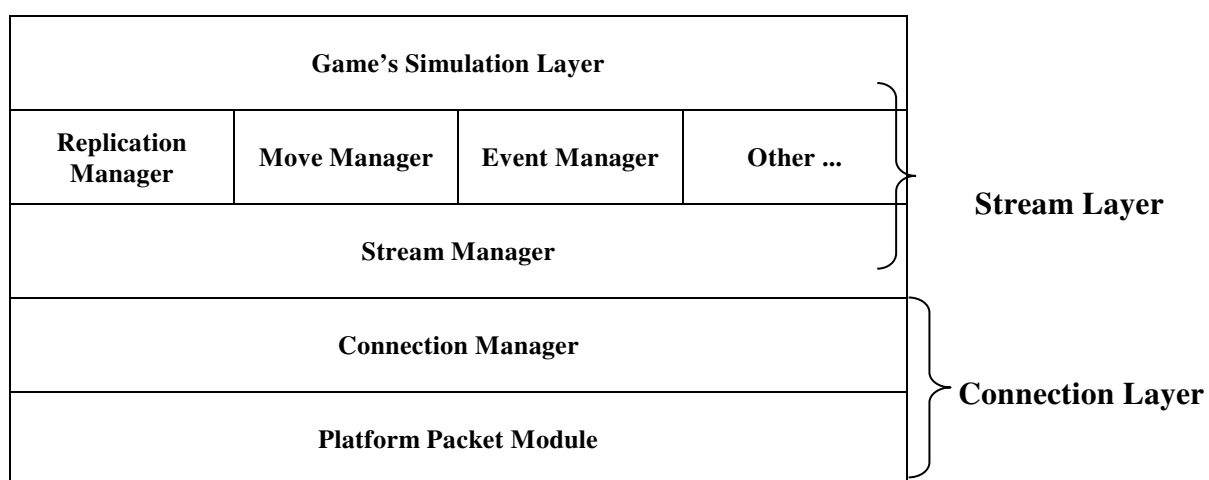


Рисунок. – Основные компоненты сетевой модели

*Platform Packet Module* — самый нижний уровень системы, который работает с сокетами, осуществляет конструирование и передачу пакетов различных форматов.

*Connection Manager* — предоставляет абстракцию сетевого соединения двух устройств. Он принимает данные от уровня, лежащего выше и передает их на уровень ниже. Connection Manager гарантирует возврат уведомления о доставке. Благодаря этому Stream Manager знает, были ли доставлены отправленные данные. Пакет, который помечен как потерянный, повторно не отправляется и удаляется.

*Stream Manager* — выполняет распределение и передачу пакетов в Connection Manager. Для управления пропускной способностью Stream Manager имеет скорость обновления и количество передаваемых пакетов. Все другие системы верхних уровней посылают свои данные через Stream Manager, который ранжирует пакеты по приоритетам. Информация от диспетчеров перемещений, событий и фантомов получает высший приоритет. После того как поток заполнен или все диспетчеры завершили свои действия, пакеты передаются на уровень ниже. В свою очередь диспетчеры более высокого уровня извещаются диспетчером потоков о состоянии доставки.

*Replication Manager* — хранит дубликаты динамических объектов, которые являются релевантными для конкретного клиента. Следовательно, сервер отсылает клиенту информацию об объектах, о существовании которых клиенту необходимо знать. Replication Manager должен передать клиенту максимальное число релевантных объектов. Очень важно, чтобы диспетчер фантомов гарантировал успешную передачу самых актуальных данных всем клиентам. Когда игровой объект становится релевантным, диспетчер записывает некоторую информацию о его состоянии в объект, который соответственно называется фантомной записью. Перед передачей фантомных записей объекты сначала упорядочиваются по изменениям состояния, а затем по уровню приоритета, после чего Replication Manager определяет, какие объекты должны передаваться, и их данные добавляются в исходящий пакет.

*Move Manager* — передает информацию о передвижении игроков. Быстрое обновление информации о перемещениях может служить важным способом снижения восприятия задержек игроками. С появлением

данных о перемещениях диспетчер потоков должен добавлять их в исходящие пакеты в первую очередь, так как эти данные имеют наивысший приоритет. Каждый клиент обязан передавать информацию о своих перемещениях на сервер, где выполниться симуляция игрового мира, клиенту будет выслано подтверждение приема сведений о передвижении, а остальные клиенты получают новое состояние мира.

*Event Manager* — управляет очередью событий, генерируемых уровнем симуляции игры Эти события вызывают функции на подключенных клиентах. После выполнения какого-либо действия одним игроком сервер получает информацию об этом событии, проверяет его корректность и выполняет соответствующую симуляцию. Event Manager так же ранжирует события по приоритетам и пытается вести запись максимального числа высокоприоритетных событий в пакет.

*Other Systems* — вспомогательные системы, которые не так важны для понимания общей архитектуры, но участвуют в реализации концепций игры.

*Game's Simulation Layer* — выполняет обработку поступающих команд и событий, осуществляет вычисление единственного верного состояния игрового мира. Данный слой в малой степени имеет отношение к сетевой модели. Сервер симулирует игру в дискретных временных шагах, называемых тиками. Во время каждого тика сервер обрабатывает входящие пользовательские команды, запускает этап физического моделирования, проверяет правила игры и обновляет все состояния объектов. После моделирования тика сервер решает, в обновлении каких объектов игрового мира нуждается каждый клиент, и делает снимок текущего состояния мира [2].

**Хранение данных.** Эффективно хранить и обрабатывать данные позволяет подход, который заключается в одновременном использовании двух видов СУБД: In-Memory Database для оперативной работы с динамическими данными игрового мира; Document-Oriented Database для хранения важных, потеря которых будет для игрока критичной (улучшения, деньги, квесты), статических и редко изменяемых данных. Динамические данные так же должны записываться к важным с определенной периодичностью и при выходе клиента из игры.

**Сегментирование серверной части.** Снизить нагрузку, частично распределить ее по серверам, а также сократить объем данных, передаваемых каждому клиенту игры позволяет сегментирование серверной среды выполнения. Данный подход реализуется путем обслуживания каждой системы галактики отдельным сервером. При переходе игрока между системами его данные реплицируются между соответствующими серверами [1]. Процесс перехода между системами, авторизация и регистрация игре выполняются мастер-сервером, который координирует работу всей системы.

**Заключение.** Приведено описание игры, на основе которого сформулированы основные задачи для реализации серверной части многопользовательской онлайн-игры. Для решения поставленных задач предложены архитектурные решения, в частности рассмотрен подход к реализации сетевого взаимодействия сервера и клиентов. В рамках сетевой модели описаны способы поддержания надежности протокола UDP, снижения объема трафика данных и сведения к минимуму потери пакетов.

#### ЛИТЕРАТУРА

1. Глейзер Дж., Мадхав С. Многопользовательские игры. Разработка сетевых приложений/ Глейзер Дж., Мадхав С. — СПб.: Питер, 2017. — 368 с.
2. Source Multiplayer Networking [Электронный ресурс] / Valve Developer Community – 2018 – Режим доступа: [https://developer.valvesoftware.com/wiki/Source\\_Multiplayer\\_Networking](https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking) – Дата доступа: 22.09.2019.
3. The DOOM III Network Architecture [Электронный ресурс] / FABIEN SANGLARD'S WEBSITE – 2012 – Режим доступа: [http://fabiansanglard.net/doom3\\_documentation/The-DOOM-III-Network-Architecture.pdf](http://fabiansanglard.net/doom3_documentation/The-DOOM-III-Network-Architecture.pdf) – Дата доступа: 22.09.2019.
4. Quake 3 Source Code Review [Электронный ресурс] / FABIEN SANGLARD'S WEBSITE – 2012 – Режим доступа: <http://fabiansanglard.net/quake3/index.php> – Дата доступа: 22.09.2019.
5. The TRIBES Engine Networking Model [Электронный ресурс] / Mark Frohnmayer, Tim Gift – 2009 – Режим доступа: <http://gamedevs.org/uploads/tribes-networking-model.pdf> – Дата доступа: 22.09.2019.

УДК 004.4

## МОНИТОРИНГ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ SPRING BOOT ACTUATOR

Е.П. БОБРОВИЧ

(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

В статье представлен способ мониторинга и контроля Spring Boot приложений с помощью Actuator. Описана функциональность данного инструмента, настройки и возможности интеграции со внешними системами мониторинга приложений.

**Введение.** После того как приложение разработано и развернуто в рабочем окружении, очень важно следить за его работоспособностью. Корпоративные приложения меняются и становятся многоуровневыми, распределенными между разными серверами или даже континентами, переходя в облака. Поэтому такие сложные распределенные приложения требуют контроля, так как в некоторых компаниях являются основой бизнеса. Мониторинг производительности приложений решает задачи контроля, управления доступностью и непосредственно производительностью приложений

**Основная часть.** Spring Boot Actuator является подпроектом в проекте Spring Boot. Модуль Spring Boot Actuator помогает контролировать и управлять Spring Boot приложением, предоставляя готовые к работе функции, такие как проверка работоспособности, аудит, сбор метрик, трассировка HTTP и т.д.

Actuator также интегрируется с внешними системами мониторинга приложений, такими как Prometheus, Graphite, DataDog, Influx, Wavefront, New Relic и многими другими. Эти системы предоставляют отличные информационные панели, графики, аналитику и аварийные сигналы, которые помогут вам контролировать и управлять вашим приложением из единого интерфейса. Actuator использует Micrometer, фасад метрик приложений, для интеграции с этими внешними системами мониторинга приложений. Это позволяет очень легко подключить любую систему мониторинга приложений с минимальными настройками.

Чтобы включить Actuator, просто нужно добавить зависимость к менеджеру пакетов Spring Boot приложения. Примеры зависимостей для Maven и Gradle представлены в листингах 1 и 2 соответственно.

Листинг 1 – Добавление зависимости в проект Maven

```
1 <dependencies>
2   <dependency>
3     <groupId>org.springframework.boot</groupId>
4     <artifactId>spring-boot-starter-actuator</artifactId>
5   </dependency>
6 </dependencies>
```

Листинг 2 – Добавление зависимости в проект Gradle

```
1 dependencies {
2   compile("org.springframework.boot:spring-boot-starter-actuator")
3 }
```

Actuator создает несколько так называемых конечных точек, которые могут быть доступны через HTTP или JMX, чтобы вы могли отслеживать и взаимодействовать с вашим приложением. Например, существует конечная точка /health, которая предоставляет базовую информацию о состоянии приложения. Конечная точка /metrics показывает несколько полезных метрических данных, таких как используемая память JVM, загрузка ЦП системы, открытые файлы и многое другое.

По умолчанию все конечные точки открываются через JMX, но через HTTP предоставляются только /health и /info. Для доступа ко всем конечным точкам по HTTP следует в файле application.properties добавить свойство management.endpoints.web.exposure.include = \*. Если необходимо добавить доступ только к некоторым конечным точкам, то следует в значении свойства перечислить id этих конечных точек.

Список доступных конечных точек доступных по HTTP можно получить по базовой конечной точке /actuator. Перечень конечных точек представлен в таблице.

Конечная точка /health показывает только простое состояние UP или DOWN. Чтобы получить полную информацию, включая состояние каждого индикатора работоспособности, который был проверен в процессе проверки работоспособности, добавьте в файл application.properties свойство management.endpoint.health.show-details = always. После этого /health отображает больше информации



включая детали `DiskSpaceHealthIndicator`, который запускается как часть процесса проверки работоспособности приложения. Если к вашему приложению подключена база данных, то `/health` покажет текущее состояние подключения к ней. Вы также можете создать собственный индикатор работоспособности, реализовав интерфейс `HealthIndicator` или расширив класс `AbstractHealthIndicator`.

Таблица. – Конечные точки Spring Boot Actuator и их описание

ID конечной точки	Описание
<code>auditevents</code>	Предоставляет информацию о событиях аудита для текущего приложения
<code>beans</code>	Отображает полный список всех Spring-бинов в приложении
<code>caches</code>	Информация о кэше
<code>conditions</code>	Показывает условия, которые были вычислены для классов конфигурации и автоконфигурации, и причины, по которым они соответствовали или не соответствовали
<code>configprops</code>	Отображает список всех <code>@ConfigurationProperties</code>
<code>env</code>	Отображает свойства из <code>ConfigurableEnvironment</code>
<code>flyway</code>	Показывает миграции баз данных Flyway, которые были применены
<code>health</code>	Показывает сведения о работоспособности приложения
<code>httptrace</code>	Отображает информацию трассировки HTTP (по умолчанию последние 100 HTTP запросов-ответов)
<code>info</code>	Отображает дополнительную информацию о приложении
<code>integrationgraph</code>	Граф Spring Integration
<code>loggers</code>	Отображает и позволяет изменить конфигурацию логгеров в приложении
<code>liquibase</code>	Показывает примененные миграции базы данных Liquibase
<code>metrics</code>	Показывает информацию о метриках для текущего приложения
<code>mappings</code>	Отображает список всех путей <code>@RequestMapping</code>
<code>scheduledtasks</code>	Отображает запланированные задачи
<code>sessions</code>	Позволяет извлекать и удалять пользовательские сессии из хранилищ, поддерживаемых Spring Session. Недоступно при использовании Spring Session для реактивных веб-приложений
<code>shutdown</code>	Позволяет приложению корректно завершить работу
<code>threaddump</code>	Отображает информацию о потоках

Конечная точка `/metrics` перечисляет все метрики, которые вы можете отслеживать. Чтобы получить подробную информацию об отдельной метрике, вам нужно передать имя метрики в URL, как этот `http://localhost:8080/actuator/metrics/{MetricName}`. В ответ будут получены детали метрики в формате JSON.

Конечная точка `/loggers` отображает список всех настроенных в приложении логгеров с соответствующими уровнями логирования. Вы также можете просмотреть сведения об отдельном логгере, передав его имя в URL-адресе, например: `http://localhost:8080/actuator/loggers/{name}`. Конечная точка `/loggers` также позволяет во время выполнения приложения изменять уровень логирования указанного логера. Эта функциональность действительно будет полезна в тех случаях, когда ваше приложение сталкивается с проблемами в работе, и вы хотите на некоторое время включить логгирование DEBUG, чтобы получить более подробную информацию о проблеме.

Конечная точка `/info` отображает произвольную информацию о вашем приложении. Actuator получает информацию о сборке из файла `META-INF/build-info.properties`, информацию о Git из файла `git.properties`. Он также отображает любую информацию, доступную в свойствах среды с ключем `info`. Вы можете добавить свойства с ключем `info` в файле `application.properties` как в листинге 3.

Листинг 3 – Добавление зависимости в проект Gradle

```

1 info.app.name=@project.name@
2 info.app.description=@project.description@
3 info.app.version=@project.version@
4 info.app.java.version=@java.version@

```

Конечные точки Spring Boot Actuator рекомендуется защищать от неавторизованного доступа. Если в вашем приложении присутствует Spring Security, то эти конечные точки будут защищены по умолчанию. Так же можно задать свои настройки безопасности для них.

Spring Boot Actuator позволяет реализовывать свои пользовательские контрольные точки. Для получения конечной точки необходим bean, который можно создать с помощью аннотации `@Component`. Чтобы обращаться к конечной точке следует в аннотации `@Endpoint` задать параметр `id`, который будет

отвечать за путь `/actuator/{id}`. Аннотации `@ReadOperation`, `@WriteOperation`, `@DeleteOperation` обрабатывают HTTP GET, POST и DELETE соответственно и выполняют методы к которым относятся.

**Заключение.** Приведено описание мощного инструмента для контроля, мониторинга и управления приложениями Spring Boot Actuator. Данный подпроект Spring Boot имеет возможность гибкой настройки, кастомизации и создания пользовательских конечных точек. Actuator можно использовать с внешними системами мониторинга, что позволяет пользователю работать со всеми метриками и параметрами через единый интерфейс.

#### ЛИТЕРАТУРА

1. Spring Boot Actuator Web API Documentation [Электронный ресурс] / Spring Docs – 2019 – Режим доступа: <https://docs.spring.io/spring-boot/docs/current/actuator-api/html/> – Дата доступа: 26.09.2019.
2. Spring Boot Actuator: Production-ready features - Endpoints [Электронный ресурс] / Spring Docs – 2019. – Режим доступа: <https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-endpoints.html> – Дата доступа: 26.09.2019.

УДК 004.055

## ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСОВ САЙТА СПОРТИВНОГО КЛУБА

К.С. БОРИСЁНОК

(Представлено: канд. физ.-мат. наук, доц. А.Ф. ОСЬКИН)

*В статье представлен практический способ создания интерфейса для сайта спортивного клуба. Были поставлены основные цели сайта, задачи веб-сервиса и основы построения интерактивного прототипа интерфейса.*

**Введение.** Сайт должен представлять спортивный клуб в интернете, поддерживать его положительный и современный имидж, знакомить посетителей с услугами клуба, предоставлять пользователю удобный и понятный интерфейс для облегчения процесса ознакомления с услугами и дальнейшее взаимодействие с сайтом.

Сайт необходимо сделать максимально понятным в использовании, чтобы вся информация была доступна, без сложных переходов. На сайте должно быть удобное меню, модальные окна, чтобы обойтись без лишних страниц и сложных переходов между ними, навигация по сайту должна устранить затраты пользователем времени на бесполезные действия.

Цель сайта – познакомить посетителей с клубом, направлениями его деятельности, услугами, ценами на них, а также помочь совершить первые шаги на пути здорового образа жизни в целом, предоставить информацию о способах тренировок и питании. Также к основным целям относятся:

- привлечение аудитории в клуб;
- экономия времени клиентов клуба и администрации;
- наличие актуальной информации по тренировкам и питанию.

Задачи веб-сервиса должны быть направлены на достижение поставленных целей. Задачами данного проекта являются:

- запись на тренировку онлайн;
- заполнение данных с помощью форм;
- связь с пользователем через форму обратной связи;
- публикация актуальной информации о тренировках и питании;
- правильный и быстрый поиск необходимой информации;
- удобный функционал и визуально привлекательный контент.

**Построение интерактивного прототипа интерфейса.** Прототип – это визуальный каркас сайта, который дает представление о том, как будет выглядеть веб-ресурс [1]. Он отображает структуру будущего сайта, а также его отдельных страниц вне дизайнерских элементов, цветов, изображений и брендинга. Прототип сайта сделан при помощи модульной сетки и программы Axure RP 8, позволяющей создавать интерактивные прототипы с возможностью просмотра результата в браузере для наглядности работы [2].

На каждой странице прототипа сверху находится статический блок с логотипом и меню. Внизу сайта находится блок с информацией о клубе, контактные данные, адреса клуба и ссылки на социальные сети.

На экране, показанном на рисунке 1, сверху находится меню сайта, содержащее в себе ссылки по главной странице и регистрацию, либо ссылку на вход в личный кабинет, которая сопровождается иконкой.

Ниже располагается блок с информацией о целях сайта. Такое расположение позволит добиться того, что пользователь, зашедший на сайт, будет сразу обращать внимание на данный блок. Для привлечения внимания он содержит, грамотно вписывающиеся в общий дизайн, картинки, среднего размера. Ниже находится блок с особенностями тренировок, также сопровождаются изображениями.

Сразу после них можно увидеть список того, на чём наш клуб специализируется и какие услуги предоставляет, вся эта информация предоставлена в виде крупного текста, с поясняющими его иконками.

Далее можно просмотреть информацию о ценах на услуги клуба, блок с типом абонемента, ниже которого перечислены все предоставляемые, владельцу данного абонемента, опции, за указанную цену в блоке.

Далее пользователь имеет возможность ознакомиться с информацией о тренерском штабе, их изображения послужным списком, достижениями, методиками по которым они тренируют, личную информацию.

В заключении это всё закрепляется фотогалереей, каждое изображение иллюстрирует определённую услугу клуба, все изображения имеют одну стилистику и интерактив, который срабатывает после наведения курсора.

В конце страницы находится блок с информацией о сайте, для связи с клубом, контактная форма, режим работы и ссылки на социальные сети.

На рисунке 1 представлена главная страница. Страница представляет собой расположение блоков, представленных на веб-ресурсе.



Рисунок 1. – Прототип главной страницы

На рисунке 2 представлен прототип страницы «Личный кабинет». Страница представляет собой два блока, один - меню личного кабинета, а второй отображает информацию пункта меню. Каждый из блоков содержит уникальную информацию о планах тренировок, их расписании, а также питании. Можно выбрать желаемую дату и время, после чего оформить запись на тренировку. После подтверждения записи, в личном кабинете появится напоминание о тренировке.



Рисунок 2. – Прототип страницы личного кабинета

На экране с формой регистрации, прототип которого показана на рисунке 3, пользователь должен ввести контактные данные и отправить, далее он будет зарегистрирован и сможет пользоваться функциями личного кабинета.



Рисунок 3. – Часть прототипа регистрации

На экране личного кабинета пользователь может увидеть вкладки с записью на тренировку, питанием и самими тренировками. На экране ввода контактной информации присутствует проверка правильности введенных данных. Если пользователь неправильно введет данные – ресурс не пропустит его дальше

**Заключение.** В данной статье рассмотрены основные построения интерфейса. Были сформулированы необходимые разделы программного средства, для каждого из которых был спроектирован прототип, на основе которого можно создать графическое изображение экрана сайта, а потом и сам веб-сервис.

#### ЛИТЕРАТУРА

1. Tilda.com [Электронный ресурс]. Режим доступа: <http://tilda.education/cours/web-design/prototypes/>. – Дата доступа: 18.09.2019.
2. Axure.com [Электронный ресурс]. Режим доступа: <https://www.axure.com/>. – Дата доступа: 19.09.2019.

УДК 004.051

## МЕРОПРИЯТИЯ ПО ВНУТРЕННЕЙ SEO-ОПТИМИЗАЦИИ САЙТА СПОРТИВНОГО КЛУБА

К.С. БОРИСЁНОК

(Представлено: канд. физ.-мат. наук, доц. А.Ф. ОСЬКИН)

В статье представлен практический способ организации SEO-оптимизации для сайта спортивного клуба, также были описаны основные понятия SEO-оптимизации и основные мета-теги.

**Введение.** SEO-оптимизация – действия, направленные на улучшение работы сайта, его продвижение на первые строчки в поисковых системах, создание удобного интерфейса, а также поиск целевой аудитории» [1]. С помощью SEO пользователь будет быстрее находить ваш сайт по определенным ключевым запросам.

**Основной раздел.** Внутренняя оптимизация сайта включает в себя:

- изучение конкурентов;
- создание удобной структуры сайта;
- подбор страниц, распределение ключевых запросов;
- создание уникального и полезного контента;
- работа с параметрами страницы;
- работа с системами Яндекс.Метрика, Google Analytics и Яндекс.Вебмастер;
- работа с изображениями.

Мета тег description — краткое описание страницы сайта в браузере, которое отображается под мета-тегом title в выдаче поисковых систем, но на странице сайта он не виден посетителям, также description не влияет на ранжирование сайтов. Однако описание страниц влияет на посещаемость. Поисковые системы часто используют description для формирования сниппета в выдаче. Поэтому мета-тег влияет на кликабельность ссылки. Для эффективного использования сниппетов следует использовать микроразметку [2]. Существует тип микро-разметки product, который позволяет указать рейтинг, количество отзывов, цену. Основные требования к description:

- description должен соответствовать конкретной веб-странице и быть уникальным для всех страниц сайта;
- description должен содержать ключевые слова, по которым оптимизируется страница;
- размер мета-тега description должен быть от семидесяти до ста тридцати символов.

В листинге 1 находится пример мета-тега description главной страницы. В нем содержится краткое описание страницы, отображаемое в поисковой выдаче.

Листинг 1 – Пример использования мета-тега description

```
<meta name="description" content="Спорт «Красота, сила» привести тело в порядок!">
```

Ключевые слова представляют собой слова, которые описывают тему сайта или текст, и используются поисковыми системами для обеспечения точного соответствия результатов поиска. Существует несколько областей сайта, в которые необходимо поместить собранные ключевые слова:

- заголовки и описание сайта;
- подзаголовки h1;
- содержимое страниц;
- внутренние и внешние ссылки;
- текстовые описания к изображениям.

На главной странице содержится информация о клубе. В листинге 2 представлен мета-тег keywords для страницы «Главная».

Листинг 2– Ключевые слова для главной страницы

```
<meta name="keywords" content="Спорт, тренажёрный, массаж, тренер, тренировки, растяжка, пресс, мышцы, худеть, питание, массаж тела, массаж ног, массаж головы, массаж рук, Crossfit, Sport">
```

Заголовок должен состоять из характерных страниц ключевых слов, собранных в короткую, читаемую, осмысленную и информативную фразу. Основные требования к заголовку на сайте следующие:

- 1) длина заголовка не длиннее семидесяти символов;
- 2) в заголовке необходимо отразить содержимое страниц;
- 3) использовать в заголовке ключевые слова;
- 4) важные слова вставлять в начале заголовка.

В листинге 3 размещен заголовок для главной страницы сайта.

Листинг 3 – Заголовок для категорий главной страницы

```
<title>Спортивный клуб высшего уровня. </title>
```

При создании дизайна сайта необходимо учитывать структуру семантики. Семантические теги созданы для наглядной разметки веб-мастером контента и удобства распознавания структуры сайта поисковыми системами. Есть несколько наиболее часто используемых тэгов для успеха в SEO-оптимизации:

– article – используется для того, чтобы поисковые роботы могли безошибочно выделять на странице основное смысловое содержание. Расчет разработчиками версии сделан на то, что индексация происходит по главному контенту. Сообщение в блоге или новая статья может быть заключена в тег <<article>>;

– section – указывает подраздел блок контента, таких как article. При этом каждая секция получает индивидуальные HTML-заголовки. Структурирование и сегментация страницы помогают поисковым алгоритмам быстро ориентироваться на просматриваемой веб-странице;

– header – как правило, в тег <<header>> заключается не только название (заголовок) сайта, но и логотип, и навигационная панель;

– footer – в данном теге располагается общая информация об авторских правах, контактные данные владельцев и / или разработчиков сайта, ссылки на другие ресурсы и так далее. Поисковики, желая найти на странице такую информацию, будут целенаправленно идти в этот тег;

– nav – демонстрирует поисковикам архитектурное вашего сайта.

Микроразметка – стандарт семантической оптимизации. Стандарт был представлен поисковиками Google, Bing и Yahoo летом 2011 года. Микроразметка позволяет роботам точнее определять и структурировать информацию на страницах сайтов. Микроразметка выполняется добавлением в HTML-код страницы специальных тегов, в которых размещается различная полезная информация [3].

Микроразметка поделена на словари. Словарь – это набор различных атрибутов для данных. Один из наиболее популярных и использованный в данном проекте – OpenGraph. Основные атрибуты словаря OpenGraph, используемые при SEO-оптимизации данного проекта:

– og:locale – указывает на язык контента и локацию сайта;

– og:type – указывает тип контента на странице (статья, новость, обзор);

– og:title – указывает метатеги title;

– og:description – указывает метатеги description;

– og:url – URL страницы;

– og:image – указываем изображение статьи, обзоры;

– og:site\_name – имя сайта.

Описанные действия по SEO-оптимизации эффективно влияют на развитие ресурса и привлечение новых пользователей. Стандартных методов SEO-оптимизации не существует, поэтому были выбраны наиболее подходящие и как можно менее затратные способы. SEO-оптимизация является самым доступным по стоимости инструментом привлечения клиентов. А эффект от нее может держаться достаточно долгое время, продолжая приносить прибыль вашей компании, даже когда деньги в продвижение уже не вкладываются.

**Заключение.** В данном разделе приводятся методы для SEO-оптимизации, позволяющие найти веб-сервис среди остальных, по ключевым словам, и тегам. Таким образом, поисковые системы с наибольшей вероятностью выдадут данный веб-сервис.

#### ЛИТЕРАТУРА

1. Seo-tag.by. [Электронный ресурс]. Режим доступа: <https://seo-tag.by/services/prodvizhenie-sajtov/seo-optimization>. – Дата доступа: 20.09.2019.
2. Мета-теги description и как его использовать. [Электронный ресурс]. Режим доступа: <https://site-ok.ru/meta-teg-description/>. – Дата доступа: 20.09.2019.
3. Микроразметка: что нужно знать SEO-специалисту о словарях и синтаксисе. [Электронный ресурс]. Режим доступа: <https://netpeak.net/ru/blog/mikrorazmetka-cto-nuzhno-znat-seo-spetsialistu-oslovaryakh-i-sintaksise/>. – Дата доступа: 20.09.2019.

УДК 004.855.5

## СРАВНЕНИЕ РАЗЛИЧНЫХ ПОДХОДОВ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

М.Ю. ВАШКЕВИЧ

(Представлено: канд. физ.-мат. наук, доц. О.Н. ПЕТРОВИЧ)

В статье рассмотрены 3 различных способа обучения нейронной сети. В частности рассматриваются их достоинства и недостатки. А так же наилучшее место для использования, каждого конкретного метода обучения.

**Введение.** Обучить нейронную сеть можно разными способами: с учителем, без учителя, с подкреплением. Каждый из этих алгоритмов обучения имеет свои достоинства и недостатки. В зависимости от задачи, которая стоит перед нейронной сетью лучше использовать тот или иной алгоритм обучения.

В случае когда необходимо выполнить конкретную задачу, и есть большое количество как входных и выходных данных то лучше всего выбрать алгоритм с глубоким обучением так разработчик может наиболее точно настроить алгоритм обучения учитывая вид входных данных и стоящую перед ним задачу.

Результат обучения нейронной сети — кластеризация изображений. При обучении с учителем нейронная сеть обучается на размеченном наборе данных и предсказывает ответы, которые используются для оценки точности алгоритма на обучающих данных. При обучении без учителя модель использует неразмеченные данные, из которых алгоритм самостоятельно пытается извлечь признаки и зависимости.

Обучение с частичным привлечением учителя представляет собой нечто среднее. Оно использует небольшое количество размеченных данных и большой набор неразмеченных. А обучение с подкреплением тренирует алгоритм при помощи системы поощрений. Агент получает обратную связь в виде вознаграждений за правильные действия. Похожим образом происходит дрессировка животных.

Для каждого способа обучения рассмотрим примеры подходящих для него данных и задач.

**Основной раздел. Обучение с учителем** (*supervised learning*) предполагает наличие полного набора размеченных данных для тренировки модели на всех этапах ее построения.

Наличие полностью размеченного датасета означает, что каждому примеру в обучающем наборе соответствует ответ, который алгоритм и должен получить. Таким образом, размеченный датасет из фотографий цветов обучит нейронную сеть, где изображены розы, ромашки или нарциссы. Когда сеть получит новое фото, она сравнит его с примерами из обучающего датасета, чтобы предсказать ответ.

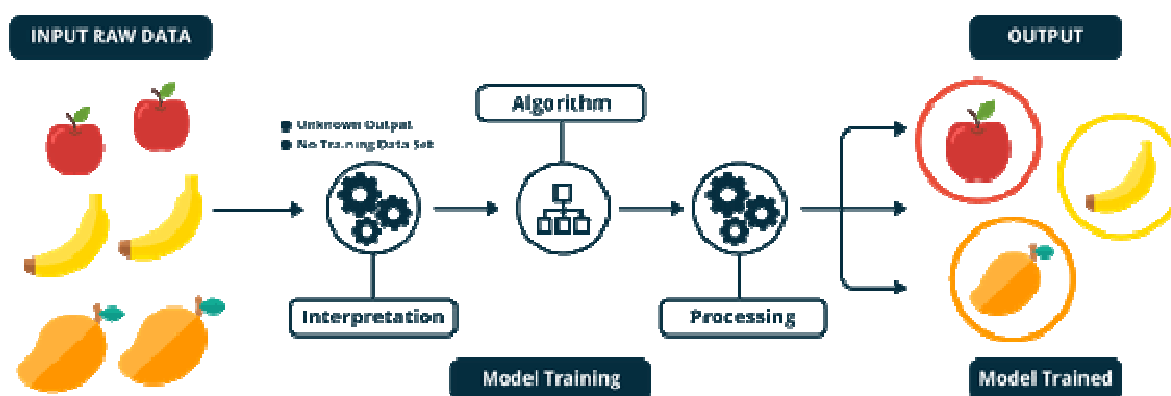


Рисунок. – Пример обучения с учителем

Пример обучения с учителем — классификация (слева), и дальнейшее ее использование для сегментации и распознавания объектов

В основном обучение с учителем применяется для решения двух типов задач: классификации и регрессии.

В задачах классификации алгоритм предсказывает дискретные значения, соответствующие номерам классов, к которым принадлежат объекты. В обучающем датасете с фотографиями животных каждое изображение будет иметь соответствующую метку — «кошка», «коала» или «черепаша». Качество



алгоритма оценивается тем, насколько точно он может правильно классифицировать новые фото с коалами и черепаками.

Более утилитарные задачи машинного обучения задействуют большое число переменных. Как пример, нейронная сеть, предсказывающая цену квартиры в Сан-Франциско на основе ее площади, местоположения и доступности общественного транспорта. Алгоритм выполняет работу эксперта, который рассчитывает цену квартиры исходя из тех же данных.

Таким образом, обучение с учителем больше всего подходит для задач, когда имеется внушительный набор достоверных данных для обучения алгоритма. Но так бывает далеко не всегда. Недостаток данных — наиболее часто встречающаяся проблема в машинном обучении на 2018 год.

**Обучение без учителя** (*unsupervised learning*) идеально размеченные и чистые данные достать не легко. Поэтому иногда перед алгоритмом стоит задача найти заранее не известные ответы. Вот где нужно обучение без учителя.

В обучении без учителя у модели есть набор данных, и нет явных указаний, что с ним делать. Нейронная сеть пытается самостоятельно найти корреляции в данных, извлекая полезные признаки и анализируя их. Кластеризация данных на основе общих признаков. В зависимости от задачи модель систематизирует данные по-разному.

Кластеризация. Даже без специальных знаний эксперта-орнитолога можно посмотреть на коллекцию фотографий и разделить их на группы по видам птиц, опираясь на цвет пера, размер или форму клюва. Именно в этом заключается кластеризация — наиболее распространенная задача для обучения без учителя. Алгоритм подбирает похожие данные, находя общие признаки, и группируют их вместе.

Обнаружение аномалий. Банки могут обнаружить мошеннические операции, выявляя необычные действия в покупательском поведении клиентов. Например, подозрительно, если одна кредитная карта используется в Калифорнии и Дании в один и тот же день. Похожим образом, обучение без учителя используют для нахождения выбросов в данных.

Ассоциации – проверяют признаки с другими признаками. Рассматривая пару ключевых признаков объекта, модель может предсказать другие, с которыми существует связь.

Автоэнкодеры принимают входные данные, кодируют их, а затем пытаются воссоздать начальные данные из полученного кода. Не так много реальных ситуаций, когда используют простой автоэнкодер. Но стоит добавить слои и возможности расширятся: используя зашумленные и исходные версии изображений для обучения, автоэнкодеры могут удалять шум из видеоданных, изображений или медицинских сканов, чтобы повысить качество данных.

В обучении без учителя сложно вычислить точность алгоритма, так как в данных отсутствуют «правильные ответы» или метки. Но размеченные данные часто ненадежные или их слишком дорого получить. В таких случаях, предоставляя модели свободу действий для поиска зависимостей, можно получить хорошие результаты.

**Обучение с подкреплением** (*reinforcement learning*) Данная система строится на системе бонусов и наказаний. За правильное решение некой задачи нейросеть получает данные которые она считает поощрением за неверное решение получает наказание. Так же как люди без обратной связи не смогли бы обучиться ничему новому, так же и нейросеть без понимая какое решение является верным, а какое неверным не смогла бы научиться решать задачи. Обучение с подкреплением действует по тому же принципу. Видеоигры — популярная тестовая среда для исследований.

Результат обучения с подкреплением — «агент» проходит трассу, не выезжая за ее пределы. Далее можно добиваться повышения скорости прохождения трассы.

Агенты ИИ пытаются найти оптимальный способ достижения цели или улучшения производительности для конкретной среды. Когда агент предпринимает действия, способствующие достижению цели, он получает награду. Глобальная цель — предсказывать следующие шаги, чтобы заработать максимальную награду в конечном итоге.

При принятии решения агент изучает обратную связь, новые тактики и решения способные привести к большему выигрышу. Этот подход использует долгосрочную стратегию — так же как в шахматах: следующий наилучший ход может не помочь выиграть в конечном счете. Поэтому агент пытается максимизировать суммарную награду.

Это итеративный процесс. Чем больше уровней с обратной связью, тем лучше становится стратегия агента. Такой подход особенно полезен для обучения роботов, которые управляют автономными транспортными средствами или инвентарем на складе.

Так же, как и ученики в школе, каждый алгоритм учится по-разному. Но благодаря разнообразию доступных методов, вопрос в том, чтобы выбрать подходящий и научить вашу нейронную сеть разбираться в среде.

**Заключение.** В заключении хочется отметить, что каждый подход в обучении нейросети будет показывать себя по-разному и следует учитывать это при выборе метода обучения. Так выбрав наиболее подходящий алгоритм для обучения можно получить более корректные данные и затратить на их получение меньше время.

## ЛИТЕРАТУРА

1. Обучение без учителя [Электронный ресурс]. – Режим доступа [https://ru.wikipedia.org/wiki/Обучение\\_без\\_учителя](https://ru.wikipedia.org/wiki/Обучение_без_учителя). – Дата обращения 24.09.2018.
2. Обучение без учителя [Электронный ресурс]. – Режим доступа [http://www.machinelearning.ru/wiki/index.php?title=Обучение\\_без\\_учителя](http://www.machinelearning.ru/wiki/index.php?title=Обучение_без_учителя). – Дата обращения 24.09.2018.
3. Обучение с учителем [Электронный ресурс]. – Режим доступа [https://ru.wikipedia.org/wiki/Обучение\\_с\\_учителем](https://ru.wikipedia.org/wiki/Обучение_с_учителем). – Дата обращения 24.09.2018.
4. Обучение с учителем [Электронный ресурс]. – Режим доступа [http://www.machinelearning.ru/wiki/index.php?title=Обучение\\_с\\_учителем](http://www.machinelearning.ru/wiki/index.php?title=Обучение_с_учителем). – Дата обращения 24.09.2018.
5. Обучение с подкреплением [Электронный ресурс]. – Режим доступа [https://ru.wikipedia.org/wiki/Обучение\\_с\\_подкреплением](https://ru.wikipedia.org/wiki/Обучение_с_подкреплением). – Дата обращения 24.09.2018.
6. Обучение с подкреплением [Электронный ресурс]. – Режим доступа [http://www.machinelearning.ru/wiki/index.php?title=Обучение\\_с\\_подкреплением](http://www.machinelearning.ru/wiki/index.php?title=Обучение_с_подкреплением). – Дата обращения 24.09.2018.

УДК 004.855.5

## УСКОРЕНИЕ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

**М.Ю. ВАШКЕВИЧ***(Представлено: канд. физ.-мат. наук, доц. О.Н. ПЕТРОВИЧ)*

*В статье рассматривается возможность ускорения обучения нейронной сети с использованием библиотеки Batch Normalization, а также возможные минусы данного ускорения.*

**Введение.** В современном мире нейронные сети находят себе всё больше применений в различных областях науки и бизнеса. Причем чем сложнее задача, тем более сложной получается нейросеть.

Обучение сложных нейронных сетей иногда может занимать дни и недели только для одной конфигурации. А чтобы подобрать оптимальную конфигурацию для конкретной задачи, требуется запустить обучение несколько раз — это может занять месяцы вычислений даже на действительно мощной машине.

В какой-то момент, знакомясь с представленным в 2015 году методом Batch Normalization от компании Google, для решения задачи связанной с распознаванием лиц, удалось существенно улучшить скорость работы нейросети.[1]

**Основной раздел.** В современном мире нейронные сети находят себе всё больше применений в различных областях науки и бизнеса. Причем чем сложнее задача, тем более сложной получается нейросеть.

Обучение сложных нейронных сетей иногда может занимать дни и недели только для одной конфигурации. А чтобы подобрать оптимальную конфигурацию для конкретной задачи, требуется запустить обучение несколько раз — это может занять месяцы вычислений даже на действительно мощной машине.

В какой-то момент, знакомясь с представленным в 2015 году методом Batch Normalization от компании Google, для решения задачи связанной с распознаванием лиц, удалось существенно улучшить скорость работы нейросети.

В данной статье будет представлена возможность совмещения дух актуальных на сегодняшний день задач — это задача компьютерного зрения и машинного обучения. В качестве проектировки архитектуры нейронной сети, будет использоваться Batch Normalization для ускорения обучения нейронной сети. Обучение же нейронной сети (написанной с использованием популярной в рамках computer vision библиотеки Caffe) проводилось на базе из 3 миллионов лиц 14 тысяч различных людей.

В данной задаче необходимо произвести классификацию на 14700 классов. Известна точность классификации на тестовой выборке: 94,5%. При этом для этого потребовалось 420 тысяч итераций обучения — а это 95 часов на видеокарте NVidia Titan X. Изначально для данной нейросети использовались некоторые стандартные способы ускорения обучения:

- 1) Увеличение learning rate
- 2) Уменьшение число параметров сети
- 3) Изменение learning rate в процессе обучения особым способом

Все методы были применены к данной сети, однако в процессе применения данных способов была обнаружена потеря точности обучения.

Рассмотрим классическую нейронную сеть с несколькими слоями. Каждый слой имеет множество входов и множество выходов. Сеть обучается методом обратного распространения ошибки, по батчам, то есть ошибка считается по какому-то подмножеству обучающей выборки.

Стандартный способ нормировки — для каждого  $k$  рассмотрим распределение элементов батча. Вычтем среднее и поделим на дисперсию выборки, получив распределение с центром в 0 и дисперсией 1. Такое распределение позволит сети быстрее обучаться, т.к. все числа получатся одного порядка. Но ещё лучше ввести две переменные для каждого признака, обобщив нормализацию следующим образом (рисунки).

Получим среднее, дисперсию. Эти параметры будут входить в алгоритм обратного распространения ошибки. Тем самым получаем batch normalization слой с  $2*k$  параметрами, который и будем добавлять в архитектуру предложенной сети для распознавания лиц.

На вход в задаче подаётся черно-белое изображение лица человека размером  $50 \times 50$  пикселей. На выходе имеем 14000 вероятностей классов. Класс с максимальной вероятностью считается результатом предсказания.

<b>Input:</b> Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$ ;	
Parameters to be learned: $\gamma, \beta$	
<b>Output:</b> $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$	
$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$	// mini-batch mean
$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2$	// mini-batch variance
$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$	// normalize
$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i)$	// scale and shift

Рисунок. – Обобщение нормализации

Используется 8 свёрточных слоёв, каждый размером 3x3. После каждой свёртки, используется ReLU:  $\max(x, 0)$ . После блока из двух свёрток идёт max-pooling с размером ячейки 2x2 (без перекрытия ячеек). Последний pooling слой имеет размер ячейки 7x7, который усредняет значения, а не берёт максимум. В итоге получается массив 1x1x320, который и подаётся на полносвязный слой.

В новой архитектуре каждый блок из двух свёрток содержит слой Batch Normalization между ними. Также пришлось удалить один свёрточный слой из второго блока, так как добавление новых слоёв увеличило расход памяти графической карты.

При этом был убран Dropout в соответствии с рекомендациями по применению BN авторов оригинальной статьи. Экспериментальная оценка.

Основная метрика — точность, делим количество правильно классифицированных изображений на количество всех изображений в тестовой выборке.

Основная сложность в оптимизации нейросети с помощью слоя Batch Normalization — подобрать learning rate и правильно его изменять в процессе обучения сети. Чтобы сеть сходилась быстрее, начальный learning rate должен быть больше, а потом снижаться, чтобы результат был точнее.

Было протестировано несколько вариантов изменения learning rate (рисунок 2).

Имя	Формула изменения learning rate	Итераций до точности 80%	Итераций до полной сходимости	Максимальная точность
<u>original</u>	$0.01 * 0.1^{\lfloor \frac{\#iter - 1}{50000} \rfloor}$	64000	420000	94,5%
<u>short step</u>	$0.055 * 0.7^{\lfloor \frac{\#iter}{11000} \rfloor}$	45000	180000	86,7%
<u>multistep without dropout</u>	$0.055 * 0.7^{Nsteps}$	45000	230000	91,3%

[x] – целая часть; #iter – номер итерации; Nsteps – шаги, заданные вручную на итерациях: 14000, 28000, 42000, 120000(x4), 160000(x4), 175000, 190000, 210000

Рисунок 2. – Сравнение скорости и точности обучения

График, демонстрирующий процесс обучения: точность на тестовой выборке в зависимости от количества выполненных итераций обучения нейронной сети:

Оригинальная сеть сходится за 420000 итераций, при этом learning rate за всё время изменяется только 2 раза на 150000-ой итерации и на 300000-ой. Такой подход был предложен автором оригинальной сети, и мои эксперименты с этой сетью показали, что этот подход оптимален.

Но если присутствует слой Batch Normalization, такой подход даёт плохие результаты — график long\_step. В результате проведения тестов выяснилось, что лучших показателей можно добиться если изменять learning rate плавно, а потом сделать несколько скачков (график multistep\_no\_dropout). График short\_step показывает, что просто плавное изменение learning rate работает хуже.

В итоге экспериментов был получен результат, что ускорить обучение можно, но точность в любом случае будет немного хуже. Если стоит задача как можно быстрее получить приемлимую точность, то BN как раз может помочь: чтобы достичь точности 80% требуется в 1,4 раза меньше времени по сравнению с оригинальной сетью (45000 итераций против 64000). Это можно использовать, например, для проектирования новых сетей и подбора параметров.

**Заключение.** Обучение с использованием Batch Normalization позволяет существенно сократить количество итераций и как следствие количество времени на обучение нейросети. Но так же это ведёт к понижению точности обучения. Как следствие необходимо выбирать подходит ли этот алгоритм улучшения для поставленной задачи или нет.

#### ЛИТЕРАТУРА

1. Нейросеть [Электронный ресурс]. – Режим доступа <https://geektimes.ru/post/101789/DES>. – Дата обращения: 24.09.2019.
2. Описание нейросетей [Электронный ресурс]. – Режим доступа <https://ru.wikipedia.org/wiki/>. – Дата обращения: 24.09.2019.
3. Описание нейросетей [Электронный ресурс]. – Режим доступа [http://elibr.psuti.ru/Gorozhanina\\_Nejronnye\\_seti\\_uchebnoe\\_posobie.pdf](http://elibr.psuti.ru/Gorozhanina_Nejronnye_seti_uchebnoe_posobie.pdf). – Дата обращения 24.09.2019.

УДК 007.003

**BEAPP – КОНСТРУКТОР ПРИЛОЖЕНИЙ ДЛЯ СООБЩЕСТВ ВКОНТАКТЕ****А.В. ВИНОГРАДОВ, А.С. ДЬЯКОВА**  
(Представлено: Т.М. ГЛУХОВА)

*В статье рассмотрены актуальность социальных сетей в продвижении бизнеса, а также необходимость разработки сервиса по созданию приложений для сообществ ВКонтакте «BEAPP». Были проанализированы основные возможности сервиса и способы его взаимодействия с аудиторией.*

За последние несколько лет сообщества в социальной сети ВКонтакте стали одним из самых важных инструментов для взаимодействия с клиентами. Через сообщество можно узнать информацию о каких-либо предстоящих мероприятиях компании, уточнить всю необходимую информацию, пообщаться с ботом и многое другое. А в недавнем времени ВКонтакте представили платформу приложений сообществ. Эти приложения становятся полноценной заменой веб-сайта или мобильного приложения. Ведь все взаимодействия происходят на уровне клиента ВКонтакте.

В каталоге приложений для сообществ в данный момент находится огромное количество различных решений, но не одно из приложений не позволяет в полной мере сочетать в себе большое количество необходимых функций.

Именно такая задача и поставлена перед «BEAPP»: вывести это взаимодействие на новый уровень.

«BEAPP» – это новое решение для SMM-администраторов социальной сети ВКонтакте. Сервис предоставляет возможность конструировать уникальные приложения-лендинги с многочисленным количеством возможностей для сообществ ВКонтакте.

«BEAPP» поможет справиться с различными задачами:

- Сбор заявок из форм. Создавайте разнообразные формы. Используйте приложение для получения заявок от пользователей.
- Консультация. Используйте приложение для консультирования ваших клиентов, размещайте различные видео, фото и другие интерактивные элементы, которые помогут получить важную информацию вашей аудитории.
- Универсальное решение. Контролируйте каждый аспект вашего приложения используя многофункциональный редактор, чтобы решить любую из поставленных вами задач.

Создание приложения, выполняющего необходимые вам функции, не потребует от вас знаний каких-либо языков программирования! Все происходит просто и интуитивно. После быстрой авторизации с помощью ВКонтакте, которая не требует ввода каких-либо данных, вы попадаете на страницу со списком подключенных вами групп.

Первым делом необходимо подключить ваше сообщество к сервису, сразу после этого автоматически будет создана главная страница приложения.

Каждое приложение сервиса состоит из страниц. Эта возможность позволяет использовать каждую страницу для своих целей. Например, страница с формой заявки на вакансию может продвигаться с помощью таргетированной рекламы и получать заявки. Но при этом, если пользователь откроет приложение через меню в сообществе, ему будет отображаться другая страница, в настройках которой установлена галочка «Стартовая страница приложения».

После создания страницы вы автоматически попадаете в «Редактор BEAPP». Каждая страница приложения состоит из различных блоков, которые можно найти в каталоге.

В данный момент в каталоге блоков доступна одна категория, состоящая из 12 блоков. По блокам в каталоге также доступен поиск. Все блоки предоставляют определенные возможности странице вашего приложения. Добавление блоков происходит по принципу «Drag and drop», достаточно просто переместить блок в необходимое место на странице. Каждому блоку доступны персональные настройки в разделе «Редактирование» и общие настройки дизайна.

Блок «Секция» используется для создания вложенных столбцов с блоками. Таким образом, вы можете создавать сложные макеты. В редактировании блока доступен выбор отображения в соотношении процентов: 66,33; 33, 66; 50, 50; 33, 33, 33; 100; 25, 25, 25, 25.

Блок «Визуальный редактор», один из самых часто используемых, он позволяет форматировать текст, изображения и многое другое. Загрузка всех изображений происходит на сервера ВКонтакте, посредством API – это позволяет пользователю загружать неограниченное количество изображений, не переживая о лимитах.

Редактор также адаптирован и под разработчиков. Если у вас есть знания HTML, JS или CSS, то вы можете использовать продвинутый блок «HTML код». Он позволяет инициализировать код в месте, где будет размещён. Таким образом, можно увеличить доступные возможности редактора. Например, вы можете разместить виджет радио или виджет с погодой на одной из страниц приложения.

Важнейший блок для взаимодействия с клиентом, который доступен в редакторе – «Форма». В редактировании блока есть полноценный конструктор форм с различными полями. Сейчас доступно несколько вариантов полей формы: текстовое поле, многострочное поле, выпадающий список, чекбокс и кнопка отправки. Каждое из полей можно полностью настроить, указать стандартное значение, подсказку для ввода и другое. Эти возможности помогают сконструировать полноценную форму для вашего приложения. Через форму вы можете собирать различные заявки, заказы и многое другое. После отправки формы клиентом, приходит оповещение на подключенные аккаунты в социальных сетях.

Блок «Изображение» позволяет добавлять на страницу изображение с настройкой высоты, ширины и расположения.

Если вам необходимо показать вашему клиенту какое-либо место на карте, то вы можете использовать блок «Google Карты». Например, вы можете указать адреса ваших заведений. В нем можно указать место, которое будет показано на карте, а также изменить увеличения карты и высоту.

Блок «Видео» позволяет добавлять на страницу видеозапись с популярных стриминговых сервисов, таких как YouTube, Вконтакте, Одноклассники, RuTube, Vimeo. Видео добавляется на страницу используя технологию Embed. В редактирование блока необходимо выбрать сервис с видео, указать ссылку на видео и настроить параметры отображения.

Для перенаправления клиента на другие сайты или социальные сети компании можно использовать блок «Кнопка». Редактор позволяет выбрать цвет, размер, дизайн кнопки. Также возможен выбор иконки для каждой кнопки и указание ее расположения и отступа от текста.

Блок «Иконка» позволяет добавлять иконки и полностью настраивать их. Вы можете указать размер, отступ, тип оформления, цвет и многое другое.

Деление вашего приложения на логические части, очень поможет пользователю ориентироваться в нем. Для этого вы можете использовать блок «Разделитель». В настройках блока доступен выбор цвета, стиля, выравнивая, ширины, отступа.

Если вам необходимо уведомить пользователя о каком-либо событии и времени его проведения, то вы можете использовать блок «Таймер». Он позволяет добавить в нужное место вашего приложения виджет обратного отсчета с полной настройкой. Например, вы можете использовать блок для того, чтобы показать, когда закончится акция, или начнется музыкальный фестиваль. В самом блоке можно указывать дату и время конца отсчета, цвета рамок, цвета цифр отсчета.

Блок «Меню» поможет вашим клиентам ориентироваться по страницам приложения или другим ссылкам, которые вы добавляете. Например, вы можете использовать блок, чтобы указать ссылки на все социальные сети. В редактировании доступно добавление пунктов меню.

Для каждого из добавленных в ваше приложение блоков доступны общие настройки дизайна. В данный момент раздел настроек позволяет добавить отступы для блока снаружи и внутри, полностью настроить вид границы блока (стиль, цвет, размер, округление), возможность указать уникальный фон в виде градиента, цвета или изображения, а также цвет текста внутри блока.

Редактор адаптирован под работу на мобильном устройстве и позволяет настроить приложение в довольно короткие сроки времени, если у вас нет доступа к компьютеру.

После наполнения страницы блоками её необходимо сохранить и подключить приложение «BEAPP» в сообщество, чтобы клиенты смогли найти его.

Следующим этапом настройки вашего приложения будет настройка уведомлений для заявок из форм. Для этого в разделе настроек можно найти специальную строку, которую необходимо отправить боту «BEAPP» в Телеграм или ВКонтакте. После этого на аккаунт, с которого пришло сообщение, будут отправляться уведомления с данными клиента и формы, которую он заполнил и отправил, как показано на рисунке 1.

Еще одной главной особенностью «BEAPP», в отличие от других доступных предложений, является взаимодействие с вашими клиентами в оффлайне. Вы можете скачать и распечатать стикер для каждой из созданных страниц, затем наклеить его на дверь или любое другое видное место в вашем заведении. Это позволит привлечь клиентов в ваше приложение. Таким образом, можно собирать различные заявки, давать дополнительную информацию и многое другое! Стикер представляет из себя QR-код, который можно сканировать через приложение ВКонтакте. Пример стикера представлен на рисунке 2.

Если вам необходимо собирать статистику вашего приложения, или же подобрать аудиторию для рекламных компаний, то вы сможете использовать поле со скриптами в настройках сайта. В это поле можно добавить код виджета Яндекс.Метрики, Вконтакте Пикселя.

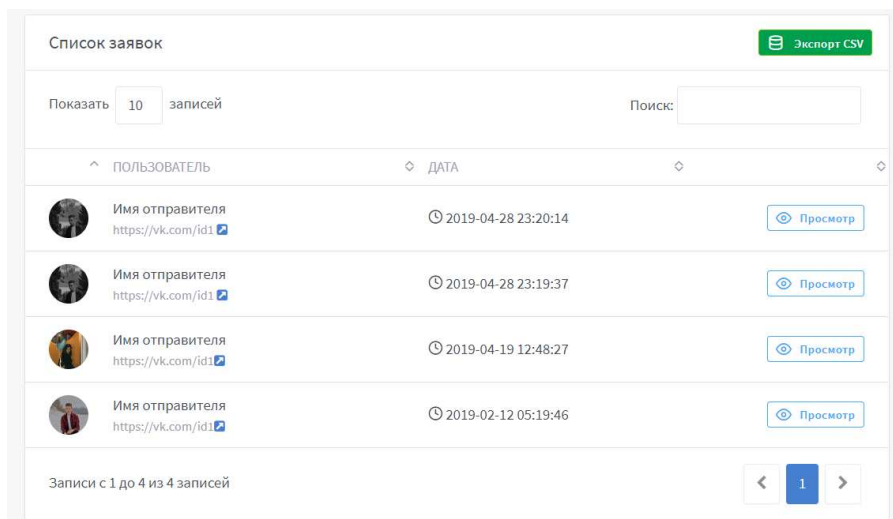


Рисунок 1. – Заявки из форм приложения



Рисунок 2. – Пример размещения QR-кода

«BEAPP» это именно то универсальное решение, которое вы искали для вашего сообщества в социальной сети ВКонтакте. Сервис поможет вам справиться с множеством задач. А самым главным будет то, что вы сможете взаимодействовать с клиентами на абсолютно новом уровне.

#### ЛИТЕРАТУРА

1. Приложения сообществ [Электронный ресурс]. – Режим доступа: <https://vk.com/blog/community-apps>. – Дата доступа: 05.08.2019.
2. ВКонтакте для бизнеса [Электронный ресурс]. – Режим доступа: <https://vk.com/business>. – Дата доступа: 20.08.2019.
3. VK Business — всё о бизнесе ВКонтакте [Электронный ресурс]. – Режим доступа: <https://vk.com/blog/business>. – Дата доступа: 21.08.2019.



УДК 007.003

## ОБЗОР SMM-ИНСТРУМЕНТОВ ДЛЯ ВКОНТАКТЕ

**А.В. ВИНОГРАДОВ, А.С. ДЬЯКОВА**  
(Представлено: Т.М. ГЛУХОВА)

*Рассматриваются основные инструменты для SMM-специалистов. Анализируются их возможности и отличия от аналогов.*

В современном мире для компании не иметь веб-сайт или страницу в социальных сетях означает не существовать вовсе. ВКонтакте является одной из крупнейших социальных сетей в мире и занимает 20 место в мире и 3 в Беларуси по посещаемости (данные Alexa Internet). Поэтому почти все современные компании имеют сообщество в этой социальной сети. Именно с этим и связано появление такого понятия как SMM.

SMM – это часть маркетинговой и коммуникационной стратегии компании, направленной на продвижение товара или услуги через различные социальные сети. Основной упор в SMM-стратегии делается на создании визуального контента, который будет распространяться в социальных сетях. Это позволяет более углублено взаимодействовать с целевой аудиторией вашего бизнеса.

Появление такого понятия как SMM привело к возникновению профессии SMM-специалист. Это тот человек, который управляет присутствием и продвижением бренда в социальных сетях. SMM-специалистом может быть фотограф, модератор или просто человек, который создает уникальный и привлекающий клиентов контент для компании.

Сейчас создается и разрабатывается огромное количество инструментов для SMM-специалистов. Они позволяют упростить их работу и вывести взаимодействие с клиентами компании на новый уровень. Примерами таких инструментов могут быть сервисы, позволяющие создавать чат-боты, проводить рассылки и многое другое.

Для написания статьи были проанализированы популярные SMM-инструменты ВКонтакте и были выделены их уникальные особенности.

Автоматизация общения с клиентами с помощью чат-ботов стала одной из самых популярных тем в 21 веке. Чат-боты уже сейчас можно создать без навыков программирования, используя готовые решения. Несколько сервисов позволяют полностью настроить чат-боты для общения за короткий промежуток времени.

Bot-vk.ru. Это современный сервис для создания чат-ботов групп и сообществ ВКонтакте. Чат-бот, созданный с помощью данного сервиса, может справиться с различными задачами, такими как консультирование пользователей, отправка актуальной информации о статусе заказа, создание тикетов в поддержку, генерализация уникальных скидков для клиента и многое другое. В возможности сервиса входит создание своих шаблонов ответов, команд, которые будут входить в меню бота, огромное количество модулей, каждый из которых можно настроить по вашим требованиям. На данный момент базовые возможности сервиса стоят 100 российских рублей для одного сообщества ВКонтакте. Пример общения пользователя с ботом представлен на рисунке 1.

Robochat.io. Робочат – это сервис, который поможет автоматизировать поддержку и продажи в вашем сообществе ВКонтакте. Сервис не сложен в освоении, и через пару минут использования вы создадите готового бота. Использование сервиса может быть бесплатным, если в вашем сообществе менее чем 50 тысяч подписчиков. В возможности проекта входят следующие функции:

- создание ответов на сообщения с определенными этапами;
- использование переменных в сообщениях;
- полный анализ пользователей бота;
- таргетинговые сообщения (в зависимости от параметров пользователя);
- API для взаимодействия с сервисом;
- плагины.

В числе клиентов Робочата — Мегафон, Телеканал ТВ-3, Ozon, Амедиатека, Booking Machine и другие официальные сообщества известных компаний.

Реклама окружает обычного человека почти везде в повседневной жизни, социальные сети не стали исключением и являются мощнейшей площадкой для продвижения. Например, музыкальные лейблы используют сообщества ВКонтакте для продвижения своих исполнителей. Существуют сервисы, которые автоматизируют этот процесс.

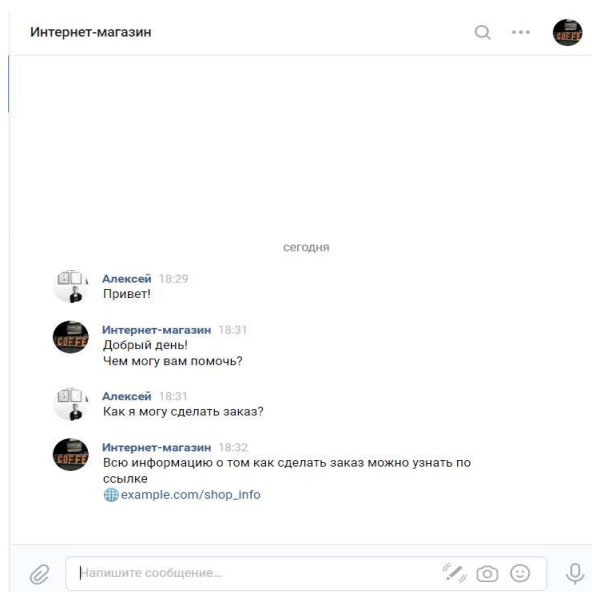


Рисунок 1. – Пример общения с ботом

Adstools.ru. Adstools – это многофункциональная платформа, разработанная для рекламодателей и владельцев сообществ ВКонтакте. Этот инструмент призван автоматизировать покупку и дальнейший анализ статистики вашей нативной рекламы. Если вы рекламодатель, то вам необходимо добавить рекламный пост на сайт и затем выбрать площадки, на которых он будет продвигаться. После публикации поста будет собираться статистика и другая информация.

Еще одним важнейшим инструментом для работы с рекламой в группах ВКонтакте является сайт Adspoiler.com. Сервис предназначен для нахождения выгодных рекламных площадок с детальной аналитикой по каждому посту, слежки за конкурентами, предусмотренной экономии рекламного бюджета. Адспойлер обладает большим списком возможностей, среди которых:

- подробная аналитика групп;
- точная статистика постов;
- фильтры поиска площадок;
- выгрузка списка id пользователей и группы;
- всегда актуальная информация.

Динамическая обложка сообщества – это новый формат контента, позволяющий отображать всю актуальную информацию, которая необходима для пользователей. В данный момент существует несколько популярных сервисов, которые позволяют создать динамическую обложку сообщества.

DuCover.ru. DuCover - это сервис предоставляющий визуальный редактор (конструктор), в котором можно настроить свою обложку для сообщества. Для того, чтобы начать пользоваться сервисом - нужно просто подключить сообщество и настроить обложку. Сервис обладает огромным количеством возможностей, из которых можно выделить:

- вывод погоды на обложку;
- пожертвования;
- JSON взаимодействие с виджетами;
- смена обложек по времени суток;
- пользовательские шрифты;
- курс валют;
- конкурсы.

Сервис предоставляет свои базовые услуги бесплатно, но за 180 российских рублей в месяц можно приобрести подписку с расширенным функционалом.

NeuTeaser.ru. «Эй, тизер!» — это конструктор дизайна для социальных сетей. Этот сервис поможет создавать тизеры, посты, сторис и другой графический контент. Сайт представляет из себя конструктор изображений с возможностью настройки шрифта, цвета, изображения и другого. В данный момент на сайте есть огромное количество шаблонов логотипов, обложек, историй.

«BEAPP» – это новое решение для SMM-администраторов социальной сети ВКонтакте. Сервис предоставляет возможность конструировать уникальные приложения-лендинги с многочисленным количеством возможностей для сообществ ВКонтакте. Пример использования конструктора

BEAPP представлен на рисунке 2.

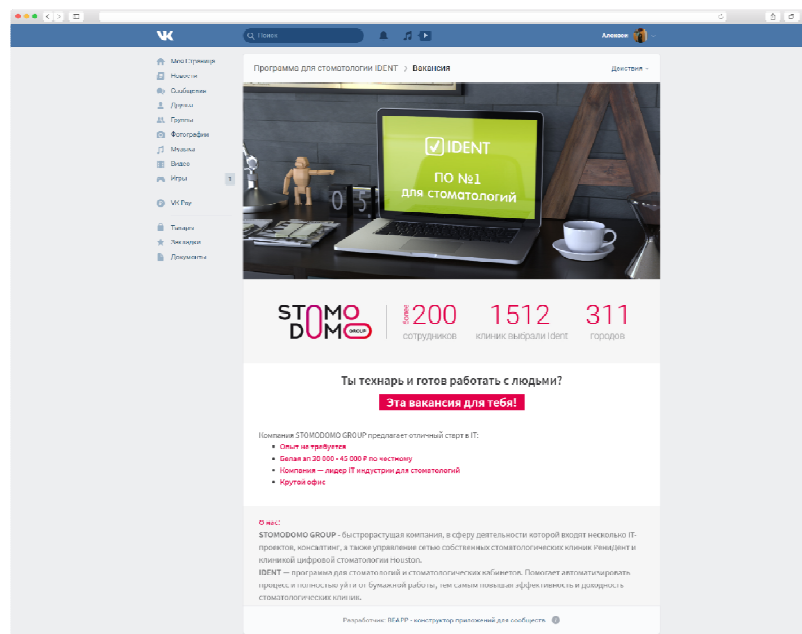


Рисунок 2. – Пример приложения созданного через конструктор «BEAPP»

«BEAPP» поможет справиться с различными задачами:

- Сбор заявок из форм. Создавайте разнообразные формы. Используйте приложение для получения заявок от пользователей.
- Консультация. Используйте приложение для консультирования ваших клиентов, размещайте различные видео, фото и другие интерактивные элементы, которые помогут получить важную информацию вашей аудитории.
- Универсальное решение. Контролируйте каждый аспект вашего приложения используя многофункциональный редактор, чтобы решить любую из поставленных вами задач.

Проанализировав и сравнив использование некоторых сервисов, «BEAPP» показал более высокие результаты конверсии в отличие от схожего сервиса «Гаммаюн».

#### ЛИТЕРАТУРА

1. Alexa - Top Sites in Belarus - Alexa [Электронный ресурс]. - Режим доступа: <https://www.alexa.com/topsites/countries/BY>. - Дата доступа: 06.09.2019.
2. Стратегия SMM продвижения, конкурентный анализ групп в соц сетях [Электронный ресурс]. - Режим доступа: <https://www.intervolga.ru/adv/smm-strategy/>. - Дата доступа: 07.09.2019.
3. Маркетинг в социальных сетях [Электронный ресурс]. - Режим доступа: [https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D0%BA%D0%B5%D1%82%D0%B8%D0%BD%D0%B3\\_%D0%B2\\_%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D1%85\\_%D1%81%D0%B5%D1%82%D1%8F%D1%85/](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D0%BA%D0%B5%D1%82%D0%B8%D0%BD%D0%B3_%D0%B2_%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D1%85_%D1%81%D0%B5%D1%82%D1%8F%D1%85/). - Дата доступа: 07.09.2019.

УДК 613.691

**ОСОБЕННОСТИ ОБУЧЕНИЯ ДЕТЕЙ ДОШКОЛЬНОГО И ШКОЛЬНОГО ВОЗРАСТА  
ПРИ РАБОТЕ С ЭВМ****А.Н. ВОЛКОВ***(Представлено: Т.С. РУДЬКОВА)*

*В данной статье рассматриваются правила обучения детей дошкольного и школьного возраста при работе с ЭВМ и особенности графического интерфейса приложения для детей.*

**Введение.** Все чаще для обучения детей в школах или детских прибегают к использованиям мобильных устройств – это позволяет сократить время на обслуживание, это более дешевое и долговечное решение, но, самое главное, это позволяет уменьшить влияние человека, как педагога. Вместе с тем появляется необходимость изучения негативного влияния ЭВМ на организм детей и особенностей привлечения внимания ребенка.

**Основной раздел.** Персональные компьютеры (ПК) – это источник электромагнитных излучений. Электромагнитное излучение, создаваемое компьютерами, при определенных условиях может представлять повышенную опасность для детей, подростков и взрослых. С целью снижения воздействия ПК на пользователя рекомендуется ограничение времени работы за компьютером, обязательные паузы во время работы и пр.

При организации рабочего места пользователя ПК необходимо иметь в виду, что минимальное расстояние от экрана до оператора около 50-70 см (длина вытянутой руки), а ближайших рабочих мест от боковой и задней стенок монитора – не менее 1,5 м, клавиатура и руки оператора также должны быть расположены на максимально возможном расстоянии от монитора.

Помещение, где устанавливается ПК, должно быть оборудовано защитным заземлением (занулением) и двухполюсной розеткой (с заземленным контактом), подключенной к электрической сети переменного тока промышленной частоты напряжением 220 В и частотой 50 Гц.

Приоритетными и наиболее безопасными при использовании являются жидкокристаллические мониторы, входящие в состав ПК.

По мнению специалистов Республиканского центра гигиены, эпидемиологии и общественного здоровья (ГУ РЦГЭиОЗ), продолжительность непрерывной работы на компьютере для детей различного возраста должна быть следующей:

- 5 лет – не более 7 минут;
- 6 лет – 10 минут;
- 7-9 лет – 15 минут;
- 10-12 лет – 20 минут;
- 13-14 лет – 25 минут;
- 15-16 лет – 30 минут.

Кроме того, между занятиями, если они предполагают более длительное нахождение ребёнка перед экраном, необходимо делать перерывы не менее 10 минут [1].

Светлые и яркие оттенки, вызывают позитивные эмоции. Доказано, что хорошее, приподнятое настроение можно создать, используя также цвет слоновой кости – светлый с золотистым оттенком, светло-бежевый, и нежно-голубой цвет [2].

Дети интересуются яркими цветами. Было проведено много разных исследований, которые показали, что в детстве ребенок много раз меняет свое предпочтение к цветовой гамме. Большинство детей до десяти лет обожают красные, розовые или желтые цвета. Как только ребенку исполняется десять лет, он уже может полюбить голубой цвет со всеми его оттенками.

Согласно исследованиям, девочки отдают предпочтение розовому, лавандовому, фиолетовому цветам. Мальчикам же больше нравятся темные и синие цвета [3]. На рисунке 1 продемонстрированы цветовые гаммы для детей с градацией по полу.

Если приложение рассчитано на детей любого пола и возраста - оптимальным решением для этого является интерфейс, который бы удовлетворял все цветовые запросы. На рисунке 2 продемонстрирована цветовая палитра цветов для детей любого пола.

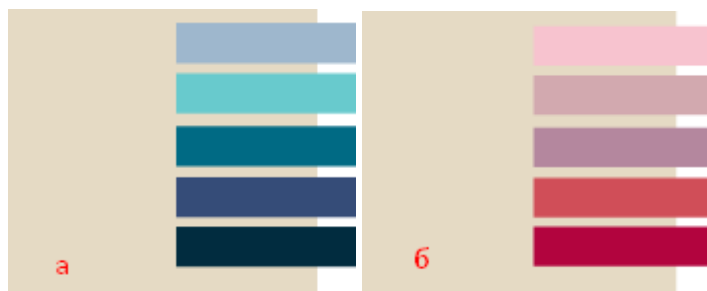


Рисунок 1. – Цветовые предпочтительный для детей:  
а – мальчики; б – девочки



Рисунок 2. – Цветовая гамма для детей любого пола

**Заключение.** В данной статье рассмотрены особенности обучения детей при работе с ЭВМ, а также некоторые аспекты выбора цветовой гаммы для интерьера или графического интерфейсам приложения.

#### ЛИТЕРАТУРА

1. Интернет-портал Helper [Электронный ресурс]. – Режим доступа: <http://www.helper.by/kompyuteri-i-zdorove.html>. – Дата доступа: 15.05.2019.
2. Интернет-портал Lock-Collor [Электронный ресурс]. – Режим доступа: <https://lookcolor.ru/populyarnye-ottenki/cvet-slonovoj-kosti-ajvori-chast-vtoraya/>. – Дата доступа: 15.05.2019.
3. Киреева, З.А. Исследование цветовых предпочтений человека и их взаимодействие с личностными особенностями / З.А. Киреева // Гуманитарные науки. – С. 98-99.

УДК 004.855.6

## РАЗВИТИЕ НЕЙРОННЫХ СВЕРТОЧНЫХ СЕТЕЙ В СОВРЕМЕННОМ МИРЕ

А.Н. ВОЛКОВ

(Представлено: Т.С. РУДЬКОВА)

*В данной статье рассматриваются основные направления развития машинного обучения и нейронных сетей, области их применения и уровень развития в современном мире.*

**Введение.** Очень часто на сегодняшний день можно услышать словосочетание «искусственный интеллект» в повседневной жизни. Машинное обучение является очень перспективной сферой развития информатики – это доказывают все новые изложения энтузиастов со всего мира, алгоритмы которых выдают поистине невероятные результаты: создание стихов и текстов песен, хоть и весьма посредственных, до создания фотографий людей которых никогда не существовало.

**Основной раздел.** Русский программист Алексей Приходько, будучи глухим с рождения, создал систему, которая управляет компьютером с помощью жестов и переводит язык людей с нарушением слуха в текст.

Разработанный алгоритм не нуждается в датчиках и дополнительных устройствах – машина работает только при помощи компьютерного зрения. Камера распознает руку в кадре, создает ее виртуальную модель и выводит символ на экран. Например, увидев кулак, нейронная сеть выдает букву «а». Помимо перевода нейросеть двигает курсор на экране — без использования мыши, только отслеживая движения ладони.

Наиболее интересное и, в тоже время очень опасное, направление развития искусственного интеллекта – «дипфейки». Это методика синтеза изображения человека, использующаяся для соединения и наложения существующих изображений и видео на исходные изображения или видеоролики [1]. Исследователи загрузили нейросети по одной фотографии и аудиозаписи, и Эйнштейн начал читать лекцию, а Распутин запел песню Бейонсе [2].

Данный пример показывает, что все одного фото может быть недостаточно для реалистичной мимики лица, но если загрузить сети коллекцию снимков человека, то получается поразительно естественная картинка даже в движении [3].

Дизайнер Константин Жабинский собрал 100 тысяч фото несуществующих людей и выложил их на «Google Диск». В базе есть люди с разными выражениями и формами лица, разного возраста и этнической принадлежности [4]. На рисунке изображены примеров таких «дипфейков».



Рисунок. – Фото людей сгенерированных нейронной сетью

Широта вредоносного использования «дипфейков» стала настолько велика, что исследователи из Университета Южной Калифорнии создали инструмент, который способен распознать несуществующую личность на видео с вероятностью 96%. Инструмент исследователей накладывает кадры ролика друг на друга и ищет любые возможные несоответствия в мимике и движениях головы. А они обычно есть — иначе на создание одного «дипфейка» уходило бы слишком много времени.

На сегодняшний день нейронное обучение применяют при разработке программного обеспечения для распознавания окружающей среды, фотоинформации и видеoinформации. Условно данную область называют компьютерным зрением. Однако нейронное обучение не ограничивается только картинками, его также можно использовать для работы с аудио- и дактилоскопическими данными (распознавание отпечатка пальца, голоса).

Компания «Google» продемонстрировала нейронную сеть, которая может определять болезнь по снимку кожи. В нейронную сеть загрузили пятьдесят тысяч случаев из практики дерматологов и научили его находить двадцать шесть болезней. Точность распознавания – до 92%.

Искусственный интеллект от компании «OpenAI» генерирует тексты на любую из заданных тем, которые по содержанию могут составлять конкуренцию любому писателю. Для создания такого инструмента понадобилось полтора миллиарда параметров и большая база данных. Разработчики обучали нейронную сеть с помощью популярной социальной сети, откуда собирали тексты с высоким рейтингом читателей. В итоге лаборатория отказалась выкладывать код и показала лишь малую его часть [5].

**Заключение.** В данной статье рассмотрены основные пути развития нейронных сетей в современном мире. Итогом данной статьи можно считать то, что потенциал развития информатики не исчерпан. Появляются все новые усовершенствованные системы в областях: медицины, литературы и мобильной разработки.

#### ЛИТЕРАТУРА

1. Wikipedia.org [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Deepfake>. – Дата обращения 19.09.2019.
2. GoogleInc. YouTube видеохостинг [Электронный ресурс]. – Режим доступа: [<https://www.youtube.com/watch?v=9Ctm4rTdVTU>] – Дата обращения 19.09.2019.
3. GoogleInc. YouTube [Электронный ресурс]. – Режим доступа: [[https://www.youtube.com/watch?v=HG\\_NZpkttXE&feature=youtu.be&has\\_verified=1](https://www.youtube.com/watch?v=HG_NZpkttXE&feature=youtu.be&has_verified=1)] – Дата обращения 19.09.2019.
4. GoogleInc. Google Disk. Облачное хранилище [Электронный ресурс] – Режим доступа: [<https://drive.google.com/drive/folders/1wSy4TVjSvtXeRQ6Zr8W98YbSuZXrZrgY>] – Дата обращения: 19.09.2019.
5. OpenAI [Электронный ресурс]. – Режим доступа: [<https://openai.com/blog/better-language-models/#sample8>] – Дата обращения: 19.09.2019.

УДК 069.53+004.41

**СОЗДАНИЕ ПРОГРАММНОЙ ОБОЛОЧКИ ИНФОРМАЦИОННОЙ СЕНСОРНОЙ ПАНЕЛИ  
«ПОЧЁТНЫЕ ГРАЖДАНЕ ГОРОДА ПОЛОЦКА»****В.В. ГАВРИЛОВ***(Представлено: канд. физ.-мат. наук, доц. А.Ф. ОСЬКИН)*

*В статье рассматриваются вопросы разработки (выбор технологии, определение функциональных возможностей) и создания программной оболочки информационной сенсорной панели (инфокиоска) «Почётные граждане города Полоцка» для одноимённой выставки Краеведческого музея — филиала Национального Полоцкого историко-культурного музея-заповедника.*

В 2017 году в Краеведческом музее Полоцка, филиале Научно-исследовательского и просветительского учреждения культуры «Национальный Полоцкий историко-культурный музей заповедник», начала работать стационарная выставка «Почётные граждане города Полоцка». Выставка рассказывает посетителям о жизни и деятельности 36-ти выдающихся граждан города. Для представления большого количества дополнительного материала различного типа (текстового, графического, аудио и видео) о Почётных гражданах ещё на этапе разработки концепции выставки было принято решение о создании и размещении в экспозиции информационной сенсорной панели (инфокиоска) [1]. Научными сотрудниками Краеведческого музея был собран и подготовлен большой объём информации о персоналиях на трёх языках (русском, белорусском и английском) для размещения в ней. Одновременно со сбором информации было реализовано аппаратное решение информационной сенсорной панели и её внедрение в экспозицию стационарной выставки. Самостоятельное создание аппаратного решения и отказ от готового напольного информационного киоска в пользу собранной на основе 27" сенсорного монитора Пуама и мини ПК Intel NUC настенной информационной сенсорной панели не только позволили сэкономить значительные денежные средства, место в небольшой по площади экспозиции выставки, но и расширили возможности её использования. Было реализовано подключение панели к мультимедиа проектору с выводом изображения на подвесной экран, что позволило использовать её для мероприятий и занятий, проводимых на базе стационарной выставки.

Непосредственно работа над программной оболочкой информационной панели началась в 2018 году. Поставленные задачи определили её функциональные возможности. Панель, размещённая в экспозиции выставки, решила следующие задачи:

- представление посетителям аудио- и видеоматериалов, созданных сотрудниками музея о Почётных гражданах;
- показ богатого фотоматериала личных архивов Почётных граждан;
- демонстрация посетителям большого количества музейных предметов из фондов музея-заповедника, не вошедших в основную экспозицию по тем или иным причинам;
- использование различных мультимедиа материалов при проведении тематических лекций, занятий и мероприятий, проходящих на базе стационарной выставки, путём вывода изображения и звука на проектор с выдвижным экраном;
- представление всей информации на различных языках [2, с. 43].

Предъявленные технические требования к разрабатываемой оболочке информационной сенсорной панели, такие как: поддержка различных операционных систем (семейства Windows и Linux); вывод текстовой, графической, аудио- и видеoinформации непосредственно из файлов в форматах, созданных пользователями без перекодировки и конвертации; возможность не только добавления новой информации, но и оперативного изменения уже внесённой, опять же посредством привычного научным сотрудникам программного обеспечения, будь-то Microsoft Word или LibreOffice для текстовых документов, Windows Movie Maker для создания видео в формате wmv (Windows Media Video) и обусловили выбор технологий при создании программного продукта.

Реализация оболочки осуществлялась в визуальной среде разработки AutoPlay Media Studio версии 8.5 [3]. Возможности данного программного продукта в полной мере соответствуют предъявленным техническим требованиям. Данная визуальная среда разработки позволяет создавать приложения для всех актуальных версий Windows (от XP до 10) и позволяет запускать созданные приложения в Linux под WINE (среда выполнения приложений для Windows в Linux). Встроенная поддержка скриптового языка программирования Lua версии 5.1 может выполнять код из внешнего файла, а не только включённого непосредственно в состав исполняемого модуля [4]. Это позволяет производить отладку логики работы сенсорной оболочки в реальном времени непосредственно на самом устройстве, так же в дальнейшем



позволит изменять и расширять функциональность средствами только языка Lua, без развёртывания среды AutoPlay Media Studio. Большие мультимедийные возможности, простота развёртывания созданного программного продукта и большой выбор расширений и дополнений (plugin и addon) не в последнюю очередь повлияли на выбор данной визуальной среды разработки (рис. 1).

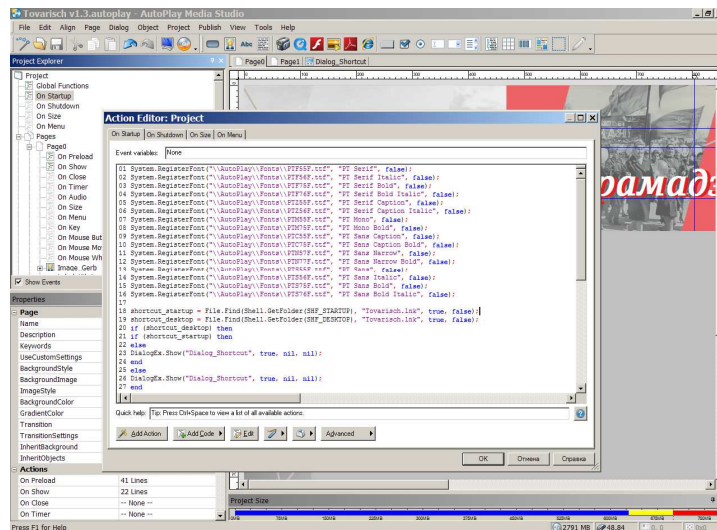


Рисунок 1. – Интерфейс визуальной среды разработки AutoPlay Media Studio

Для представления информации о каждом из Почётных граждан реализованы основные блоки размещения различных типов контента (рис. 2):

- текстовые (фамилия, имя, отчество и годы жизни; биография персоналии);
- графические (портретная фотография Почётного гражданина; фотоальбом — личные фотографии с текстовыми пояснениями; изображения личных предметов, не вошедших в экспозицию, из фондов музея-заповедника);
- мультимедиа (аудиозапись, озвучивающая представленную текстовую информацию на странице о Почётном гражданине, позволяющая одновременно с прослушиванием просматривать графический материал; видеofilm или интервью с ним) [2, с. 44–45].



Рисунок 2. – Основной информационный экран о персоналии программной оболочки информационной сенсорной панели «Почётные граждане города Полоцка» с подключёнными блоками контента различного типа

Подключение блоков с различными типами информации, как и подключения разных языковых версий, в оболочке происходит автоматически при добавлении сотрудниками соответствующего контента. Созданные с помощью внешних редакторов текстовые, графические, аудио- или видеофайлы помещаются непосредственно в файловую структуру оболочки, в папки соответствующих персоналий, а программа уже проверяет их наличие и согласно заданным настройкам подключает различные типы контента, если они присутствуют, в блоки пользовательского интерфейса в соответствии с заданной логикой размещения (листинг).

Данная реализация характеризуется лёгкостью внесения и изменения информации, что позволяет научным сотрудникам музея сосредоточиться непосредственно на создании качественного контента, а не на освоение тонкостей работы и обслуживания программой оболочки.

```

found_img = File.Find( "AutoPlay\\Docs\\"..tovarisch.."\\img\\", "*.jpg", false, false);
if (found_img) then
  albom = INIFile.GetValue("AutoPlay\\Docs\\"..tovarisch.."\\..tovarisch.."_"..index.."ini", "albom", "0");
  Label.SetText("Label_FotoAlbom", albom);
  Hotspot.SetEnabled("Hotspot_FotoAlbom", true);
end

found_audio = File.Find("AutoPlay\\Docs\\"..tovarisch.."\\..tovarisch.."_"..index.."wma", false, false);
if (found_audio) then
  Plugin.SetVisible("Audio", true);
  MediaPlayer.Load("Audio", "AutoPlay\\Docs\\"..tovarisch.."\\..tovarisch.."_"..index.."wma");
  audio_play = true;
end

found_video = File.Find("AutoPlay\\Docs\\"..tovarisch.."\\", "*.wmv", false, false);
if (found_video) then
  Plugin.SetVisible("Video", true);
  MediaPlayer.Load("Video", found_video[1]);
end

```

**Листинг. – Пример кода Lua, проверяющего наличие соответствующего контента (графического, аудио, видео) и подключающего его, если он найден**

Проектирование и разработка графического интерфейса оболочки изначально производилась для конкретной информационной сенсорной панели с разрешением Full HD 1920×1080. Однако позже было решено добавить поддержку других разрешений. С помощью языка Lua была решена задача изменения размещения графических элементов интерфейса для разнообразных разрешений пропорций экрана как 16:9, так и 4:3.

За время эксплуатации информационной сенсорной панели в программу оболочки вносились изменения и дополнения, на сегодняшний день актуальная версия оболочки 1.3. Так, была добавлена возможность прослушивания аудиоинформации при просмотре всего графического материала о персоналии; использование собственных шрифтов, регистрируемых в систему при запуске оболочки; упрощено развёртывание оболочки на инфокиоске — размещение ярлычков на рабочем столе, добавление в автозапуск операционной системы; улучшена функция добавления языковых версий контента и т. д.

Информационная сенсорная панель «Почётные граждане города Полоцка», являясь неотъемлемой частью одноимённой стационарной выставки, позволяет посетителю ознакомиться более полно с информацией о людях, внесших огромный вклад в развитие города. Представленные в ней текстовые, графические, аудио- и видеоматериалы открывают различные аспекты жизни и творчества почётных граждан города Полоцка. Удобный сенсорный интерфейс и разнообразность контента позволили ей занять важное место в экспозиции выставки.

Программное решение, применённое для создания данной оболочки, позволяет использовать её, наполненную соответствующим контентом, не только в этой экспозиции, но и для других временных выставок.

#### ЛИТЕРАТУРА

1. Воднева, I. П. Навуковая канцэпцыя выстаўкі «Ганаровыя грамадзяне горада Полацка» / I. П. Воднева, С. В. Гаўрылава // Матэрыялы навукова-практычнай канферэнцыі : па выніках

- навукова-даследчай работы супрацоўнікаў Нацыянальнага Полацкага гісторыка-культурнага музея-запаведніка ў 2016 г.) / уклад. Т. У. Явіч.— Мінск: Медысонт, 2019.— С. 27—30.
2. Гаврилов, В. В. Информационная сенсорная панель «Почётные граждане города Полоцка» в экспозиции стационарной выставки Краеведческого музея // Матэрыялы навукова-практычнай канферэнцыі: (па выніках навукова-даследчай работы супрацоўнікаў Нацыянальнага Полацкага гісторыка-культурнага музея-запаведніка ў 2018 г.).— Мінск: Медысонт, 2019.— С. 40—48.
  3. Indigo Rose — Software Deployment Tools for Windows Developers. — [Электронны рэсурс]. — Режим доступа: <https://www.indigorose.com/autoplay-media-studio/>. — Дата доступа: 20.09.2019.
  4. Gavrilov V. Lua multiparadigm programming language: application and opportunities / Gavrilov V., Oskin A. // EUROPEAN AND NATIONAL DIMENSION IN RESEARCH. TECHNOLOGY = ЕВРОПЕЙСКИЙ И НАЦИОНАЛЬНЫЙ КОНТЕКСТЫ В НАУЧНЫХ ИССЛЕДОВАНИЯХ : Electronic collected materials of XI Junior Researchers' Conference, Novopolotsk, May 23—24, 2019 / Polotsk State University; ed. D. Lazouski [et al.].— Novopolotsk, 2019.— P. 164—165.

УДК 069.53+004.514

## РАЗРАБОТКА ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ ПРОГРАММНОЙ ОБОЛОЧКИ ИНФОРМАЦИОННОЙ СЕНСОРНОЙ ПАНЕЛИ «ПОЧЁТНЫЕ ГРАЖДАНЕ ГОРОДА ПОЛОЦКА»

**В.В. ГАВРИЛОВ**

(Представлено: канд. физ.-мат. наук, доц. А.Ф. ОСЬКИН)

В статье рассматриваются вопросы разработки и создания графического интерфейса пользователя программной оболочки информационной сенсорной панели (инфокиоска) «Почётные граждане города Полоцка» Краеведческого музея — филиала Национального Полоцкого историко-культурного музея-заповедника.

В 2017 году в Краеведческом музее Полоцка — филиале Национального Полоцкого историко-культурного музея-заповедника открылась стационарная выставка «Почётные граждане города Полоцка», представляющая историю послевоенного (с 1962 года) города по сегодняшний день через жизнь и творчество Почётных граждан. Для представления большого количества дополнительной информации о персоналиях была создана и установлена информационная сенсорная панель.

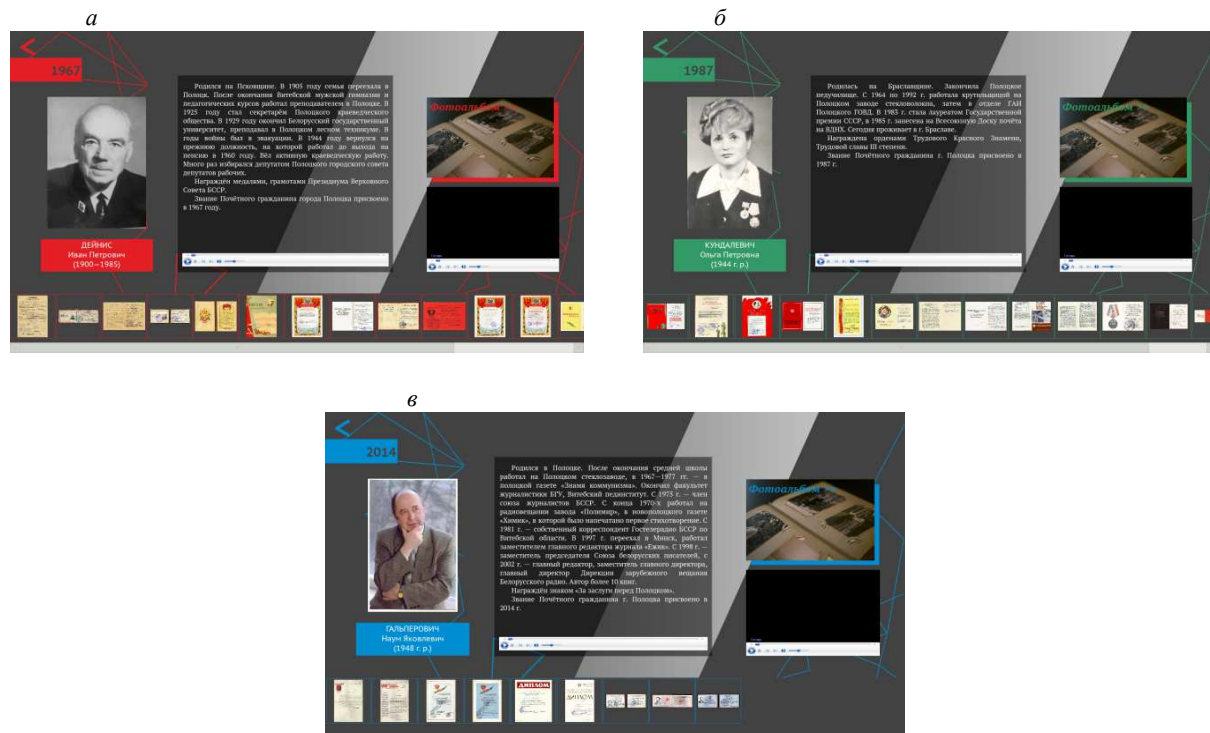
Материалы, отобранные для размещения в информационной сенсорной панели, — а это фото-портреты почётных граждан, документы, заверяющие факт присвоения звания, удостоверения, почётные грамоты, фотографии наград и т. д. — также как и общая тематика и художественное решение выставки, предопределили общую форму их представления. Начальный экран оболочки был создан в виде Доски почёта, классический вид которой был заложен в советское время, на нём размещены портретные фотографии с фамилией, именем и отчеством Почётного гражданина (рис. 1).



**Рисунок 1. – Начальный экран программной оболочки информационной сенсорной панели «Почётные граждане города Полоцка»**

Разработка графического интерфейса пользователя программной оболочки информационной сенсорной панели «Почётные граждане города Полоцка» осуществлялась в визуальной среде разработки AutoPlay Media Studio [1].

Общее дизайнерское решение оболочки инфокиоска было предопределено художественным решением всей экспозиции стационарной выставки. Для начального и основных экранов программной оболочки использовались элементы художественного решения, созданного минскими дизайнерами для стен экспозиции, что позволило интерактивной сенсорной панели стать единым целым со всей экспозицией стационарной выставки. Цветовое решение разных разделов выставки было перенесено в интерфейс программной оболочки. Все персоналии были разбиты на три группы по времени присвоения им почётного звания. Каждая группа имеет своё цветовое решение для фона, интерактивных элементов и элементов навигации основного экрана, в соответствии с художественным решением выставки (рис. 2).



**а** – с 1962 по 1967 г.; **б** – с 1970 по 1995 г.; **в** – с 2003 по сегодняшний день

**Рисунок 2. – Основной информационный экран оболочки с разным цветовым решением графических элементов интерфейса в соответствии со временем присвоения звания Почётного гражданина персоналиям [2]**

Для представления информации о каждом из Почётных граждан реализованы основные блоки размещения различных типов контента.

1) Текстовый контент включает в себя блоки: фамилия, имя, отчество и годы жизни; биография Почётного гражданина.

Для шрифтового решения текстовых блоков и всей программной оболочки инфокиоска была использована бесплатная шрифтовая система PT Sans, PT Serif и PT Mono, разработанная фирмой Пара-тайп [3]. Чтобы не зависеть от наличия данного набора шрифтов в операционной системе конкретного информационного киоска, они были внедрены в ресурсы программной оболочки и регистрируются в системе при её запуске, а при выходе регистрация удаляется (после перезагрузки системы).

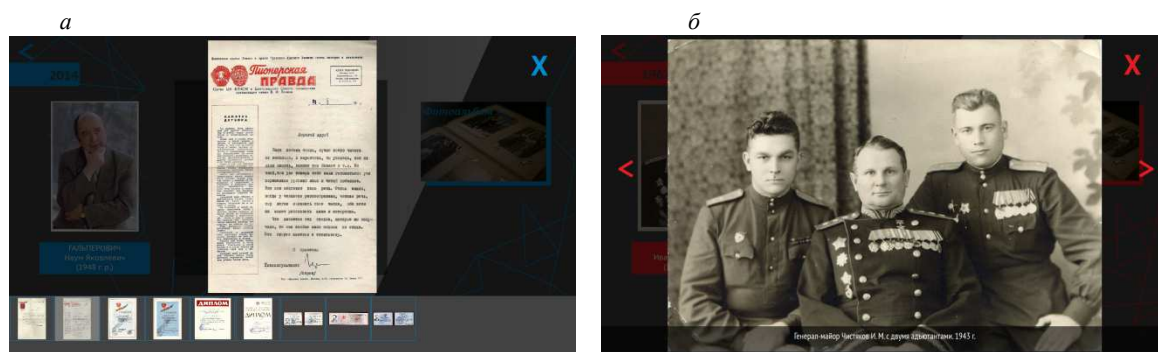
2) Графический контент представлен несколькими блоками: портретная фотография Почётного гражданина; фотоальбом — личные фотографии с текстовыми пояснениями; изображения личных предметов, не вошедших в экспозицию, из фондов музея-заповедника.

Наличие графического материала нескольких типов обусловило использование двух разных модулей для его отображения. Так, для представления музейных предметов, не вошедших в экспозицию, на основном экране был подключён модуль ThumbList Object Plugin (рис. 3, а), а для представления архива личных фотографий персоналии и аннотаций к ним было создано слайд-шоу с использованием модуля SlideShow Object (рис. 3, б)

3) Блоки мультимедиа: аудиозапись, озвучивающая представленную биографию Почётного гражданина, с возможностью одновременного просмотра графического материала; видеоконтент (фильм о персоналии или интервью с ней).

Для представления мультимедиа (как видео-, так и аудиоконтента) в оболочке используется модуль Media Player Object Plugin со стандартным, привычным для всех пользователей операционной системы Windows интерфейсом Windows Media Player, что позволяет пользователям с лёгкостью управлять воспроизведением данного типа контента.

На проектирование графического интерфейса программной оболочки наложило отпечаток её функциональное назначение как сенсорной оболочки инфокиоска. Так, все элементы интерфейса, предназначенные для выбора или перехода по контенту, увеличены и имеют активные области, которые по площади перекрывают размер самого элемента, что является удобным при управлении информационной сенсорной панели касанием пальцев (рис. 4).



**а** – представление музейных предметов, не вошедших в экспозицию, из фондов музея-заповедника;  
**б** – представление фотоальбома личных фотографий с аннотацией

Рисунок 3. – Представление графического материала

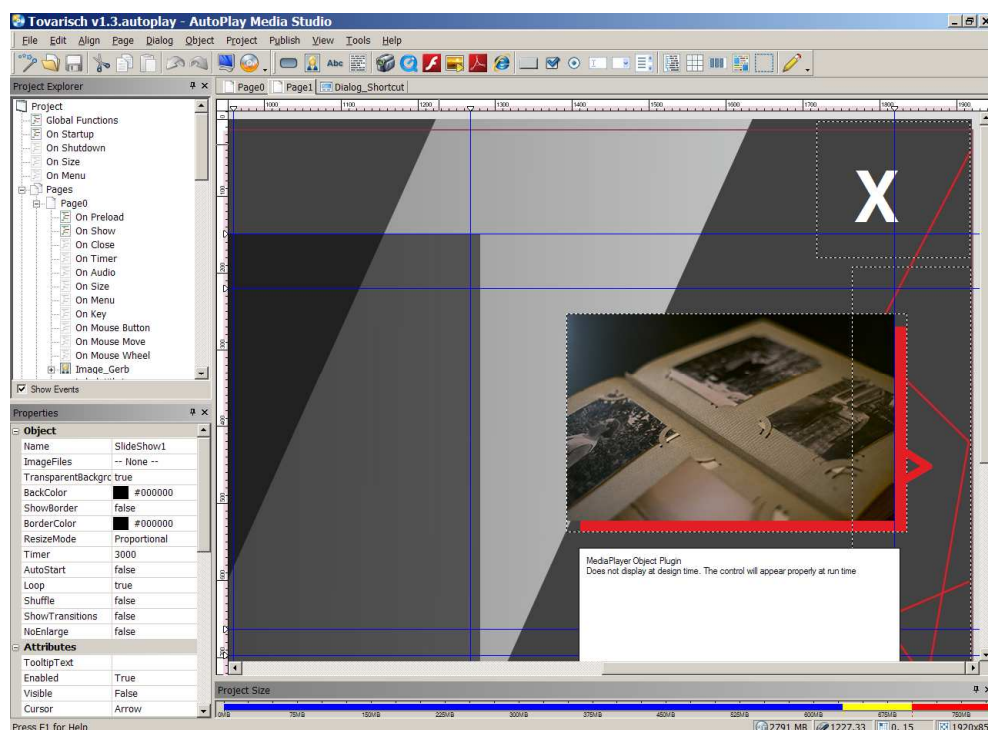


Рисунок 4. – Пунктиром обозначены созданные активные области для выбора элементов (визуальная среда разработки AutoPlay Media Studio версии 8.5)

Лаконичность и простота навигации обусловили востребованность информационной сенсорной панели «Почётные граждане города Полоцка» посетителями одноимённой выставки Краеведческого музея Полоцка.

#### ЛИТЕРАТУРА

1. Indigo Rose — Software Deployment Tools for Windows Developers [Электронный ресурс]. – Режим доступа: <https://www.indigorose.com/autoplay-media-studio/>.— Дата доступа: 26.09.2019.
2. Почётные граждане города Полоцка. Краеведческий музей [Электронный ресурс]. – Режим доступа: <http://local.polotsk.museum.by/node/41428>.— Дата доступа: 26.09.2019.
3. Паратайн. Шрифтовая система PT Sans, PT Serif и PT Mono [Электронный ресурс]. – Режим доступа: <http://rus.paratype.ru/pt-sans-pt-serif>.— Дата доступа: 26.09.2019.

УДК 004.021

## АЛГОРИТМ КНУТА-МОРРИСА-ПРАТТА НА ЯЗЫКЕ PYTHON

А.П. ГАЙДЕЛЬ

(Представлено: канд. физ.-мат. наук, доц. О.Н. ПЕТРОВИЧ)

В данной статье рассматриваются реализация алгоритма поиска подстроки в строке кнута-морриса-пратта на языке python

**Введение.** Часто в задачах поиска информации одной из важнейших задач является поиск точно заданной подстроки в строке. Дело в том, что очевидный способ, который все формулирует как «взять и поискать», является отнюдь не самым эффективным, а для такой низкоуровневой и сравнительно частовывозываемой функции это немаловажно. Поэтому в данной статье рассмотрены некоторые варианты реализации поиска подстроки в строке, в частности, алгоритм Кнута-Морриса-Пратта.

**Основной раздел.** Примитивный алгоритм поиска подстроки в строке основан на переборе всех подстрок, длина которых равна длине шаблона поиска, и посимвольном сравнении таких подстрок с шаблоном поиска. По традиции шаблон поиска или образец принято обозначать как needle (англ. «иголка»), а строку, в которой ведётся поиск — как haystack (англ. «стог сена»). На языке Python примитивный алгоритм выглядит так:

```

index = -1
for i in xrange(len(haystack)-len(needle)+1):
    success = True

    for j in xrange(len(needle)):
        if needle[j]<>haystack[i+j]:
            success = False
            break

    if success:
        index = i
        break
print index

```

## Листинг 1. – Простой алгоритм поиска

Обозначим  $n=|haystack|$ ,  $m=|needle|$ . Простейший алгоритм поиска даже в лучшем случае проводит  $n-m+1$  сравнений; если же есть много частичных совпадений, скорость снижается до  $O(n*m)$ .

Рассматриваемый далее алгоритм хотя и имеет невысокую скорость на «хороших» данных, но это компенсируется отсутствием регрессии на «плохих». Алгоритм Кнута-Морриса-Пратта является одним из первых алгоритмов с линейной оценкой в худшем случае. Прежде чем перейти к описанию алгоритма, необходимо рассмотреть понятие префикс-функции.

Префикс-функция строки  $\pi(S,i)$  – это длина наибольшего префикса строки  $S[1..i]$ , который не совпадает с этой строкой и одновременно является ее суффиксом. Проще говоря, это длина наиболее длинного начала строки, являющегося также и ее концом. Для строки  $S$  удобно представлять префикс функцию в виде вектора длиной  $|S|-1$ . Можно рассматривать префикс-функцию длины  $|S|$ , положив  $\pi(S,1)=0$  (табл.).

Таблица. – Пример префикс функции для строки «abcdabcabcdabcdab»

S[i]	a	b	c	d	a	b	c	a	b	c	d	a	b	c	d	a	b
$\pi(S,i)$	0	0	0	0	1	2	3	1	2	3	4	5	6	7	4	5	6

Предположим, что  $\pi(S,i)=k$ . Отметим следующие свойства префикс-функции.

1. Если  $S[i+1]=S[k+1]$ , то  $\pi(S,i+1)=k+1$ .
2.  $S[1..\pi(S,k)]$  является суффиксом строки  $S[1..i]$ . Действительно, если строка  $S[1..i]$  оканчивается строкой  $S[1..\pi(S,i)]=S[1..k]$ , а строка  $S[1..k]$  оканчивается строкой  $S[1..\pi(S,k)]$ , то и строка  $S[1..i]$  оканчивается строкой  $S[1..\pi(S,k)]$ .

3.  $\forall j \in (k,i)$ ,  $S[1..j]$  не является суффиксом строки  $S[1..i]$ . В противном случае было бы неверным предположение  $\pi(S,i)=k$ , так как  $j>k$ .

Рассмотренные свойства позволяют получить алгоритм вычисления префикс-функции. Пусть  $\pi(S,i)=k$ . Необходимо вычислить  $\pi(S,i+1)$ .

1. Если  $S[i+1]=S[k+1]$ , то  $\pi(S,i+1)=k+1$ .
2. Иначе, если  $k=0$ , то  $\pi(S,i+1)=0$ .
3. Иначе положить  $k:=\pi(S,k)$  и перейти к шагу 1.

Ключевым моментом для понимания сути алгоритма является тот факт, что если найденный на предыдущем шаге суффикс не может быть расширен на следующую позицию, то мы пытаемся рассматривать меньшие суффиксы до тех пор, пока это возможно.

```
def prefix(s):
    v = [0]*len(s)
    for i in xrange(1,len(s)):
        k = v[i-1]
        while k > 0 and s[k] <> s[i]:
            k = v[k-1]
        if s[k] == s[i]:
            k = k + 1
        v[i] = k
    return v
```

### Листинг 2. – Алгоритм вычисления префикс-функции на языке Python

Покажем, что время работы алгоритма составляет  $O(n)$ , где  $n=|S|$ . Заметим, что асимптотику алгоритма определяет итоговое количество итераций цикла while. Это так, поскольку без учета цикла while каждая итерация цикла for выполняется за время, не превышающее константу. На каждой итерации цикла for  $k$  увеличивается не более чем на единицу, значит максимально возможное значение  $k=n-1$ . Поскольку внутри цикла while значение  $k$  лишь уменьшается, получается, что  $k$  не может суммарно уменьшиться больше, чем  $n-1$  раз. Значит цикл while в итоге выполнится не более  $n$  раз, что дает итоговую оценку времени алгоритма  $O(n)$ .

Рассмотрим алгоритм Кнута-Морриса-Пратта, основанный на использовании префикс-функции. Как и в примитивном алгоритме поиска подстроки, образец «перемещается» по строке слева направо с целью обнаружения совпадения. Однако ключевым отличием является то, что при помощи префикс-функции мы можем избежать заведомо бесполезных сдвигов.

Пусть  $S[0..m-1]$  – образец,  $T[0..n-1]$  – строка, в которой ведется поиск. Рассмотрим сравнение строк на позиции  $i$ , то есть образец  $S[0..m-1]$  сопоставляется с частью строки  $T[i..i+m-1]$ . Предположим, первое несовпадение произошло между символами  $S[j]$  и  $T[i+j]$ , где  $i < j < m$ . Обозначим  $P = S[0..j-1] = T[i..i+j-1]$ . При сдвиге можно ожидать, что префикс  $S$  сойдется с каким-либо суффиксом строки  $P$ . Поскольку длина наиболее длинного префикса, являющегося одновременно суффиксом, есть префикс-функция от строки  $S$  для индекса  $j$ , приходим к следующему алгоритму:

1. Построить префикс-функцию образца  $S$ , обозначим ее  $F$ .
2. Положить  $k = 0$ ,  $i = 0$ .
3. Сравнить символы  $S[k]$  и  $T[i]$ . Если символы равны, увеличить  $k$  на 1. Если при этом  $k$  стало равно длине образца, то вхождение образца  $S$  в строку  $T$  найдено, индекс вхождения равен  $i - |S| + 1$ . Алгоритм завершается. Если символы не равны, используем префикс-функцию для оптимизации сдвигов. Пока  $k > 0$ , присвоим  $k = F[k-1]$  и перейдем в начало шага 3.
4. Пока  $i < |T|$ , увеличиваем  $i$  на 1 и переходим в шаг 3.

Возможная реализация алгоритма Кнута-Морриса-Пратта на языке Python выглядит так:

```
def kmp(s,t):
    index = -1
    f = prefix(s)
    k = 0
    for i in xrange(len(t)):
        while k > 0 and s[k] <> t[i]:
            k = f[k-1]
        if s[k] == t[i]:
            k = k + 1
        if k == len(s):
            index = i - len(s) + 1
            break
    return index
```

### Листинг 3. – Возможная реализация алгоритма Кнута-Морриса-Пратта на языке Python



**Заключение.** Надеюсь, кто-то посчитает эту статью полезной для себя. Алгоритм Кнута-Морриса-Пратта не единственный быстрый алгоритм поиска, но он достаточно быстрый (для задач типа олимпиадных) и при этом простой. Алгоритм Бойера-Мура близок к Кнута-Морриса-Пратта по сложности и для определенного круга задач (где образец не содержит повторяющихся фрагментов) он быстрее. Но зато для другого круга задач (где образец и строка поиска содержат повторяющиеся последовательности, как в примерах к статье) он заметно уступает алгоритму Кнута-Морриса-Пратта.

#### ЛИТЕРАТУРА

1. Wikipedia [Электронный ресурс] Алгоритм Кнута — Морриса — Пратта. Режим доступа: [https://ru.wikipedia.org/wiki/Алгоритм\\_Кнута\\_-\\_Морриса\\_-\\_Пратта](https://ru.wikipedia.org/wiki/Алгоритм_Кнута_-_Морриса_-_Пратта). Дата доступа: 11.09.19.
2. Py-algorithm [Электронный ресурс] Алгоритм Кнута-Морриса-Пратта (КМП). Режим доступа: [http://py-algorithm.blogspot.com/2013/04/blog-post\\_19.html/](http://py-algorithm.blogspot.com/2013/04/blog-post_19.html/). Дата доступа: 15.09.19.
3. Intuit [Электронный ресурс] Алгоритмы поиска в тексте. Режим доступа: [https://www.intuit.ru/studies/professional\\_skill\\_improvements/2056/courses/504/lecture/1146](https://www.intuit.ru/studies/professional_skill_improvements/2056/courses/504/lecture/1146). Дата доступа: 18.09.19.

УДК 004.051

## СЕРВИС AWS LAMBDA

А.П. ГАЙДЕЛЬ

(Представлено: канд. физ.-мат. наук, доц. О.Н. ПЕТРОВИЧ)

*В данной статье рассматриваются вычислительная платформа AWS Lambda — ключевой компонент бессерверных вычислений — ее плюсы и минусы, возможности и предоставляемые сервисы.*

**Введение.** Все чаще люди переходят на AWS Lambda ради масштабируемости, производительности, экономии и возможности обрабатывать миллионы запросов в месяц. Для этого не нужно управлять инфраструктурой, на которой работает сервис. А автоматическое масштабирование позволяет обслуживать тысячи одновременных запросов в секунду.

**Основной раздел.** AWS Lambda — это событийно-ориентированный сервис бессерверных вычислений, который позволяет выполнять код без выделения и администрирования серверов и дополнять другие сервисы AWS на основе пользовательской логики. Lambda автоматически реагирует на различные события (так называемые триггеры), например, на HTTP-запросы через Amazon API Gateway, изменение данных в корзинах Amazon S3 или таблицах Amazon DynamoDB; либо можно запустить свой код через вызовы API, используя AWS SDK и переходы между состояниями в AWS Step Functions.

Lambda выполняет код на высокодоступной вычислительной инфраструктуре и полностью отвечает за администрирование нижележащей платформы, включая обслуживание серверов и операционной системы, выделение ресурсов, автоматическое масштабирование, мониторинг кода и ведение журналов. То есть вам достаточно загрузить свой код и настроить, как и когда он должен выполняться. В свою очередь, сервис позаботится о его запуске и обеспечит высокую доступность вашего приложения.

AWS Lambda — это удобная вычислительная платформа, подходящая для решения множества задач, разумеется, если язык и среда выполнения вашего кода поддерживаются сервисом. Если вы хотите сосредоточиться на коде и бизнес-логике, поручив обслуживание серверов, выделение ресурсов и масштабирование стороннему поставщику за разумные деньги, вам точно стоит перейти на AWS Lambda.

Lambda идеально подходит для создания программных интерфейсов, а если использовать сервис вместе с API Gateway, можно значительно сократить расходы и быстрее выйти на рынок. Есть различные способы использования функций Lambda и варианты организации бессерверной архитектуры — каждый сможет выбрать что-то подходящее с учетом поставленной цели.

Lambda позволяет выполнять широкий спектр задач. Так, благодаря поддержке CloudWatch можно создавать отложенные задания и автоматизировать отдельные процессы. Нет никаких ограничений по характеру и интенсивности использования сервиса (учитываются расход памяти и время), и вам ничто не мешает планомерно работать над полноценным микросервисом на основе Lambda.

Здесь можно создавать сервис-ориентированные действия, которые не выполняются постоянно. Типичный пример — масштабирование изображений. Даже в случае распределенных систем функции Lambda не теряют своей актуальности.

Как и большинство сервисов AWS, Lambda предоставляется по принципу общей ответственности AWS и клиента по части безопасности и соблюдения нормативных требований. Этот принцип снижает операционную нагрузку на клиента, поскольку AWS берет на себя задачи обслуживания, администрирования и контроля компонентов сервиса — от операционной системы хоста и уровня виртуализации до физической безопасности объектов инфраструктуры.

Если говорить конкретно об AWS Lambda, то AWS отвечает за управление нижележащей инфраструктурой, связанными базовыми сервисами, операционной системой и платформой приложений. В то время как клиент несет ответственность за безопасность своего кода, хранение конфиденциальных данных, контроль доступа к ним, а также к сервису и ресурсам Lambda (Identity and Access Management, IAM), в том числе в пределах используемых функций.

Основное преимущество Lambda заключается в том, что, выполняя функцию от вашего имени, сервис сам выделяет необходимые ресурсы. Вы можете не тратить время и силы на администрирование систем и сосредоточиться на бизнес-логике и написании кода.

Сервис Lambda разделен на две плоскости. Первая — плоскость управления. Согласно Википедии, плоскость управления (control plane) — это часть сети, отвечающая за транспортировку сигнального трафика и маршрутизацию. Она является главным компонентом, принимающим глобальные решения о выделении, обслуживании и распределении рабочих нагрузок. Кроме того, плоскость управления выступает в роли сетевой топологии поставщика решения, отвечающей за маршрутизацию трафика и управление им.

Вторая плоскость — плоскость данных. У нее, как и у плоскости управления, свои задачи. Плоскость управления предоставляет API для управления функциями (CreateFunction, UpdateFunctionCode) и контролирует взаимодействие Lambda с другими сервисами AWS. Плоскость данных управляет API вызовов (Invoke API), который запускает функции Lambda. После вызова функции плоскость управления выделяет либо выбирает существующую, заранее подготовленную для этой функции среду выполнения, а затем выполняет в ней код.

AWS Lambda поддерживает множество языков программирования, включая Java 8, Python 3.7, Go, NodeJS 8, .NET Core 2 и другие, через соответствующие среды выполнения. AWS регулярно их обновляет, распространяет исправления безопасности и выполняет прочие операции по обслуживанию этих сред. Lambda позволяет использовать и другие языки при условии, что вы сами внедрите соответствующую среду выполнения. И тогда уже вам придется заниматься ее обслуживанием, в том числе следить за безопасностью.

Каждая функция работает в одной или нескольких выделенных средах, которые существуют лишь в течение жизненного цикла этой функции, а затем уничтожаются. В каждой среде одновременно выполняется лишь один вызов, но она используется повторно, если возникает множество серийных вызовов одной и той же функции. Все среды выполнения работают на виртуальных машинах с аппаратной виртуализацией — на так называемых microVM. Каждая microVM назначается конкретной учетной записи AWS и может многократно использоваться средами для выполнения различных функций в этой учетной записи. MicroVM упаковываются в структурные блоки аппаратной платформы Lambda Worker, которой владеет и управляет AWS. Одна и та же среда выполнения не может использоваться разными функциями, равно как microVM уникальны для разных учетных записей AWS (рис. 1).

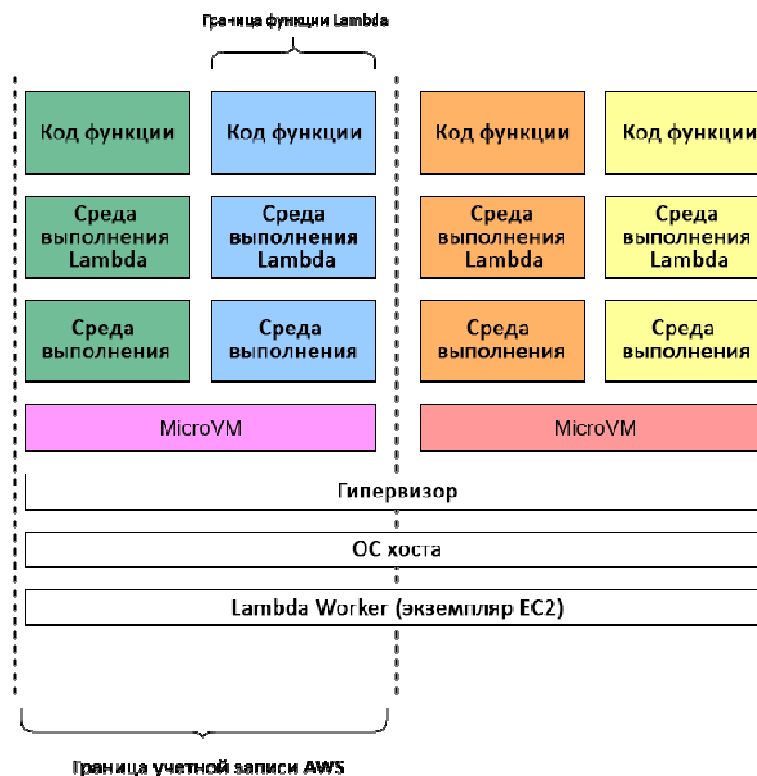


Рисунок 1. – Модель изоляции в AWS Lambda

Изоляция сред выполнения реализована с помощью нескольких механизмов. На высшем уровне каждой среды присутствуют отдельные копии следующих компонентов:

- Код функции
- Любые слои Lambda, выбранные для функции
- Среда выполнения функции
- Минимальное пользовательское пространство на базе Amazon Linux

Для изоляции разных сред выполнения применяются следующие механизмы:

- cgroups — ограничение доступа к ресурсам ЦП, памяти, пропускной способности накопителей и сети для каждой среды выполнения;

- namespaces — объединение в группы ID процессов, ID пользователей, сетевых интерфейсов и других ресурсов, которыми управляет ядро Linux. Каждая среда выполнения работает в своем пространстве имен;
- seccomp-bpf — ограничение системных вызовов, которые можно использовать в среде выполнения;
- iptables и routing tables — изоляция сред выполнения между собой;
- chroot — предоставление ограниченного доступа к нижележащей файловой системе.

В сочетании с проприетарными технологиями изоляции AWS перечисленные механизмы гарантируют надежное разграничение сред выполнения. Изолированные таким образом среды не могут обращаться к данным других сред и изменять их.

Хотя несколько сред выполнения одной учетной записи AWS могут выполняться на одной microVM, ни при каких обстоятельствах microVM не могут совместно использоваться разными учетными записями AWS. Для изоляции microVM в AWS Lambda используется всего два механизма: экземпляры EC2 и Firecracker. Изоляция гостей (guest isolation) в Lambda на основе экземпляров EC2 применяется с 2015 года. Firecracker — это новый гипервизор с открытым исходным кодом, специально разработанный AWS для бессерверных рабочих нагрузок и представленный в 2018 году. Физическое оборудование, на котором выполняются microVM, совместно используется рабочими нагрузками разных учетных записей.

Хотя среды выполнения Lambda уникальны для разных функций, в них можно повторно вызывать одну и ту же функцию, то есть среда выполнения может просуществовать несколько часов, прежде чем будет уничтожена.

В каждой среде выполнения Lambda также имеется файловая система с разрешением на запись, доступная через каталог /tmp. К его содержимому нельзя обращаться из других сред выполнения. Что касается сохранения состояний процессов, записанные в /tmp файлы существуют в течение всего жизненного цикла среды выполнения. За счет этого возможно аккумулирование результатов нескольких вызовов, что особенно полезно для таких затратных операций, как загрузка моделей машинного обучения.

Интерфейс Invoke API можно задействовать в двух режимах: в режиме событий и в режиме «запрос — ответ». В режиме событий вызов добавляется в очередь для последующего выполнения. В режиме «запрос — ответ» функция вызывается мгновенно с предоставленной полезной нагрузкой, после чего возвращается ответ. И в том и в другом случае функция выполняется в среде Lambda, но с различными путями полезной нагрузки.

Во время вызовов типа «запрос — ответ» полезная нагрузка поступает от API обработки запросов (API Caller), такого как AWS API Gateway или AWS SDK, в балансировщик нагрузки, а затем — в службу выполнения вызовов Lambda (Invoke Service). Последняя определяет подходящую среду для выполнения функции и передает туда полезную нагрузку, чтобы завершить вызов. Балансировщик нагрузки получает трафик с TLS-защитой через Интернет. Трафик в пределах сервиса Lambda — после балансировщика нагрузки — проходит через внутренний VPC в определенном регионе AWS.

Вызовы по событиям могут выполняться незамедлительно либо добавляться в очередь. В некоторых случаях очередь реализована с помощью сервиса Amazon SQS (Amazon Simple Queue Service), который передает вызовы в службу выполнения вызовов Lambda посредством внутреннего опрашивающего процесса (poller). Передаваемый трафик защищен TLS, при этом какое-либо дополнительное шифрование данных, хранящихся в Amazon SQS, не предусмотрено.

Вызовы по событиям не возвращают ответы — любую ответную информацию Lambda Worker попросту игнорирует. Вызовы на основе событий Amazon S3, Amazon SNS, CloudWatch и других источников обрабатываются сервисом Lambda в режиме событий. Вызовы из потоков Amazon Kinesis и DynamoDB, вызовы очередей SQS, балансировщика нагрузки приложений и API Gateway обрабатываются в режиме «запрос — ответ».

Вы можете производить мониторинг и аудит функций Lambda с помощью различных механизмов и сервисов AWS, включая следующие:

- Amazon CloudWatch — собирает различные статистические данные, такие как количество запросов, продолжительность выполнения запросов и число запросов, завершившихся ошибкой.
- Amazon CloudTrail — позволяет вести журналы, непрерывный мониторинг и сохранять сведения об активности в учетной записи, связанные с вашей инфраструктурой AWS. У вас будет полная хронология действий, выполненных с помощью консоли AWS Management Console, AWS SDK, инструментов командной строки и других сервисов AWS.
- AWS X-Ray — обеспечивает полную видимость всех этапов обработки запросов в вашем приложении на основе карты его внутренних компонентов. Позволяет анализировать приложения в ходе разработки и в производственной среде.

– AWS Config — вы сможете отслеживать изменения конфигурации функций Lambda (включая их удаление) и сред выполнения, тегов, имен обработчиков, размера кода, распределения памяти, настроек времени ожидания и параметров параллелизма, а также роли выполнения Lambda IAM, подсети и привязки групп безопасности.

**Заключение.** Событийно-ориентированная архитектура — это не просто ускорение или улучшение существующих бизнес-процессов, это другой бизнес. Раньше невозможно было реализовать такую архитектуру, не было фундамента в виде облачных платформ, но теперь это реальное технологическое и стратегическое преимущество.

#### ЛИТЕРАТУРА

1. Blog.kr [Электронный ресурс] AWS LAMBDA. Событийно-ориентированная архитектура. – Режим доступа: <https://blog.kr.pp.ru/post/2018-12-24/>. – Дата доступа: 10.09.19.
2. Wikipedia [Электронный ресурс] Событийно-ориентированное программирование. – Режим доступа: [https://ru.wikipedia.org/wiki/Событийно-ориентированное\\_программирование/](https://ru.wikipedia.org/wiki/Событийно-ориентированное_программирование/). – Дата доступа: 20.09.19.
3. Aws.amazon [Электронный ресурс] AWS Lambda. – Режим доступа: <https://aws.amazon.com/ru/lambda/>. – Дата доступа: 21.09.19.

УДК 338.45.01

## ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО ПОИСКУ ПАРТНЁРОВ ДЛЯ ЗАНЯТИЯ СПОРТОМ

А.Ю. ГЕРЕЦ

(Представлено: канд. тех. наук, доц. И.Б. БУРАЧЁНОК)

*В данной статье исследована перспективность разработки и использования мобильного приложения по поиску партнёров для занятия спортом. Проект имеет высокую социальную направленность.*

У инновационных продуктов достаточно короткий жизненный цикл, так как быстро появляются аналоги со сложим либо улучшенным функционалом. В отличие от российского рынка, на отечественном пока еще нет высокой конкуренции и, в общем, мало кому интересно развивать социальные проекты. Подобные приложения широко распространены за рубежом. В Российской Федерации преимущественно в виде социальных сетей и приложений брендов, например, Nike.

Использование разработанного приложения позволит повысить осознанность занятий спортом, приобщив молодежь (большинство молодых людей сидят в гаджетах и являются активными пользователями интернета), а также откроет возможность пользователям недорого арендовать спортивные площадки.

Логико-структурный подход (ЛСП) на сегодняшний день эффективен на всех фазах жизненного цикла (внедрение на рынок, рост, зрелость и спад) проекта, особенно при его идентификации, разработке и мониторинге. Данный подход широко используется в разнообразных проектах (вывод на рынок новой продукции, проведение модернизации продуктов), осуществляемых многими международными, правительственными, коммерческими организациями.

Далее в таблице 1 представим логико-структурную матрицу по созданию приложения по поиску партнеров для занятия спортом в Республике Беларусь.

Таблица 1. – Логико-структурная матрица приложения

Логика участия	Объективно проверяемые индикаторы достижения	Источники и средства верификации	Допущения
Общие цели	Улучшить здоровье молодого поколения	Популяризация приложения в школах и университетах, тематических группах, таргетинг	Замкнутость людей, предпочитают зависать в гаджетах
Цель проекта	Вовлечение молодых людей в спорт, увеличение уровня их коммуникации в офф-лайн	приложения быстро меняются, нужна оперативная техподдержка, авторизация возможна через соцсеть или мессенджеры	мало желающих заниматься спортом, отсутствие интерактивных спорт-мероприятий
Ожидаемые результаты	Заработок на подписке за пользование всем функционалом приложения	Люди будут покупать подписку Увеличиться посещаемость спортивных мероприятий У организаторов спорт-мероприятий снизятся издержки на продвижение	Малые продажи «подписки», сложности с поиском площадок для бесплатного спортивного времяпрепровождения, мало организаторов бесплатных спорт-мероприятий
Мероприятия	Поиск спонсоров на спортивных мероприятия, своевременное уведомление пользователей о бесплатных спортивных мероприятиях	Налаживание связей с крупными брендами, образовательными госучреждениями, магазинами спортивного инвентаря	Много отказов от участия для стартапа приложения. Не возможность сразу организовать увлекательное мероприятие. Анонс мероприятий может приходиться запоздало

На рисунке 1 представим дерево проблем для реализации проекта по выведению на рынок приложения.

Основными проблемами в сложившейся ситуации является неизвестность разработчиков, а, соответственно, крупные бренды будут не готовы к сотрудничеству. Сразу не получится динамично анонсировать мероприятия, так как их мало проводится в городах. Если приглашать на «пригородные» мероприятия необходимо будет продумывать доставку либо предлагать экономный вариант добраться до места назначения.

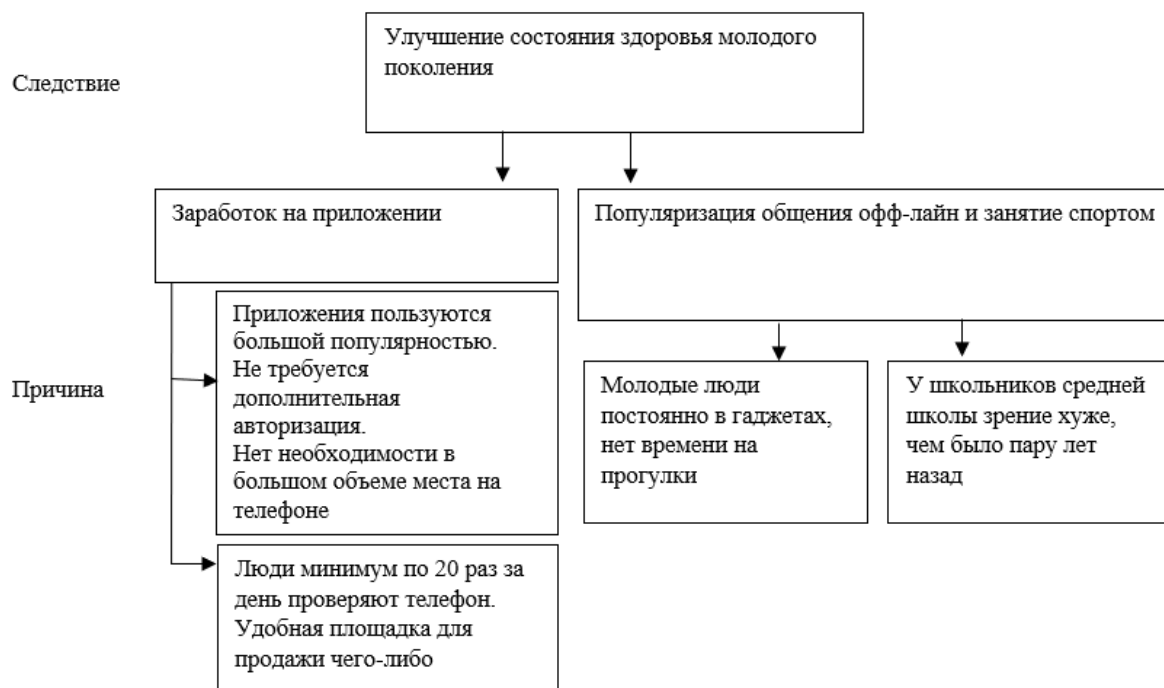


Рисунок 1. – Дерево проблем реализации проекта

Для решения поставленных проблем составим наглядное дерево целей.



Рисунок 2. – Дерево целей

Ключевая цель снизить время на поиски единомышленников для занятия спортом. Привлекать молодых людей к активной жизни и минимизировать их нахождение в онлайн.

В настоящее время молодежь привыкла проводить время в социальных сетях и интернет пространстве, а также ходить по клубам, барам и другим заведениям тем самым ухудшая свое здоровье. Также в Республике Беларусь наблюдается недостаточное количество мест для занятия спортом, что отражается слабой активности населения. Низкая платежеспособность граждан отражается и на неактивном использовании существующих площадок.

Структура разбиения работ позволяет участникам проекта и всем заинтересованным лицам достичь ясного представления о конечной цели проекта. Как правило, структура разбиения (декомпозиции) работ представляет собой иерархическую структуру последовательной декомпозиции проекта на под-проекты, пакеты работ различного уровня, пакеты детальных работ.

Таблица 2 – Структура работ по созданию спортивного приложения

Логика вмешательства	Объективно проверяемые индикаторы	Источники проверки	Допущения
Общая цель	Повысить уровень здоровья молодежи 14-30 лет	Количество участников на спортивных мероприятиях	Не выполнение запланированных показателей, срывы проведения спортивных мероприятий, слабые на интерактив мероприятия. Спортивные встречи могут проходить на «разбитых» площадках
Цель проекта	Количество пользователей, скачиваний, количество людей, оформивших ежемесячную подписку	Сервисы статистики. Средства, зачисленные на платформу	Слабое продвижение, непонятный интерфейс
Результаты	Ежемесячный доход в 30,0 тыс. руб. Количество организованных мероприятий, количество посетивших пользователей мероприятий	в приложении будет выдаваться «промокод», который нужно будет называть на мероприятии.	Низкая активность рекламодателей. Низкий уровень интерактивности в приложении. Нежелание организовывать мероприятия на безвозмездной основе.
Действия	Запуск рекламы, можно внедрить реферальную систему (за 3 приглашенных друга месячная подписка в подарок или комплекс упражнений)	Конверсия в скаченное приложение слабый рост пользователей	Плохо настроенный таргет-рекламы (будет дана не в том месте где находится целевая аудитория). Не работает рефсистема

По данным таблицы 2 следует, что главной целью на уровне миссии является улучшение состояния здоровья молодежи. Непосредственно для приложения объективным показателем успешности проекта станет рост числа скачиваний и количество людей, которые оформят платную подписку на приложение. Как результат мобильное приложение позволит ежемесячно получать выручку в размере 30 тыс. руб. Однако на первоначальном этапе обязательно потребуется настроить таргетинговую рекламу, а в последующем непосредственно по спортивным мероприятиям будет рекламироваться приложение, а также заработает «сарафанное радио» и в тематических разделах Google Play и App Store приложение будет выдаваться по поиску в первой десятке.

На рисунке 3 представим функционально-ориентированную систему разбиения работ (ССР) по созданию спортивного приложения как инновационного продукта в Республике Беларусь.



Рисунок 3 – Функционально-ориентированная ССР спортивного приложения



Создание и запуск спортивного приложения состоит из четырех блоков работ: разработка бизнес-идеи и изыскание финансирования, программирование, разработка дизайна приложения, анонсирование приложения. Для проекта привлечен беспроцентный заем. Основными пользователями разрабатываемого приложения будут учащиеся и студенты, основная рекламная площадка придется на государственные и частные учебные заведения. Параллельно с этим будет идти реклама по социальным сетям, где находится целевая аудитория (молодежь). Данный проект весьма актуален для государства, так как спортом занимается незначительное число современной молодежи, спортивные площадки пустуют в виду отсутствия желающих и высокой стоимости их аренды для обычных студентов и учащихся. Приложение позволит собирать людей по спортивным интересам и совместно арендовать помещение, а также проводить бесплатные мероприятия по вовлечению молодежи в спорт.

**Заключение.** Таким образом, в данной статье представлены основные принципы вывода на рынок мобильного приложения, обоснованы ресурсы для его реализации, рассмотрены возможные последствия использования молодежью. Детальное изучение представленных в данной статье вопросов позволило исключить ошибки при разработке собственного проекта.

#### ЛИТЕРАТУРА

1. Система дистанционного бизнес-образования – Инновационный проект: понятия, основные этапы создания и реализации [Электронный ресурс]. – Режим доступа: <http://www.tehnpark.bybusiness/207.html>. – Дата доступа: 02.09.2019.

УДК 339.137.22

## ОЦЕНКА РЫНКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО ПОИСКУ ПАРТНЁРОВ ДЛЯ ЗАНЯТИЯ СПОРТОМ

**А.Ю. ГЕРЕЦ**

(Представлено: канд. тех. наук, доц. И.Б. БУРАЧЁНОК)

*В данной статье рассмотрены наиболее известные спортивные приложения на постсоветском пространстве, их функционал и интерфейс, проведена сравнительная характеристика стоимости платной подписки, возможностей приложений, дан финансовый прогноз реализации разрабатываемого мобильного приложения по поиску партнёров для занятия спортом.*

По всему миру активно продвигается идея здорового образа жизни. А быть профессиональным спортсменом считается престижным. Однако сегодня, в молодежной среде спорт далеко не в приоритете, а, наоборот, современные молодые люди всячески пытаются увильнуть от различных физических нагрузок, что приводит к сколиозу, ожирению и иным проблемам со здоровьем. Всего этого можно избежать занятиями спортом, а для этого нужна командная поддержка. В виду того, что молодежь много времени проводит в гаджетах – поиск информации через них не составляет труда. Поэтому, целесообразным является разработка приложения, в котором будет собраны все актуальные спортивные мероприятия, а также привлечены заинтересованные люди. В результате они смогут кооперироваться между собой и начать заниматься спортом.

Далее осуществим анализ имеющихся на сегодняшний день приложений и сервисов на постсоветском пространстве. Внешний вид сервиса **TeamKraft** [1] при помощи которого пользователи могут узнать о спортивных мероприятиях или организовать их сами представлен на рисунке 1.

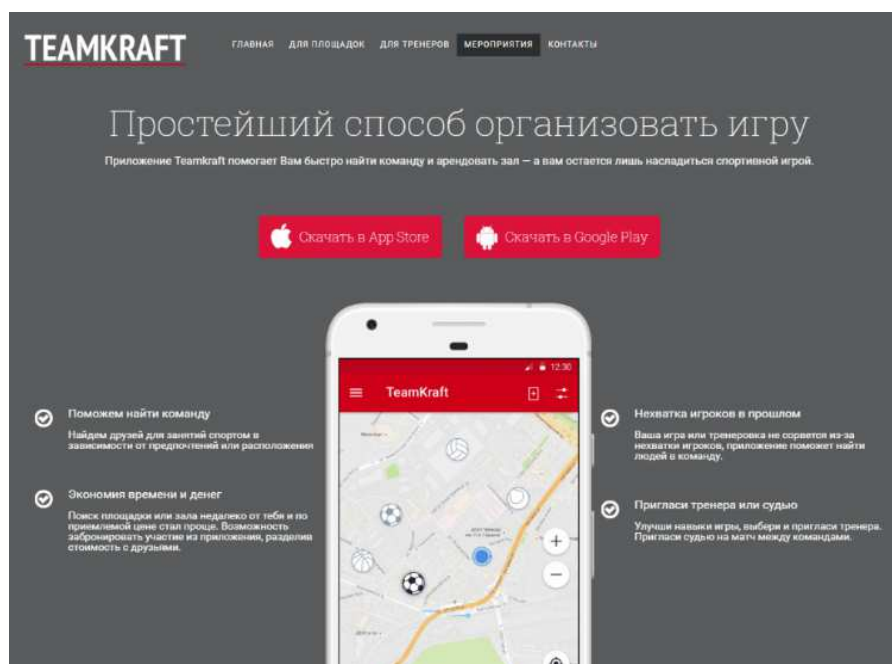


Рисунок 1. – Внешний вид сервиса *TeamKraft*

Мобильное приложение TeamKraft помогает легко и просто спланировать свой активный отдых,знакомиться с новыми людьми и участвовать в командных играх (футбол, волейбол, баскетбол, картинг, пейнтбол, велосипедные прогулки и теннис).

Сейчас в г. Москве уже сформирована достаточная база площадок и игроков, чтобы поиск команды и бронирование зала для игры занимали пару минут. Сегодня пользователи из любого региона могут добавлять любимые площадки на карту прямо в приложении, создавать на них мероприятия и приглашать на них своих друзей. В ближайшее время любители спортивного досуга, вне зависимости от уровня подготовленности, могут за пару минут найти новые интересные виды спорта и присоединиться к играм.

Приложение **Ridewithme** помогает организовывать и находить спортивные мероприятия [2]. Оно позволяет спортсменам находить напарников и соперников для занятий любимым видом спорта. Что позволяет исключать неполные составы, одиночных пробежек или несостоявшихся поездок, например, в горы. Внешний вид интерфейса Ridewithme представлен на рисунке 2.

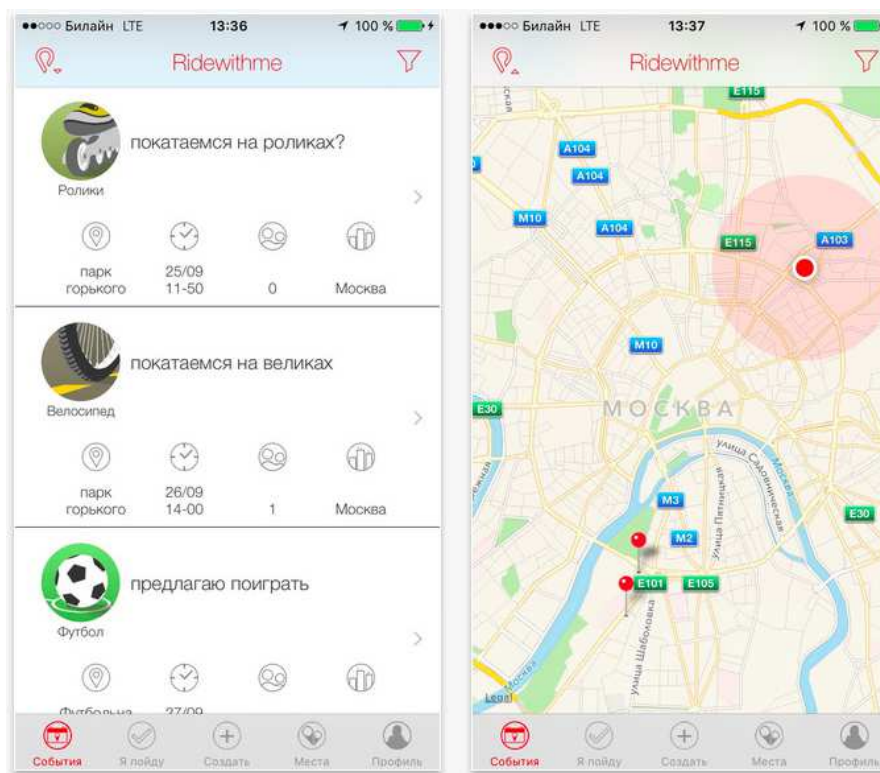


Рисунок 2. – Интерфейс *Ridewithme*

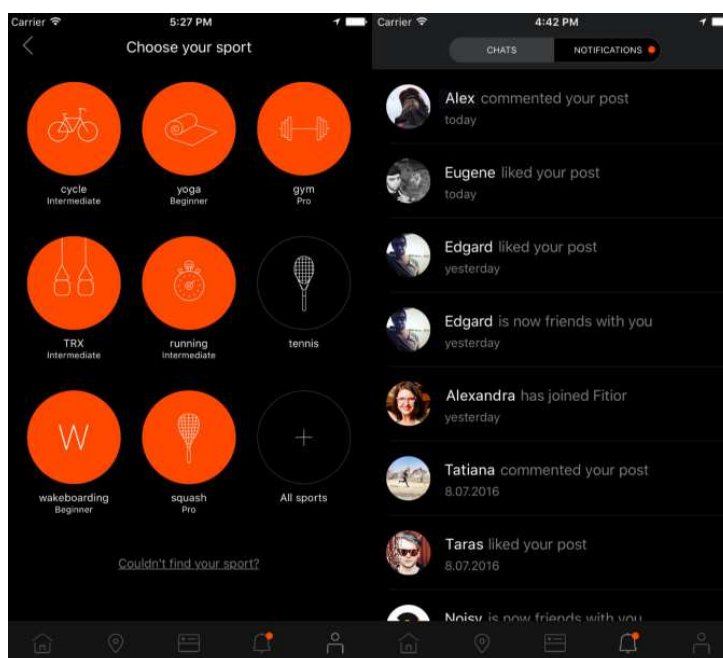
В представленном приложении можно создавать события или присоединяться к другим пользователям прямо со своего смартфона. Для ленты отображаемых событий предусмотрены фильтры, такие как вид спорта, город или страна. При создании события пользователь может указывать всю необходимую информацию, включая дату, время, место, вид спорта, а также персональное описание, которое позволяет сделать каждое событие уникальным. Внутри созданного события предусмотрены комментарии для обсуждения аспектов грядущего мероприятия, к примеру, на кого возложена обязанность принести чай и «печеньки». События, на которые подписан пользователь, хранятся во вкладке «Мои события» для удобства мониторинга возможных изменений со стороны организатора. Плюсом станет то, что в приложении предусмотрены напоминания. Чтобы найти дорогу до места встречи, пользователю не нужно переходить в навигационные программы и вбивать адрес. В Ridewithme предусмотрено, что сверху каждой страницы события есть карта, нажав на которую, пользователь получает маршрут до пункта назначения. Java-приложение, разработано в США. Оно помогает пользователю найти партнёра для занятий спортом. Сервис может быть полезен для тех, кто не хочет заниматься спортом в одиночестве. Приложение фильтрует потенциальных партнёров по спортивным целям и геолокации. Кроме того, пользователи могут наполнить собственные профили другой информацией, которая поможет им найти подходящего партнёра – рассказать о своих интересах, предпочтениях в музыке, добавить фотографию и несколько фактов о себе. Сервис работает также, как и популярное приложение для знакомств Tinder. Пользователь свайпами обозначает, понравился ему тот или иной человек или нет. Если два человека «отметили» друг друга, они могут пообщаться в чате, узнать друг друга получше и назначить время первой тренировки.

Еще одно приложение **Nike+ Run Club**, которое помогает, как отслеживать личные успехи и делиться ими с другими, так и искать напарников по занятиям спортом [3]. Внешний вид приложения Nike+ Run Club представлен на рисунке 3.

Функционал приложения очень широк – это целая соцсеть для любителей спорта, где каждый может найти то, что нужно ему: просто использовать как органайзер, создать программу тренировок, найти поблизости того, с кем можно заниматься спортом вместе и даже выкладывать фотографии с тренировок по аналогии с Instagram.

Рисунок 3 – Внешний вид приложения *Nike+ Run Club*

В прошлом году в России появилась новая спортивная соцсеть **Fitior**, позволяющая найти людей поблизости для совместных занятий спортом [4]. Никаких спортивных фиш в нем нет, но можно искать людей по определенным видам спорта. Идея создания такой соцсети просто великолепна, но пока что она насчитывает очень небольшое количество пользователей, потому поиск напарника может быть затруднительным. Внешний вид интерфейса спортивной соцсети Fitior представлен на рисунке 4.

Рисунок 4 – Интерфейс социальной сети *Fitior*

Дейтинг-сервис Badoo в этом вопросе ушел еще дальше – приложение доступно для скачивания на базе iOS и Android. Теперь, человек оказавшись в одном помещении с тем, кто тоже пользуется приложением, получит push-уведомление. Даже если в этот момент пользователь не смотрит на экран, этого человека можно будет потом легко найти и увидеть, сколько всего раз пересекались и есть ли общие друзья. Эту фишу можно отлично использовать и для поиска друзей по занятиям спортом.

В целом получается, что спортивные приложения не только объединяют людей по спортивным интересам, но и служат некой социальной сетью для общения. В приложения есть возможность делиться

собственными достижениями в спорте, находить друзей для занятия спортом даже если приехал как турист в другую страну.

Далее в таблице 1 представим данные о конкурентах.

Таблица 1. – Анализ конкурентов

Наименование услуги	Team Kraft	Ride with me	Nike+ Run Club	Fitior	Средняя розничная цена конкурентов	Минимальная розничная цена конкурентов	Розничная цена организации
Одно скачивание платной версии, руб.	4,08	3,07	2,95	3,56	3	3	3,15
Коллоборация, руб.		476,41		584,69	531	476	472,50

По данным представленным в таблице следует, что все четыре конкурента предоставляют возможность платной подписки на приложение, которое дает расширенные возможности. Средняя цена конкурентов 3,0 руб., а предлагаемое к созданию приложение будет обходиться в 3,15 руб., что немного больше (+5%). В отношении коллоборации (взаимного пиара) отметим, что такая возможность есть только в двух приложениях. В приложении Nike+ Run Club коллоборации нет, так как это приложение больше для расширения силы влияния бренда Nike, популяризации продукции организации. Коллоборация возможна в мобильном приложении Ride with me и Fitior. Средняя цена привлечения для спонсора является 531 руб., а минимальная цена 476 руб., но приложение пока еще не работает в Республике Беларусь.

В заключении в таблице 2 представим данные проведенного анализа мобильных приложений емкости и доли рынка.

Таблица 2. – Анализ рынка схожих мобильных приложений

Год	2022	2023	2024	2025	2026	2027	2028	2029
Одно скачивание платной версии								
Емкость рынка, тыс. руб.	3553,5	3600,1	3788,2	3939,7	4117,0	4322,9	4539,0	4811,4
Доля рынка, %	3	4	4	4	4	4	4	3
Темп роста рынка, %	3	3	3,5	4	4,5	5	5	6
Объем продаж (нат. ед. / ден. ед.)	47 340	66 717	70 424	70 424	70 424	70 424	70 424	70 424
Количество конкурентов	4	4	5	6	6	6	7	7
Количество покупателей, тыс. чел.	1870,3	1926,4	1993,8	2073,5	2166,9	2275,2	2389,0	2532,3
Коллоборация								
Емкость рынка, руб.	314421	323854	335189	348596	364283	382497	401622	425719
Доля рынка, %	18	20	23	25	28	30	33	37
Темп роста рынка, %	3,5	4	5	5	6	6	7	7
Объем продаж (нат. ед. / ден. ед.)	254160	285930	301815	301815	301815	301815	301815	301815
Количество конкурентов	2	2	2	3	3	3	3	4
Количество основных покупателей	4 005	4 126	4 270	4 441	4 641	4 873	5 116	5 423

Из таблицы видно, что емкость рынка мобильных приложений к 2022 г. составит 3,6 млн. руб., а к 2029 г. уже 4,8 млн. руб. Для Республики Беларусь рынок мобильных приложений пока еще новый, молодой и с низкой конкуренцией. К этому стоит добавить более дешевое производство продукта, чем в ближнем зарубежье. Предлагаемое к реализации мобильное приложение по платным установкам соста-

вит порядка 3-4% в год, а темп рост рынка колеблется в пределах 3-6% в год. Конкуренция в сфере спорта будет 4-7 шт. и с каждым годом будут возрастать, а, соответственно, рынок будет насыщаться.

По направлению коллаборация емкость рынка составляет 314,4-425,7 тыс. руб. В виду того, что коллаборация на отечественном рынке в мобильном сегменте не высока, то станет возможным занять долю рынка 18% в 2027 г. И 37% в 2029 г. Число возможных партнеров колеблется 4,0 – 5,4 тыс. шт. (физические (блогеры, ИП) и юридические лица).

**Заключение.** Таким образом из проведенного анализа разрабатываемого мобильного приложения и трех конкурентов следует, что рынок мобильных приложений в Республике Беларусь перспективен: конкуренция нет, рынок мобильных приложений находится в начале своего зарождения. Разрабатываемое мобильное приложение по поиску партнёров для занятия спортом направлено на глобальную миссию – улучшение состояние здоровья молодежи, что является социально ориентированной миссией и способно в долгосрочной перспективе снизить количество погибших от онкологических и сердечных заболеваний. Из сильных преимуществ и возможностей можно отметить низкий уровень конкуренции и динамичное развитие мобильного сегмента приложений.

#### ЛИТЕРАТУРА

1. TeamKraft – сервис для организации спортивных событий [Электронный ресурс]. – Режим доступа: <https://vc.ru/tribuna/21085-teamkraft>. – Дата доступа: 01.09.2019.
2. Ridewithme – сервис для организации спортивных событий [Электронный ресурс]. – Режим доступа: <https://vc.ru/tribuna/11643-ridewithme>. – Дата доступа: 01.09.2019.
3. Nike+ Run Club [Электронный ресурс]. – Режим доступа: [https://www.nike.com/au/en\\_gb/c/running/nike-run-club](https://www.nike.com/au/en_gb/c/running/nike-run-club). – Дата доступа: 01.09.2019.
4. Fitior – социальная сеть для любителей спорта [Электронный ресурс]. – Режим доступа: <https://vc.ru/tribuna/18392-fitior>. – Дата доступа: 01.09.2019.

УДК 62-5

## ПОВОРОТНЫЙ МЕХАНИЗМ ДЛЯ АНТЕННЫ С GPS ДАТЧИКОМ И УПРАВЛЕНИЕМ ПО LAN

**В.И. ГОРДЕЁНОК***(Представлено: И.С. РУСЕЦКИЙ)*

*В данной статье рассматривается метод разработки поворотного устройства с GPS датчиком и возможностью управлять положением антенны удаленно от рабочего места при помощи LAN соединения.*

**Введение.** Главной задачей было максимально точно изучить всю элементные базы сторонних производителей и найти им альтернативу, чтобы привести разрабатываемое устройство уменьшению размеров корпуса, снижение веса самого устройства, иметь более высокую скорость поворота антенны. Третьей задачей является перекрытие образовавшихся недостатков устройства, встроив и запрограммировав устройство на поддержку LAN соединения

**Основной раздел.** Как правило, для разработки устройства нам нужно определить наши технические характеристики, на основе которых и будет во многом определена элементная база разрабатываемого устройства, а также должны быть определены рабочие условия данного устройства.

Технические данные:

- Исполнительная часть управление моторам имеет мощность 500Вт  $\pm 10\%$ ;
- Разъем проводного соединения LAN;
- Наличие модуля GPS;
- Устройство имеет несколько координат поворота антенны;
- С помощью ручки ПО ВЕРТИКАЛИ обеспечивается управление по азимуту от 0 до 360°;
- С помощью ручки ПО ГОРИЗОНТАЛИ обеспечивается управление по углу места от 0 до 90°.

Рабочие условия применения:

- Диапазон рабочих температур от минус 20 до плюс 60°C;
- Атмосферное давление от 84 до 106 кПа;
- Относительная влажность воздуха 80% при температуре не более 50°C.

Габаритные размеры устройства не должны превышать 190x110x95мм, масса должна быть не более 3кг.

Питание осуществляется от источника постоянного тока с напряжением 24В.

Для поворотного устройства были выбраны следующие основные элементы:

- Энкодер-OCD-EIB1B-10V-IXARC;
- Микроконтроллер-ATmega328p-32KB Flash 2.5V/3.3V/5V-Vercial;
- Дисплей-WH1602A-5V-WINSTAP;
- Мотор-ZY1016-350W 24V2750 RPM-Funsport;
- Модуль GPS - Beitан BN-880.

Структурная схема поворотного устройства представлена на рисунке 1.

Принцип работы прибора основан на обработке сигналов, передаваемых с управляющих устройств на контроллер, с последующим поворотом устройства. Устройство состоит из управляющей части, контроллера и исполнительного устройства.

Исполнительное устройство состоит из мостового усилителя и драйвера управления направлением вращения. Необходимый сигнал поступивший на контроллер обрабатывается и затем передается на моторы устройства.

Данное поворотное устройство предназначено для точного позиционирования антенн, Вращение мотора синхронизировано с положением поворотного переключателя на блоке управления. Это достигается за счет использования высокоточных синхронных двигателей. После завершения рабочего цикла прибор автоматически переходит в режим энергосбережения с минимальным портеблением тока. Переход в рабочее состояние происходит при любом аоздействии на органы управления.

Исходя из структурной схемы, можно определить, принцип работы устройства.

Сигнал с энкодера поступает на микроконтроллер, который управляет информацией, отображаемой на дисплее, исполнительными устройствами через усилитель, которые осуществляют поворот антенну по горизонтали и вертикали.

С микроконтроллера обработанный сигнал поступает на дисплей, на котором будет отображаться угол поворота антенны.

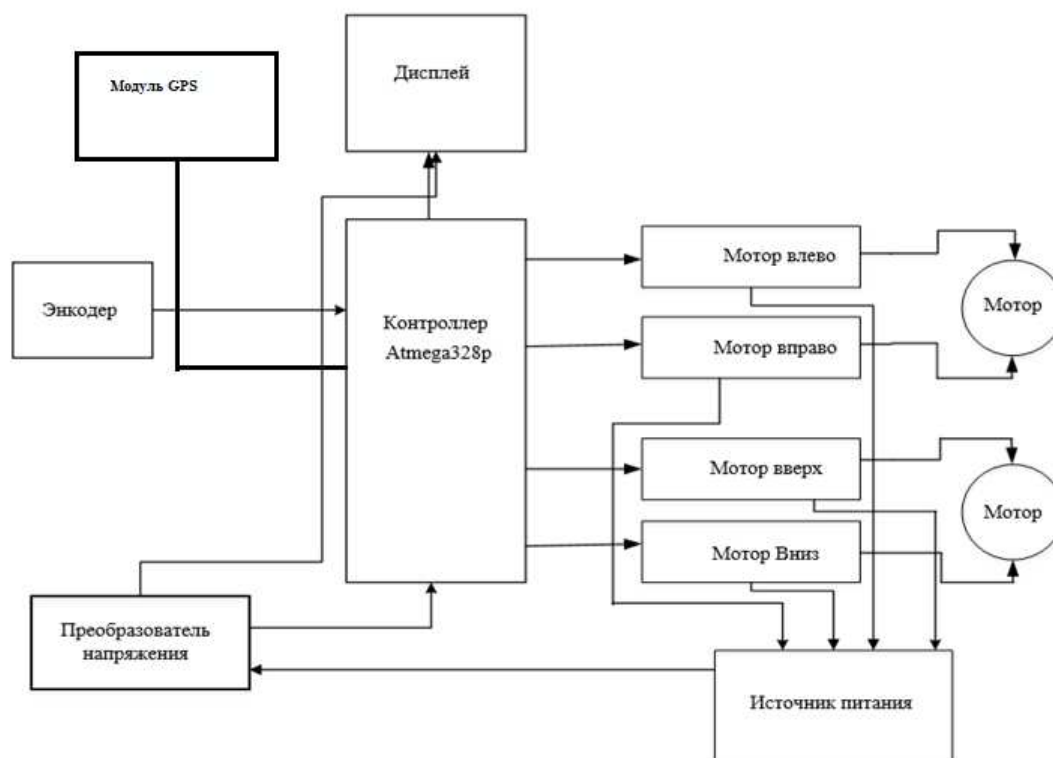


Рисунок 1. – Структурная схема поворотного устройства

После тщательного анализа имеющихся данных, структурной схемы можно приступать к разработке функциональной схемы, на которой будет больше компонентов устройства, а также само устройство будет разбито на отдельные блоки для удобства при составлении схемы электрической- принципиальной.

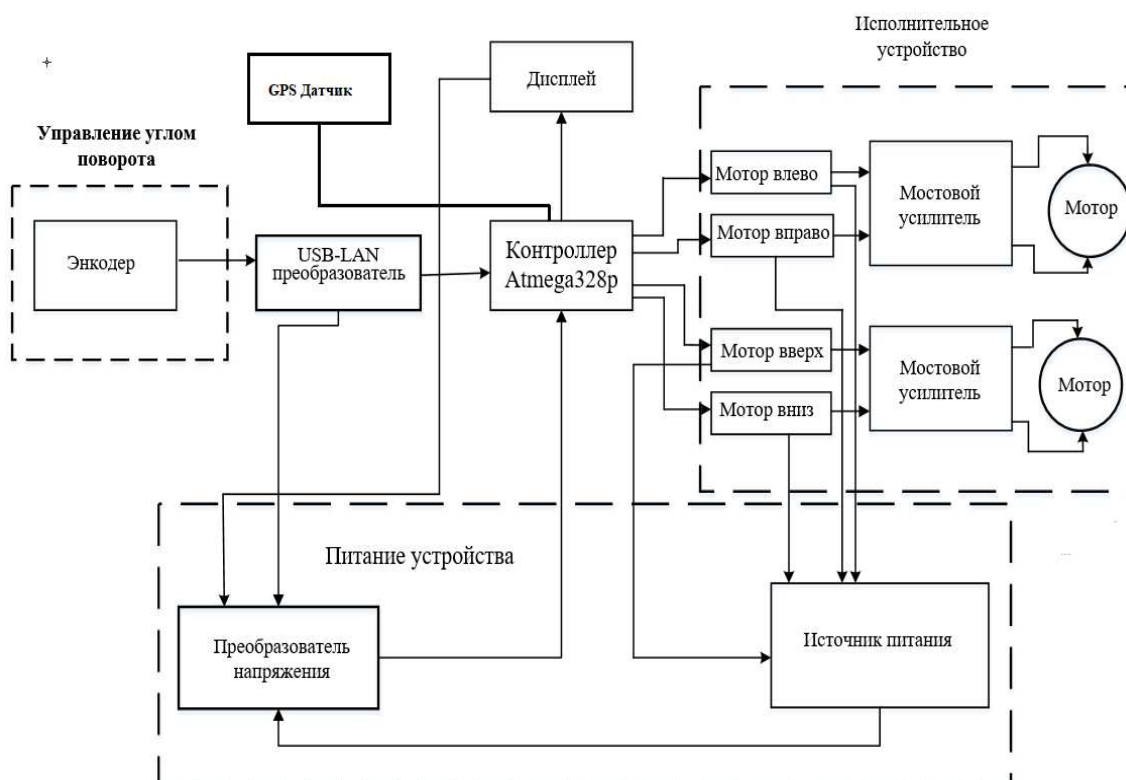


Рисунок 2. – Функциональная схема поворотного устройства



Датчик GPS в данном устройстве необходим для:

- Точного определения местоположения антенны;
- Точного определения направления антенны;
- Точного азимутального положения (если антенна была перемещена человеком в другое место)

**Заключение.** В результате выполнения статьи по теме «Поворотный механизм с GPS датчиком и управлением по LAN» были получены следующие результаты:

- Сформулированы требования к разрабатываемому устройству
- Произведена разработка структурной схемы устройства.
- Произведена разработка функциональной схемы устройства.

После осуществления вышеуказанных пунктов, было произведено экономическое обоснование дипломного проекта, рассчитана полная стоимость изделия. Информация об аналогах показала, что разработанное устройство более конкурентоспособно по цене, и не уступает им по характеристикам. Следовательно, производство данного устройства можно считать экономически и технически целесообразным.

#### ЛИТЕРАТУРА

1. Материал из docplayer. Поворотное устройство для антенны [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docplayer.ru/31343592-Povorotnoe-ustroystvo-dlya-antenny.html> – Дата доступа: 20.09.2019.
2. Материал из electrosam – информационно познавательный сайт. Энкодеры. Виды и работа. Особенности и применение. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/enkodery/> – Дата доступа: 20.09.2019.
3. Материал из Википедии — свободной энциклопедии. Дисплей [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://ru.wikipedia.org/wiki/Дисплей>. –Дата доступа: 20.09.2019

УДК 004.273

АНАЛИЗ СОВРЕМЕННОЙ АРХИТЕКТУРЫ МОБИЛЬНОГО ПРИЛОЖЕНИЯ  
ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ АНДРОИД

Е.Н. ГОРОВОЙ

(Представлено: М.В. ДЕКАНОВА)

В данной статье рассматривается архитектура и общие подходы к реализации мобильного приложения для операционной системы Android, выделяются основные слои и уровни архитектуры, необходимые для создания качественного проекта.

**Введение.** Процесс разработки приложения представляет собой написание огромного количества кода, создания новых экранов и модулей, зачастую сопровождается изменением изначальных требований, добавлением новых разработчиков к работе. Правильный выбор базовой архитектуры приложения позволит создать легко масштабируемый и модифицируемый проект, позволит уменьшить время, требующееся новым разработчикам, чтобы присоединиться к работе над проектом, обеспечит организованность и чистоту кода.

Подход, позволяющий реализовать хорошо организованный и легко модифицируемый проект, должен отвечать следующим требованиям:

- независимость от способа предоставления данных;
- независимость от пользовательского интерфейса;
- независимость от используемых библиотек;
- тестируемость.

**Независимость от способа представления данных.** Данные могут предоставляться через веб-сервис, или храниться в локальной базе данных. Бизнес-логика не зависит от способа предоставления данных.

**Независимость от пользовательского интерфейса.** Пользовательский интерфейс может быть изменен без изменения остальной части системы, также возможно создать несколько реализаций интерфейса, например, для мобильной версии, и версии для Android TV.

**Независимость от используемых библиотек.** Архитектура приложения не должна зависеть от наличия каких-либо библиотек, это позволит использовать их в качестве инструмента, а не встраивать систему в ограничения, создаваемые ими.

**Тестируемость.** Бизнес-логика может быть протестирована без использования пользовательского интерфейса, базы данных, веб-сервера или любого другого внешнего элемента [2].

Исходя из данных требований можно выделить четыре основных слоя приложения:

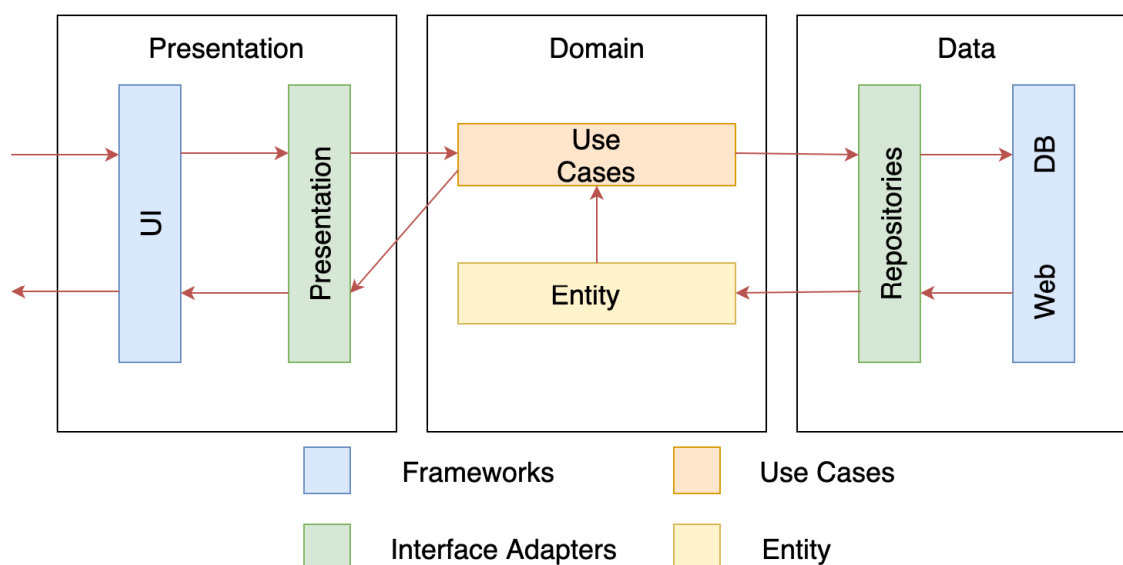
- Entities. Слой данных;
- Use Cases (Interactors). Бизнес-логика приложения;
- Interface Adapters. Адаптеры между бизнес-логикой и пользовательским интерфейсом, а также между бизнес-логикой и внешними данными.
- Frameworks. Внешний слой, в нем находится все остальное: пользовательский интерфейс, база данных, http-клиент и другое.

Расположив данные слои приложения на уровне можно построить схему процесса обработки данных в приложении (рисунок). Событие пользователя идет в Presenter, тот передает в Use Case. Use Case делает запрос в Repository. Repository получает данные из источника данных, создает Entity, передает его в UseCase. Так Use Case получает все нужные ему Entity. Затем, применив их и свою логику, получает результат, который передает обратно в Presenter. А тот, в свою очередь, отображает результат в пользовательском интерфейсе [1].

Рассмотрим уровни приложения подробнее.

**Data.** Все данные, необходимые для приложения, поставляются из этого слоя через реализацию Repository (интерфейс находится в domain — слое бизнес-логики), который использует Repository Pattern со стратегией, которая, в свою очередь, выбирает нужный источник данных, например, веб-сервис или базу данных [3].

Например, для получения конкретного пользователя по ключу источником данных выбирается база данных, если пользователь уже загружен в неё, в противном случае отправляется запрос на получение данных в веб-сервис, и сохраняется в базу данных.



**Рисунок. – Схема процесса обработки данных в приложении**

Главная идея всего заключается в том, что происхождение данных является понятным для клиента, которого не волнует, поступают данные из памяти, базы данных или веб-сервиса, ему важно только то, что данные будут получены и доступны.

**Domain.** На данном уровне происходит вся бизнес-логика приложения и преобразование данных в данные, подходящие для отображения, если это требуется. Все внешние компоненты используют интерфейсы для связи с бизнес-объектами.

**Presentation.** Presenters на данном уровне связываются с Use Cases для получения данных для отображения и дальнейшей их передачи на пользовательский интерфейс. Так же на данном уровне происходят все анимации и отображение данных.

Стоит заметить, что существует множество паттернов, для реализации уровня представления данных, наиболее популярные из них – Model-View-Presenter (MVP), Model-View-ViewModel (MVVM), а также Model-View-Controller (MVC).

**Закключение.** Архитектура приложения является одной из самых важных вещей в разработке программы. Изменение архитектуры – это сложная и длительная задача с использованием уже существующего кода, которая может привести к появлению многих новых ошибок в работе приложения. Поэтому нужно быть осторожным при выборе архитектуры еще на этапе разработки приложения.

В этой статье рассмотрен один из подходов к разработке мобильных приложений, а именно подход «Clean Architecture», определяющий слои, разделяющие уровни мобильного приложения, что, в свою очередь, улучшает тестируемость кода. Модули становятся независимыми друг от друга, что дает возможность их самостоятельной разработки и повторного использования, а также упрощает внедрение новых функций в проект.

#### ЛИТЕРАТУРА

1. A Guided Tour inside a clean architecture code base. [Электронный ресурс]. Режим доступа: <https://proandroiddev.com/a-guided-tour-inside-a-clean-architecture-code-base-48bb5cc9fc97>. – Дата доступа: 15.09.2019.
2. Basic concepts of software architecture patterns in Android. [Электронный ресурс]. Режим доступа: <https://android.jlelse.eu/basic-concepts-of-software-architecture-patterns-in-android-c76e53f46cba>. – Дата доступа: 16.09.2019.
3. Clean architecture for Android with Kotlin: a pragmatic approach for starters. [Электронный ресурс]. Режим доступа: <https://antonioleiva.com/clean-architecture-android/>. – Дата доступа: 16.09.2019.

УДК 004.273

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОДХОДОВ РЕАЛИЗАЦИИ УРОВНЯ ПРЕДСТАВЛЕНИЯ  
ДАННЫХ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Е.Н. ГОРОВОЙ

(Представлено: М.В. ДЕКАНОВА)

В данной статье рассматриваются и сравниваются основные подходы к реализации уровня представления данных мобильного приложения для операционной системы Андроид.

**Введение.** Существует множество подходов к реализации уровня представления данных, в данной статье рассматриваются три основных: Model View Controller (MVC), Model View Presenter (MVP), а также Model View ViewModel (MVVM). Данные подходы имеют общую цель – отделить логику приложения от отображения, но реализация отличается, и выбор зачастую зависит от взглядов конкретного разработчика.

Рассмотрим понятия общие для трех подходов, а именно Model и View.

Под Моделью, обычно понимается часть, содержащая в себе функциональную бизнес-логику приложения. Модель должна быть полностью независима от остальных частей продукта. Модельный слой ничего не должен знать об элементах дизайна, и каким образом он будет отображаться. Достигается результат, позволяющий менять представление данных, то, как они отображаются, не трогая саму Модель.

В обязанности Представления входит отображение данных, полученных от Модели. Однако, представление не может напрямую влиять на модель. Можно говорить, что представление обладает доступом «только на чтение» к данным.

**Model View Controller.** Подход MVC можно охарактеризовать двумя пунктами:

- Представление — это визуальная проекция модели;
- Контроллер — это соединение между пользователем и системой.

Ниже представлена диаграмма, иллюстрирующая идеологию подхода (рисунок 1).

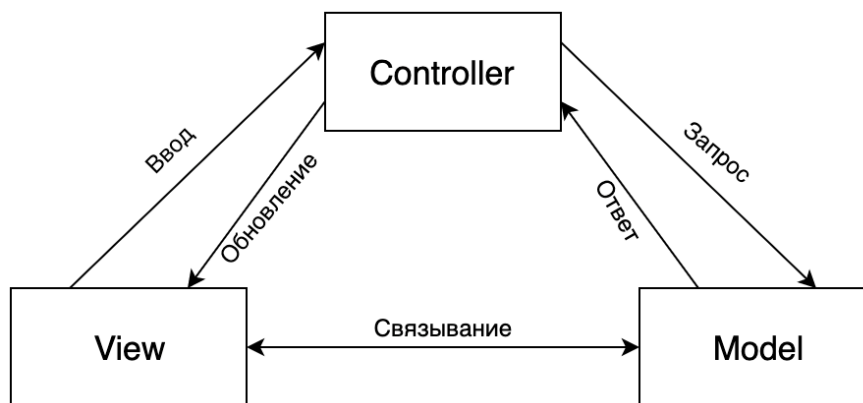


Рисунок 1. – Диаграмма подхода MVC

Характеризующим элементом данного подхода является то, что представление связано с моделью, а основной идеей – разделить вводом и выводом данных. Контроллер принимает входные данные, а представление – выходные, однако большое число операций происходит и между ними [1].

Из плюсов можно выделить малое количество кода, по сравнению с другими подходами. Из минусов – слабое разделение ответственности, сложная инкапсуляция, сложное юнит-тестирование.

Данный подход хорошо подходит только для небольших проектов так как очень плохо масштабируется.

**Model View Presenter.** Решение этих проблем кроется за созданием абстрактного интерфейса для представления. Таким образом данная архитектура облегчает юнит-тестирование, а также может многократно использоваться, потому что представление может реализовать несколько интерфейсов.

Характеризующим элементом данного подхода является то, что view закрыта интерфейсом от presenter'a, а главной идеей – сделать presenter независимым от view. Диаграмма идеологии подхода MVP представлена на рисунке 2.

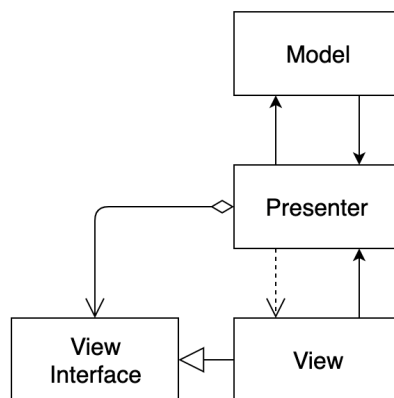


Рисунок 2. – Диаграмма подхода MVP

Каждое представление должно реализовывать соответствующий интерфейс. Интерфейс представления определяет набор функций и событий, необходимых для взаимодействия с пользователем. Презентер должен иметь ссылку на реализацию соответствующего интерфейса. Логика представления должна иметь ссылку на экземпляр презентера. Все события представления передаются для обработки в презентер и практически никогда не обрабатываются логикой представления (в т.ч. создание других представлений) [2].

Таким образом можно выделить следующие плюсы: презентер легко тестируется, презентер не зависит от представления, таким образом, презентер можно повторно использовать для других представлений.

Из минусов: необходимость создавать и поддерживать интерфейсы для представлений и лишний шаблонный код.

Данная архитектура хорошо масштабируется, тестируется, и хорошо подходит для больших проектов.

**Model View ViewModel.** ViewModel так же, как и презентер, содержит логику пользовательского интерфейса. Когда пользователь нажимает на кнопку, это событие направляется в ViewModel, который затем решает, что с ним делать. ViewModel может преобразовывать данные из модели так, чтобы они были легко отображены на экране. Часто информация, содержащаяся в модели, не может непосредственно использоваться на экране. Это наиболее вероятно, когда у вас нет полного контроля над моделью. Например, если вы получаете данные от сторонних веб-сервисов или же из базы данных существующего приложения[3].

Основным отличием данного подхода является то, что ViewModel не может общаться со View напрямую. Вместо этого она представляет легко связываемые свойства и методы в виде команд. View может привязываться к этим свойствам, чтобы получать информацию из ViewModel и вызывать на ней команды (методы). Это не требует того, чтобы View знала о ViewModel.

Плюсы подхода: легко тестируется и масштабируется, View и ViewModel не зависят друг от друга.

Минусы: возрастает сложность кода, так как методы, в отличие от MVP, вызываются не напрямую, а происходит подписка на изменение состояния, или отправленные команды.

Данный подход также подходит для больших проектов, так как легко масштабируется и имеет хорошую тестируемость.

**Заключение.** В заключении стоит отметить, что строго придерживаться только одному подходу – не всегда лучший выбор, ведь главная цель – это отделить представление от бизнес-логики и логики, которая их связывает. Несмотря на то, что внедрение разделения ответственности требует усилий, это хороший способ повысить качество кода в целом, сделать его масштабируемым, легким в понимании и надежным.

## ЛИТЕРАТУРА

1. Model-View-Controller. [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Model-View-Controller>. – Дата доступа: 16.09.2019.
2. Common Android Architectures (MVC vs MVP vs MVVM) [Электронный ресурс]. Режим доступа: <https://medium.com/@mr.anmolsehgal/common-android-architectures-mvc-vs-mvp-vs-mvvm-afd8461e1fee>. – Дата доступа: 17.09.2019.
3. MVP vs MVVM: A Review of Patterns for Android. [Электронный ресурс]. Режим доступа: <https://thinkmobiles.com/blog/mvp-vs-mvvm-android-patterns/>. – Дата доступа: 17.09.2019.

УДК 004.4'236

## JSX – УДОБНАЯ ТЕХНОЛОГИЯ ПРИ РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЙ

**Р.П. ГУСЕВСКИЙ***(Представлено: канд. тех. наук, доц. И.Б. БУРАЧЕНОК)*

*В статье рассмотрены особенности технологии JSX. Описан процесс транспилирования файлов формата JSX в JavaScript-код.*

На первоначальном этапе своего развития «веб» состоял только из статических HTML-страниц и при открытии адреса, сервер возвращал контент страницы. Верстка, стили, скрипты – все было в общей массе. С развитием интернет-технологий стало появляться все больше одностраничных веб-приложений, где основная часть логики сосредоточена на клиентской стороне. На любой Uniform Resource Locator — унифицированный указатель ресурса (URL), сервер отдает один и тот же статичный HTML, а JavaScript-код в браузере пользователя определяет текущий адрес и делает запрос на сервер за нужными данными. При клике на ссылку, страница не перезагружается целиком, а лишь обновляется URL, а JavaScript-код делает новый запрос для следующей страницы. Появление веб-приложений в корне поменяло подход к разработке. Появились новые специальные фреймворки, предназначенные для создания веб-приложений с большим количеством кода и сложной логикой пользовательского интерфейса. Рендеринг (процесс получения изображения по модели с помощью компьютерной программы) данных теперь происходит на клиенте, в ресурсах страницы хранятся статические html-шаблоны, которые заполняются данными, полученные с сервера. Наиболее подходящим способом организации такой массы кода стал *компонентный подход*. Каждый более или менее независимый блок оформляется в отдельный компонент, со своими стилями, шаблоном и JavaScript (JS), а потом из этих блоков собирается страница целиком. Этот подход реализуется большинством современных JS-фреймворков, но с одним отличием: Angular, Ember поощряют создание отдельного файла с шаблоном, а React предлагает писать HTML внутри JS [1]. Технология, которая позволяет нам писать HTML внутри JS – это JavaScript XML (JSX) технология.

JSX – это технология, которая была представлена React. Данная технология расширяет синтаксис JavaScript, которое выглядит подобно XML. Многие думают, что использование JSX это как смесь HTML и JavaScript, но это не совсем так, мы также можем производить манипуляции и с CSS. JSX в основном используется в качестве строительных блоков в React. По своей сути JSX ни что иное как компонента, в которой мы можем контролировать и верстку, и логику, и стили. Все находится в одном месте, что уже дает огромное преимущество в разработке,

Рассмотрим пример написания HTML-элемента в JSX. Для примера рассмотрим обозначения h1-тега, который содержит в себе содержимое в виде строки, как показано на рисунке 1.

```
const element = <h1>Hello, world!</h1>
```

**Рисунок 1. – H1-тег, объявленный в JSX**

Данный синтаксис очень напоминает странную смесь JavaScript с HTML, но в реальности это все только JavaScript. А то, что похоже на HTML, на самом деле является «синтаксическим сахаром» для определения компонентов и их позиционирования внутри разметки. Внутри JSX выражения, атрибуты могут быть довольно легко вставлены (рисунок 2).

```
const myId = 'test'  
  
const element = <h1 id={myId}>Hello, world!</h1>
```

**Рисунок 2. – Вставка атрибутов в JSX**

Самое важное знать, что когда у атрибута присутствует знак тире, то JSX конвертирует его в camelCase нотацию. Также следует помнить о двух специальных случаях: вместо class используется className, а вместо атрибута for используется htmlFor. Так как это зарезервированные слова в JavaScript.

Далее рассмотрим пример JSX, который включает два компонента в div-тег. Часть кода JSX с двумя пользовательскими компонентами показана на рисунке 3.

```
<div>

  <BlogPostsList />

  <Sidebar />

</div>
```

Рисунок 3. – Часть кода JSX с двумя пользовательскими компонентами

Существует ряд правил при объявлении компонент. Тег всегда должен быть закрыт, так как это скорее XML, чем HTML. Поэтому в этом случае используется самозакрывающийся тег. Как описывалось ранее, в React в качестве компонент принято использовать файлы с расширением `jsx`. Как и во всех популярных фреймворках, в React можно генерировать определенную компоненту и возвращать её представление в виде `html`. За отображение представления в компоненте JSX отвечает функция `render`. У функции `render`, имеется одно ограничение, она может вернуть только один `html`-элемент. Так что в случае, когда вам надо вернуть две компоненты, нужно обязательно добавить родителя. В роли может быть вообще абсолютно любой тег. Данный случай очень часто встречается на практике и из-за этого React-разработчики придумали компоненту-обертку для таких случаев и назвали `React.Fragment` или пустой тег при помощи которого можно «оборачивать» один или более возвращаемых узлов (рисунок 4).

```
<React.Fragment></React.Fragment>

или

<></>
```

Рисунок 4. – Синтаксис написания обертки `React.Fragment`

Не маловажным аспектом является транпиляция JSX. Под транпиляцией понимается процесс преобразования кода, написанного с помощью следующих версий языка или на диалектах JavaScript в некий стандартный вариант, понимаемый всеми браузерами [2]. Браузер не может понимать JS-файлы, которые содержат JSX-код. Сначала они должны быть трансформированы в обычный JavaScript. Ранее было упомянуто что JSX опционален, потому что для каждой JSX строки, есть соответствующая JavaScript альтернатива и именно в нее трансформируется JSX. Для примера, следующие две конструкции полностью равнозначны (рисунок 5, 6).

```
ReactDOM.render (

  ReactDOM.div (

    { id: 'test' },

    ReactDOM.h1 (null, 'A title'),

    ReactDOM.p (null, 'A paragraph')

  ),

  document.getElementById ('myapp')

)
```

**Рисунок 5. – Код в формате JS**

А это уже JSX:

```
ReactDOM.render(  
  
  <div id="test">  
  
    <h1>A title</h1>  
  
    <p>A paragraph</p>  
  
  </div>,  
  
  document.getElementById('myapp')  
  
)
```

**Рисунок 6. – Код в формате JSX**

Первоначально рассмотрим наиболее простой случай. Сложный синтаксис на чистом JS сравним с использованием JSX. Во время написания, самым популярным способом является транспиляция с использованием Babel. Babel – это транспайлер, переписывающий код новых стандартов JavaScript в код на предыдущем стандарте ES5, который поддерживается на всех браузерах [3]. Вывод переменной JS в тэге p представлен на рисунке 7.

```
const paragraph = 'A paragraph'  
  
ReactDOM.render(  
  
  <div id="test">  
  
    <h1>A title</h1>  
  
    <p>{paragraph}</p>  
  
  </div>,  
  
  document.getElementById('myapp')  
  
)
```

**Рисунок 7. – Вывод переменной JS в тэге p**

JSX также в себя включает работу с JavaScript. Всякий раз, когда нужно в представление компоненты добавить какой-либо JS-код (функции, перебирающие методы и многое другое), его необходимо объявлять в фигурных скобках { }.

Для более простого понимания можно считать JSX как Javascript-объект. React создает элемент (JavaScript-объект) при помощи функции createElement (рисунок 8). Данные объекты носят название «React-элементы».

```
const element = React.createElement(  
  'h1',  
  {className: 'greeting'},  
  'Hello, world!'  
)
```



**Рисунок 8. – Пример создания «React-элемента»**

React читает эти объекты, использует их для построения дерева DOM и хранит их в актуальном состоянии. На таких объектах и строится примитивное React-приложение.

Рассмотрим работу CSS в JSX. Без различных сторонних библиотек разработчики React навязывают написание стилей в атрибуте style. С одной стороны, это кажется ужасным решением, но если посмотреть то, как обстоит работа с CSS подробнее, то оказывается данный подход имеет свои плюсы. Во-первых, вместо принятия строки, содержащей свойства CSS, JSX атрибут style принимает только объект. Пример определения стилей в JSX показан на рисунке 9. Это означает то, что определение свойства происходит в объекте:

```
var divStyle = {  
  
  color: 'white'  
  
}  
  
ReactDOM.render(<div style={divStyle}>Hello World!</div>, mountNode)
```

**Рисунок 9 – Пример определения стилей в JSX**

CSS значения, которые объявляются в JSX, немного другие, чем чистый CSS: ключевые имена пишутся, как и атрибуты в camelCase нотации, значения – это просто строки, а также каждое «определение» отделяется при помощи запятой. Также JSX позволяет компонентам полностью инкапсулировать их стили. Данный подход покрывает примитивные случаи в объявлениях стилей, но, если описывать анимации либо медиа запросы, код становится нагроможденным при объявлении в JSX сложных CSS-объектов.

В JSX довольно частой операцией является назначение значений атрибутам. Вместо того, чтобы делать это вручную, как показано на рисунке 10.

```
<div>  
  
  <BlogPost title={data.title} date={data.date} />  
  
</div>
```

**Рисунок 10 – Назначение значений атрибутов вручную**

Для того, чтобы избежать таких длительных объявлений, на помощь приходит «оператор расширения» или «spread operator». Оператор расширения – позволяет инициализировать части литерала массива из итерируемого выражения (например, другого литерала массива) или развернуть выражение в несколько аргументов (в вызовах функций) [4]. Используем оператор расширения для назначения значений атрибутов (рисунок 11). Свойства из объекта data используются как атрибуты в автоматическом порядке.

```
<div>  
  
  <BlogPost {...data} />  
  
</div>
```

**Рисунок 11 – Назначение значений атрибутов при помощи оператора расширения**

Таким образом, можно сделать вывод, что JSX является очень доступной технологией в освоении и очень практичной для разработки и проектирования веб-приложений, так как приложение будет состоять из маленьких строительных блоков, так называемыми React-компонентами. И при помощи JSX формата код станет более читабельным, структурированным, а возможность хранить все в одном месте дает возможность верно и правильно закладывать архитектуру приложения под определенные потребности веб-приложения.

**Заключение.** При разработке программного обеспечения большинство людей находятся в поисках тех или иных технологий, которые будут подходить по их критериям. Если рассматривать данную технологию в качестве использования её вместе с ReactJS, то она имеет ряд преимуществ, которые можно дополнить различными библиотеками, что поможет быстро, а самое главное качественно разработать программное обеспечение.

#### ЛИТЕРАТУРА

- 1 Habr. Сайт интересных статей по программированию. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/311226>. – Дата доступа: 24.08.2019.
- 2 Happy JavaScripting. Сайт для изучения JavaScript. [Электронный ресурс]. – Режим доступа: <https://kodaktor.ru>. – Дата доступа: 25.08.2019.
- 3 Тостер – вопросы и ответы для IT-специалистов [Электронный ресурс]. – Режим доступа: <https://toster.ru/q/393850>. – Дата доступа: 10.09.2019.
- 4 TechNet Microsoft [Электронный ресурс]. – Режим доступа: [https://technet.microsoft.com/ru-ru/dn919259\(v=vs.90\)](https://technet.microsoft.com/ru-ru/dn919259(v=vs.90)). – Дата доступа: 12.09.2019.

УДК 004.588

## ВАЖНОСТЬ ОСНОВНЫХ CSS-ТЕХНОЛОГИЙ НА ПРИМЕРЕ РАЗРАБОТЫВАЕМОГО ИГРОВОГО ВЕБ-ПРИЛОЖЕНИЯ «ОБУЧЕНИЕ ВОЗМОЖНОСТЯМ CSS»

**Р.П. ГУСЕВСКИЙ**

(Представлено: канд. тех. наук, доц. И.Б. БУРАЧЕНОК)

*Приводится обоснование необходимости в разработке современных игровых приложений. Анализируются используемые технологии при создании веб-приложений. Подробно рассмотрены CSS-технологии, с целью их использования для демонстрации в виде увлекательной игры «Обучение возможностям CSS»*

Большинство людей проводят свое свободное время в играх, особенно дети и подростки. Есть такой подход к обучению, как подача материала для изучения в игровой форме, которая будет полезна и интересна людям любого возраста. В рамках данной темы ведется разработка кроссплатформенного, игрового веб-приложения «Обучение возможностям CSS».

Рассмотрим, что такое CSS. Это Каскадные таблицы стилей, которые являются следствием дальнейшего развития HTML и дают нам возможность перейти на следующий уровень представления информации. Таблицы стилей позволяют разделить смысловое содержимое странички и его оформление. При изучении CSS можно столкнуться с рядом проблем: с различными стилями, свойствами, а также с технологиями для верстки блоков и таблиц. Что можно сказать о HTML – это язык разметки, который является инструментом для создания гиперссылок, вставки изображений, таблиц и много другого [1].

В современном мире веб-технологии и языки программирования развиваются очень быстрыми темпами, соответственно увеличивается и количество веб-разработчиков, а также желающих пользователей, которые хотят изучать или развивать свои навыки программирования. Однако при изучении веб-технологий многие сталкиваются с различными проблемами, и тратят много времени на поиски решений той или иной проблемы. Для того, чтобы облегчить изучение CSS, предлагается разработать веб-приложение, которое ориентировано на обучение сложным свойствам CSS и его технологиям верстки: flexbox, grid layout, работа с анимацией, позиционирование элементов на веб-странице при помощи свойства position. Благодаря визуализации и игровой подаче, рассматриваемые технологии станут доступнее для изучения и запоминания материала. Вряд ли можно стать программистом только играя. Но можно получить положительный заряд на старте, изучить основные алгоритмические конструкции и логику программирования, приобрести навыки планирования действий при решении практических задач [2].

При реализации данного веб-приложения планируется использование следующих технологий: ReactJS – в качестве фреймворка на клиентской стороне, для разработки серверной части – программная платформа Node.js и фреймворк Express, а также язык программирования Javascript. ReactJS – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React может использоваться для разработки одностраничных и мобильных приложений. Его цель – предоставить высокую скорость, простоту и масштабируемость. В качестве библиотеки для разработки пользовательских интерфейсов React часто используется с другими библиотеками, такими как Redux [3]. Node.js – программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения [4]. Express – это минималистичный и гибкий веб-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений [5]. JavaScript – это полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах [6].

Приложение будет состоять из двух страниц: основная страница с перечисленными пунктами изучаемых технологий и различных свойств CSS и динамически подгружаемая страница с самой игрой, которая характеризует выбранную технологию.

Далее осуществим описание технологий верстки и свойств, которые используются в приложении и с которыми чаще всего у людей возникают проблемы. В CSS имеется 4 самых проблемных технологии: flexbox, grid layout, работа с анимацией, позиционирование элементов на веб-странице при помощи свойства position.

Flexbox – это новый способ располагать блоки на странице. Это технология, созданная именно для раскладки элементов, в отличие от float-ов. С помощью Flexbox можно легко выравнивать элементы по горизонтали и по вертикали, менять направление и порядок отображения элементов, растягивать блоки

на всю высоту родителя или прибавить их к нижнему краю [7]. Но есть один нюанс, что эта технология поддерживается не всеми браузерами:

- 1) IE: 11.0, 10.0 -ms-;
- 2) Firefox: 28.0, 18.0 -moz-;
- 3) Chrome: 29.0, 21.0 -webkit-;
- 4) Safari: 6.1 -webkit-;
- 5) Opera: 12.10 -webkit-;
- 6) iOS Safari: 6.1 -webkit-;
- 7) Opera Mini: 8;
- 8) Android Browser: 4.4, 4.1 -webkit-;
- 9) Chrome for Android: 44.

Проблемы с данной технологией возникают тогда, когда огромную роль играет поддержка старых браузеров, в таком случае нужно отказаться от данной технологии. Однако на данный момент развитие идет в сторону использования новых технологий нежели поддержки старых браузеров. Следует отметить, что многие пользователи не знают какие свойства существуют и как ими пользоваться для правильного расположения элементов. Имеется как старый синтаксис, так и новый, и у каждого из них имеются свои нюансы и с этим тоже необходимо разобраться. Таким образом, в разрабатываемом приложении обязательно будут рассмотрены все перечисленные аспекты.

Следующей технологией для рассмотрения была выбрана технология Grid layout. Grid layout – приносит инструмент двумерного макета в веб, с возможностью размещения элементов в строках и столбцах. CSS-сетка может использоваться для достижения многих различных макетов. Он разделяет страницу на крупные регионы, или определяет отношения с точки зрения размера, положения и слоя, между частями элемента управления, построенных из HTML-примитивов.

Как и таблицы, Grid layout дает возможность автору позиционировать элементы в колонках (columns) и строках (rows). Тем не менее, в отличие от таблиц, Grid layout не имеет контентной структуры, поэтому включение в таблицы большого количества макетов невозможно [8].

CSS3-анимация придаёт сайтам динамичность. Она оживляет веб-страницы, улучшая взаимодействие с пользователем. В отличие от CSS3-переходов, создание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.

CSS3-анимация может применяться практически для всех html-элементов, а также для псевдо-элементов :before и :after. При создании анимации не стоит забывать о возможных проблемах с производительностью, так как на изменение некоторых свойств требуется много ресурсов [9].

Далее рассмотрим свойство CSS position, которое устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице. Данное свойство принимает 4 основных значения:

- 1) Absolute – указывает, что элемент абсолютно позиционирован, при этом другие элементы отображаются на веб-странице словно абсолютно позиционированного элемента и нет. Положение элемента задается свойствами left, top, right и bottom, также на положение влияет значение свойства position родительского элемента. Так, если у родителя значение position установлено как static или родителя нет, то отсчет координат ведется от края окна браузера. Если у родителя значение position задано как fixed, relative или absolute, то отсчет координат ведется от края родительского элемента;
- 2) Fixed – по своему действию это значение близко к absolute, но в отличие от него привязывается к указанной свойствами left, top, right и bottom к точке на экране и не меняет своего положения при прокрутке веб-страницы.
- 3) Relative – положение элемента устанавливается относительно его исходного места. Добавление свойств left, top, right и bottom изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения;
- 4) Static – элементы отображаются как обычно. Использование свойств left, top, right и bottom не приводит к каким-либо результатам.
- 5) Inherit – наследует значение родителя [10].

Таким образом, вышеописанные технологии и свойства CSS изначально будут включены в разрабатываемое веб-приложение с целью исключить слабые места разработчиков, а также ознакомить и дать правильную базу знаний пользователям для работы с этими технологиями.

**Заключение.** Все описанные в статье нюансы, будут учтены в разрабатываемом веб-приложении. В перспективе планируются доработки данного приложения с помощью добавления новыми технологиями CSS.

## ЛИТЕРАТУРА

1. Web2Studio – Яркие решения [Электронный ресурс]. – 2019. – Режим доступа: – <http://webstudio2u.net/ru/programming/127-programming.html>. – Дата доступа: 07.09.2019
2. Игры, которые учат программированию / Хабрахабр [Электронный ресурс]. – 2019. – Режим доступа: – <https://m.habrahabr.ru/post/273003>. – Дата доступа: 07.09.2019.
3. ReactJS / Википедия [Электронный ресурс]. – 2019. – Режим доступа: – <https://ru.wikipedia.org/wiki/React>. – Дата доступа: 08.09.2019.
4. NodeJS / Википедия [Электронный ресурс]. – 2019. – Режим доступа: – <https://ru.wikipedia.org/wiki/Node.js>. – Дата доступа: 08.09.2019.
5. Express – фреймворк веб-приложений Node.js / [Электронный ресурс]. – 2019. – Режим доступа: – <http://expressjs.com/ru>. – Дата доступа: 08.09.2019.
6. Основы Javascript / MDN [Электронный ресурс]. – 2019. – Режим доступа: – [https://developer.mozilla.org/ru/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics). – Дата доступа: 08.09.2019.
7. CSS 3 flexbox – описание всех свойств модуля, примеры верстки / [Электронный ресурс]. – 2019. – Режим доступа: – <https://html5book.ru/css3-flexbox>. – Дата доступа: 12.09.2019.
8. CSS Grid Layout / MDN [Электронный ресурс]. – 2019. – Режим доступа: – [https://developer.mozilla.org/ru/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid_Layout). – Дата доступа: 13.09.2019.
9. CSS3-анимация (свойство animation) [Электронный ресурс]. – 2019. – Режим доступа: – <https://html5book.ru/css3-animation>. – Дата доступа: 14.09.2019.
10. Position / htmlbook.ru [Электронный ресурс]. – 2019. – Режим доступа: – <http://htmlbook.ru/css/position>. – Дата доступа: 14.09.2019.

УДК 519.72

**ВОЗМОЖНЫЕ ВАРИАНТЫ ШИФРОВАНИЯ ТЕКСТОВЫХ ДАННЫХ,  
АНАЛИЗ И ВЫБОР ОПТИМАЛЬНОГО ТИПА ШИФРОВАНИЯ ТЕКСТОВЫХ ДАННЫХ  
ДЛЯ ДАЛЬНЕЙШЕЙ РЕАЛИЗАЦИИ****Е. ДЕНИСОВА***(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)*

*В статье рассматриваются различные варианты шифрования текстовых данных. В ходе анализа был выбран оптимальный вариант шифрования текстовых данных. Проектирование приложения для шифрования и дешифрования текстовых данных модифицированным методом Владимира Сизова Проведены исследования по актуальности разработки данного приложения.*

**Введение.** Люди издавна использовали шифрование – как способ защиты информации. Идея скрыть в тексте тайные значения и сообщения почти так же стара, как и само искусство письма. За века своего существования человечество придумало множество способов хранения тайны.

**Основной раздел.** Степень изученности методов шифрования достаточно высокая. Каждый год создается множество программ, литературы, посвящённым криптографическим системам. Криптографическая система – семейство преобразований шифра и совокупность ключей. Существуют Симметричные и Асимметричные криптосистемы.

Симметричные криптосистемы (с секретным ключом – SecretKeySystems) – данные криптосистемы построены на основе сохранения в тайне ключа шифрования. Процессы шифровки и расшифровки используют один и тот же ключ. Секретность ключа является постулатом.

Асимметричные криптосистемы (системы открытого шифрования – с открытым ключом PublicKeySystems) – смысл данных криптосистем состоит в том, что для шифровки и расшифровки используются разные преобразования. Одно из них – зашифрование – является абсолютно открытым для всех. Другое же – расшифрование – остается секретным.

Рассмотрим возможные варианты шифрования текстовых данных:

1. Шифр XOR - это алгоритм шифрования данных с использованием исключительной дизъюнкции. Алгоритм шифрования заключается в «наложении» последовательности случайных чисел на текст, который необходимо зашифровать, Последовательность случайных чисел называется гамма-последовательность, и используется для шифрованной и расшифровки данных;

2. Шифр Цезаря - один из наиболее простых и широко известных алгоритмов шифрования текстовых данных. Алгоритм шифрования Цезаря заключается в замене каждого символа входящего сообщения на символ, который находится на некотором константном расстоянии с правой или левой стороны. Расстояние при этом называют - ключом;

3. Шифр Скитала - шифрование текста при помощи деревянного цилиндра и пергамента, также известен как шифр Древней Спарты. Для шифрования текст используется цилиндр фиксированного диаметра, на который наматывается узкая полоска пергамента. Сообщение записывают вдоль цилиндра, а затем разматывают, в итоге получается зашифрованное сообщение, которое можно расшифровать, применяя цилиндр того же диаметра. При этом диаметр цилиндра выступает в роли ключа шифрования;

4. Квадрат Полиция - метод шифрования текстовых данных с помощью замены символов. Для шифрования используется квадратная таблица, в которую вписаны все буквы шифруемого алфавита. Если букв больше, то можно их совмещать или добавлять ячейки с произвольными знаками.

5. Метод шифрования текста функцией косинуса Владимиром Сизовым предложен впервые в 2005 на международной конференции Рускрипто. По координатной оси X расставляются компьютерные символы в любом порядке. Каждому символу соответствует свой порядковый номер от 1 до 256. По оси Y расставляем те же самые символы в любом (таком же или другом) порядке. Им также присваивают порядковые номера от 1 до 256. Функция, посимвольно переводит исходный текст в зашифрованный текст.

Исходя из вышеперечисленных вариантов можно сделать вывод, что каждый алгоритм имеет свои плюсы и минусы. Все вышеперечисленные алгоритмы были изучены и запрограммированы, но чтобы добиться надежного шифрования текстовых данных, необходимо привнести новизну в существующие алгоритмы. Таким образом был создан модифицированный метод шифрования Владимира Сизова. Метод был модифицирован произвольной периодической функцией с любым числом секретных ключей и аperiodической неограниченной функции типа обобщенного степенного ряда Фробениуса.

Актуальность разработки модифицированного метода шифрования В. Сизова текстовых данных заключается в возможности использовать для шифрования произвольные периодические и неограниченные непериодические функции.

Основу разрабатываемого приложения составляет возможность шифровать и дешифровать текстовые данные с помощью усовершенствованного метода шифрования В. Сизова. Аналогов подобного приложения найдено не было. Метод усовершенствовать также использованием аperiodической неограниченной функцией применением более десятка секретных ключей для отдельного шифрования каждого второго, каждого третьего, каждого третьего символа отдельными ключами-это делает метод более защищенным от внешних атак злоумышленников.

**Заключение.** В ходе данного исследования и анализа были сделаны выводы о выборе оптимального шифрования текстовых данных и обоснование актуальности создания приложения для шифрования данных несколькими ключами с помощью аperiodических функций и сложными периодически и функциями.

#### ЛИТЕРАТУРА

1. Сизов, В.П. Криптографические алгоритмы на основе тригонометрических функций [Электронный ресурс]. – URL: <https://www.ruscrypto.ru/association/archive/rc2005.html>. – Дата доступа: 22.09.2019.
2. Каминский, Л.П. Информационные технологии / Л.П. Каминский, В.А. Степанов // Тригонометрическая криптография. – Красноярск : Сибирский федеральный университет, 2005. – С. 38–41. – Дата доступа: 23.09.2018.

УДК 519.72

МОДИФИЦИРОВАННЫЙ МЕТОД В. СИЗОВА  
НА СЛУЧАЙ НЕПЕРИОДИЧЕСКОЙ НЕОГРАНИЧЕННОЙ ФУНКЦИИ

Е. ДЕНИСОВА

(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)

В статье рассматривается усовершенствованный алгоритм метода шифрования текстовых данных Владимира Сизова.

**Введение.** Шифрование— обратимое преобразование информации в целях сокрытия от неавторизованных лиц, с предоставлением, в это же время, авторизованным пользователям доступа к ней. Защищенность данных пользователей в современном мире очень актуальна. Взять как самый простой пример - пароли от интернет-банкингов. Пароли не просто так даны, они необходимы для защиты. Но также необходимо защищать и свои текстовые данные.

**Основная часть.** В данной работе рассмотрим модифицированный метод В. Сизова на случай непериодической неограниченной функции. Функция представляет собой сумму дробно-степенных функций. Аргумент функции один и тот же, представляет собой аффинное преобразование порядкового номера шифруемого символа

$$Z(i) = (i + w_0)t + fi$$

где  $Z$  – аргумент функции;

$i$  – порядковый номер в шифруемой фразе;

$w_0$  – параметр частотной модуляции или сдвиг порядковых номеров;

$t$  – момент времени;

$fi$  – начальная фаза.

За исключением порядкового номера  $i$ , все остальные параметры – параметры ключа действительные числа двойной точности.

Функция шифрования имеет вид

$$Y = [X + F] \bmod P. \quad (3)$$

Функция дешифрования имеет вид

$$X = [Y - F] \bmod P, \quad (4)$$

где  $F(i, a_1, a_2, a_3, b_1, b_2, b_3, w_0, t, fi) =$

$$= a_1(Z)^{b_1} + a_2(Z)^{b_2} + a_3(Z)^{b_3} = \quad (5)$$

$$= a_1((i + w_0)t + fi)^{b_1} + a_2((i + w_0)t + fi)^{b_2} + a_3((i + w_0)t + fi)^{b_3}.$$

Функция  $F$  имеет 9 ключей:  $a_1, a_2, a_3$  – амплитуды,  $b_1, b_2, b_3$  – показатели дробно-степенных функций,  $w_0$  – параметр частоты модуляции,  $t$  – момент времени,  $fi$  – начальная фаза.

Корректность формулы шифрования (3) и дешифрования (4) обсуждались с заведующим кафедры высшей математики, доцентом, кандидатом физ.-мат. наук А.А. Козловым.

Алгоритм шифрования можно разделить на несколько шагов:

– Исходный текст заносится в символьный массив массив `char str[m+1]`. Порядковый номер каждого символа в ASCII [4] заносится в `baza[i]`;

– Согласно алгоритму применяется формула Сизова

$baza2[i] = baza[i] + 0.5(double)$  (действ. 2-й точности);

–  $baza1[i] = [baza2[i] + F(Z)] \bmod P$  (целая переменная).

Особенностью выработанного алгоритма является следующая модификация метода:

Если  $baza1[i] < C$ , то  $baza1[i] = baza1[i] + p - 1$ .

Шифрованный текст хранится в массиве чисел `baza1[i]`

Алгоритм дешифрования:



1. Извлекаем из массива  $baza1[i]$  целое число – шифрованный символ.
2. Образует действительное число из значения  $baza1[i]$  по формуле  $s2=baza1[i]+0,5$  (действ. 2-й точности) (данная часть алгоритма совпадает с алгоритмом Сизова).
3. Вычисляем функцию  $Z(i)$  (действ. 2-й точности).
4. Составляем разность:  $s3 = [s2 - s1] \bmod F$ , что соответствует массиву  $baza3[i]$  (целая переменная), так как  $s2 = x + F(Z)$ ,  $s1 = F(Z)$ .  $s3 = x$  (целая переменная),  $s3$  – целочисленный массив поскольку от действительной разности  $s2 - s1$  взята целая часть, затем остаток по  $\bmod F$ .
5. Применяем модернизированный алгоритм Сизова:  
Если  $baza3[i] < C$ , то  $baza3[i] = baza3[i] + p - 1$ .  
Получим векторный массив, в котором число элементов  $m$  (число символов сообщения) должно совпадать с исходным массивом  $baza[i]$ .
6. На конечном этапе можно обратить порядковый номер каждого символа фразы по таблице ASCII в сам символ.

Исходя из вышеперечисленных вариантов можно сделать вывод, что каждый алгоритм имеет свои плюсы или минусы. Все вышеперечисленные алгоритмы были изучены и запрограммированы, но чтобы добиться надежного шифрования текстовых данных, необходимо привести новизну в существующие алгоритмы. Таким образом был выбран метод шифрования Владимира Сизова. Но метод был усовершенствован аperiodической неограниченной функцией, то есть обобщенным степенным рядом Фробениуса

**Заключение.** В ходе данного изучения метода шифрования был усовершенствован алгоритм шифрования по методу В. Сизова, что привело данный алгоритм к большей защищенности от несанкционированных атак.

#### ЛИТЕРАТУРА

1. Сизов, В.П. Криптографические алгоритмы на основе тригонометрических функций [Электронный ресурс]. – URL: <https://www.ruscyrto.ru/association/archive/rc2005.html>. - Дата доступа: 22.09.2019.
2. Каминский, Л.П. Информационные технологии / Л.П. Каминский, В.А. Степанов // Тригонометрическая криптография. – Красноярск : Сибирский федеральный университет, 2005. – С. 38–41. – Дата доступа: 23.09.2018.
- 1.

УДК 004.588

**РАЗРАБОТКА ТРЕНИРОВОЧНОЙ ПЛАТФОРМЫ ДЛЯ НАЧИНАЮЩИХ СПЕЦИАЛИСТОВ  
В СФЕРЕ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ****А.В. ДЫСИН***(Представлено: канд. физ.-мат. наук, доц. О.В. ГОЛУБЕВА)*

*В данной статье описана концепция разработки тренировочной платформы по тестированию программного обеспечения. Проанализированы наиболее подходящие инструменты для разработки приложения. Описаны ключевые моменты в работе программы.*

В последнее время популярность индустрии информационных технологий набирает все большие обороты. Количество компаний, занимающихся услугами в сфере информационных технологий с каждым годом становится все больше. Также растет и спрос на разработчиков и специалистов по тестированию программного обеспечения.

Многие ИТ компании организуют свои лаборатории, в которых с нуля обучают будущих специалистов, однако, не каждый желающий получить высокооплачиваемую должность достоин обучаться за счет компании в таких лабораториях.

Неопытные джуниор-разработчики заполняют рынок труда в сфере информационных технологий. Многим начинающим тестировщикам и разработчикам не так легко найти свою первую профессию, даже после успешного окончания высших, средне-специальных учебных заведений, а также специализированных курсов. Работодатели, как правило, не готовы нанимать к себе в штат сотрудников без коммерческого опыта. Таким образом, появилась необходимость помочь начинающим и молодым специалистам получить опыт, необходимый в будущей работе.

Одной из самых популярных и самых востребованных профессий в информационной индустрии на сегодняшний день является тестировщик программного обеспечения. Следовательно, предметом разработки для тренировочной платформы является набор сервисов, которые позволят будущим специалистам оттачивать свои профессиональные навыки.

Тестирование – это процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатом работ с целью определить, что они соответствуют описанным требованиям, показать, что они подходят для заявленных целей и для определения дефектов [1].

Тестировщик программного обеспечения – это опытный специалист, принимающий участие в тестировании компонента или системы [1]. Если выразить образно главную цель тестировщика, то она будет звучать так: «понимать, что в настоящий момент необходимо проекту, получает ли проект это необходимое в должной мере и, если нет, как изменить ситуацию к лучшему» [2].

В обязанность специалиста по тестированию входит поиск вероятных ошибок и сбоев в функционировании объекта тестирования (системы, компонента и т.д.). Тестировщик моделирует различные ситуации, которые могут возникнуть в процессе использования предмета тестирования, чтобы разработчики смогли исправить обнаруженные ошибки до выхода продукта в реализацию к реальным пользователям.

В данной статье хотелось бы поговорить о разработке платформы для тестирования. Данная платформа будет содержать закрытые ссылки на ресурсы, в которых будет содержаться как теоретическая информация по теме, так и случаи из практики разработчиков и тестировщиков.

Как и в любой сфере информационных услуг, в тестировании около 70% составляет практика, то есть умение тестировщиков находить ошибки в приложениях. Таким образом, будет разработана игровая платформа, в которой за ограниченное время нужно будет найти максимальное количество дефектов. Это своего рода тренировочная база для будущих специалистов в сфере тестирования.

Для обеспечения структурированного обучения тренировочная платформа будет разделена на три модуля: «Тестирование Web-приложений», «Тестирование мобильных приложений», «Тестирование Desktop-приложений». В зависимости от выбранной специализации, будет возможность тренировать свои навыки по отдельности.

Для обеспечения проверки правильности выполнения заданий, будут разработаны модальные окна, в которых пользователь будет указывать следующие данные:

- компонент, в котором была обнаружена ошибка;
- вид тестирования;
- критичность ошибки.

Помимо практических задач, с которыми сталкивается специалист по тестированию на протяжении своего рабочего дня, большой объем работ занимает написание тестовой документации (чек-листы, тест-кейсы, отчеты о дефекте, отчеты по тестированию, тест-планы и т.д.).

Отчет о дефекте – документ, содержащий отчет в любом недостатке в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию [1]. Данный документ должен содержать правильную, единую терминологию, описывающую элементы пользовательского интерфейса и события данных элементов, приводящих к возникновению ошибки. В общем случае, состоит из следующих полей:

- шапка (краткое описание проблемы, проект, компонент приложения, версия, серьезность, приоритет, статус, автор, назначение);
- окружение;
- описание (шаги воспроизведения, фактический результат, ожидаемый результат);
- дополнения (прикрепленный файл) [3].

В рамках разработки данного сервиса будет возможность описывать, как полагает тестировщику, найденные ошибки и сохранять их в личном кабинете. Прототипом выступает популярная платная система Atlassian Jira, используемая в коммерческих проектах, которая позволяет описывать, отслеживать и создавать отчеты по найденным в системе или компоненте ошибкам [4].

Платформа будет разработана с использованием популярного фронтенд-фреймворка Vue.js, базы данных MySQL и Node.js.

Vue.js — JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Легко интегрируется в проекты с использованием других JavaScript-библиотек. Может функционировать как веб-фреймворк для разработки одностраничных приложений в реактивном стиле [5].

Необходимыми качествами тестировщика являются логическое мышление, внимательность, любознательность, хорошая память, умение учиться и адаптироваться к существующим задачам, быстро переключаться с одного типа задач на другой. Не менее важны терпение, усидчивость, творческий подход и умение работать в команде. Кроме того, тестировщик выступает одновременно и как пользователь, и как эксперт, а потому должен иметь определенный склад мышления: уметь воспроизводить поведение пользователя продукта и анализировать поведение системы, входящие параметры и полученные результаты с точки зрения инженера. Таким образом, после занятий на тренировочной платформе будущие специалисты по тестированию также смогут потренировать необходимые для работы качества.

В заключение хотелось бы отметить, что данная платформа в дальнейшем будет развиваться и помогать не только тестировщикам становиться высококвалифицированными специалистами, но и разработчикам программного обеспечения.

#### ЛИТЕРАТУРА

1. Standard glossary of terms used in Software Testing [Электронный ресурс]. – 2014. – Режим доступа: <https://www.rstqb.org/ru/istqb-downloads.html>. – Дата доступа: 23.09.2019
2. Куликов, С.С. Тестирование программного обеспечения: базовый курс / С. С. Куликов. — Минск: Четыре четверти, 2017. – 312 с.
3. Центр подготовки IT специалистов [Электронный ресурс] / Bug-report. – 2016. – Режим доступа: <https://qalight.com.ua/baza-znaniy/bug-report>. – Дата доступа: 24.09.2019
4. Atlassian Jira Software [Электронный ресурс]. – 2016. – Режим доступа: <https://www.atlassian.com/ru/software/jira> – Дата доступа: 24.09.2019
5. The Progressive JavaScript Framework [Электронный ресурс]. – 2014. – Режим доступа: <https://vuejs.org/> - Дата доступа: 25.09.2019

УДК 004.738.52

**ПРЕИМУЩЕСТВА АВТОМАТИЗАЦИИ SEO  
НА ПРИМЕРЕ ПОПУЛЯРНЫХ ИНСТРУМЕНТОВ АНАЛИЗА****А.В. ДЫСИН***(Представлено: канд. физ.-мат. наук, доц. О.В. ГОЛУБЕВА)*

*В статье рассмотрены основные понятия по SEO. Проанализированы наиболее популярные инструменты для автоматизации продвижения сайтов в сети Интернет. Выделены главные преимущества данного процесса*

Возрастающая конкуренция на рынке интернет-маркетинга приводит к росту требований к digital-агентствам. Рутинные операции автоматизируются, высвобождается время на нестандартные работы, которые способствуют комплексному оказанию услуги по продвижению сайта. Если раньше часть задач была неподъемной для бюджета проекта по трудозатратам, то сегодня это возможно выполнить за несколько минут с однозначной точностью.

SEO продвижение – это комплекс мер по внутренней и внешней оптимизации для поднятия позиций сайта в результатах выдачи поисковых систем по определённым запросам пользователей, с целью увеличения сетевого трафика и потенциальных клиентов и последующей монетизации этого трафика [1].

Каждый специалист, который так или иначе работал с SEO, знает, что продвижение Web-сайта является сложным и трудоёмким процессом. Необходимо тщательно планировать и выполнять стратегии компании, следить за тенденциями к изменениям рынка.

Однако не стоит забывать о том, что углубленные знания по SEO продвижению мешают использовать готовые инструменты и программное обеспечение для упрощения процесса, особенно когда ведется работа с несколькими крупными поставщиками информационных услуг.

Весь процесс SEO слишком сложный, чтобы все это было автоматизировано, однако определенные аспекты продвижения определенно могут быть. Автоматизация SEO продвижения помогает освободить значительное время для других творческих и административных работ.

Ключевым моментом здесь является понимание того, что необходимо автоматизировать, а что в автоматизации не нуждается. Например, такие задачи, как мониторинг сайта и сбор данных, должны быть автоматизированы, чтобы информация собиралась в пути. Затем SEO-специалист должен принимать решения на основе собранных данных.

Задачи, которые необходимо автоматизировать:

- таск-менеджмент;
- аудит внутренней оптимизации сайта;
- сбор семантического ядра и группировка запросов;
- регулярный мониторинг позиций;
- регулярный мониторинг работоспособности сайта и изменений на нем;
- подготовка отчетности [1].

Также, стоит отметить процессы, которые не должны быть автоматизированы, так как компания получит не только некачественный контент, но и получит штраф за создание спам-ссылок. К таким процессам можно отнести создание контента и построение органических ссылок.

Лучше всего найти комбинацию инструментов и программного обеспечения, которые помогают выполнить работу за минимальное время.

Рассмотрим инструмент SEMrush для расширенного поиска ключевых слов и отслеживания. Исследование ключевых слов является основой любой хорошей SEO-программы. В конце концов, цель заключается в том, чтобы нужная аудитория находила компанию через поиск. Хотя бесплатный инструмент, такой как Google Keyword Planner – отличный инструмент для начала, SEMrush – это один из инструментов, который сэкономит SEO-специалисту много времени и сил, когда дело дойдет до исследования ключевых слов [2].

С помощью инструмента исследования ключевых слов SEMrush есть возможность определить лучшие ключевые слова конкурента компании, наблюдать изменения позиций в доменах, раскрывать ключевые слова с длинными хвостами для собственного сайта и компаний PPC [2].

Объединение инструмента Magic Keyword и анализатора ключевых слов позволяет выполнять исследование ключевых слов быстрее, чем с помощью комбинации инструментов. Можно легко сортировать тысячи ключевых слов и открывать ключевые слова с длинным хвостом, не имея дело с несколькими электронными таблицами [3].

Анализатор ключевых слов сортирует выбранные ключевые слова в соответствии с объемом, KD и потенциалом щелчка. С помощью инструмента Magic Key Tool можно выбрать комбинацию фраз, и инструмент предложит комбинации длинного хвоста с точными фразами, которые ищут пользователи [3].

Если компания работает с клиентами, то для создания настраиваемых отчетов пригодится инструмент SEMrush. Все, что необходимо сделать, это выбрать специальные виджеты, относящиеся к текущему отчету, и настроить параметры. Конструктор отчетов автоматически обновляет информацию всякий раз, когда появляется необходимость создать отчет.

Также можно планировать отчеты или использовать их в качестве шаблонов для будущих отчетов.

Cyfe для мониторинга «все в одном». Одна из проблем SEO заключается в том, что всегда есть много информации для отслеживания.

Cyfe – это всеохватная бизнес-панель, которая позволяет отслеживать важные маркетинговые показатели, включая показатели SEO и SEM. Сила Cyfe заключается в способности вытащить данные из нескольких источников в одну приборную панель.

Предварительно созданные виджеты Cyfe позволяют легко отслеживать любые показатели, которые могут отслеживаться. Например, с помощью виджета Moz можно отслеживать полномочия веб-сайта в поиске без необходимости использования платной учетной записи Moz [4].

Добавив виджет для Google Search Console, компания получает обновления на собственном сайте в результатах поиска Google. Также есть возможность отслеживать производительность своего сайта в разных поисковых системах.

Screaming Frog SEO – это паук для обхода и аудита сайта. Суть в SEO – это знать, что исправить на сайте, чтобы повысить производительность в поиске. Это означает, что есть необходимость в отличном процессе аудита сайта, особенно в том, который можно повторить. Для данной цели Screaming Frog SEO spider – наилучший инструмент.

Screaming Frog – это инструмент сканирования в Интернете, который проверяет неработающие ссылки, перенаправляет аудит, анализирует названия страниц, метаданные и обнаруживает дублирующийся контент. Он также может генерировать XML-файлы Sitemap и интегрироваться с Google Analytics, а также есть возможность настроить его для регулярных расписаний [5].

Таким образом, для того, чтобы получить наилучшие результаты от данного процесса, все равно необходимо вложить время и усилия. Однако можно использовать отличные инструменты, которые делают работу проще и быстрее.

SEO-продвижение эффективно влияет на развитие любого бизнеса и им может воспользоваться абсолютно любая компания, вне зависимости от рода деятельности и сферы реализации. Заниматься SEO-продвижением должны профессионалы. Незнание или неопытность может привести к не самым приятным последствиям.

Инструмент исследования ключевых слов SEMrush – отличный вариант для интеллектуального исследования ключевых слов, в то время как Ahrefs помогает легко отслеживать обратные ссылки.

Screaming Frog SEO Spider помогает быть в курсе производительности сайта благодаря углубленному обходу. Для отслеживания всех показателей сайта в одном месте можно положиться на Cyfe.

## ЛИТЕРАТУРА

1. Artox media digital group [Электронный ресурс] / Автоматизация SEO, Инструменты, позволяющие автоматизировать seo-задачи. – 2017. – Режим доступа: <https://seo.artox-media.ru/wiki/avtomatizatsiya-seo.html>. – Дата доступа 22.09.2019.
2. Универсальный набор инструментов для профессионалов интернет-маркетинга [Электронный ресурс]. – 2016. – Режим доступа: <https://www.semrush.com/blog/>. – Дата доступа: 22.09.2019.
3. Инструменты для мониторинга своей ниши, анализа конкурентов и улучшения поискового трафика [Электронный ресурс]. – 2016. – Режим доступа: <https://ahrefs.com/blog/>. – Дата доступа: 23.09.2019
4. Cyfe: All-In-One Business Dashboard [Электронный ресурс]. – 2016. – Режим доступа: <https://www.cyfe.com/blog/>. – Дата доступа: 23.09.2019
5. The SEO Spider is a desktop program you can install locally on PC, Mac or Linux which crawls websites' links, images, CSS, script and apps to evaluate onsite SEO [Электронный ресурс]. – 2016. – Режим доступа: <https://www.screamingfrog.co.uk/seo-spider/blog/>. – 24.09.2019.

УДК 004.432.2

## НОВЫЕ ВОЗМОЖНОСТИ СОВРЕМЕННОГО ВЕБ-ПРОГРАММИРОВАНИЯ

А.В. ЕПАНЕШНИКОВ

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

Статья посвящена современным технологиям веб-программирования и использованию новых технических возможностей в реальных проектах по созданию программного обеспечения. В статье описаны новейшие фреймворки для создания клиентской части веб-приложения, а также рассказано о программной реализации веб-приложения на технологии React. Целью создания программной реализации является наглядная демонстрация преимуществ инновационных технологий веб-программирования в создании и внедрении веб-приложений над устаревшими технологиями.

**Введение.** В настоящее время веб-программирование развивается с космической скоростью. Новые технологии и идеологии разрабатываются не только специализированными компаниями и корпорациями, но и «рядовыми» программистами, имеющими потребность в создании новой технологии для дальнейшего использования, как в своих целях, так и для общества.

Эффективное применение современных технологий и идеологий веб-программирования способствует не только повышению качества программного продукта, но и экономии трудозатрат, в том числе финансовых ресурсов, производительности и безопасности приложения, легкой расширяемости проектов и многому другому.

**Постановка задачи.** Для исследования новых возможностей современного веб-программирования были поставлены следующие задачи:

1. Рассмотреть новейшие технологии программирования на языке Javascript (ECMAScript 6), а также зависимости клиент-серверного приложения.
2. Сравнить технологии.
3. Продемонстрировать возможности лучшей технологии, разработав программный продукт (в данном случае – интерфейс сетевого устройства).
4. Сделать выводы о преимуществах новых технологий.

**Технологии веб-программирования.** Сравнение разных подходов в веб-программировании по выбранным критериям приведено в таблице. Данные для сравнения отобраны из официальных сайтов разработчиков и информационных источников [2, 3, 4].

Критерий для сравнения	AngularJS	React	Серверный рендеринг без использования шаблонов
Поддержка JSX	нет	да	нет
Клиентский рендеринг	да	да	нет
Возможность переноса компиляции шаблонов на сервер	нет	да	нет
Привязана к структуре и библиотекам DOM	да	нет	да
Размер исходного файла	небольшой	большой	небольшой
Размер информации, передаваемой с сервера на клиент при каждом запросе клиента	средний	малый	большой
Имеет внедрение зависимости	да	Не всегда*	нет
Абстрактный клиентский роутинг**	нет	да	нет

\* - чтобы в React подключить внедрение зависимостей, например, NPM (Node Package Manager), необходимо установить и использовать сборщик модулей. [1] Самый популярный и наиболее эффективный, на взгляд автора статьи – Webpack.

\*\* - имеется в виду абстракция клиентского роутинга от html-файлов и появление объектно-ориентированной иерархии и явной зависимости компонентов Javascript в роутинге.

AngularJS — JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC шаблона, а также упрощение тестирования и разработки. Фреймворк работает с HTML, содержащим дополни-

тельные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных. [5]

React - разработанный компаниями Facebook и Instagram JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Большинство пользователей React считают его Представлением в идеологии MVC. React был разработан, чтобы решить большую проблему: написание больших приложений с данными, которые меняются с течением времени. [4]

Серверный рендеринг без использования шаблонов – это подход, при котором роль клиентской части веб-приложения состоит лишь в отображении HTML-файлов, посланных с сервера клиенту, в браузере пользователя. Серверная же роль заключается в генерации HTML-файлов, подходящих под запросы клиента, обработке всех данных и распределении ролей в формировании представления (Модель и Контроллер, а также частично Представление в идеологии MVC).

**Программная реализация приложения на React.** Рассмотрим реализацию интерфейса сетевого устройства с использованием React (рис. 1). В данном примере показана страница сетевых настроек. С помощью соответствующих кнопок можно добавлять и удалять строки с IP-адресами, шлюзами и статическими маршрутами.

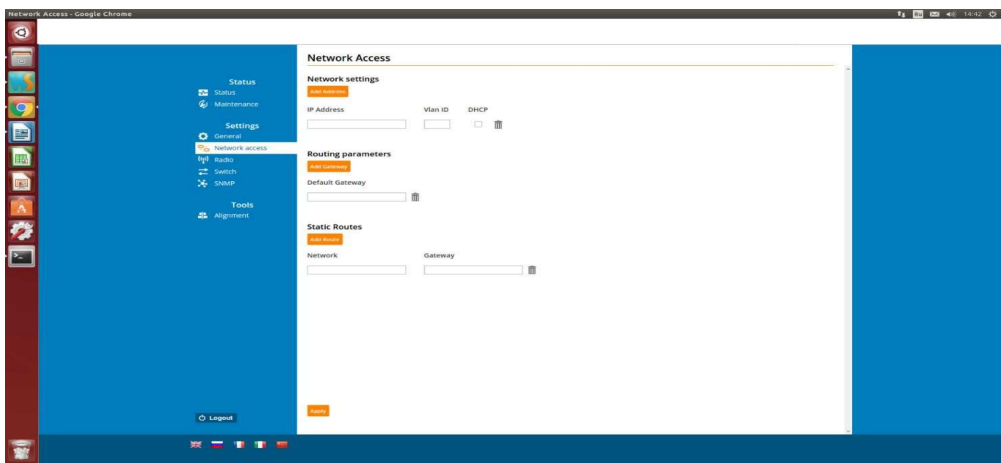


Рисунок 1. – Интерфейс сетевого устройства, реализованный с помощью React

При добавлении или удалении строки перерисовывается только соответствующие элементы, а количество самих строк и данные внутри них хранятся в состоянии родительского React-компонента страницы (NetworkAccess). Для отображения, эти данные передаются далее дочерним компонентам – в данном случае, строкам NetworkSettingsRow (рис. 2).

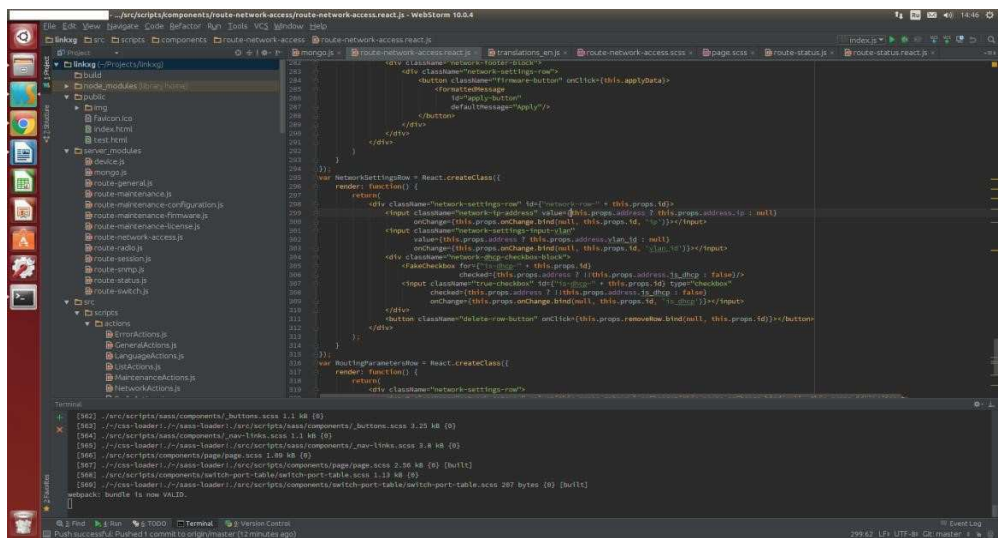


Рисунок 2. – Программный код React-компонента NetworkSettingsRow

Необходимо отметить, что хоть дочерние компоненты рендерятся как DOM-элементы, но внутри кода они полностью контролируются за счёт состояний родительского компонента, свойств, переданных дочерним элементам, а также обработчикам их изменений. Таким образом, не обращаясь к DOM-иерархии напрямую из javascript, можно управлять любыми компонентами в веб-представлении любого веб-приложения, использующего React, не осуществляя бесполезную и неявную ручную выборку элементов из DOM (например, итерирование по массиву, возвращенному методом `document.getElementsByClassName(...)`).

**Заключение.** Рассмотренные современные технологии веб-программирования и программная реализация приложения на React дают понять, что использование современных и быстро развивающихся технологий веб-программирования необходимо при разработке конкурентоспособного программного обеспечения. Устаревшие технологии в настоящее время подходят для очень узкого круга задач, в большинстве случаев они применяются в компаниях с большим объемом предыдущих разработок на старых технологиях, вследствие чего переход на новые технологии означает переписывание практически всего кода «с нуля». Однако, новые технологии позволяют писать код для много более масштабируемой разработки, которая к тому же не замыкается на уникальных компетенциях в непопулярных или устаревших фреймворках, или же просто на коде с большой вложенностью зависимостей и непродуманной оптимизацией. Более того, использование, например, React или AngularJS позволяет улучшить отзывчивость интерфейса, ускорить переход между страницами благодаря клиентскому роутингу, а кастомизация и шаблонизация интерфейса значительно упрощает доработку интерфейса.

#### ЛИТЕРАТУРА

1. Документация React [Электронный ресурс]. – Режим доступа: <https://facebook.github.io/react/docs/getting-started.html>.
2. Versus. Сравнение AngularJS и React [Электронный ресурс]. Режим доступа: <http://versus.com/ru/react-vs-angularjs>.
3. AngularJS [Электронный ресурс]. – Режим доступа: <https://angularjs.org/>
4. Краткое руководство по React JS [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/248799/>
5. AngularJS – Википедия, свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/AngularJS>.



УДК 004.432.2

## АНАЛИЗ СОВРЕМЕННЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

**А.В. ЕПАНЕШНИКОВ***(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)*

*Работа посвящена анализу современных языков программирования. Выделяются и описываются главные и характерные их особенности, а также отличия от популярных в настоящее время языков программирования.*

**Введение.** Технология программирования не стоит на месте и постоянно развивается с учетом новых потребностей в сфере IT-технологий. Существуют два главных требования для языков программирования: их близость к обычному профессиональному языку определенной сферы деятельности и простота реализации. Противоречивость требований стимулирует непрерывный процесс создания новых языков и доработки существующих.

**Основной раздел.** Язык программирования – система обозначений для описания алгоритмов и структур данных, определенная искусственная формальная система, средствами которой можно выражать алгоритмы [1].

В настоящее время ежегодно число языков программирования пополняется новыми. На сегодняшний день известно более двух с половиной тысяч языков программирования. Ниже рассматриваются несколько самых передовых языков программирования, каждый из которых решает свою определенную задачу.

Язык Dart, разработанный компанией Google, как один из языков веб-программирования.

Как и в JavaScript, в Dart используются синтаксисы и ключевые слова, похожие на те, которые используются в языке C. Однако одним существенным различием является то, что в то время как JavaScript основывается на прототипах, объекты в Dart определяются с помощью классов и интерфейсов, как в C++ или Java [2].

Идея его создания основана на том, чтобы сделать язык таким же привычным, гибким и динамичным как JavaScript. Но пока, что этот язык мало где возможно использовать потому, что разработан для запуска на клиенте, либо на сервере.

Следующим новым языком программирования, разработанный компанией Google является Go. Этот язык общего назначения, создан для легкого программирования, подходящий для всего – от разработки приложений до программирования систем.

Язык больше похож на C или C++, чем на Java или C#. Однако, как и последние языки, Go включает в себя такие современные функции, как очистка памяти от ненужных данных, отражение текущих процессов выполнения и поддержку параллельности [2].

В языке Go понятия классов, конструкторов и деструкторов совершенно отсутствуют. Однако в нем присутствуют структуры, заимствованные из языка C, к которым возможна привязка функций, в связи с этим на языке Go можно создавать программы базирующихся на объектно-ориентированном программировании. Наличие функции очищение памяти от ненужных данных упрощает работу, по сравнению с C и C++. Язык Go еще в разработке, но тем не менее с ним можно уже работать.

Веб-программирование сложно тем, что для клиента, сервера, баз данных и т. д. свои отдельные языки программирования. Инженеры разработали новый язык Ора. Этот язык созданный с идеей того, чтобы связать в себя интерфейс пользователя, логику сервера и базу данных ввода-вывода. Компилятор самостоятельно решает, где запущена программа и выполняет написанный код.

Среда выполнения Ора объединяет собственный веб-сервер и систему управления базой данных, которые не могут быть заменены самостоятельными альтернативами. Ора поставляется бесплатно и на данный момент доступен для 64-х битных Linux и Mac OS X платформ, другие же порты пока разрабатываются [2].

С другой стороны, в каждом языке существуют свои недостатки, которые ставят созданную на этом языке систему менее безопасной.

Для решения этой проблемы ученые нашли способ защиты такой, что позволяет защитить и веб-сайты, и компьютеры пользователей, разработав систему, позволяющую использовать несколько языков программирования при создании одной программы.

Система Wуvern является чем-то вроде компилятора одного мета-языка, в состав которого входят все известные и популярные языки программирования. Эта система позволяет программистам разрабатывать веб-страницы и приложения, используя напрямую все возможности C, PHP, HTML, CSS,

JavaScript и других языков, и избегая необходимости использования некоторых искусственных приемов, которые являются источниками потенциальной опасности [3].

Данная система не единственная, примерами подобных программ являются – Scheme, ProteaJ, Spoofox и OJ. Они также решают вышеперечисленные задачи.

В продолжение темы, был разработан новый язык программирования под названием Sketch. Особенность языка в том, что он позволяет при написании программ опускать некоторые участки кода, заполняя эти промежутки самостоятельно при этом оптимизируя получившиеся конструкции.

Основной идеей, реализованной в виде компилятора языка Sketch, является метод поиска оптимального решения определенной задачи. Известно, что любая логическая и математическая задача может быть решена несколькими путями, количество которых, в некоторых случаях, может стремиться к бесконечности [4].

Еще одним новым языком программирования стал – язык Wolfram Alpha.

Wolfram Language является языком, основанным на обширной базе знаний при помощи которой можно создавать достаточно сложные приложения, не прибегая к процессу традиционного программирования. Wolfram Language является языком, опирающимся на огромную базу знаний, в которой заключены все типовые вычислительные операции [5].

**Заключение.** Рассмотрев современные языки программирования, некоторые из которых внедряются в программирования, некоторые еще дорабатываются можно сделать выводы о том, что в настоящее время, имея базу из двух с половиной тысяч языков программирования нет ни одного универсального. Пока такая задача не стоит, поэтому на этом этапе можно выделить систему Wyvern, система, позволяющую использовать несколько разных языков программирования при создании одной программы. Можно предположить, что хотя бы один из рассмотренных языков заменит в будущем популярных в настоящее время языков программирования.

#### ЛИТЕРАТУРА

1. Языки программирования [Электронный ресурс]. – URL: <http://life-prog.ru>. – Дата обращения: 08.12.2014.
2. McALLISTER N. InfoWorld. «10 языков программирования, которые могут перевернуть мир IT» // Компьютерные вести [Электронный ресурс]. – URL: <http://www.kv.by/content/325498-10yazykovprogramirovaniya-kotorye-mogut-perevernut-mir-it>. – Дата обращения: 08.12.2014.
3. Wyvern – новая система, позволяющая использовать несколько разных языков программирования при создании одной программы // DailyTechInfo [Электронный ресурс]. – URL: <http://www.dailytechinfo.org/infotech/6167-wyvern-novaya-sistema-pozvolyayuschaya-ispolzovat-neskolko-raznyh-yazyko-programirovaniya-pri-sozdanii-odnoy-programmy.html>. – Дата обращения: 08.12.2014.
4. Sketch – новый язык программирования, способный самостоятельно оптимизировать и завершать незаконченные участки кода // DailyTechInfo [Электронный ресурс]. – URL: <http://www.dailytechinfo.org/infotech/5691-sketch-novyy-yazyk-programirovaniya-sposobnyu-samostoyatelnooptimizirovat-i-zavershat-nezakonchennye-uchastki-koda.html>. – Дата обращения: 09.12.2014.
5. Язык Wolfram Alpha – революция в области программирования // DailyTechInfo [Электронный ресурс]. URL: <http://www.dailytechinfo.org/infotech/5709-yazyk-wolfram-alpha-revolyuciya-voblasti-programirovaniya.html> – Дата обращения: 09.12.2014.

УДК 004.021

## РАССМОТРЕНИЕ ПРОГРЕССИВНЫХ СЕГМЕНТАЦИОННЫХ АЛГОРИТМОВ

И.И. ИВАНОВ

(Представлено: канд. физ.-мат. наук, доц. О.В. ГОЛУБЕВА)

В статье рассматриваются современные алгоритмы сегментации изображений для выделения областей или объектов для последующей обработки и анализа. Два описанных алгоритма выделены с точки зрения эффективности применения при выделении однородных областей изображений.

**Введение.** Одной из базовых задач классификации объектов является сегментация изображения для выделения необходимых данных. В данном направлении интенсивно применяются методы оптимизации и аналитической геометрии. Алгоритмы выделения однородных областей изображения значительно упрощают классификацию объектов в области машинного обучения и повышают результативность распознавания.

**Основной раздел.** Одним из основных способов сегментации на сегодняшний день является применение алгоритма сегментации по водоразделам (WaterShed). Алгоритм работает с изображением как с функцией от двух переменных – координат пикселя.

Значением функции может быть интенсивность или модуль градиента. Для наибольшего контраста можно взять градиент от изображения. Если по оси OZ откладывать абсолютное значение градиента, то в местах перепада интенсивности образуются хребты, а в однородных регионах – равнины. После нахождения минимумов функции, идет процесс заполнения “водой”, который начинается с глобального минимума. Как только уровень воды достигает значения очередного локального минимума, начинается его заполнение водой. Когда два региона начинают сливаться, строится перегородка, чтобы предотвратить объединение областей [1]. Вода продолжит подниматься до тех пор, пока регионы не будут отделяться только искусственно построенными перегородками, что проиллюстрировано на рисунке 1.

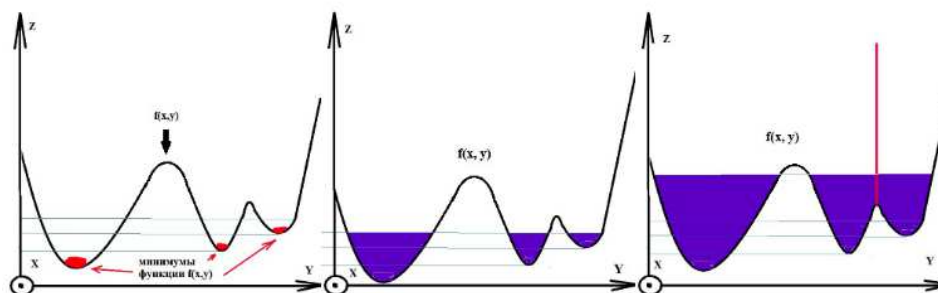


Рисунок 1. – Иллюстрация процесса заполнения водой

Такой алгоритм может быть полезным, если на изображении небольшое число локальных минимумов, в случае же их большого количества возникает избыточное разбиение на сегменты. Например, если непосредственно применить алгоритм, получим много мелких деталей, пример чего представлен на рисунке 2.



Рисунок 2. – Изображение после сегментации алгоритмом WaterShed

Чтобы избавиться от избытка мелких деталей, можно задать области, которые будут привязаны к ближайшим минимумам. Перегородка будет строиться только в том случае, если происходит объединение двух регионов с маркерами, в противном случае будет происходить слияние этих сегментов. Такой подход убирает эффект избыточной сегментации, но требует предварительной обработки изображения для выделения маркеров, которые можно обозначить интерактивно, что представлено на рисунке 3.



Рисунок 3. – Изображение после сегментации алгоритмом WaterShed с использованием маркеров

В результате работы алгоритма мы получаем маску с сегментированным изображением, где пиксели одного сегмента помечены одинаковой меткой и образуют связную область. Основным недостатком данного алгоритма является использование процедуры предварительной обработки для картинок с большим количеством локальных минимумов (изображения со сложной текстурой и с обилием различных цветов).

Алгоритм сегментации по водоразделам показывает хорошие результаты в обработке изображений с простой текстурой. Для группировки объектов с близкими признаками используется алгоритм MeanShift.

В основе идеи алгоритма лежит объединение пикселей со схожими признаками в один сегмент, получая на выходе изображение с однородными областями.

Например, в качестве координат в пространстве признаков можно выбрать координаты пикселя  $(x, y)$  и компоненты RGB пикселя. Изобразив пиксели в пространстве признаков, можно заметить сгущения в определенных местах.

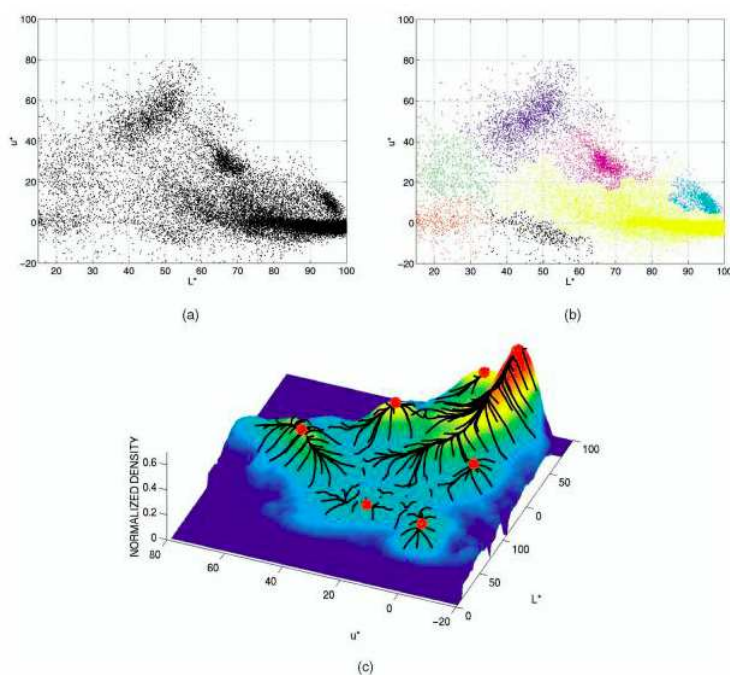


Рисунок 4. – (a) Пиксели в двухмерном пространстве признаков.  
(b) Пиксели, пришедшие в один локальный максимум, окрашены в один цвет.  
(c) Функция плотности, максимумы соответствуют местам наибольшей концентрации пикселей [2]

Чтобы легче было описывать сгущения точек, вводится функция плотности:

$$f(\vec{x}) = \frac{1}{Nh^d} \sum_{i=1}^N K\left(\frac{\vec{x}-\vec{x}_i}{h}\right)$$

$\vec{x}_i$  – вектор признаков  $i$ -го пикселя,  $d$  — количество признаков,  $N$  — число пикселей,  $h$  — параметр, отвечающий за гладкость,  $K\left(\frac{\vec{x}}{h}\right)$  — ядро.

Максимумы функции расположены в точках сгущения пикселей изображения в пространстве признаков. Пиксели, принадлежащие одному локальному максимуму, объединяются в один сегмент. Получается, чтобы найти к какому из центров сгущения относится пиксель, надо шагать по градиенту функции для нахождения ближайшего локального максимума.

При выборе в качестве признаков координат пикселей и интенсивностей по цветам в один сегмент будут объединяться пиксели с близкими цветами и расположенные недалеко друг от друга. Соответственно, если выбрать другой вектор признаков, то объединение пикселей в сегменты уже будет идти по нему. Например, если убрать из признаков координаты, то небо и озеро будут считаться одним сегментом, так как пиксели этих объектов в пространстве признаков попали бы в один локальный максимум.

Если объект, который хотим выделить, состоит из областей, сильно различающихся по цвету, то MeanShift не сможет объединить эти регионы в один, и наш объект будет состоять из нескольких сегментов. Но зато хорошо справиться с однородным по цвету предметом на пестром фоне.

**Заключение.** Рассмотренные способы сегментации изображений могут применяться для решения очень схожих задач. Использование реализации алгоритма MeanShift обоснованно для кластеризации пикселей, близких по цвету и координатам. MeanShift также используется при реализации алгоритма слежения за движущимися объектами [3]. WaterShed же показывает свою эффективность для изображений с простой текстурой. Таким образом, алгоритм сегментации следует выбирать, конечно, исходя из точной специфики конкретной задачи, однако оба из способов эффективны в своих направлениях, имея математическое обоснование. Применение описанных алгоритмов в паре может повысить энергоэффективность системы для обнаружения широкого спектра типов объектов, включая объекты в движении.

#### ЛИТЕРАТУРА

1. Р. Гонсалес, Р. Вудс Цифровая обработка изображений, Москва: Техносфера, 2005. – 1072 с.
2. D. Comaniciu, P. Meer Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. – 603 с.
3. D. Comaniciu, V. Ramesh, P. Meer Real-Time Tracking of Non-Rigid Objects Using Mean Shift, Conference on CVPR, 2000. – 4 с.

УДК 004.021

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

И.И. ИВАНОВ

(Представлено: канд. физ.-мат. наук, доц. О.В. ГОЛУБЕВА)

В статье рассматриваются наиболее важные современные алгоритмы машинного обучения. Анализ сфокусирован на помощи в выборе правильного подхода исходя из специфики решаемых задач в рамках предиктивной аналитики.

**Введение.** Алгоритмы машинного обучения можно описать как обучение целевой функции  $F$ , которая наилучшим образом соотносит входные переменные  $X$  и выходную переменную  $Y$ .

Неизвестно, что из себя представляет функция  $F$ . Ведь при её непосредственном наличии, она бы использовалась напрямую, а не предполагалась и уточнялась с помощью различных алгоритмов обучения.

Наиболее распространённой задачей в машинном обучении является предсказание значений  $Y$  для новых значений  $X$ . Это называется прогностическим моделированием, и главная цель — сделать как можно более точное предсказание.

**Основная часть. Случайный лес** — очень популярный и эффективный алгоритм машинного обучения. Это разновидность ансамблевого алгоритма, называемого бэггингом.

**Бутстрэп** является эффективным статистическим методом для оценки какой-либо величины вроде среднего значения. Вы берёте множество подвыборок из ваших данных, считаете среднее значение для каждой, а затем усредняете результаты для получения лучшей оценки действительного среднего значения.

В бэггинге используется тот же подход, но для оценки всех статистических моделей чаще всего используются деревья решений. Тренировочные данные разбиваются на множество выборок, для каждой из которой создаётся модель. Когда нужно сделать предсказание, то его делает каждая модель, а затем предсказания усредняются, чтобы дать лучшую оценку выходному значению.

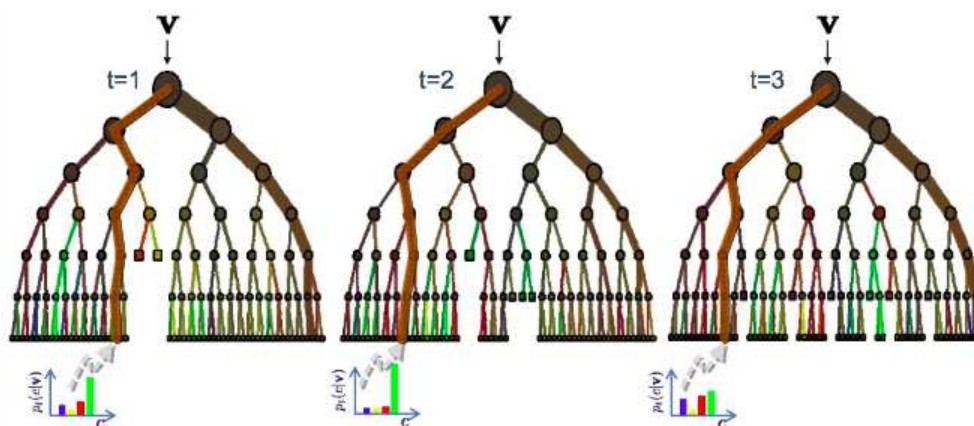


Рисунок 1. – Графическое представление алгоритма случайного леса

В алгоритме случайного леса для всех выборок из тренировочных данных строятся деревья решений. При построении деревьев для создания каждого узла выбираются случайные признаки. В отдельности полученные модели не очень точны, но при их объединении качество предсказания значительно улучшается. [1]

Если алгоритм с высокой дисперсией, например, деревья решений, показывает хороший результат на ваших данных, то этот результат зачастую можно улучшить, применив бэггинг.

**Бустинг** — это семейство ансамблевых алгоритмов, суть которых заключается в создании сильного классификатора на основе нескольких слабых. Для этого сначала создаётся одна модель, затем другая модель, которая пытается исправить ошибки в первой. Модели добавляются до тех пор, пока тренировочные данные не будут идеально предсказываться или пока не будет превышено максимальное количество моделей.

**AdaBoost** был первым действительно успешным алгоритмом бустинга, разработанным для бинарной классификации. Именно с него лучше всего начинать знакомство с бустингом. Современные методы вроде стохастического градиентного бустинга основываются на AdaBoost.

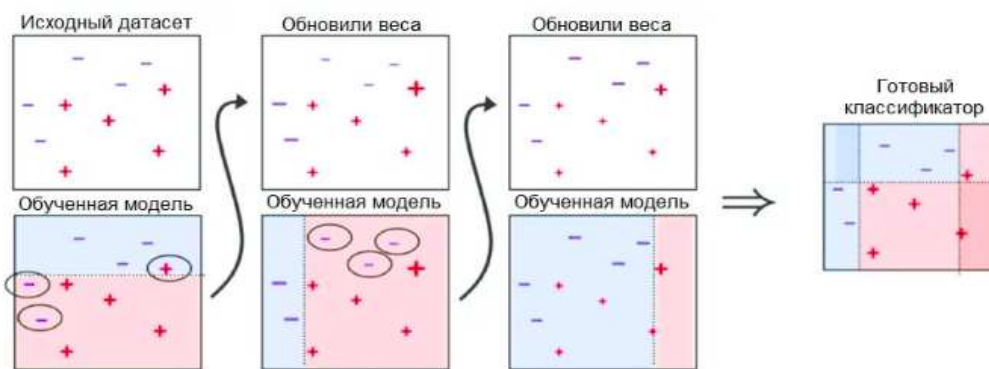


Рисунок 2. – Графическое представление алгоритма AdaBoost

AdaBoost используют вместе с короткими деревьями решений. После создания первого дерева проверяется его эффективность на каждом тренировочном объекте, чтобы понять, сколько внимания должно уделить следующее дерево всем объектам. Тем данным, которые сложно предсказать, даётся больший вес, а тем, которые легко предсказать, — меньший. Модели создаются последовательно одна за другой, и каждая из них обновляет веса для следующего дерева. После построения всех деревьев делаются предсказания для новых данных, и эффективность каждого дерева определяется тем, насколько точным оно было на тренировочных данных.

Так как в этом алгоритме большое внимание уделяется исправлению ошибок моделей, важно, чтобы в данных отсутствовали аномалии.

**Метод опорных векторов**, вероятно, один из наиболее популярных и обсуждаемых алгоритмов машинного обучения.

Гиперплоскость — это линия, разделяющая пространство входных переменных. В методе опорных векторов гиперплоскость выбирается так, чтобы наилучшим образом разделять точки в плоскости входных переменных по их классу: 0 или 1. В двумерной плоскости это можно представить как линию, которая полностью разделяет точки всех классов. Во время обучения алгоритм ищет коэффициенты, которые помогают лучше разделять классы гиперплоскостью.

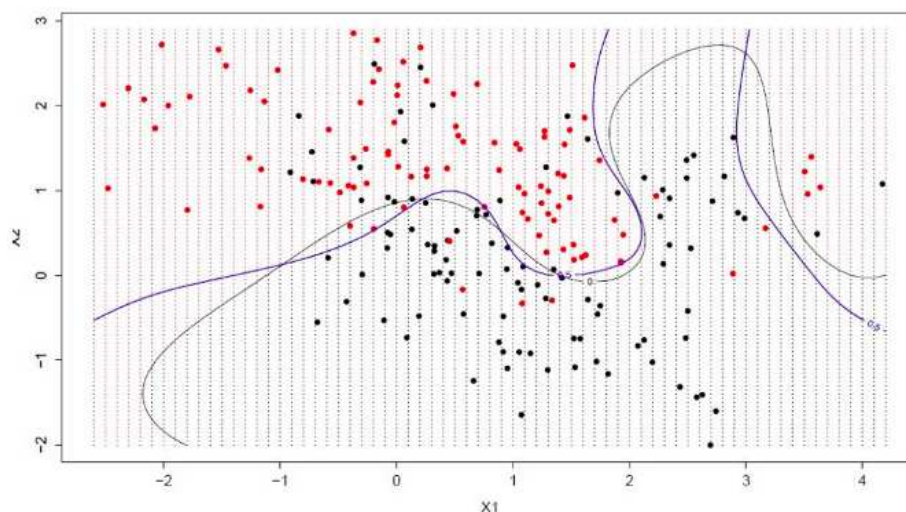


Рисунок 3. – Графическое представление метода опорных векторов в частном двумерном случае

Расстояние между гиперплоскостью и ближайшими точками данных называется разницей. Лучшая или оптимальная гиперплоскость, разделяющая два класса, — это линия с наибольшей разницей. Только эти точки имеют значение при определении гиперплоскости и при построении классификатора. Эти точки называются опорными векторами. Для определения значений коэффициентов, максимизирующих разницу, используются специальные алгоритмы оптимизации. [2]

Метод опорных векторов, наверное, один из самых эффективных классических классификаторов, на который определённо стоит обратить внимание.

**Заключение.** Рассмотренные алгоритмы машинного обучения являются одними из самых эффективных на сегодняшний день. Примитивные алгоритмы, исключенные из рассмотрения, могут являться лишь пробным решением с малой точностью.

Метод опорных векторов является выбором многих успешных моделей с высокой точностью классификации в многомерном пространстве. Алгоритм может играть роль твердой базы для большинства задач и является практически универсальным.

Методы бустинга и бутстрэпа же рекомендуются к детальному рассмотрению при специфичности проблемы для решения.

Выбор конкретного алгоритма должен опираться на анализ следующих характеристик решаемой задачи:

- Размер, качество и характер данных;
- Доступное вычислительное время;
- Срочность задачи;
- Конечная специфика применения данных.

#### ЛИТЕРАТУРА

1. Multiclass Decision Forest [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-decision-forest>. – Дата обращения: 03.09.2019.
2. One-Class Support Vector Machine [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/one-class-support-vector-machine>. – Дата обращения: 02.09.2019.



УДК 003.26

## СОВРЕМЕННАЯ КОМПЬЮТЕРНАЯ СТЕГАНОГРАФИЯ В ПРОГРАММИРОВАНИИ

И.В. ИСАКОВ

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

*В данной работе рассмотрим и проанализируем основные направления современной компьютерной стеганографии, её обобщенную модель и схему построения.*

Стеганография – это наука о скрытой передаче информации, при этом сам факт передачи остается в тайне. Криптография же, осуществляет скрытие информации путём её шифрования. Так же, в отличие от криптографии, стеганография скрывает наличие самого передаваемого сообщения. Данное скрытие осуществляется различными способами, для которых общей чертой является то, что скрываемое сообщение встраивается в некоторый простой, не привлекающий внимание объект. После чего, объект открыто передается адресату. И как мы уже знаем, наличие скрытой связи остается незаметным. Так же, дабы дополнить стеганографию, совместно с ней обычно используют методы криптографии.

В компьютерной стеганографии передаваемые данные встраивают в цифровые, которые как правило имеют аналоговую природу, то беж – речь, видео, изображения, аудиофайлы. В компьютерной стеганографии имеются два основных направления – связанное и не связанное с цифровой обработкой сигнала/ [1]

**Направления стеганографии.** Современная стеганография — цифровая и компьютерная. Последнюю можно разделить на три больших направления. Первое — это собственно тайнопись, или методы сокрытия одних файлов (которые принято называть сообщением) внутри других («контейнера»). После заполнения сообщением контейнер внешне меняется незаметно и полностью сохраняет свою функциональность. Второе направление изучает методы добавления к сообщению скрытых или стеганографических меток (stegomarks). Это незаметные без специальной обработки метки, идентичные для всех файлов одного человека или устройства. Например, такие стегометки записываются в цифровые фотографии для того, чтобы можно было доказать их авторство. Крэкеры иногда оставляют стегометки в лицензионных ключах. Они зашиты на уровне алгоритма генерации, а потому сохраняются при попытке изменить интерфейс кейгена и выдать его за свой. Третье направление — внедрение в сообщение цифровых отпечатков (digital fingerprints). В отличие от стегометок, эти скрытые знаки уникальны для каждого сообщения. Они служат в основном для защиты интересов правообладателей, позволяя отследить распространение контента. К примеру, многие интернет-магазины внедряют цифровые отпечатки в продаваемые книги и музыкальные композиции. В них кодируется информация о дате продажи и аккаунте купившего (имя, IP-адрес и прочее). Если купленные файлы позже появятся среди торрентов или на файлообменниках, то правообладатели смогут установить распространителя нелегального контента. Для этого будет достаточно считать из контрафактного файла вкрапленный цифровой отпечаток. Использует ли наш любимый онлайн-сервис стегометки? Это легко проверить. Достаточно купить два экземпляра одного и того же произведения с разных аккаунтов и сделать побайтное сравнение файлов. Разница между ними и покажет скрытые метки. Если же файлы скачались идентичные (и их хеши полностью совпадают), то стегометок внутри нет. [2]

**Строение стегосистемы.** Стегосистема является совокупностью средств и методов, которые используются для формирования скрытого канала передачи данных. Стегосистема выполняет задачу встраивания и выделения сообщений из другой информации. Обобщенная модель стегосистемы представлена на рис.1 Она состоит из множества основных элементов [3]:

**Пустой контейнер** – контейнер? не содержащий встроенного сообщения; Контейнер – любая информация, предназначенная для сокрытия сообщений;

**Стего-контейнер** – содержит встроенную информацию;

**Встроенное сообщение** – собственно само сообщение, которое встраивается в контейнер;

**Стегоканал** – канал передачи стего-сообщения;

**Стегоключ** – в зависимости от уровня защиты может быть несколько ключей. Данный ключ предназначен для сокрытия информации;

**Прекодер** – предназначен для того чтобы преобразовать сообщение? которое необходимо скрыть, к удобному для встраивания в сигнал-контейнер виду;

**Стегокодер** – устройство, с помощью которого осуществляется вложение сообщения в другие данные, учитывая их модели;

**Стегодетектор** – определяет наличие стегосообщения;



Рисунок 1. – Обобщенная модель стегосистемы

При построении стегосистемы, должны соблюдаться следующие положения, требования [3]:

- Стегосообщение обязано быть устойчивым к каким-либо искажениям, особенно злонамеренным. При передаче изображение либо другой контейнер может подвергаться всяко-разным трансформациям: увеличиваться или уменьшаться, изменяться может сам формат и т.д. Так же может произойти сжатие, при чём, могут использоваться алгоритмы сжатия, при которых имеет место потеря данных.
- Свойства контейнера должны быть модифицированы таким образом, дабы невозможно было выявить какие-либо изменения при визуальном контроле. Данное требование определяет уровень качества сокрытия встраиваемого сообщения. Стоит отметить, что для обеспечения беспрепятственного прохождения стегосообщения по каналу связи, оно никак не должно привлечь внимание атакующего.
- Для сохранения целостности встраиваемого сообщения необходимо использование кода с исправлением ошибки.
- Для повышения надежности встраиваемое сообщение должно быть продублировано.
- Потенциальный противник должен быть лишен каких-либо технических и иных преимуществ в распознавании или раскрытии содержания тайных сообщений.
- Противник имеет полное представление о стегосистеме и деталях ее реализации. Единственной информацией, которая остается неизвестной потенциальному противнику, является ключ, с помощью которого только его держатель может установить факт присутствия и содержание скрытого сообщения.
- Противник, каким-то образом узнавший о факте существования скрытого сообщения, не должен иметь возможности извлечь подобные сообщения в других данных до тех пор, пока ключ хранится в тайне.

Схема стеганосистемы как системы связи представлена на рисунке 2.

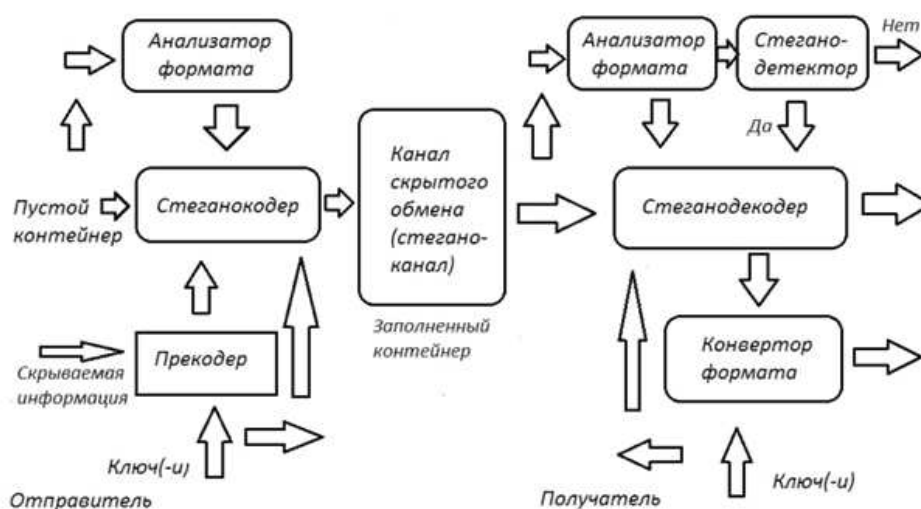


Рисунок 2. – Схема стегосистемы как системы связи

**Заключение.** Компьютерная стеганография играет важную роль в обеспечении информационной безопасности современного мира, когда существуют такие нерешенные проблемы как защита авторского права, защита прав на личную тайну, организация электронной торговли, компьютерная пре-

ступность и кибертерроризм. Стеганография может обеспечивать помехоустойчивую аутентификацию мультимедийной информации, контроль целостности данных, защиту прав собственника мультимедийной информации, отслеживание распространения информации. Основная проблема современной цифровой стеганографии – отсутствие стандартов. При наличии достаточного числа известных алгоритмов, отсутствует организованная практика их применения. Надеюсь, что в ближайшие годы в результате инициатив молодых специалистов определится в каком-то виде стандарт стеганографических средств защиты информации.

#### ЛИТЕРАТУРА

1. Садов, В.С. Компьютерная стеганография (Конспект лекций)/В.С. Садов – Минск: БГУ, 2010. – 211 с.
2. Шнайер, Б. Секреты и ложь. Безопасность данных в цифровом мире / Б. Шнайер – СПб.: Питер, 2003. – 368 с.
3. Грибунин, В.Г. Цифровая стеганография/ В.Г. Грибунин – Москва: «Академия», 2009. – 265 с.

УДК 003.042

АТАКИ НА СТЕГАНОГРАФИЧЕСКУЮ СИСТЕМУ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ  
И ПРОТИВОДЕЙСТВИЕ ИМ

И.В. ИСАКОВ

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

В данной работе исследуем перечень возможных атак на стегосистему цифровых водяных знаков, а способы противодействия этим атакам.

**Классификация атак на ЦВЗ.** ЦВЗ должны удовлетворять противоречивым требованиям визуальной (аудио) незаметности и робастности к основным операциям обработки сигналов. В дальнейшем без потери общности будем предполагать, что в качестве контейнера используется изображение.

Обратимся к системе встраивания сообщений путем модификации младшего значащего бита (LSB) пикселей. Практически любой способ обработки изображений может привести к разрушению значительной части встроенного сообщения. Например, рассмотрим операцию вычисления скользящего среднего по двум соседним пикселям  $\frac{(a+b)}{2}$ , являющуюся простейшим примером низкочастотной фильтрации. Пусть значения пикселей  $a$  и  $b$  могут быть четными или нечетными с вероятностью  $p = \frac{1}{2}$ . Тогда и значение младшего значащего бита изменится после усреднения в половине случаев. К тому же эффекту может привести и изменение шкалы квантования, скажем, с 8 до 7 бит. Аналогичное влияние оказывает и сжатие изображений с потерями. Более того, применение методов очистки сигналов от шумов, использующих оценивание и вычитание шума, приведет к искажению подавляющего большинства бит скрытого сообщения.[1] Существуют также и гораздо более губительные для ЦВЗ операции обработки изображений, например, масштабирование, повороты, усечение, перестановка пикселей. Ситуация усугубляется еще и тем, что преобразования стегосообщения могут осуществляться не только нарушителем, но и законным пользователем, или являться следствием ошибок при передаче по каналу связи.

Сдвиг на несколько пикселей может привести к не обнаружению ЦВЗ в детекторе. Аналоговые видеоманитофоны, как правило, несколько сдвигают изображение из-за неравномерности вращения двигателя лентопротяжного механизма или изнашивания ленты. Сдвиг может быть незаметен для глаза, но привести к разрушению ЦВЗ.

Теперь рассмотрим атаки, специфичные для систем ЦВЗ. Можно выделить следующие категории атак против таких стегосистем[3].

1. Атаки против встроенного сообщения — направлены на удаление или порчу ЦВЗ путем манипулирования стего. Входящие в эту категорию методы атак не пытаются оценить и выделить водяной знак. Примерами таких атак могут являться линейная фильтрация, сжатие изображений, добавление шума, выравнивание гистограммы, изменение контрастности и т. д.

2. Атаки против стегодетектора — направлены на то, чтобы затруднить или сделать невозможной правильную работу детектора. При этом водяной знак в изображении остается, но теряется возможность его приема. В эту категорию входят такие атаки, как аффинные преобразования (то есть масштабирование, сдвиги, повороты), усечение изображения, перестановка пикселей и т. д.

2. Атаки против протокола использования ЦВЗ — в основном связаны с созданием ложных ЦВЗ, ложных стего, инверсией ЦВЗ, добавлением нескольких ЦВЗ.

4. Атаки против самого ЦВЗ — направлены на оценивание и извлечение ЦВЗ из стегосообщения, по возможности без искажения контейнера. В эту группу входят такие атаки, как атаки сговора, статистического усреднения, методы очистки сигналов от шумов, некоторые виды нелинейной фильтрации [4] и другие.

Надо заметить, что рассматриваемая классификация атак не является единственно возможной и полной. Кроме того, некоторые атаки (например, удаление шума) могут быть отнесены к нескольким категориям.

В соответствии с этой классификацией все атаки на системы встраивания ЦВЗ могут быть разделены на четыре группы:

- 1) атаки, направленные на удаление ЦВЗ;
- 2) геометрические атаки, направленные на искажение контейнера;
- 3) криптографические атаки;
- 4) атаки против используемого протокола встраивания и проверки ЦВЗ.

### Методы противодействия атакам на системы ЦВЗ.

В простейших стегосистемах ЦВЗ при встраивании используется псевдослучайная последовательность, являющаяся реализацией белого гауссовского шума и не учитывающая свойства контейнера. Такие системы практически неустойчивы к большинству рассмотренных выше атак. Для повышения робастности стегосистем можно предложить ряд улучшений[2].

В робастной стегосистеме необходим правильный выбор параметров псевдослучайной последовательности. Известно, что при этом системы с расширением спектра могут быть весьма робастными по отношению к атакам типа добавления шума, сжатия и т. п. Так считается, что ЦВЗ должен обнаруживаться при достаточно сильной низкочастотной фильтрации (7х7 фильтр с прямоугольной характеристикой). Следовательно, база сигнала должна быть велика, что снижает пропускную способность стегоканала. Кроме того, используемая в качестве ключа ПСП должна быть криптографически безопасной.

Причиной нестойкости систем ЦВЗ с расширением спектра к подобным атакам объясняется тем, что используемая для вложения последовательность обычно имеет нулевое среднее. После усреднения по достаточно большому количеству реализаций ЦВЗ удаляется. Известен специальный метод построения водяного знака, направленный против подобной атаки. При этом коды разрабатываются таким образом, чтобы при любом усреднении всегда оставалась не равная нулю часть последовательности (статическая компонента). Более того, по ней возможно восстановление остальной части последовательности (динамическая компонента). Недостатком предложенных кодов является то, что их длина увеличивается экспоненциально с ростом числа распространяемых защищенных копий. Возможным выходом из этого положения является применение иерархического кодирования, то есть назначения кодов для группы пользователей. Некоторые аналогии здесь имеются с системами сотовой связи с кодовым разделением пользователей (CDMA).

Различные методы противодействия предлагались для решения проблемы прав собственности. Первый способ заключается в построении необратимого алгоритма ЦВЗ. ЦВЗ должен быть адаптивным к сигналу и встраиваться при помощи однонаправленной функции, например, хэш-функции. Хэш-функция преобразует 1000 бит исходного изображения  $V$  в битовую последовательность  $b_i, i = 1 \dots 1000$ . Далее, в зависимости от значения  $b_i$  используется две функции встраивания ЦВЗ. Если  $b_i = 0$ , то используется функция  $v_i(1 + aw_i)$ , если  $b_i = 1$ , то функция  $v_i(1 - aw_i)$ , где  $v_i$  -  $i$ -й коэффициент изображения,  $w_i$  -  $i$ -й бит встраиваемого сообщения. Предполагается, что такой алгоритм формирования ЦВЗ предотвратит фальсификацию.

Второй способ решения проблемы прав собственности заключается во встраивании в ЦВЗ некоторой временной отметки, предоставляемой третьей, доверенной стороной. В случае возникновения конфликта лиц, имеющее на изображении более раннюю временную отметку, считается настоящим собственником.

Один из принципов построения робастного ЦВЗ заключается в адаптации его спектра. В ряде работ показано, что огибающая спектра идеального ЦВЗ должна повторять огибающую спектра контейнера. Спектральная плотность мощности ЦВЗ, конечно же, намного меньше. При такой огибающей спектра Винеровский фильтр дает наилучшую оценку ЦВЗ из возможных: дисперсия значений ошибки достигает дисперсии значений заполненного контейнера. На практике адаптация спектра ЦВЗ возможна путем локального оценивания спектра контейнера. С другой стороны, методы встраивания ЦВЗ в области преобразования достигают этой цели за счет адаптации в области трансформанты.[4]

Для защиты от атак типа аффинного преобразования можно использовать дополнительный (опорный) ЦВЗ. Этот ЦВЗ не несет в себе информации, но используется для «регистрации» выполняемых нарушителем преобразований. В детекторе ЦВЗ имеется схема предсказания, выполняющая обратное преобразование. Здесь имеется аналогия с используемыми в связи тестовыми последовательностями. Однако, в этом случае атака может быть направлена именно против опорного ЦВЗ. Другой альтернативой является вложение ЦВЗ в визуально значимые области изображения, которые не могут быть удалены из него без существенной его деградации. Наконец, можно разместить стего в инвариантных к преобразованию коэффициентах. Например, амплитуда преобразования Фурье инвариантна к сдвигу изображения (при этом меняется только фаза).

Другим методом защиты от подобных атак является блочный детектор. Модифицированное изображение разбивается на блоки размером 12х12 или 16х16 пикселей, и для каждого блока анализируются все возможные искажения. То есть пиксели в блоке подвергаются поворотам, перестановкам и т. п. Для каждого изменения определяется коэффициент корреляции ЦВЗ. Преобразование, после которого коэффициент корреляции оказался наибольшим, считается реально выполненным нарушителем. Таким образом появляется возможность как бы обратить внесенные нарушителем искажения. Возможность такого подхода основана на предположении о том, что нарушитель не будет значительно исказить контейнер (это не в его интересах).

**Заключение.** Подведя итоги, можно с уверенностью сказать, что от каждой атаки можно защититься зная все тонкости данной атаки, а также всегда стоит закрывать уязвимости своей системы и проводить атаки против неё тестируя на устойчивость к ним. Проводя атаки на свою же ЦВЗ мы анализируем её уязвимость к данной атаке которую в итоге мы можем закрыть. Полностью обеспечить защищенность своей ЦВЗ невозможно, ведь пока мы закрываем одну уязвимость, злоумышленник ищет новую, и эта гонка между злоумышленником и специалистом по компьютерной безопасности будет вечной. Самый верный способ защиты это «нападение», если мы сами умеем выявлять уязвимости, то мы и успешно сможем их закрыть, это остается главным аспектом деятельности специалистов КБ – изучение и практика атак, и разработка систем защиты.

#### ЛИТЕРАТУРА

1. Садов, В.С. Компьютерная стеганография (Конспект лекций)/В.С. Садов – Минск: БГУ, 2010. – 211 с.
2. Шнайер, Б. Секреты и ложь. Безопасность данных в цифровом мире / Б. Шнайер – СПб.: Питер, 2003. – 368 с.
3. Грибунин, В.Г. Цифровая стеганография/ В.Г. Грибунин – Москва: «Академия», 2009. – 265 с.
4. Шелухин, О.И. Стеганография. Алгоритмы и программная реализация/ О.И. Шелухин /С.Д. Канаев – Москва:«Горячая линия - Телеком», 2017. – 592 с.

УДК 62.785

## ПРИНЦИПЫ И ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ОХРАННОЙ СИГНАЛИЗАЦИИ В СОСТАВЕ ИНТЕГРИРОВАННОЙ СИСТЕМЫ БЕЗОПАСНОСТИ АДМИНИСТРАТИВНОГО ЗДАНИЯ

**Е.В. КОНЯЕВ***(Представлено: И.С. РУСЕЦКИЙ)*

В статье рассмотрена специфика и принципы проектирования интегрированной охранной системы для крупных административных объектов. При рассмотрении вопроса подразумевается, что данная система будет объединена с другими компонентами интегрированной системы безопасности.

**Введение.** Построение эффективной системы безопасности является актуальной проблемой для многих крупных компаний. Главной сложностью при построении целостной системы является обеспечение централизованного контроля всех систем. Ввиду большого количества оборудования могут возникнуть сложности в управлении, что приведет к нарушению функционирования системы безопасности. Очевидным и эффективным решением поставленной задачи является сведение всех систем в одну единую интегрированную систему безопасности.

Интегрированная система безопасности представляет собой комплекс оборудования, функционирующий по единым протоколам, использующий общие линии связи и базы данных, управляемый из единого программного ядра. Такая интеграция разрешает производить настройку автоматических реакций одной подсистемы на события, фиксируемые другой подсистемой. Список таких реакций называется сценарием, а количество сценариев и их сложность однозначно говорят о техническом уровне данной охранной системы. Однако нецелесообразно строить охранную сигнализацию как полностью автоматическую систему, так как невозможно предусмотреть все реально встречающиеся на конкретном предприятии ситуации. Необходимо автоматизировать лишь стандартные процессы, но окончательное решение по наиболее важным аспектам безопасности объекта должен все же принимать человек. Главное назначение интегрированной системы безопасности состоит в максимальном упрощении контроля за обстановкой на объекте, предоставлении ему обработанной четкой информации.

Рассмотрим особенности проектирования интегрированной системы на объекте, а также более углубленно один из компонентов данной системы, а именно проектирование охранной сигнализации. В качестве объекта защиты будет выступать административное здание, принадлежащее крупной промышленной организации.

**Принцип работы охранной сигнализации.** Простейшая система охранной сигнализации административного здания состоит из следующих обязательных элементов: прибора приемно-контрольного охранного, технических средств охраны объекта, устройства оповещения и блока передачи информации на пункт центрального наблюдения (рис.). Прибор приемно-контрольный охранный постоянно производит контролирование всех подключенных шлейфов, в которые, в свою очередь, включены технические средства охраны: датчики и извещатели различного назначения. Контроль происходит путём замера потребляемого шлейфом тока. В нормальном состоянии потребление тока имеет постоянное значение. При выходе из строя либо срабатывании хотя бы одного охранного датчика, в шлейфе моментально изменится ток потребления и данное изменение регистрирует прибор приемно-контрольный. Последний формирует сигнал тревоги и выдает его на устройство оповещения и передает сигнал на пункт централизованного наблюдения.



Рисунок. – Структурная схема охранной сигнализации

**Особенности проектирования.** Главный принцип, который должен соблюдаться при проектировании и монтаже систем охранной сигнализации, состоит в обязательном оснащении техническими средствами охранной сигнализации всех уязвимых мест объекта, защищаемого с помощью системы охранной сигнализации.

При проектировании в первую очередь стоит провести детальный анализ охраняемого объекта. Обусловлено это тем, что практически каждый объект является уникальной единицей, со своей спецификой и требованиями к реализации проектируемой системы. При разработке плана системы безопасности объект стоит делить на импровизированные зоны, включающие в себя подходящие по расположению помещения. Обусловлено это тем, что при расстановке технических средств охранной сигнализации на первом этаже необходимо учитывать возможность выема окон злоумышленником с последующим проникновением, возможность взлома дверей, как внешних, так и внутренних. Вместе с этим аналогичный уровень защиты необходимо поддерживать и на последнем этаже, так как злоумышленник может произвести разрушение стекла используя страховочные приспособления, только уже со стороны крыши.

Далее стоит определиться с количеством независимых рубежей. Согласно руководящему документу по тактике оснащения объекта техническими средствами охраны [1, 2] обязательным условием обеспечения повышенного и высокого уровней защиты объектов (помещений) подгрупп АI, АII и БII должно являться наличие многорубежной системы охранной сигнализации. Актуальность данного условия не поддается сомнению, поскольку большая часть административных зданий ввиду своей площади подходят под данные категории защиты. В частности для объектов, относящихся к подгруппе АI, рекомендуется применять три независимых рубежа.

Первым рубежом является рубеж охраны периметра здания. Периметральная охрана здания заключается в защите уязвимых мест, с которыми нарушитель может взаимодействовать для попадания на территорию охраняемого объекта. Чтобы предотвратить такие случаи, защищают внешние двери и окна, независимо от их противовзломной стойкости. Для этого используются магнитоконтактные охранные извещатели, формирующие извещение о тревоге при размыкании магнитных контактов извещателя, а также извещатели контроля разбития стекла, срабатывающие при регистрации звуковых колебаний в помещении, генерируемых стеклом при его разрушении под воздействием механического удара [3, с.59].

Второй рубеж содержит так называемы «ловушки»: контроль объема помещений и открытие дверей внутри здания. «Ловушки» необходимы, чтобы исключить возможность злоумышленника остаться на объекте защиты после сдачи его на охрану. Контроль объема помещений осуществляется оптоэлектронными пассивными инфракрасными извещателями, реагирующими на изменение уровня инфракрасного излучения в результате перемещения нарушителя в зоне обнаружения [3].

К третьему рубежу допускается относить дополнительные средства защиты мест скопления большого количества материальных ценностей. Например, на третьем рубеже защиты защищают сейфы с помощью емкостных, магнитоконтактных, либо и тех и других совместно работающих датчиков.

Дополнительным фактором при проектировании системы будет являться дальнейшая возможность объединения с другими системами, такими как системы контроля и управления доступом и видеонаблюдения.

#### ЛИТЕРАТУРА

1. СТБ 1250-2000. Охрана объектов и физических лиц. Термины и определения. – Минск : Госстандарт, 2001. – 13 с.
2. РД 28/3.006. Технические средства и системы охраны. Тактика применения технических средств охранной сигнализации». – Департамент охраны МВД РБ, 2005. – 51 с.
3. Рыжова, В.А. Проектирование и исследование комплексных систем безопасности. / В.А.Рыжова. – СПб: НИУ ИТМО, 2012. – 157 с.



УДК 316

**БЕЗОПАСНОСТЬ КАК ТЕОРЕТИЧЕСКИЙ КОНСТРУКТ****Е.В. КОНЯЕВ, К.В. САВИЦКИЙ**  
(Представлено: И.С. РУСЕЦКИЙ)

*В статье рассматривается внутренняя и внешняя безопасность как теоретический конструкт в современном социокультурном дискурсе, а также в нормативно-правовых актах Республики Беларусь.*

Существование угроз безопасности сопровождало общественному развитию на протяжении многих веков. Трансформационные процессы в современном обществе не являются исключением, поскольку они оказывают амбивалентное влияние на социальное развитие, воспроизводя традиционные и добавляя новые угрозы для различных сфер безопасности.

Для изучения влияния происходящих изменений и в целом сферы безопасности, требуется теоретическое обоснование, в связи с чем необходимо ответить на несколько важных вопросов: что такое безопасность и какова её структура.

В «Толковом словаре русского языка» С. И. Ожегова и Н. Ю. Шведовой слово «безопасность» трактуется как «состояние, при котором не угрожает опасность, есть защита от опасности» [4]. Данная трактовка рассматриваемого понятия, безусловно, является верной, но в то же время довольно абстрактной. Наиболее общее определение безопасности приводится в «Словаре социальных наук», вышедшем при поддержке ЮНЕСКО. В данном словаре Даниэль Лернер отмечает, что «безопасность – это отсутствие физической опасности (danger) или защита от неё» [12].

Обратимся к нормативным правовым актам Республики Беларусь и толкованию в них термина «безопасность». Наиболее развернутое определение, объединяющее все её сферы, даётся в документе под названием «Основные термины и понятия в экологической сфере», в котором содержится следующее: «Безопасность – состояние защищенности жизненно важных интересов личности, общества и государства от внутренних и внешних угроз или опасностей» [6].

Если попытаться выделить общее в различных дефинициях безопасности, можно обозначить два основных подхода: безопасность, как состояние спокойствия, свободы, уверенности, идентичности, отсутствия реальных угроз и атак; безопасность, как непрерывная деятельность индивидов, локальных сообществ, государств, альянсов и международных организаций в создании желаемого безопасного состояния.

При рассмотрении безопасности с позиции угроз и атак, актуальным становится теоретическая схема анализа объективных и субъективных аспектов угроз, которую предложил швейцарский политолог Дэниел Фрай. Схема акцентирует внимание на следующих моментах: состоянием отсутствия безопасности является ситуация, когда большой уровень угроз воспринимается как нормальное состояние дел; состояние обсессии наступает в тот момент, когда незначительная угроза воспринимается как большая; состояние ложной безопасности, т.е. такое положение дел, когда угроза, имеющая большой уровень опасности, воспринимается в качестве малозначимой; состояние безопасности, т.е. такое состояние дел, когда внутренние и внешние угрозы трактуются правильно.

В социокультурном дискурсе посвященном проблематике различных областей и уровней безопасности актуализирован вопрос взаимосвязи терминов: политика безопасности и стратегия безопасности. Существуют различные позиции: 1) данные термины являются синонимами; 2) политика безопасности вытекает из стратегии безопасности, а не наоборот. Для правильного понимания данных терминов необходимо их подробное рассмотрение, а также оценка взаимосвязи первичного термина с вытекающим из него понятием.

Политика безопасности включает в себя действия субъекта в какой-либо области для достижения конкретных целей. Из этого следует, что первоначально политикой безопасности должны быть определены цели в конкретной области. С позиции реалистов, при рассмотрении традиционного понимания роли государства в обеспечении безопасности, можно прийти к выводу, что политика безопасности является прерогативой государства в области международных отношений. В связи этим в научном дискурсе актуализируются такие дефиниции как национальная безопасность и безопасность государства.

В рамках концепции национальной безопасности обеспечение безопасности является элементом государственной политики относительно действий, направленных на создание оборонного потенциала в целях предотвращения и противодействия угрозам различного рода. Также политика национальной безопасности может быть определена, как элемент политики государства в рамках практической деятельности исполнительной власти с целью создания и использования оборонного потенциала для реализации целей, и задач, вытекающих из целей политики безопасности.

В теоретическом аспекте политика безопасности (security polisy) – это: деятельность государства во внутренней сфере направленная на организацию правильного функционирования органов государственной власти, стабилизацию общественного порядка и защиту жизни и здоровья граждан; деятельность государства во внешней сфере направленная на анализ и нейтрализацию угроз со стороны других государств, международных институций, а также частных и физических лиц; политика безопасности может быть направлена на различные сферы функционирования государства (например, военную, экономическую, энергетическую, информационную или экологическую) [8].

Политика национальной безопасности – элемент государственной политики относительно действий, связанных с созданием и использованием оборонного потенциала в целях предотвращения и противодействия угрозам различного рода в политике государственной безопасности [11].

Политика безопасности – система ценностей государства (союза), гарантирующая функционирование центра принятия решений (правительства), определяющего долгосрочные цели, в различных областях его деятельности [9].

Вышеперечисленные определения политики безопасности опираются на деятельность государства (правительства), из чего можно сделать вывод о его определяющей роли, в том числе и при конструировании политики безопасности. Государство определяет и формулирует цели, на основе которых разрабатываются стратегии обеспечения безопасности.

Для конкретизации сферы политики безопасности необходимо выделить функции политики в данной области. К общим функциям относятся следующие: единообразие законодательства о безопасности; формирование социальных установок всех субъектов; обеспечение индивидам чувства безопасности и развития; прогнозирование угроз безопасности для государств и союзов; создание условий для реализации национальных интересов; международное сотрудничество и реализация обязательств в рамках союзов; защита духовного и материального культурного наследия; защита окружающей среды (воздуха, воду и почвы); поиск новых инструментов и способов противостояния современным угрозам.

Исходя из функций политики безопасности, безопасность можно разделить на следующие составляющие: национальная безопасность и международная безопасность [9].

Национальная безопасность в социальном значении является одной из основных потребностей для индивидов, социальных групп и государств, т.к. она является главной целью деятельности структуры государственной власти и местного самоуправления. В функциональном значении обеспечение национальной безопасности является главной национальной миссией общества в целом и его госаппарата для достижения других целей.

Можно выделить три функции национальной безопасности: защита национальных интересов и ценностей от существующих и потенциальных угроз; создание внутренних и внешних условий для свободного развития личности, государства, общества; решение кризисных ситуаций и потенциальных проблем, которые возникают в процессе взаимодействия в рамках глобальных трансформаций.

Национальная безопасность включает в себя структурную, функциональную и системную составляющие. Соответственно, это может быть отражено в следующем определении: «национальная безопасность – это комплексная организация и подготовка государства к обеспечению безопасности в правовой, политической, стратегической, военной, гражданской, образовательной, дипломатической, экономической, социальной и интеграционной сферах».

В «Словаре терминов в области национальной безопасности» содержится одно из всеобъемлющих определений национальной безопасности, в котором она рассматривается как процесс или состояние, обеспечивающее функционирование государства в основных сферах, обеспечивающих существование, развитие и свободу реализации национальных интересов в определённой среде безопасности, путём принятия вызовов, использования возможностей и снижения рисков, а также предотвращения всех видов угроз в пользу своих интересов [10].

Национальная безопасность зачастую отождествляется с государственной безопасностью, что может быть связано с экзистенциальной потребностью социальных групп организовываться в государства. Также обеспечение национальной безопасности является одной из основных функций государства. В нормативных правовых актах Республики Беларусь данные понятия рассматриваются как идентичные [1; 2].

Национальная безопасность также может рассматриваться как национальный интерес, который отождествляется с совместным существованием государства и общества (населения страны), т.е. защитой целостности и политической независимости. Кроме того, что национальная безопасность рассматривается как совместное существование, она выступает фактором, который влияет на охрану таких феноменов как суверенитет, социальная и экономическая системы, престиж на международной арене, идеология, уровень жизни и пр.

Среди ценностей, обеспечиваемых национальной безопасностью, стоит выделить четыре основные: выживание, территориальная целостность, суверенитет и уровень жизни. Однако государства самостоятельно решают, какие ценности должны быть защищены в рамках национальной политики в области

безопасности, в связи с этим определяется стратегия для достижения намеченной цели. В свою очередь, стратегия зависит от национальных интересов государства и его особенностей, вызовов и угроз, с которыми оно сталкивается или может столкнуться в перспективе, а также от существующих возможностей государства как для развития, так и для реализации политики безопасности.

Существует точка зрения, согласно которой должны учитываться все вышеперечисленные аспекты, а национальная безопасность в этом случае трактуется как состояние и процесс жизни, существования и развития в любых условиях функционирования государства.

Обеспечение эффективной внутренней безопасности является важным элементом внутренней стратегии и политики в рамках национальной безопасности. Внутренняя безопасность должна быть обеспечена в соответствии с ожиданиями определённого качества уровня жизни граждан, условий для экономического и социального развития, культуры и состояния окружающей среды, а также возможностей развития цивилизации.

Эффективность защиты в значительной степени зависит от типа и степени внутренней угрозы. Внутренние угрозы национальной безопасности – это отдельный вид угроз на появлении определённого стечения обстоятельств, которые сочетают в себе условия различного рода, находящиеся во всех областях её внутренней активности, которые могут негативно сказаться на стабильности и гармоничном развитии государства в основных сферах, действие которых направлено на борьбу с ослаблением или утратой соответствующих позиций в международной среде [12].

Одной из важнейших составляющих внутренней безопасности является общественная безопасность. В нормативных правовых актах Республики Беларусь общественная безопасность трактуется как: урегулированная нормами права система общественных отношений, целями функционирования которой являются обеспечение личного и имущественного спокойствия, безопасности, создание условий для реализации прав граждан, обеспечения функционирования государственных органов, организаций [5]; состояние защищённости жизни, здоровья и благополучия граждан, а также ценностей общества от возможных опасностей и угроз, способных нанести им ущерб [4].

Основными категориями данной сферы безопасности, на наш взгляд, являются: защищённость ценностей, личное спокойствие и создание условий для реализации прав граждан и деятельности государственных органов.

В рамках внутренней политики общественная безопасность включает в себя различные сферы, что может быть обусловлено многогранностью повседневной общественной жизни. Среди них можно выделить политическую, экономическую, экологическую, информационную, демографическую, религиозную, социальную, образовательную и другие сферы [3].

Национальная безопасность – это не только внутренняя безопасность, но и внешняя. Именно эти две составляющие образуют одно целое функциональной структуры национальной безопасности в реальной повседневности. В науке область внешней безопасности служит полем для анализа событий, происходящих в международной сфере и в рамках альянсов, которые направлены на противодействие современным угрозам.

Обеспечение внешней безопасности обусловлено желанием актора (в данном случае – государства) сохранять безопасность в случае агрессии, также основано на уверенности, что актор не будет являться объектом атак, а в случае возникшей агрессии получит немедленную помощь со стороны других государств – союзников.

Суть внешней безопасности сводится к пониманию реалистами национальной безопасности, её обеспечение сфокусировано на том, чтобы не допустить политического, военного, экономического, идеологического, культурного или иного давления на государство.

Одним из основных аспектов обеспечения внешней безопасности является профессиональная, ответственная, компетентная и прозрачная внешняя политика государства на международной арене, а также крепкие и надёжные связи между странами, международными организациями и другими субъектами международных отношений.

Таким образом, внешняя безопасность должна быть основана на взаимопонимании и взаимоуважении акторов в рамках международных отношений, а также требовать от них активного и эффективного взаимодействия в союзах, альянсах и объединениях.

Международная безопасность является ещё одной областью теории безопасности, которая включает в себя региональную и глобальную безопасность. Основной задачей международной безопасности в XXI веке является сохранение мира во всём мире. Данная область теории безопасности обладает своим терминологическим аппаратом, в рамках которого безопасность рассматривается как процесс или состояние международной (региональной, континентальной, глобальной) среды, гарантирующей его стабильность и возможность развития, полученную в результате предпринимаемых действий.

Международная безопасность характеризуется отсутствием существующих угроз и субъективных страхов, оказывающих влияние на защиту определённых государственных и международных ценностей,

посредством норм, институтов и инструментов, обеспечивающих мирное урегулирование конфликтов, а также устранение потенциальных угроз. Как и национальная безопасность, международная является не только состоянием, но и процессом, дающим возможность жизни и развития индивида, государства, региона, континента, мира.

Влияние на международную безопасность оказывает как политика внешней безопасности государств, так и функционирующие международные структуры [7]. Следует отметить, что международная безопасность не представляет совокупность безопасностей отдельных государств, так как кроме ценностей отдельных государств включает в себя ценности глобальной системы безопасности (стабильность, мир, международный порядок, сотрудничество). Иными словами, международная безопасность – состояние и процесс, в котором государства и международное сообщество могут свободно достигать намеченные цели, не опасаясь внешних угроз (политической, военной, экономической и др.).

Кроме традиционной реалистской концепции военно-политической безопасности, в рамках международной безопасности дополнительно можно выделить такие виды безопасности как: экономическую, геополитическую, экологическую, информационную, институциональную, демографическую, культурную и социальную.

Таким образом, безопасность как теоретический конструкт состоит из таких взаимосвязанных понятий как: национальная безопасность, общественная безопасность, международная безопасность и др. В связи с многообразием подходов к трактовке «безопасности» необходимо отметить необходимость его операционализации и интерпретации в социологических исследованиях.

Потребность в конкретизации понятия «безопасность» и его социологическое осмысление обусловлено существенными изменениями в системе международной безопасности. В связи с распадом bipolarного мира вероятность глобальных и региональных конфликтов снизилась. Однако в XX и XXI вв. произошло возрождение большого количества давних споров и конфликтов локального значения, которые основываются на национальной, этнической, религиозной почвах. Кроме того, для современной ситуации на международной арене характерны пограничные конфликты, а также различного рода социальные и экономические кризисы.

На сегодняшний день мы не можем говорить о состоянии полноценной международной безопасности, так как находимся в состоянии переходного периода (в экономическом, технологическом, политическом, социокультурном и демографическом аспектах), характерными чертами которого являются нестабильность, неопределённость и непредсказуемость.

#### ЛИТЕРАТУРА

1. Концепция национальной безопасности Республики Беларусь, утвержденная Указом Президента Республики Беларусь от 9 ноября 2010 г. № 575
2. Модельный Закон «О безопасности», принятого постановлением Межпарламентского Комитета Республики Беларусь, Республики Казахстан, Кыргызской Республики, Российской Федерации и Республики Таджикистан от 15 октября 1999 г. № 9-9. – Ст. 1.
3. Мясникович М. В. Национальная безопасность Республики Беларусь. Современное состояние и перспективы / М. В. Мясникович, В. В. Пузиков. – Минск: Экономика и право, 2003. – 451 с.
4. Ожегов, С. И. Толковый словарь русского языка / С. И. Ожегов, Н. Ю. Шведова. – М. : А ТЕМП, 2006. – 944 с.
5. Об утверждении Положения о применении систем безопасности и телевизионных систем видеонаблюдения [Электронный ресурс] : постановление Совета Министров Респ. Беларусь, 11 дек. 2012 г., № 1135// Национальный правовой Интернет-портал Республики Беларусь. – Режим доступа: [http://pravo.by/upload/docs/op/C21201135\\_1355432400.pdf](http://pravo.by/upload/docs/op/C21201135_1355432400.pdf). – Дата доступа: 05.11.2017.
6. Рекомендации «Основные термины и понятия в экологической сфере» [Электронный ресурс] : прин. постановлением Межпарл. Ассамблеи государств – участников СНГ от 27 ноября 2015 г. № 43–10 // Национальный центр правовой информации Республики Беларусь. – Режим доступа: [http://www.etalonline.by/?type=text&regnum=n21500095#load\\_text\\_none\\_1\\_](http://www.etalonline.by/?type=text&regnum=n21500095#load_text_none_1_). – Дата доступа: 05.03.2018.
7. Czaputowicz, J. Bezpieczeństwo europejskie : koncepcje, instytucje, implikacje dla Polski / pod red. J. Czaputowicza. – Warszawa : Ararat, 1997. – 290 s.
8. Opara, S. Mały słownik politologii / pod red. S. Opara i zesp. – Toruń : Wydawnictwo Adam Marszałek, 2007. – S. 120.
9. Pokruszyński, W. Bezpieczeństwo teoria i praktyka / W. Pokruszyński. – Józefów: Wydawnictwo Wyższej Szkoły Gospodarki Euroregionalnej im. Alcide De Gasperi, 2012. – 271 s.
10. Słownik terminów z zakresu bezpieczeństwa narodowego / Warszawa: AON, 2002. – S.16.
11. Słownik terminów z zakresu bezpieczeństwa narodowego/ kier. Jacek Pawłowski. – Warszawa: AON, 2002. – S. 99.
12. Zięba, R. Bezpieczeństwo narodowe i międzynarodowe u schyłku XX wieku / R. Zięba, D.B. Bobrow, E. Haliżak. – Warszawa: Wydawnictwo Naukowe SCHOLAR, 1997. – S. 3.

УДК 004.056.5

## ПРОЕКТИРОВАНИЕ СТЕГОСИСТЕМЫ, ОСНОВАННОЙ НА СОКРЫТИИ ТЕКСТОВЫХ ДАННЫХ В ИЗОБРАЖЕНИЯХ ПРИ ПОМОЩИ ДИСКРЕТНЫХ ПРЕОБРАЗОВАНИЙ

А.В. КОХАНОВСКИЙ

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

В статье представлен практический способ создания надёжной стегосистемы с нестандартным способом сокрытия информации. Цель данной работы – построить систему, основанную на сокрытии текстовых сообщений в изображениях, изучить атаки и выяснить, насколько пригодна такая система для практического применения. Задача решалась путём разбиения основной программы на библиотеки и вспомогательные подпрограммы. Программа написана на языке программирования С#.

**Введение.** В статье пойдёт речь о сокрытии данных в изображениях с помощью одного из методов стеганографии. Под цифровой стеганографией понимается сокрытие одной информации в другой. Причем сокрытие это должно реализоваться таким образом, чтобы, во-первых, не были утрачены свойства и некоторая ценность скрываемой информации, а во-вторых, неизбежная модификация информационного носителя не только не уничтожила смысловые функции, но и на определенном уровне абстракции даже не меняла их. Тем самым факт передачи одного сообщения внутри другого не выявляется традиционными методами.

**Принцип построения стеганографической системы и описание алгоритма сокрытия данных.** Стандартная стеганографическая схема сохраняется вне зависимости от технологии, которой она реализуется. Задачу встраивания и выделения сообщений из другой информации выполняет стегосистема. Стегосистема состоит из следующих основных элементов (рис.).



Рисунок. – Структурная схема типичной стегосистемы

В качестве данных может использоваться любая информация: текст, сообщение, изображение и т. п. Контейнер - любая информация, предназначенная для сокрытия тайных сообщений. Пустой контейнер - контейнер без встроеного сообщения; заполненный контейнер или стегоконтейнер, содержащий встроенную информацию. Встроенное (скрытое) сообщение - сообщение, встраиваемое в контейнер. Стеганографический канал или просто стегоканал - канал передачи стего. Стегоключ или просто ключ - секретный ключ, необходимый для сокрытия информации. В зависимости от количества уровней защиты (например, встраивание предварительно зашифрованного сообщения) в стегосистеме может быть один или несколько стегоключей.

Данные, содержащие скрытое сообщение, могут подвергаться преднамеренным атакам или случайным помехам. Как показано на рисунке 1, в стегосистеме происходит объединение двух типов информации так, чтобы они могли быть различимы двумя принципиально разными детекторами. В качестве одного из детекторов выступает система выделения ЦВЗ, в качестве другого – человек [1].

Стеганографические методы замены неустойчивы к любым искажениям, а применение операции сжатия с потерями приводит к полному уничтожению всей секретной информации, скрытой методом НЗБ в изображении. Более устойчивыми к различным искажениям, в том числе сжатию, являются методы, которые используют для сокрытия данных не временную область, а частотную [2].

Подавляющее большинство методов компьютерной стеганографии базируется на двух ключевых принципах:

- файлы, которые не требуют абсолютной точности (в данном случае файлы с изображением), могут быть видоизменены без потери своей функциональности;
- органы чувств человека неспособны надежно различать незначительные изменения в модифицированных таким образом файлах и/или отсутствует специальный инструментарий, который был бы способен выполнять данную задачу.

В основе базовых подходов к реализации методов компьютерной стеганографии в рамках той или иной информационной среды лежит выделение малозначительных фрагментов этой среды и замена существующей в них информации информацией, которую необходимо скрыть.

Рассматриваемый в данной работе, алгоритм Коха-Жао для встраивания информации использует частотную область контейнера и заключается в относительной замене величин коэффициентов дискретного косинусного преобразования (ДКП) [3].

Изображение разбивается на блоки размерностью  $8 \times 8$  (в нашем случае на блоки размером  $2 \times 2$ ,  $3 \times 3$  и  $4 \times 4$ ) пикселей и к каждому блоку применяется ДКП. Каждый блок пригоден для записи одного бита информации.

Алгоритм скрытия будет заключаться в следующем:

- итерируем изображение двойным массивом с шагом в 8;
- на каждой итерации создаем временный массив  $8 \times 8$  пикселей, каждым элементом которого будет набор трех цветов пикселя;
- применяем ДКП к этому массиву, и получаем массив коэффициентов размером  $8 \times 8$ ;
- выбираем 2 коэффициента и высчитываем их разность по модулю;
- если разность меньше или равна 25, то присваиваем первому коэффициенту положительное значение второго + константа, либо тоже самое, но со знаком минус (это называется передача бита);
- если разность меньше либо равна -25, то выполняем те же действия только для второго коэффициента;
- далее применяется обратное ДКП чтобы перевести наши коэффициенты обратно в пространственную область;
- после чего копируем новые значения цветов в изображение.

Достоинства метода:

- устойчивость к JPEG-компрессии с малым коэффициентом сжатия

Недостатки:

- заметное визуальное искажение изображения-контейнера при большом пороговом значении разницы между коэффициентами ДКП блоков;
- малый объем сообщения, который можно встроить.

**Заключение.** Исследования в области стеганографии очень перспективное направление защиты информации, так как в современном мире задача передачи секретной информации стоит наравне со скрытым общением, т.е. скрытия факта передачи сообщений. Поэтому, необходимо продолжать исследованиями в этой области для поиска новых, эффективных, методов или улучшения существующих. В данной работе был рассмотрен метод встраивания информации в изображения. Метод самый перспективный, однако, требует доработки в плане пропускной способности, и вероятности правильного извлечения встроженных бит информации. Эффективность применения ДКП в данном методе для сжатия изображений объясняется тем, что оно хорошо моделирует процесс обработки изображения в системе человеческого зрения (СЧЗ), отделяет «значимые» детали от «незначимых». Значит, его более целесообразно применять в случае активного нарушителя. Программный продукт реализован и готов к использованию с возможностью доработки.

#### ЛИТЕРАТУРА

1. Matsui K., Tanaka K., and Nakamura Y. Digital signature on a facsimile document by recursive MH coding / Symposium On Cryptography and Information Security, 1989.
2. Садов, В. С. Компьютерная стеганография / В. С. Садов. – М: МГВРК, 2012. – 289 с.
3. Грибунин, В. Г. Цифровая стеганография / В. Г. Грибунин. – М.: СОЛОН-Пресс, 2002. – 272 с.

УДК 004.056.5

**ОПРЕДЕЛЕНИЕ СТЕПЕНИ ПРИГОДНОСТИ КОНТЕЙНЕРА  
ДЛЯ СТЕГАНОГРАФИЧЕСКОЙ МОДИФИКАЦИИ В СТЕГОСИСТЕМЕ,  
ОСНОВАННОЙ НА СОКРЫТИИ ТЕКСТОВЫХ ДАННЫХ В ИЗОБРАЖЕНИЯХ  
ПРИ ПОМОЩИ ДИСКРЕТНЫХ ПРЕОБРАЗОВАНИЙ**

**А.В. КОХАНОВСКИЙ**

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

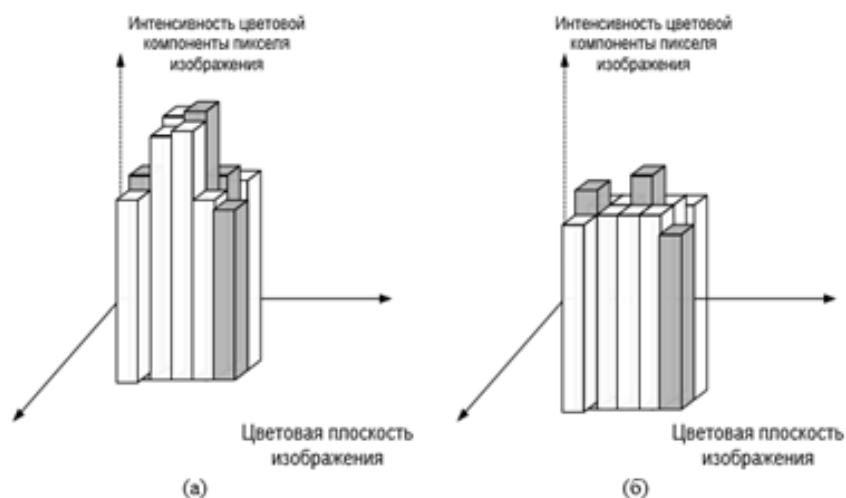
В статье определена степень пригодности контейнера для стеганографической модификации и представлены некоторые виды атак на стегосистему. Цель данной работы – построить систему, основанную на скрытии текстовых сообщений в изображениях, а также изучить атаки и выяснить, насколько пригодна такая система для практического применения. Задача решалась путём разбиения основной программы на библиотеки и вспомогательные подпрограммы. Программа написана на языке программирования C#.

**Введение.** Стеганографической модификации могут подвергаться как пространственные, так и частотные параметры контейнера-изображения. Для стеганографической модификации пригодны все эти параметры, но перед процедурой встраивания необходимо оценить пределы модификации параметров того или иного контейнера, а также вносимые при этом искажения. Пропускная способность, надёжность, устойчивость стеганографической системы во многом будет определяться степенью модификации контейнера-изображения [1].

**Оценка стеганостойкости реализованной системы.** Каждое изображение обладает уникальными свойствами, которые можно положить в основу их разделения на классы.

1. Изображения с небольшим количеством цветов (4-16) и большими областями, заполненными одним цветом.
2. Изображения, построенные на компьютере, с плавными переходами цветов.
3. Фотореалистичные изображения.
4. Фотореалистичные изображения с наложением деловой графики.
5. Картографические изображения.
6. Космические изображения.

Для обеспечения незаметности факта скрытия данных в контейнерах-изображениях необходимо выбирать изображения, содержащие области с резкими переходами цветов, границы объектов, так как в них можно в большей степени скрыть небольшие изменения интенсивностей цветовых компонент пикселя контейнера по отношению к соседним пикселям (рис.). Модифицированные пиксели на рисунке выделены серым цветом. Видно, что даже незначительное изменение интенсивности какой-нибудь цветовой компоненты отдельных пикселей, в областях, заполненных одним цветом (рис. 1, б), увеличивает возможность визуального или компьютерного детектирования факта стеганографического скрытия данных.



**а** – область резкого перехода цветов; **б** – область, заполненная одним цветом

**Рисунок. – Цветовые плоскости модифицированных областей изображения**

Анализ содержимого контейнеров-изображений целесообразно проводить при помощи математического преобразования, позволяющего выделить его частотные параметры и оценить вклад отдельных частот в состав изображения. Одним из таких преобразований является дискретное косинусное преобразование [2], применяемое к изображениям посредством окон размеров  $n \times n$  пикселей. Дискретное косинусное преобразование позволяет получить информацию о резких и плавных границах цветов изображения, областях, заполненных одним цветом или с градиентным изменением цвета и др.

Оценку степени пригодности изображения для стеганографической модификации целесообразно проводить в два этапа.

На первом этапе изображения делятся на классы согласно относительному весу их пространственных частот. Положительные результаты получены при использовании спектральной классификации изображений, позволяющей провести их разбиение на 8 классов [3]. Классы с 1-го по 3-й описывают изображения с наибольшим относительным весом низких частот; классы с 4-ого по 7-ой разграничивают изображения по спектральным составляющим, сосредоточенным в областях близких к низкочастотному и/или высокочастотному диапазонам; восьмой класс отделяет изображения, имеющие равномерный спектр в пределах всего рассматриваемого диапазона.

На втором этапе оценивается пропускная способность изображения-контейнера путем исключения непригодных для модификации областей, например, заполненных одним цветом или градиентным изменением цвета. Обнаружить такие области на изображении возможно при анализе его спектрального состава (наличие границ на изображении приводит к увеличению вклада средних и высоких частот), так и при оценке изменения интенсивности пикселя по отношению к соседним пикселям.

В методе Коха-Жао для обеспечения незаметности факта скрытия данных в контейнерах-изображениях нежелательно использовать изображения с небольшим количеством цветов и большими областями, заполненными одним цветом (в частности, белым). Так как при этом, после шифрования на изображении будет наглядно виден факт скрытия информации.

Выделяются следующие типы атак для метода Коха-Жао:

1. Атаки против встроенного сообщения, направленные на удаление или порчу встроенной информации путем манипулирования заполненным контейнером. Входящие в эту категорию методы атак не нацелены на оценку и выделение сообщения. Примерами таких атак в данной работе является сжатие изображений.

2. Атака против встроенного сообщения, направленная на использование фильтров. При этом сообщение в изображении остается, но теряется возможность его приема. В эту категорию входят такие атаки, как атака с использованием эффектов в различных фоторедакторах.

3. Атака против встроенного сообщения, направленная на использование геометрических преобразований. Данная категория атак связана с усечением и изменением размерности нашего изображения.

4. Атака, направленные на изменении яркости и контрастности изображения. Данный вид атак связан с существенным изменением изображения.

**Заключение.** При выполнении данного проекта были выделены основные результаты и выводы:

1. Произведен анализ метода Коха-Жао, используемый для встраивания данных в изображения.

2. В работе были произведены многочисленные эксперименты, которые позволили дать определенные рекомендации по выбору атак для данного метода.

3. Метод Коха-Жао является достаточно устойчивым к существенному изменению изображений (контрастность, яркость и т.д.) и к геометрическим преобразованиям (размерность, повороты изображений и т.д.), связанных с ними.

4. При использовании в качестве атак фильтров и сжатия в формате JPEG зашифрованного изображения гарантировано разрушается встроенное сообщение.

5. Эффективность применения ДКП в данном методе для сжатия изображений объясняется тем, что оно хорошо моделирует процесс обработки изображения в системе человеческого зрения (СЧЗ), отделяет «значимые» детали от «незначимых». Значит, его более целесообразно применять в случае активного нарушителя.

6. В ходе тестирования программы при изменении разности коэффициентов (P) и сравнении полученных результатов скрытия и извлечения информации, можно сделать вывод, что чем больше значение P, тем стеганосистема, созданная на основе данного метода, является более стойкой к компрессии, однако качество изображения при этом значительно ухудшается.

7. Программа реализована и готова к использованию с возможностью ее доработки.



ЛИТЕРАТУРА

1. Садов, В. С. Компьютерная стеганография / В. С. Садов. – М: МГВРК, 2012. – 289 с.
2. Тихоненко, С.Г. Формирование критерия качества фильтрованных изображений / С. Г. Тихоненко // Тезисы докладов IX республиканской научной конференции студентов и аспирантов РБ, Гродно, 26-27 мая 2004 г. В 8-ми частях / Гродненский гос. университет; редкол.: А. И. Жук. – Гродно, 2004. – Ч.6. – С. 241– 243. 40.
3. Чернявский, А.Ф. Оценка информационных потерь при фильтрации изображений / А.Ф. Чернявский, С.Г. Тихоненко, В.С. Садов // Информатика. – 2005. – № 3(7). – С. 52–59.

УДК 004.4

АРХИТЕКТУРНЫЕ РЕШЕНИЯ РЕАЛИЗАЦИИ КЛИЕНТСКОЙ ЧАСТИ  
МОБИЛЬНОЙ ММО-ИГРЫ В КОСМИЧЕСКОЙ СТИЛИСТИКЕ

Р.Р. КРАСЬКО

(Представлено: канд. техн. наук, доц. А.Ф.ОСЬКИН)

В статье формулируется постановка задачи для клиентской части мобильной ММО-игры в космической стилистике, концепция построения клиентского приложения. Предлагается подход к программной реализации данного приложения для мобильных устройств под управлением системы Android, приводится описание основных структурных компонентов клиентского приложения.

**Введение.** Онлайн игры являются одними из самых востребованных направлений игровой индустрии в настоящее время, по статистике можно выделить несколько наиболее популярных жанров: MOBA (Multiplayer Online Battle Arena, «многопользовательская онлайн боевая арена»), шутеры, ММО и MMORPG, стратегии, азартные онлайн-игры [1].

Многопользовательские онлайн-игры и игры в космической стилистике показали свою востребованность на примере таких популярных игр под персональные компьютеры как EVE Online, Elite: Dangerous, Star Wars: The Old Republic, Star Citizen. Следовательно, у пользователей мобильных устройств космическая MMORPG с элементами квеста и аркады способна получить популярность.

**Описание игры.** Мир игры представляет собой галактику, которая состоит из множества систем, а те в свою очередь представляют собой некоторое количество планет, которые вращаются вокруг звезды. На планетах игрок может брать квесты, покупать/продавать товары, улучшать/чинить оборудование. Главный герой выступает в роли капитана космического корабля на котором он передвигается по игровому миру. Также в распоряжении игрока будет находиться виртуальная валюта, за которую он сможет улучшать/чинить оборудование, комплектовать команду, вести торговлю и др. За выполнение заданий и участия в сражениях игроку начисляется опыт, который используется для улучшения характеристик персонажа. Бои с другими игроками будут проходить в пошаговом режиме. В игре будут присутствовать 3 расы: зеленые, красные, синие. При старте игры пользователь может присоединиться к одной из вышеперечисленных рас. В зависимости от выбора у игрока будут различные стартовые характеристики.

**Игровые экраны.** На основе описания игры в мобильном клиенте можно выделить следующие игровые экраны:

«Стартовый экран» является входной точкой для приложения. При его старте загружаются и инициализируются все игровые ресурсы. Данные могут быть получены как от сервера, так и из локального кэша. После окончания подготовительных работ игрок переходит на экран «Главное меню».

Экран «Главное меню» предоставляет возможность переходить на экраны «Регистрация и авторизация», «Настройки», «О приложении», а также позволяет выйти из приложения.

Экран «Регистрация и авторизация». Вход возможен под аккаунтами Facebook, Google+, Twitter. Если игрок уже существует, то пользователь появляется на планете (экран «Планета»), которую он посещал последний раз. В противном случае он переходит на экран «Создание персонажа».

Экран «Настройки» предоставляет возможность настроить уведомления, вибрацию, громкость музыки и звуковых эффектов.

Экран «О приложении» содержит краткую информацию о текущей версии игры.

Экран «Создание персонажа» предоставляет пользователю возможность выбора имени, пола, расы и иконки персонажа. После выбора характеристик игрок появляется на случайной планете в галактике.

Экран «Планета». Позволяет переходить на экраны «Магазин», «Мастерская», «Доска объявлений», «Бар», «Система», «Корабль».

Экран «Магазин» позволяет за виртуальную валюту покупать комплектующие для корабля.

Экран «Мастерская» позволяет чинить и модифицировать комплектующие для корабля.

Экран «Доска объявлений» содержит список доступных игроку квестов для выполнения.

Экран «Бар» позволяет за виртуальную валюту нанимать команду.

Экран «Система» представляет собой некоторое количество планет, которые вращаются вокруг звезды. Управление кораблем осуществляется с помощью виртуального джойстика в левом нижнем углу. Скорость и маневренность корабля зависит от его технических характеристик. В любой момент игры можно перейти на экраны «Галактика», «Корабль», «Радар». При нажатии на чужой корабль появляется возможность открыть экраны «Торговля», «Бой», «Чат».

Экран «Галактика» представляет собой мини-карту галактики и позволяет совершить полет в любую систему, которая доступна пользователю. Дальность полета зависит от технических характеристик корабля и навыков команды.

Экран «Корабль» позволяет просмотреть комплектующие корабля, характеристики персонажа и команды, список активных квестов.

Экран «Радар» представляет собой мини карту текущей системы.

Экран «Бой». Бои с другими игроками будут проходить в пошаговом режиме, суть которого сводится к поочередному выполнению каких-либо определенных действий, например, использование оружия, ремонтного дроида и т.д.

Экран «Торговля» позволяет произвести обмен между двумя реальными игроками.

Экран «Чат» представляет текстовый чат между двумя реальными игроками.

#### **Функциональные требования:**

- загрузка и отображение текстур, спрайтов, шрифтов и частиц, построение пользовательского интерфейса;

- загрузка звуковых ресурсов, воспроизведение фоновой музыки и звуковых эффектов;

- сетевое взаимодействие с сервером будет осуществляться по протоколам UDP, TCP и HTTP. Первый служит для передачи данных в реальном времени, где важна максимальная скорость и допустима потеря пакетов. Используется при взаимодействии игроков в системе во время полета. Протокол TCP служит для надежной передачи данных, где потеря пакетов недопустима. Используется в пошаговом бою, при выполнении квестов, обмене товарами с другими игроками, текстовом чате и т.д. HTTP протокол используется для взаимодействия с сервером вне игровой логики: внутриигровые платежи, авторизация, регистрация.

- хранение данных;

- обработка задержек на стороне клиента: интерполяция на стороне клиента, прогнозирование на стороне клиента, расчет траектории движения, прогнозирование и переигровка шагов клиента.

- получение и обработка уведомлений от сервера, которые сообщают пользователю о различных событиях, происходящих в игре;

- текстовый чат для общения пользователей во время игры;

- покупка виртуальной валюты за реальные деньги;

- интеграция бонусной рекламы (получение каких-либо виртуальных вознаграждений за просмотр рекламы).

**Компоненты мобильного клиента.** Для реализации мобильного клиента выбран паттерн MVP (Model-View-Presenter) т.к. он позволяет отделить бизнес логику от отображения, что облегчит автоматическое модульное тестирование, правку и поддержку кода.

Работа с графикой и звуком будет осуществлена с помощью LibGDX, который является Java фреймворком, предоставляющий кроссплатформенный API для разработки 2D игр и приложений реального времени [2]. Данный фреймворк имеет хорошую документацию, предоставляет кроссплатформенное API для проведения внутриигровых платежей, дает возможность использовать как низкоуровневые (OpenGL) инструменты работы с графикой, так и высокоуровневые, полностью бесплатен и имеет открытый код.

Для работы с сетью будет использоваться фреймворк Netty. Данный фреймворк совместим с LibGDX, позволяет работать с UDP, TCP и HTTP, имеет хорошую документацию и полностью бесплатен [3]. В качестве протокола сериализации будет выступать Protocol Buffer.

Хранение данных реализуется библиотекой Realm. Библиотека представляет собой бесплатную встраиваемую NoSql базу данных для мобильных устройств, которая имеет высокую скорость работы [4].

Работа с уведомлениями, рекламой и социальными сетями выполняется через Firebase SDK, который предоставляет следующие инструменты:

Firebase Cloud Messaging – кроссплатформенное решение для работы с нотификациями для Android, iOS и веб приложений, которое является полностью бесплатным.

Firebase Auth – сервис, который позволяет проводить аутентификацию через Facebook, GitHub, Twitter, Google или адрес электронного ящика.

Adwords – сервис для интеграции рекламы в мобильные приложения.

На рисунке представлена диаграмма компонентов, на которой размещены основные компоненты мобильного клиента.

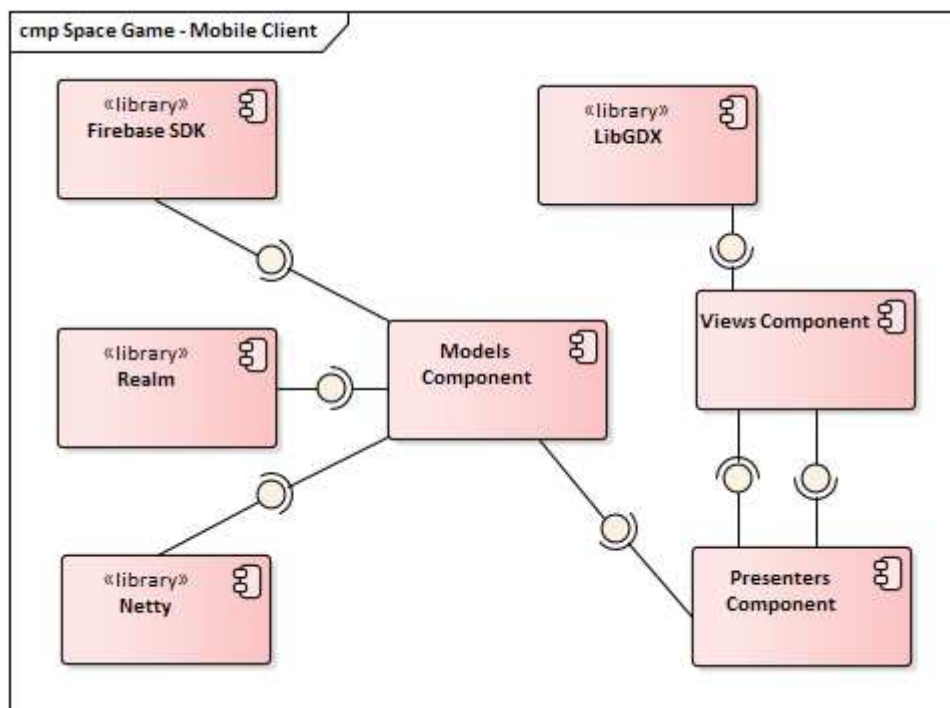


Рисунок. – Диаграмма компонентов мобильного клиента

**Заключение.** Сформулирована постановка задачи для клиентской части мобильной MMO-игры в космическом сеттинге. Предложен подход к программной реализации данного приложения для мобильных устройств под управлением системы Android. Архитектурным паттерном был выбран MVP. В качестве инструмента для работы с графикой и звуком было решено использовать LibGDX. Сетевое взаимодействие с сервером будет осуществляться с помощью фреймворка Netty. Для хранения локальных данных будет использоваться библиотека Realm. Интеграция с сервисами от Google будет реализована через Firebase SDK.

#### ЛИТЕРАТУРА

1. Тренды онлайн-игр в 2017 году и дальше [Электронный ресурс] / Plarium © 2019 – Режим доступа: <https://plarium.com/ru/sovety-mm0/trendy-onlayn-igr-2017/> – Дата доступа: 26.09.2019.
2. LiGDX Features [Электронный ресурс] / LibGDX © 2019. – Режим доступа: <http://www.libgdx.ru/2013/08/goals-features.html> – Дата доступа: 26.09.2019.
3. Netty [Электронный ресурс] / Netty © 2019. – Режим доступа: <https://netty.io/> © 2018. – Дата доступа: 26.09.2019.
4. Realm Features [Электронный ресурс] / Realm © 2019. – Режим доступа: <https://realm.io/products/realm-database/> – Дата доступа: 26.09.2019.

УДК 004.4

## ИСПОЛЬЗОВАНИЕ ЧИСТОЙ АРХИТЕКТУРЫ ПРИ РАЗРАБОТКЕ ANDROID ПРИЛОЖЕНИЙ

Р.Р. КРАСЬКО

(Представлено: канд. техн. наук, доц. А.Ф.ОСЬКИН)

В статье описываются особенности разработки мобильного Android приложения с использованием чистой архитектуры. Предлагается набор инструментов для программной реализации данного подхода.

**Введение.** Архитектура приложения является важнейшим аспектом при разработке программного обеспечения - она должна быть надежной, стабильной, гибкой в тестировании, легко расширяться и изменяться. В то же время архитектура должна быть понятной для разработчиков с разным уровнем подготовки и опыта для обеспечения удобства сопровождения.

**Чистая архитектура.** При использовании чистой архитектуры весь код приложения разделен на уровни, которые придерживаются одного правила: внутренний уровень не должен ничего знать про внешний. При этом внутренний уровень содержит бизнес логику, а внешний ее реализацию в зависимости от платформы. Данный подход должен отвечать и другим требованиям:

- независимость от инструментов. Архитектура не должна полагаться на существование какой-либо библиотеки. Это позволяет использовать с минимальными затратами изменять реализацию бизнес-логики [1];
- тестируемость, бизнес-логика должна быть тестируемой без любых внешних элементов [1];
- независимость от базы данных, бизнес-логика не должна быть привязана к конкретным базам данных [1];
- независимость от любого внешнего агента, бизнес-логика не должна знать ничего о взаимодействии в внешними источниками данных [1].

На рисунке предоставлены различные уровни и компоненты архитектуры и их взаимодействие.

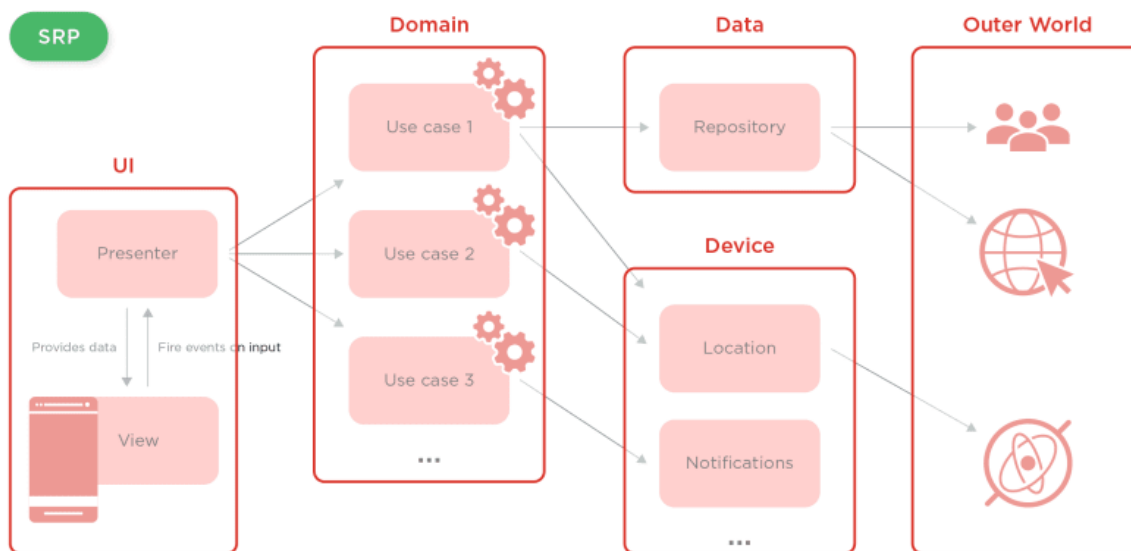


Рисунок. – Уровни и компоненты чистой архитектуры

### Уровень представления (Presentation/UI):

- для данного слоя выбран паттерн MVP(Model View Presenter), также возможно использование MVVM;
- «View» представляет собой реализацию паттерна «Passive View» и предоставляет собой набор интерфейсов, которые должны быть реализованы компонентами из Android SDK (Fragment, Activity, Adapter, Custom View) [2];
- «Presenter» является посредником между пользовательским интерфейсом и бизнес логикой. Должен быть платформонезависимым для улучшения тестируемости кода [2];

– при тестировании используются «JUnit» и «Mockito». Для интеграционных тестов также используется «Esspresso».

**Уровень бизнес-логики (Domain):**

– Состоит из вариантов использования (Use Cases), которые представляют собой самые примитивные части бизнес логики приложения для максимальной простоты и возможности их дальнейшего переиспользования;

– При тестировании используются «JUnit» и «Mockito».

**Уровень данных (Data):**

– паттерн репозиторий отвечает за интерфейс, по которому варианты использования из доменного уровня могут получать необходимые данные [3];

– на этом уровне возможна реализация кеширования;

– при тестировании используются «JUnit», «Mockito», «Robolectric».

**Платформозависимый слой (Device):**

– содержит реализации платформозависимой функциональности: уведомления, сенсоры, различные менеджеры и так далее.

**Взаимодействие между слоями, навигация:**

– внедрение зависимостей с использованием Dagger 2;

– поток данных между слоями осуществляется с помощью RxJava;

– для навигации используется Cicerone;

**Заключение.** Описаны особенности использования чистой архитектуры при разработке приложений для мобильных Android приложений. Предложен набор инструментов для реализации данного подхода. Данная архитектура является гибкой в поддержке, простой в тестировании и очень удобной при работе в больших командах с различным уровнем специалистов.

#### ЛИТЕРАТУРА

1. Чистая архитектура на Android и iOS [Электронный ресурс] / Appttractor © 2019 – Режим доступа: <https://appttractor.ru/develop/chistaya-arhitektura-na-android-i-ios.html> – Дата доступа: 26.09.2019.
2. Model-view-presenter [Электронный ресурс] / Wikipedia © 2019. – Режим доступа: <https://en.wikipedia.org/wiki/Model-view-presenter> – Дата доступа: 26.09.2019.
3. Repository [Электронный ресурс] / Martinfowle © 2019. – Режим доступа: <https://martinfowler.com/eaaCatalog/repository.html> – Дата доступа: 26.09.2019.

УДК 004.8

## ОБРАБОТКА ДАННЫХ ДЛЯ МАШИННОГО ОБУЧЕНИЯ

М.А. ШЕВЦОВ

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

*Предварительная обработка и очистка данных являются важными задачами, которые обычно необходимо выполнить, прежде чем набор данных можно будет эффективно использовать для машинного обучения. В статье рассматриваются способы повышения эффективности данных.*

**Введение.** Необработанные данные зачастую искажены и ненадежны, и в них могут быть пропущены значения. Использование таких данных при моделировании может приводить к неверным результатам. Эти задачи являются частью процесса обработки и анализа данных группы и обычно подразумевают первоначальное изучение набора данных, используемого для определения и планирования необходимой предварительной обработки. Данная статья направлена на описание разных проблем и методов их решений при обработке данных для машинного обучения.

**Основной раздел.** Реальные данные собираются для последующей обработки из разных источников и процессов. Они могут содержать ошибки и повреждения, негативно влияющие на качество набора данных. Типичные проблемы с качеством данных:

— **Неполнота:** данные не содержат атрибутов, или в них пропущены значения.

— **Шум:** данные содержат ошибочные записи или выбросы.

— **Несогласованность:** данные содержат конфликтующие между собой записи или расхождения.

Качественные данные — это необходимое условие для создания качественных моделей прогнозирования. Чтобы избежать появления ситуации «мусор на входе, мусор на выходе» и повысить качество данных и, как следствие, эффективность модели, необходимо провести мониторинг работоспособности данных, как можно раньше обнаружить проблемы и решить, какие действия по предварительной обработке и очистке данных необходимы.

Стандартные методы мониторинга работоспособности данных:

- Количество записей.
- Количество атрибутов (или компонентов);
- Типы данных атрибута (номинальные, порядковые или непрерывные).
- Количество пропущенных значений.
- Правильность формата данных.

**1) Несогласованные записи данных** (проверьте допустимость диапазона значений).

При обнаружении проблем с данными необходимо выполнить обработку, которая зачастую включает очистку пропущенных значений, нормализацию данных, дискретизацию, обработку текста для удаления и/или замены внедренных символов, которые могут влиять на выравнивание данных, смешанные типы данных в общих полях и пр. Главные задачи предварительной обработки данных: **Очистка данных** — восполнение пропущенных значений, обнаружение и удаление искаженных данных и выбросов.

При работе с пропущенными значениями лучше сначала определить причину их появления в данных, что поможет решить проблему.

Методы обработки пропущенных значений:

- **Удаление:** удаление записей с пропущенными значениями.
- **Фиктивная подстановка** — замена пропущенных значений фиктивными, например, подстановка значения unknown (неизвестно) вместо категориальных или значения 0 вместо чисел.
- **Подстановка среднего значения:** пропущенные числовые данные можно заменить средним значением.
- **Подстановка часто используемого элемента:** пропущенные категориальные значения можно заменить наиболее часто используемым элементом.
- **Подстановка по регрессии:** использование регрессионного метода для замены пропущенных значений регрессионными.

**2) Преобразование данных** — нормализация данных для снижения измерений и искажений. Нормализация данных позволяет масштабировать числовые значения в указанном диапазоне. Ниже представлены распространенные методы нормализации данных.

— **Нормализация по методу минимакса:** линейное преобразование данных в диапазоне, например, от 0 до 1, где минимальное и максимальное масштабируемые значения соответствуют 0 и 1 соответственно.

— **Нормализация по Z-показателю:** масштабирование данных на основе среднего значения и стандартного отклонения: деление разницы между данными и средним значением на стандартное отклонение.

— **Десятичное масштабирование:** масштабирование данных путем удаления десятичного делителя значения атрибута.

**3) Уплотнение данных** — создание выборки данных или атрибутов для упрощения обработки данных.

Существуют различные методы, с помощью которых вы можете уменьшить размер данных для упрощения обработки данных. В зависимости от размера данных и домена вы можете применить такие методы:

1. **Выборка записей:** создание выборки записей данных и выбор репрезентативного подмножества из общего набора данных.

2. **Выборка атрибутов:** выбор в данных набора важнейших атрибутов.

3. **Агрегирование:** разделение данных на группы и хранение числовых значений для каждой группы.

**4) Дискретизация данных** — преобразование непрерывных атрибутов в категориальные. Данные можно дискретизировать, преобразовав непрерывные значения в номинальные атрибуты или интервалы. Это можно сделать несколькими способами.

1. **Группирование равной ширины:** разделение диапазона всех возможных значений атрибута в группы (N) одинакового размера с последующим присвоением значений, относящихся к ячейке с соответствующим номером.

2. **Группирование равной высоты:** разделение всех возможных значений атрибута в группы (N), содержащие одинаковое количество экземпляров, с последующим присвоением значений, относящихся к ячейке с соответствующим номером.

**5) Очистка текста** — удаление внедренных символов, которые могут нарушать выравнивание данных, например, внедренных символов табуляции в файле с разделителем-табуляцией, внедренных новых линий, которые могут разбивать записи, и пр.

3. **Текстовые поля в табличных данных** могут содержать символы, сбивающие выравнивание столбцов или границы записей (или и то и другое вместе). Например, табуляции, внедренные в файл с разделителем-табуляцией, могут сбить выравнивание столбцов, а внедренные символы новой строки могут разорвать линии записей. Неправильная кодировка текста приводит при его чтении или записи к потере информации, появлению нечитаемых символов, например, нуль-символов, и может также помешать разбору текста. Чтобы очистить текстовые поля, исправить выравнивание и извлечь структурированные текстовые данные из неструктурированных или полу-структурированных, могут потребоваться тщательные разбор и редактирование текста.

**Заключение.** Предварительная обработка данных важная часть в процессе работы с алгоритмами машинного обучения, помогая подготовить данные для обучения в дальнейшем получая намного более качественные результаты чем с данными которые были поставлены в чистом виде.

## ЛИТЕРАТУРА

1. microsoft.com [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/prepare-data>: Дата доступа: 19.09.2019.
2. kdnuggets.com [Электронный ресурс]. Режим доступа: <https://www.kdnuggets.com/2018/12/six-steps-master-machine-learning-data-preparation.html>: Дата доступа: 19.09.2019.
3. machinelearningmastery.com [Электронный ресурс]. Режим доступа: <https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/>: Дата доступа: 19.09.2019.



УДК 004.8

## ЗАДАЧИ, РЕШАЕМЫЕ С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

М.А. ШЕВЦОВ

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

Задачи, связанные с машинным обучением можно разделить на то какой вид прогнозирования или предсказания они будут совершать. В этой статье мы рассмотрим основные виды задач, решаемые с помощью машинного обучения.

**1) Задача классификации** (англ. **classification problem**) — имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

В математической статистике задачи классификации называются также задачами дискриминантного анализа. В машинном обучении задача классификации решается, в частности, с помощью методов искусственных нейронных сетей при постановке эксперимента в виде обучения с учителем.

Типы классов:

- Двух классовая классификация. Наиболее простой в техническом отношении случай, который служит основой для решения более сложных задач.
- Много классовая классификация. Когда число классов достигает многих тысяч, задача классификации становится существенно более трудной.
- Непересекающиеся классы.
- Пересекающиеся классы. Объект может относиться одновременно к нескольким классам.
- Нечёткие классы. Требуется определять степень принадлежности объекта каждому из классов, обычно это действительное число от 0 до 1.

**Основные задачи:**

- Определение принадлежности объекта к определенному классу.

**2) Кластерный анализ** (англ. **cluster analysis**) — процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в однородные группы. Задача кластеризации относится к статистической обработке, а также к широкому классу задач обучения без учителя.

Независимо от предмета изучения применение кластерного анализа предполагает следующие этапы:

- Отбор выборки для кластеризации. Подразумевается, что имеет смысл кластеризовать только количественные данные.
- Определение множества переменных, по которым будут оцениваться объекты в выборке, то есть признакового пространства.
- Вычисление значений той или иной меры сходства (или различия) между объектами.
- Применение метода кластерного анализа для создания групп сходных объектов.
- Проверка достоверности результатов кластерного решения.

**Основные задачи:**

- Разработка типологии или классификации.
- Исследование полезных концептуальных схем группирования объектов.
- Порождение гипотез на основе исследования данных.
- Проверка гипотез или исследования для определения, действительно ли типы (группы), выделенные тем или иным способом, присутствуют в имеющихся данных.

**3) Регрессионный анализ** (англ. **regression analysis**) — метод моделирования измеряемых данных и исследования их свойств. Данные состоят из пар значений зависимой переменной (переменной отклика) и независимой переменной (объясняющей переменной).

**Основные задачи:**

- Определение степени детерминированности вариации критериальной (зависимой) переменной предикторами (независимыми переменными).
- Предсказание значения зависимой переменной с помощью независимой(-ых).
- Определение вклада отдельных независимых переменных в вариацию зависимой.

Регрессионный анализ нельзя использовать для определения наличия связи между переменными, поскольку наличие такой связи и есть предпосылка для применения анализа.

**Заключение.** Каждый тип задач классифицируется по типу решаемых задач, сферы применения и алгоритмам, которые поддерживают решение данной проблемы, но важно помнить, что многие задачи разделены весьма условной линией и могут быть взаимозаменяемы в определённых условиях.

#### ЛИТЕРАТУРА

1. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. Множественная регрессия = Applied Regression Analysis. — 3-е изд. — М.: «Диалектика», 2007. — С. 912.
2. Шлезингер М., Главач В. Десять лекций по статистическому и структурному распознаванию. — Киев: Наукова думка, 2004.
3. Mitchell T. Machine Learning. — McGraw-Hill Science/Engineering/Math, 1997
4. Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell (1983), Machine Learning: An Artificial Intelligence Approach, Tioga Publishing Company.

УДК 004.021

## POSTMAN КАК СОВРЕМЕННЫЙ ИНСТРУМЕНТ ДЛЯ ТЕСТИРОВАНИЯ API

Н.О. ШЕРШНЕВ

(Представлено: канд. техн. наук, доц. А.Ф.ОСЬКИН)

В данной статье рассматривается современный способ тестирования программного обеспечения, а именно серверной части приложения (API – application programming interface).

**Введение.** Тестирование является неотъемлемой частью создания программного продукта. Основной целью тестирования является исследование программного обеспечения с целью получения информации о качестве продукта. Можно выделить следующие типы программных ошибок, далее приведены примеры для каждого из них:

- функциональные ошибки (не работает добавление товаров в корзину заказов, невозможно оставить комментарий к записи пользователя, не работает поиск по ключевым словам);
- визуальные ошибки (текст не помещается в границах обозначенной области, не отображается картинка, неверный размер или шрифт текста);
- логические ошибки (возможно указать дату рождения в будущем, оформить заказ, при этом не оставив контактные данные для связи);
- ошибки контента (орфографические или пунктуационные ошибки);
- ошибки удобства пользования (сброс заполненных полей формы при валидации данных, отсутствие возможности сохранить свой текущий прогресс, перегруженный интерфейс);
- ошибки безопасности (SQL-инъекции); [1]

Существует большое количество инструментов для тестирования программного обеспечения. Мы остановимся более подробно на тестировании API приложения по средствам Postman.

**Основной раздел.** Postman является одним из самых популярных инструментов, используемых в тестировании API. Данное приложение позволяет разработчикам легко создавать, обмениваться, тестировать и документировать API. Это достигается за счет того, что пользователи могут создавать и сохранять HTTP-запросы, а так же читать их ответы.

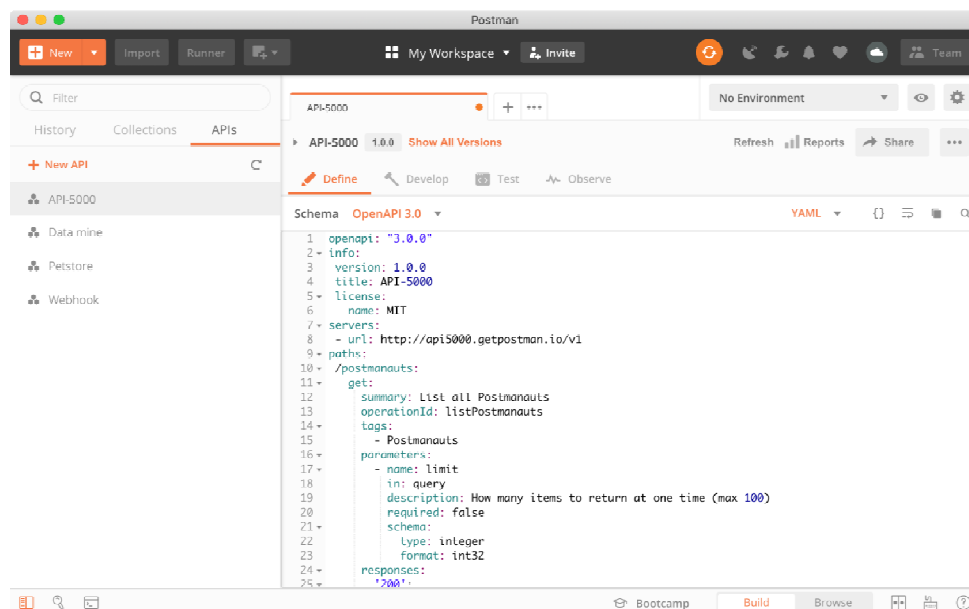


Рисунок 1. – Интерфейс приложения Postman

Postman обладает следующими преимуществами:

- доступность использования – для того, чтобы использовать Postman достаточно авторизоваться при помощи своей учетной записи, что упрощает доступ к файлам в любое время и в любом месте;

- использование коллекций – Postman позволяет пользователям создавать коллекции для вызовов API. Каждая коллекция может иметь внутри себя подразделы, каждый в свою очередь может содержать несколько запросов. Такая возможность помогает лучше организовать ваши тестовые инструкции;
- совместная работа – коллекции можно импортировать и экспортировать, что упрощает обмен файлами. Прямая ссылка так же может быть использована для обмена коллекциями;
- создание окружений – наличие нескольких окружений способствует меньшему повторению тестов, поскольку можно использовать одну и ту же коллекцию, но для разного окружения;
- создание тестов – для каждого вызова API можно добавить контрольные точки, такие как проверка на успешное состояние ответа HTTP, что помогает обеспечить покрытие тестами;
- автоматизированное тестирование – при помощи Collection Runner или Newman тесты можно запускать в несколько итераций, экономя время для повторяющихся тестов;
- отладка – консоль Postman помогает проверить, какие данные были получены, что облегчает отладку тестов. [2]

Для того чтобы выполнить GET запрос, нам нужно сделать несколько простых действий:

- 1) выбрать GET запрос из списка предложенных типов запросов;
- 2) в строку адреса нужно вставить URL адрес API, которое мы будем использовать для вызова, в нашем случае это <https://jsonplaceholder.typicode.com/users>;
- 3) нажимаем на кнопку send;
- 4) после отправки запроса мы должны увидеть успешный результат его выполнения, об этом свидетельствует статус 200 ОК;
- 5) ниже вы можете увидеть результат выполнения HTTP запроса (JSON файл, содержащий массив пользователей).

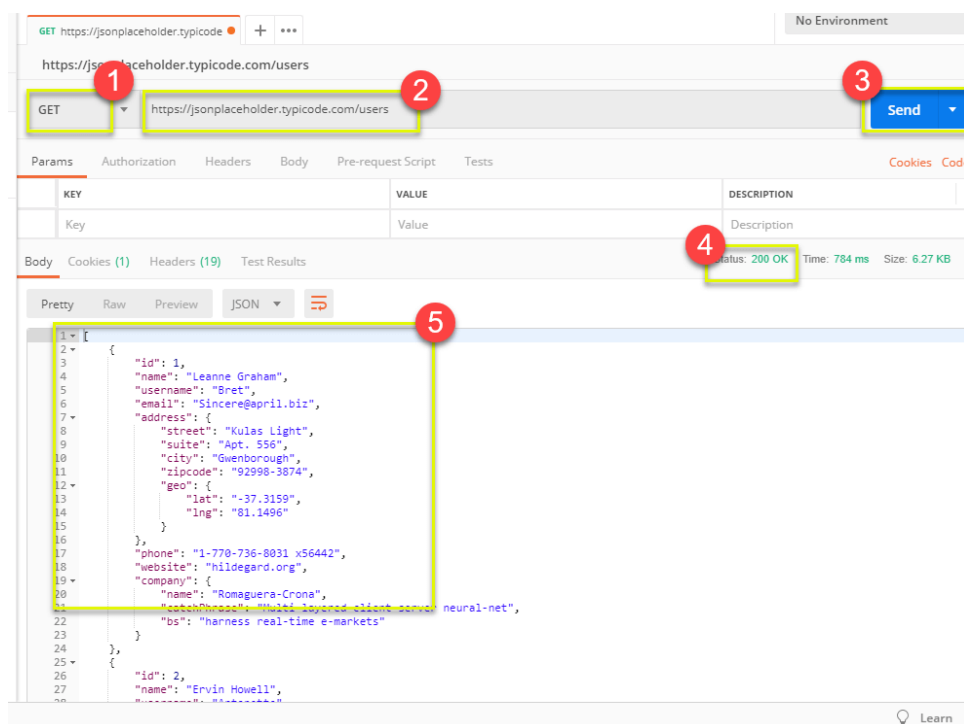


Рисунок 2. – Пример выполнения GET запроса в Postman

**Закключение.** В данной статье было рассмотрено приложение Postman, плюсы использования данного инструмента для тестирования API, а также пример выполнения HTTP запроса к API.

#### ЛИТЕРАТУРА

1. Википедия [Электронный ресурс] Тестирование программного обеспечения. – Режим доступа: [https://en.wikipedia.org/wiki/Тестирование\\_программного\\_обеспечения](https://en.wikipedia.org/wiki/Тестирование_программного_обеспечения). – Дата доступа: 25.09.19.
2. Guru99 [Электронный ресурс] Postman Tutorial for Beginners with API Testing Example. – Режим доступа: <https://guru99.com/postman-tutorial.html>. – Дата доступа: 25.09.19.

УДК 004.021

**СПОСОБЫ ИЗВЛЕЧЕНИЯ ДАННЫХ, ИСПОЛЬЗУЮЩИЕ ТЕХНОЛОГИЮ ВЕБ-СКРАПИНГА.  
DIFFBOT КАК ГОТОВОЕ РЕШЕНИЕ ДЛЯ ИЗВЛЕЧЕНИЯ ДАННЫХ****Н.О. ШЕРШНЕВ***(Представлено: канд. техн. наук, доц. А.Ф.ОСЬКИН)*

*В данной статье рассматривается подход к получению данных из сети Интернет Web-scraping. Приводится пример готового решения (Diffbot).*

**Введение.** Информация играет чрезвычайно важную роль в жизни человека. Ведь, как известно: «Кто владеет информацией, тот владеет миром». В процессе развития современного общества, все большие объемы несортированных данных хранятся в сети Интернет в открытом доступе. Поэтому при создании нового веб-ресурса перед разработчиком стоит выбор: использовать готовые данные из сети Интернет, или прикладывать усилия для создания собственного контента. Это обусловило создание и разработку технологий, которые бы позволяли собирать полезную информацию в интернете, анализировать ее и предоставлять в структурированном виде конечному пользователю.

**Основной раздел.** Такой подход к извлечению полезных данных называется веб-скрапингом, который подразумевает под собой создание такого программного обеспечения, которое позволило бы получать пользователю всю необходимую информацию с одного или нескольких интернет-ресурсов.

Существующие веб-скраперы имеют узконаправленную специализацию и зачастую создаются для конкретно веб-ресурса, что предполагает под собой большие человеческие усилия для автоматизации процессов получения и дальнейшего преобразования информации к структурированному виду.

Веб-скраперы имеют широкое применение в разных сферах жизни человека. При помощи веб-скраперов можно решить следующие задачи: мониторинг данных о погоде, сбор личных данных пользователей, поиск вакансий, работа с частными объявлениями по поиску жилья, отслеживание цен на товары в различных магазинах и т.д.

Среди готовых решений для скрапинга веб-сайтов стоит отметить следующие:

- веб-сервисы, которые работают через API (DiffBot, Embedly и др.).
- проекты с открытым кодом (Goose, Goutte, Morph, Scrapy и др.) [1].

Далее рассмотрим основные способы анализа и извлечения данных, использующие технологию веб-скрапинга:

1) **Copy-and-paste.** В некоторых случаях, лучшим решением для получения данных из сети Интернет является простое копирование необходимой информации пользователем. Так же этот способ полезен в том случае, когда веб-скрапер не может преодолеть защиту веб-сайта от машинной автоматизации;

2) **Text pattern matching.** Довольно простой, однако мощный подход к извлечению информации с веб-сервиса может быть основан на использовании регулярных выражений;

3) **HTTP programming.** Данный способ основывается на отправке http-запросов к удаленному веб-серверу, используя программирование сокетов;

4) **DOM parsing.** Программа, созданная на основе данного подхода, встраивается в полноценный веб-браузер, например, Internet Explorer или Mozilla Firefox. Система управления браузером анализирует веб-страницы в DOM-дереве, на основе которых программа получает необходимую информацию;

5) **Computer vision web-page analysis.** Существуют попытки создания такого программного обеспечения, которое использует машинное обучение и компьютерное зрение, идентифицирует и извлекает информацию с веб-страниц;

6) **HTML parsing.** Многие веб-сайты содержат коллекции страниц, которые были сгенерированы автоматически из таких источников, как базы данных. Данные обычно кодируются в похожие страницы с помощью общих скриптов и шаблонов. В процессе интеллектуального анализа данных, программа, которая определяет такие шаблоны, извлекает их содержимое и переводит в понятную форму называют оболочкой или wrapper; [2]

7) **Web-scraping.** Существует программное обеспечение, которое может быть использовано для настройки веб-скрапинг решений. Такое программное обеспечение может автоматически распознать структуру веб-страницы, что освобождает от написания веб-скрапинг кода; [2]

Среди готовых решений для веб-скрапинга стоит отметить DiffBot. DiffBot – это эффективное и революционное решение, которое предоставляет возможность автоматического извлечения структури-

рованных, нормализованных и точных данных из любой точки мира. DiffBot использует передовую технологию искусственного интеллекта и набор API для анализа веб-страниц и извлечения данных. Данное программное обеспечение снабжено всеобъемлющей диаграммой знаний, которая содержит точную и подробную информацию о различных объектах, найденных в Интернете, таких как люди, места, компании, организации, предприятия, продукты, статьи и обсуждения. Пользователи могут легко и точно запросить этот граф знаний, чтобы отобразить любые необходимые им данные. Кроме того, Diffbot определяет, как информация и объекты связаны друг с другом, облегчая пользователям понимание данных, предоставляемых программным обеспечением, и их использование для достижения своих конкретных целей. [4]



Рисунок. – Diffbot лучшее готовое решение для веб-скрапинга

Важно отметить следующие плюсы использования Diffbot:

- автоматическое определение страниц – при помощи API Analyze осуществляется автоматический поиск и извлечение всех продуктов, статей, обсуждений, видео или изображений при сканировании любого сайта;
- подробные данные о продуктах – Product API автоматически возвращает полную информацию о продукте, включая все данные о ценах, идентификаторы продуктов, марки и полные таблицы спецификации;
- чистый текст и HTML – статьи, обсуждения, описания продуктов и подписи к изображениям возвращаются в виде чистого текста и HTML;
- структурированный поиск – поиск структурированного контента на лету из любого сканирования.

**Заключение.** В данной статье рассмотрен Web-scraping как инструмент извлечения полезных данных из сети Интернет, основные способы анализа и обработки данных, назначение и применение в повседневной жизни человека. Приведен пример готового решения для веб-скрапинга Diffbot.

#### ЛИТЕРАТУРА

1. Википедия [Электронный ресурс] Web-scraping Режим доступа: [https://en.wikipedia.org/wiki/Web\\_scraping](https://en.wikipedia.org/wiki/Web_scraping). Дата доступа: 25.09.19.
2. Habrahabr [Электронный ресурс] Web-scraping с помощью Python. Режим доступа: <https://habrahabr.ru/post/280238/>. Дата доступа: 25.09.19.
3. Habrahabr [Электронный ресурс] Web-scraping при помощи Node.js. Режим доступа: <https://habrahabr.ru/post/301426/>. Дата доступа: 25.09.19.
4. Finances online [Электронный ресурс] Diffbot Review. Режим доступа: <https://reviews.financesonline.com/p/diffbot/#overview-benefits>. Дата доступа: 25.09.19.

УДК 004.771

## ПОДХОДЫ К РАЗРАБОТКЕ ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ ДЛЯ УДАЛЕННОГО УПРАВЛЕНИЯ КОМПЬЮТЕРОМ ПРИ ПОМОЩИ ANDROID УСТРОЙСТВА

*И.А. ШУМЯНЦЕВ**(Представлено: канд. техн. наук, доц. А.Ф.ОСЬКИН)*

*В статье представлен практический способ создания интерфейса приложения для удаленного управления компьютером при помощи android устройства. Проведён анализ технологий, наиболее подходящих, для разработки данного приложения. Прототип интерфейса реализован на основе Windows Presentation Foundation.*

**Введение.** На сегодняшний день, любое приложение должно быть основано на общеподдерживаемых стандартах разработки, а также обладать понятным для пользователя интерфейсом. Разрабатываемое приложение для удаленного управления компьютером при помощи android устройства должно позволять производить работу с компьютером посредством wi-fi или любой другой сети при помощи android устройства.

В данном проекте процесс создания интерфейса приложения служит для прямого взаимодействия приложения с сетью, сочетающей в себе всю необходимую информацию для вывода непосредственно в приложение. Интерфейс создаётся путем добавления необходимых компонентов и последующей их настройки. В интерфейсе любого приложения есть компоненты, обеспечивающие взаимодействие программы с пользователем.

**Выбор технологий разработки интерфейса.** Для разработки данного приложения выбран язык программирования C#.

C# – объектно-ориентированный язык программирования. Разработан в 1998-2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270 [1].

Для разработки Windows-версии нами была использована система для построения клиентских приложений Windows Presentation Foundation (WPF) – система для построения клиентских приложений Windows. Достоинства системы — высокая визуальная привлекательность. Графическая (презентационная) подсистема входит в состав .NET Framework (начиная с версии 3.0) и использует язык XAML [2].

Для разработки приложения была выбрана среда разработки Visual Studio 2017, т.к. она поддерживает синтаксис большинства языков программирования и множество плагинов для них. Кроме того, это одна из лучших сред для разработки десктопных приложений.


Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и VisualSourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Visual Studio построена на архитектуре, поддерживающей возможность использования встраиваемых дополнений – плагинов от сторонних разработчиков, что позволяет расширять возможности среды разработки [3].

**Проектирование интерфейса.** Расположение элементов пользовательского интерфейса не должно предоставлять затруднений при работе с приложением. Для создания удобного интерфейса необходимо придерживаться чёткого визуального разделения областей экрана.

Было принято решение о вынесении всех пунктов управления в так называемые Grid-блоки с последующим распределением в Text-блоки и Button-блоки. Таким образом, перед пользователем наглядно будет отображаться вся необходимая информация, а элементы управления приложением будут отображаться понятно и корректно.

При запуске программы на компьютере должно появиться окно, с помощью которого можно будет запускать, и останавливать работу сервера. Макет представлен на рисунке 1.

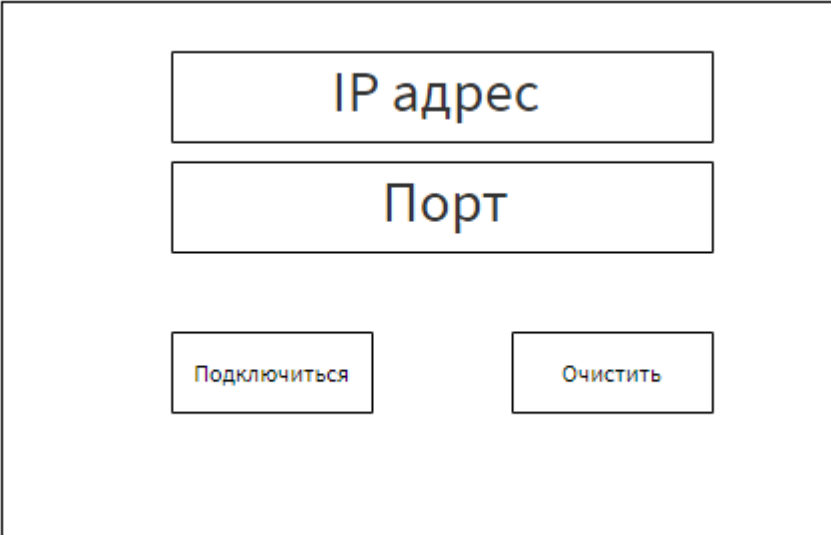


Макет серверного приложения. В центре экрана расположены три текстовых поля: «IP адрес», «Порт» и «Статус». Под полем «IP адрес» и полем «Порт» находятся две кнопки: «Старт» и «Отмена». Поле «Статус» расположено под кнопками.

**Рисунок 1. – Макет серверного приложения**

Поле «IP адрес» содержит в себе информацию о IP. Поле «Порт» содержит случайно сгенерированный порт. Поле «Статус» содержит в себе информацию об активности сервера. Кнопка «Старт» создает новые потоки, в которых реализованы функции для отправки и приема данных. Кнопка «Отмена» закрывает все запущенные потоки (кроме главного) и сокеты.

В клиентской части проекта есть два окна. В первом происходит ввод необходимой информации для подключения к серверу (рисунок 2), а во втором окне происходит само управление (рисунок 3).



Макет меню подключения к серверу. В центре экрана расположены три текстовых поля: «IP адрес», «Порт» и «Подключиться». Под полем «IP адрес» и полем «Порт» находятся две кнопки: «Подключиться» и «Очистить». Поле «Подключиться» расположено под кнопками.

**Рисунок 2. – Макет меню подключения к серверу**

В поле «IP адрес» необходимо ввести IP сервера, к которому нужно подключиться. В поле «Порт» необходимо ввести порт, заданный на сервере. При нажатии на кнопку «Очистить» происходит очистка вышеперечисленных полей. При нажатии на кнопку «Подключиться» на сервер отправляются данные о телефоне, а именно его IP адрес.

В это окно выводится изображение рабочего стола. Левая область экрана, которая занимает 10% от всей ширины экрана, осуществляет функции стандартных кнопок мыши. Управлять курсором можно при помощи перемещения пальца по любой части экрана.



90%	10%
Изображение	Левая кнопка мыши
	Правая кнопка мыши

Рисунок 3. – Макет окна управления

**Заключение.** Правильным подходом при разработке интерфейса приложения является использование современных технологий. Именно поэтому было принято решение использовать WPF. Это, в первую очередь, экономит время при разработке интерфейса, а также ресурсы на обработку данных. Это является очень актуальным, при большом количестве информации, которая подлежит постоянному изменению.

Рассмотрены технологии WPF и XAML, при взаимодействии, позволяют создать адаптивный интерфейс, любого уровня сложности, при этом сохраняя его функциональные возможности и привлекательный внешний вид. По итогам данной работы, была описана возможность их применения к построению интерфейса для удобной работы пользователя.

#### ЛИТЕРАТУРА

1. METANIT.COM Сайт о программировании [Электронный ресурс]. Режим доступа: <https://metanit.com/sharp/>. Дата доступа: 23.09.2019.
2. Microsoft.com. Xamarin [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/xamarin/>. Дата доступа: 23.09.2019.
3. Microsoft.com — свободной энциклопедии. Visual Studio [Электронный ресурс]. Режим доступа: <https://visualstudio.microsoft.com/ru/?rr=https%3A%2F%2Fwww.google.com%2F>. – Дата доступа: 23.09.2019.

УДК 004.771

**ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ УДАЛЕННОГО УПРАВЛЕНИЯ КОМПЬЮТЕРОМ  
ПРИ ПОМОЩИ ANDROID УСТРОЙСТВА****И.А. ШУМЯНЦЕВ***(Представлено: канд. техн. наук, доц. А.Ф.ОСЬКИН)*

*В статье рассматривается проектирование системы для работы с компьютером при помощи android устройства. Проведён анализ технологий, наиболее подходящих, для разработки данного приложения. Проведены исследования по актуальности разработки данного приложения.*

**Введение** Программы удаленного управления компьютером позволяют дистанционно управлять другим компьютером через Интернет или в локальной сети. Это удобно, когда необходимо помочь не очень опытному пользователю, например, родственнику или другу, плохо разбирающемуся в компьютере что-либо сделать на нем. Такие программы еще удобно использовать для удаленной работы, например, из дома для подключения в офис и, наоборот – для доступа к своему домашнему ПК, для системного администрирования целого парка компьютеров и серверов.

Дистанционное управление компьютером позволяет получить полный контроль над удаленной машиной, а также приложениями и файлами. Наиболее общие для многих программ удаленного управления ПК функции – файловый менеджер, голосовой или текстовый чат и, непосредственно, удаленное управление компьютером.

Технология удаленного подключения к ПК открывает широкий круг возможностей, как для корпоративных, так и для частных пользователей, которые хотят оставаться активными и мобильными и в то же время иметь оперативный доступ к рабочим и домашним компьютерам из любой точки мира.

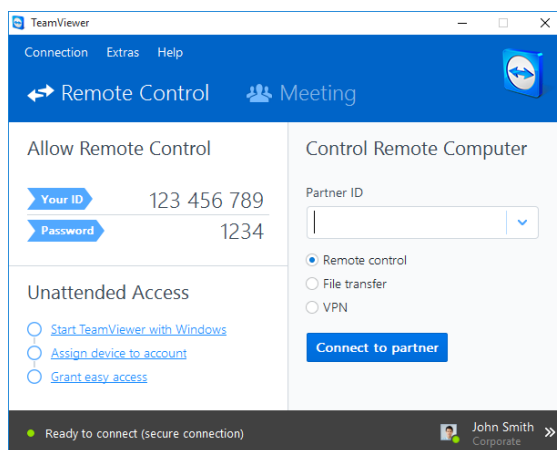
В наши дни удаленный доступ к компьютеру и преимущества, которые он предоставляет, давно вышел за рамки простого удобства. Теперь это необходимость, и бизнес все большего количества компаний зависит от технологий удаленного подключения к компьютерам. Удаленная техподдержка, системное администрирование, бизнес-конференции онлайн, дистанционное обучение – наиболее широкие сферы применения данной технологии.

При разработке данного приложения реализована передача изображения с монитора компьютера на экран телефона, также присутствует управление с телефона курсором и кнопками мыши.

**Анализ актуальности приложения.** Проведем сравнительный анализ программ удаленного управления, выделим их преимущества и недостатки.

TeamViewer

Одна из самых популярных программ для удаленного доступа, ее можно быстро скачать и установить или сразу запустить, без установки, с этим сможет справиться даже не очень опытный пользователь. При запуске программа отображает окно с ID и паролем для доступа к данному компьютеру, а также TeamViewer позволяет подключиться к другому компьютеру, задав его ID и пароль. Пример пользовательского интерфейса приведен на рисунке 1.



**Рисунок 1. – Пользовательский интерфейс приложения «TeamViewer»**

Преимущества: в программе доступно несколько основных режимов работы –это удаленное управление, передача файлов, чат, демонстрация своего рабочего стола. Программа позволяет настроить круглосуточный доступ к компьютеру, это будет удобно для системного администрирования. Скорость работы вполне достойная, есть версии для всех мобильных платформ, для различных операционных систем, что очень радует. Простой и вполне понятный интерфейс плюс ряд дополнительных утилит для расширения функционала программы, будут полезны для служб удаленной поддержки.

Недостатки: хоть программа и является бесплатной, но только для не коммерческого использования, а также при работе с ней более 5 минут возникает ряд трудностей, например, TV может заблокировать сеанс удаленного подключения, распознав его как коммерческое использование. Для круглосуточного удаленного доступа или администрирования нескольких компьютеров, компьютерной сети, придется платить за дополнительные модули программы. Стоимость программы высокая.

#### LiteManager

Простая, но довольно таки мощная по возможностям программа, состоит из двух частей, первая это Server который нужно установить или запустить на удаленном компьютере и Viewer, который позволяет управлять другим компьютером. Для работы программа требует немного больше навыков и опыта от управляющего, хотя работа сервером даже проще чем в TeamViewer, сервер можно один раз установить и больше не каких действий от пользователя не нужно, ID будет всегда постоянный, его даже можно задать самому вручную, что очень удобно для запоминания. Версия LiteManager Free является бесплатной для личного и коммерческого использования. Пример пользовательского интерфейса приведен на рисунке 2.

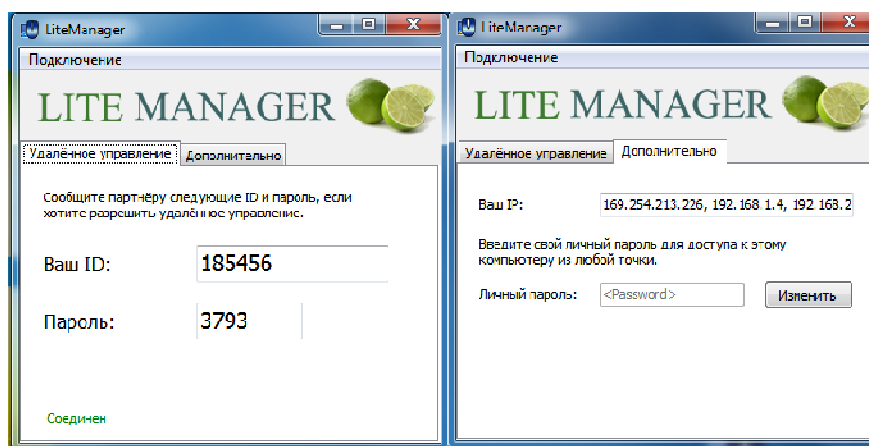


Рисунок 2. – Пользовательский интерфейс приложения «LiteManager»

Преимущества: в программе помимо основных режимов удаленного доступа: удаленного управления, передачи файлов, чата, диспетчера задач, редактора реестра, есть и уникальные функции, например, инвентаризация, запись экрана, удаленная установка. Программа бесплатна для использования на 30-ти компьютерах, ее можно использовать для круглосуточного доступа без каких-либо дополнительных модулей. Отсутствуют какие-либо ограничения по времени работы. Есть возможность настройки своего собственного ID сервера для настройки корпоративной службы поддержки. В программе нет каких-либо ограничений по времени работы и блокировок.

Недостатки: не хватает клиента под мобильные платформы или другие системы, есть ограничения на 30 компьютеров в бесплатной версии, для администрирования большего количества необходимо приобрести лицензию. Некоторые, специфичные режимы работы доступны только в Pro версии.

#### RAdmin

Одна из первых программ удаленного управления и известная в своем кругу, не мог ее не упомянуть, больше предназначенная для системного администрирования, основной акцент сделан на безопасности. Программа состоит из двух: компонент сервера и клиента. Требуется установки, неопытному пользователю будет не просто с ней разобраться, программа предназначена в основном для работы по IP адресу, что не совсем удобно для оказания технической поддержки через Интернет. Программа платная, но обладает бесплатным тестовым периодом. Пример пользовательского интерфейса приведен на рисунке 3.

Преимущества: у программы высокая скорость работы, особенно в хорошей сети, благодаря видео драйверу захвата рабочего стола, повышенной надежностью и безопасностью. Встроена технология Intel AMT, что позволяет подключаться к BIOS удаленного компьютера и настраивать его. Реализованы только основные режимы работы удаленное управление, передача файлов, чат и т.д.

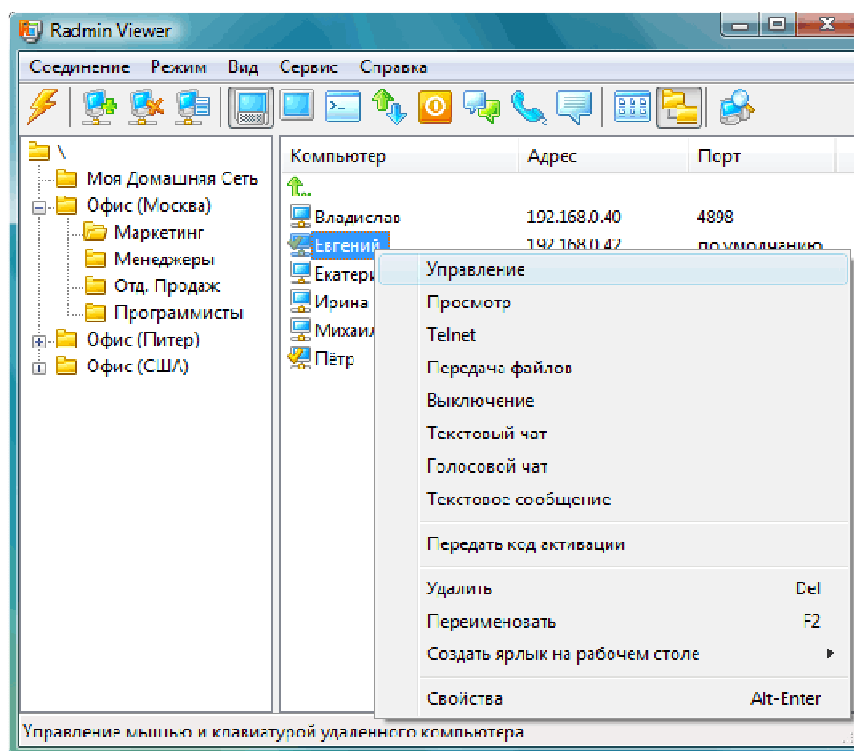


Рисунок 3. – Пользовательский интерфейс приложения «RAdmin»

Недостатки: почти нет возможности для работы без IP адреса, т.е. соединиться по ID. Отсутствует клиент для мобильных систем. Нет бесплатной версии, только тестовый период 30 дней. Для работы с программой необходимы навыки опытного пользователя. При подключении видео драйвер может отключать графическую оболочку Aero, иногда мигает экран.

Рассмотренные аналоги не в полной мере удовлетворяют оптимальному функционалу разрабатываемого приложения. Выгодное отличие разрабатываемого приложения от аналогов заключается в том, что в данном приложении нагрузка на сеть значительно ниже, чем у вышеперечисленных аналогов. Также стоит отметить, что данное приложение значительно меньше использует ресурсы компьютера (процессор, оперативная память).

**Средства решения задачи.** Необходимо выбрать протокол для сетевого взаимодействия между программой-сервером и программой-клиентом.

Сетевые сокеты делятся на два основных типа – TCP и UDP. Выбор сокета полностью зависит от функционала приложения и задач, которое оно будет решать.

TCP – это протокол, использующий принцип надежного соединения. Это означает, что вы устанавливаете соединение между двумя компьютерами, и затем пересылаете данные между ними подобно тому, как если бы вы записывали информацию в файл на одном компьютере, а на другом – считывали бы ее из того же файла. При этом соединение считается надежным и последовательным – то есть, вся информация, которую вы посылаете, гарантированно должна дойти до получателя в том же порядке, в каком была отправлена. Также TCP соединение можно считать непрерывным потоком данных – протокол сам заботится о разбивке данных на пакеты и пересылке их по сети [1].

UDP – один из ключевых элементов TCP/IP, набора сетевых протоколов для Интернета [2]. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. Протокол был разработан Дэвидом П. Ридом в 1980 году [3].

В данном случае использование TCP протокола является недопустимым, так как наша программа завязана на сетевом взаимодействии в реальном времени. Для многих аспектов программы, например, нажатых пользователем кнопок мыши или изменений в состоянии экрана, неважно, что происходило секунду назад, а важно только наиболее актуальное состояние компьютера. Сетевая часть данного приложения построена следующим образом: каждую итерацию цикла сервер посылает клиенту отчет в виде картинка, в то время как клиент посылает серверу определенные команды, которые должны работать без задержек (нажатие кнопок, перемещение курсора), и каждую итерацию сервер обрабатывает полученные данные, обновляя текущее состояние экрана и посылает скриншот в виде массива байт

клиенту, чтобы тот отрисовал пользователю новый кадр. Итак, в данном приложении, если пакет будет потерян при передаче по сети, все серверные процессы приостанавливаются на время, пока пакет не будет доставлен заново. На клиентской стороне картинка замирает, и на сервер также не будут доходить данные, так как сервер не может принимать новые пакеты. Когда потерянный пакет наконец доходит, в нем содержится уже устаревшая информация, которая уже является неактуальной. К тому же после этого приходят и все те пакеты, которые накопились в очереди за время ожидания, и их всех нужно обработать за одну итерацию цикла.

Исходя из выполненного анализа, было принято решение использовать сетевой протокол UDP.

Исходя из приведенных характеристик разработана функциональная структура программного обеспечения, которая состоит из следующих подсистем:

1. подсистема управления потоками – отвечает за корректное распределение потоков приложения. Предназначена для выполнения нескольких задач приложением одновременно;
2. подсистема управления сетевыми сокетами – отвечает за открытие сокетов для подключения клиентов. Благодаря данной системе происходит соединение между клиентом и сервером;
3. подсистема управления передачи данных – отвечает за передачу данных между клиентом и сервером;
4. подсистема управления приема данных – отвечает за прием данных между клиентом и сервером;
5. подсистема управления подключенными клиентами – отвечает за управления подключенными клиентами;
6. подсистема вывода информации – служит для отображения информации о сервере.

Функциональная структура системы представлена на рисунке 4.

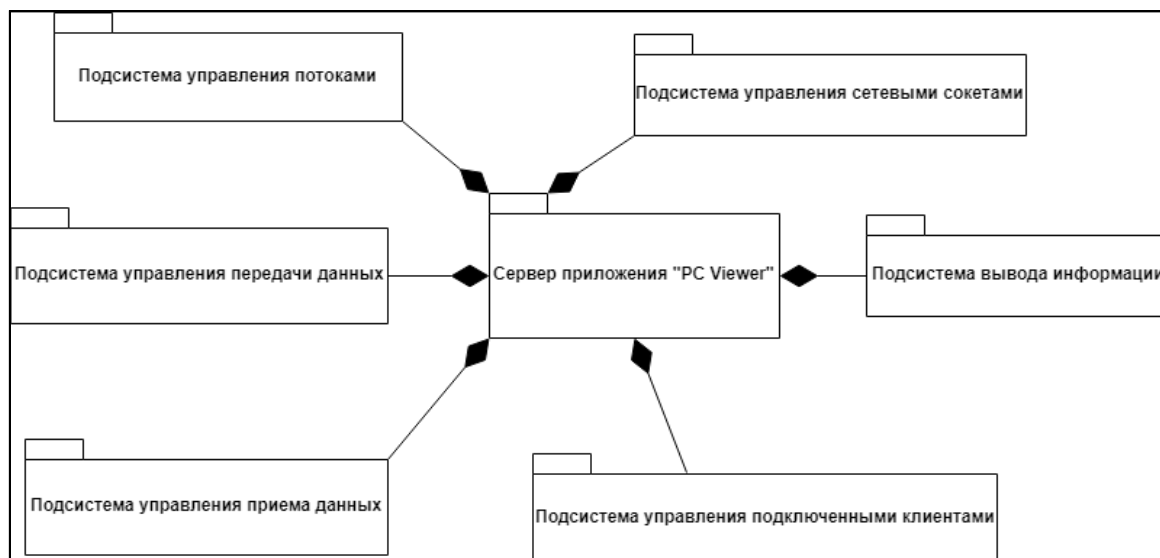


Рисунок 4. – Функциональная структура приложения

**Заключение.** В ходе данного исследования было спроектировано программное обеспечение наиболее подходящая по архитектурным особенностям под основу для разработки сетевого приложения для управления персональным компьютером при помощи мобильного устройства. При этом разработанная архитектура, является универсальной для любого современного языка программирования, а также оставляет возможности для реализации необходимого функционала.

#### ЛИТЕРАТУРА

1. OPEN.NET. TCP/IP [Электронный ресурс]. – Режим доступа: <https://www.opennet.ru/docs/RUS/tcpip/>. – Дата доступа: 25.09.2019.
2. Vanderboot.com. UDP [Электронный ресурс] – Режим доступа: <http://www.vanderboot.ru/tcp-ip/tcp-udp.php>. – Дата доступа: 25.09.2019.
3. Материал из «Professor Web» – электронные данные. Протокол UDP [Электронный ресурс]. – Режим доступа: [https://professorweb.ru/my/csharp/web/level4/4\\_5.php](https://professorweb.ru/my/csharp/web/level4/4_5.php). – Дата доступа: 25.09.2019.

УДК 004.02

## МЕТОД ОБНАРУЖЕНИЯ АНОМАЛИЙ В ПОТОКАХ ДАННЫХ СЕТЕВОГО ТРАФИКА

**М.Ю. МАКАРЫЧЕВ, В.А. МАКАРЫЧЕВА**  
(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

*В данной статье рассматривается метод обнаружения сетевых аномалий с применением статистических алгоритмов над вейвлет-коэффициентами (аппроксимаций и детализаций), полученных путем вейвлет-анализа из последовательности коэффициентов сетевого трафика.*

**Введение.** Характеристики трафика, проходящего по магистральным сетям связи, содержат в себе информацию о состоянии сетевого оборудования и подсетей, примыкающих к основной магистрали. Аномалии, зафиксированные в этом трафике, подлежат обязательной обработке. Они могут свидетельствовать о сетевых атаках, таких как DDoS (Distributed Denial of Service), отсутствии связи в некоторых сегментах сети, неисправности оборудования и других неполадках. Таким образом, своевременное обнаружение аномалий может гарантировать стабильную работу сети. В статье рассматривается метод обнаружения аномалий сетевого трафика, относящийся к методам кратномасштабного анализа.

**Основной раздел.** Для обнаружения аномалий предлагается использовать вейвлет-преобразование сигнальной кривой, отображающей зависимость «трафик-время». Одним из преимуществ вейвлет-преобразования является то, что оно дает возможность проанализировать сигнал в частотно-временной области и позволяет исследовать аномальный процесс на фоне остальных компонент [1].

Рассмотрим обнаружение аномалий сетевого трафика на основе дискретного вейвлет-преобразования с применением статистических критериев. Для адаптации этого способа к анализу трафика в реальном времени используется техника трех скользящих окон  $W_1$ ,  $W_2$ ,  $W_3$ . Первое окно является окном сравнения, второе и третье – окнами обнаружения. Пусть размер каждого окна будет  $\omega_1$ ,  $\omega_2$  и  $\omega_3$  выбранных временных единиц соответственно, причем  $\omega_1 > \omega_2 > \omega_3$ . Тогда в произвольный момент времени  $t$  начало окна  $W_3$  будет находиться в точке  $t$ , в нем будут содержаться  $\omega_3$  значений трафика от  $t - \omega_3$  до  $t$ , в окне  $W_2 - \omega_2$  значений от  $t - \omega_3 - \omega_2$  до  $t - \omega_3$ , а в окне  $W_1 - \omega_1$  значений от  $t - \omega_3 - \omega_2 - \omega_1$  до  $t - \omega_3 - \omega_2$ .

Выполнив быстрое вейвлет-преобразование для выборок внутри каждого из окон в каждый момент времени  $t_j$ , будет вычисляться на некотором масштабном уровне  $j$  набор коэффициентов для окна  $W_1$  – аппроксимации  $\{a_{1x}, a_{2x}, \dots, a_{nx}\}_{t,j}$  и детализации  $\{d_{1x}, d_{2x}, \dots, d_{nx}\}_{t,j}$ , для окна  $W_2$  – аппроксимации  $\{a_{1y}, a_{2y}, \dots, a_{my}\}_{t,j}$  и детализации  $\{d_{1y}, d_{2y}, \dots, d_{my}\}_{t,j}$  и для окна  $W_3$  – аппроксимации  $\{a_{1z}, a_{2z}, \dots, a_{kz}\}_{t,j}$  и детализации  $\{d_{1z}, d_{2z}, \dots, d_{kz}\}_{t,j}$ . Причем количество коэффициентов  $n$  на уровне  $j$  в окне  $W_1$  будет определяться выражением  $n = \frac{\omega_1}{2^j}$ , в окне  $W_2$  – выражением  $m = \frac{\omega_2}{2^j}$ , а в окне  $W_3$  – выражением  $k = \frac{\omega_3}{2^j}$ .

Эти коэффициенты будут проверяться по статическим критериям, и на основе принятия или отклонения статистических гипотез будет выноситься решение о кардинальном различии в анализируемых параметрах между окнами  $W_1$ ,  $W_2$  и  $W_3$ , а следовательно, о наличии аномалии или же наоборот – их отсутствии.

Анализ статистических характеристик коэффициентов аппроксимации и детализации показывает, что плотность распределения вероятностей мгновенных значений этих коэффициентов хорошо описывается нормальным распределением [2]. Для обнаружения аномалий, выражающихся в изменении дисперсии, предлагается использовать критерий Бартлетта, а для обнаружения величины среднего значения – критерий Кохрена-Кокса.

Критерий Бартлетта предложен для обнаружения изменений в дисперсиях выборок окон  $W_1$ ,  $W_2$  и  $W_3$ . В каждый момент времени (положении окон)  $t$  на масштабном уровне  $j$  выдвигаются две статистические гипотезы о равенстве дисперсий трех выборок  $\{d_{1x}, d_{2x}, \dots, d_{nx}\}_{t,j}$ ,  $\{d_{1y}, d_{2y}, \dots, d_{my}\}_{t,j}$  и  $\{d_{1z}, d_{2z}, \dots, d_{kz}\}_{t,j}$ : нулевая  $H_0: \sigma_{1,t,j}^2 = \sigma_{2,t,j}^2 = \sigma_{3,t,j}^2$  и альтернативная  $H_1: \sigma_{1,t,j}^2 \neq \sigma_{2,t,j}^2 \neq \sigma_{3,t,j}^2$ .

Алгоритм обнаружения выбросов на основе анализа аномального изменения дисперсий записывается как  $\chi_{t,j}^2 = M_{t,j} \left[ 1 + \frac{1}{6} \left( \frac{1}{n-1} + \frac{1}{m-1} + \frac{1}{k-1} \right) - \frac{1}{N} \right]^{-1}$ . Введем обозначения:

$$M_{t,j} = N \ln \left[ \frac{1}{N} \left( (n-1)S_{1,t,j}^2 + (m-1)S_{2,t,j}^2 + (k-1)S_{3,t,j}^2 \right) \right] - \left[ (n-1) \ln S_{1,t,j}^2 + (m-1) \ln S_{2,t,j}^2 + (k-1) \ln S_{3,t,j}^2 \right]$$

$$N = n + m + k - 3$$

$S_{1,t,j}^2 = \frac{1}{n-1} \sum_{i=1}^n (d_{ix} - \bar{d}_x)^2$  – выборочная дисперсия выборки последовательности деталей на масштабном уровне  $j$  в окне  $W_1$ .

$S_{2,t,j}^2 = \frac{1}{m-1} \sum_{i=1}^m (d_{iy} - \bar{d}_y)^2$  – выборочная дисперсия выборки последовательности деталей на масштабном уровне  $j$  в окне  $W_2$ .

$S_{3,t,j}^2 = \frac{1}{k-1} \sum_{i=1}^k (d_{iz} - \bar{d}_z)^2$  – выборочная дисперсия выборки последовательности деталей на масштабном уровне  $j$  в окне  $W_3$ .

$\bar{d}_x = \frac{1}{n} \sum_{i=1}^n d_{ix}$  – выборочное среднее выборки последовательности деталей на масштабном уровне  $j$  в окне  $W_1$ .

$\bar{d}_y = \frac{1}{m} \sum_{i=1}^m d_{iy}$  – выборочное среднее выборки последовательности деталей на масштабном уровне  $j$  в окне  $W_2$ .

$\bar{d}_z = \frac{1}{k} \sum_{i=1}^k d_{iz}$  – выборочное среднее выборки последовательности деталей на масштабном уровне  $j$  в окне  $W_3$ .

Нулевая гипотеза опровергается в пользу альтернативной, в случае если  $\chi_{t,j}^2 > \chi_{\alpha,2}^2$ , где  $\chi_{\alpha,2}^2$  –  $\alpha$ -квантиль распределения хи-квадрат с двумя степенями свободы.

Для обнаружения изменений среднего значения выборок аппроксимаций  $\{a_{1x}, a_{2x}, \dots, a_{nx}\}_{t,j}$ ,  $\{a_{1y}, a_{2y}, \dots, a_{my}\}_{t,j}$  и  $\{a_{1z}, a_{2z}, \dots, a_{kz}\}_{t,j}$  предложен критерий Кохрена-Кокса. Данный критерий можно применять только для двух выборок, поэтому его можно применить только для окон  $W_1$  и  $W_2$ , т.к. критерий предлагается в [2] для обнаружения долговременных низкочастотных аномалий. Далее будет представлен алгоритм обнаружения выбросов на основе анализа аномального изменения среднего значения выборки на примере окон  $W_1$  и  $W_2$ .

Статистикой критерия является  $Y = \frac{1}{S_{t,j}} \left( \frac{1}{n} \sum_{i=1}^n a_{ix} - \frac{1}{m} \sum_{i=1}^m a_{iy} \right)$ . Введем, как и прежде, обозначения:

$S_{1,t,j}^2 = \frac{1}{n-1} \sum_{i=1}^n (a_{ix} - \bar{a}_x)^2$  – выборочная дисперсия выборки последовательности аппроксимаций на масштабном уровне  $j$  в окне  $W_1$ .

$S_{2,t,j}^2 = \frac{1}{m-1} \sum_{i=1}^m (a_{iy} - \bar{a}_y)^2$  – выборочная дисперсия выборки последовательности аппроксимаций на масштабном уровне  $j$  в окне  $W_2$ .

$$S_{t,j}^2 = \frac{S_{1,t,j}^2}{n} + \frac{S_{2,t,j}^2}{m}$$

– суммарная взвешенная дисперсия выборок аппроксимаций для окон  $W_1$  и  $W_2$ .

$\bar{a}_x = \frac{1}{n} \sum_{i=1}^n a_{ix}$  и  $\bar{a}_y = \frac{1}{m} \sum_{i=1}^m a_{iy}$  – выборочные средние выборок последовательности аппроксимаций на масштабном уровне  $j$  в окне  $W_1$  и  $W_2$  соответственно.

С учетом введенных обозначений статистика сводится к виду  $Y = \frac{1}{S_{t,j}} (\bar{a}_y - \bar{a}_x)$ .

Критические (пороговые) значения статистики вычисляются по формуле:  $t'_\alpha = \frac{f_1 t_\alpha(\nu_1) + f_2 t_\alpha(\nu_2)}{f_1 + f_2}$ ,

где  $f_1 = \frac{S_{1,t,j}^2}{n}$ ,  $f_2 = \frac{S_{2,t,j}^2}{m}$ ;  $t_\alpha(\nu)$  –  $\alpha$ -квантиль распределения Стьюдента с  $\nu$  степенями свободы ( $\nu_1 = n - 1$  и  $\nu_2 = m - 1$ ).

Для каждого статистического критерия используются два порога, исходя из уровня значимости:  $\alpha = 0.05$  и  $\alpha = 0.01$ . Превышение верхнего порога на каком-либо уровне вейвлет-разложения означает наличие аномалии. В случае превышения нижнего порога, производится дальнейшая декомпозиция по следующему уровню разложения, и для коэффициентов этого уровня опять будут проверяться статистические критерии.

Техника применения трех (а не двух) скользящих окон обоснована тем, что результат обнаружения зависит от размера окна обнаружения и длительности аномалии [3]. Размеры окон сравнения и обнаружения предполагается устанавливать экспериментально.

**Заключение.** Рассмотренный метод основывается на вейвлет-представлении временного ряда сетевого трафика и статистических алгоритмах обнаружения аномалий с двумя порогами. Метод может быть применен для класса задач обнаружения аномалий в компьютерных и телекоммуникационных сетях.

#### ЛИТЕРАТУРА

1. Максименко, Г.А. Метод обнаружения аномалий потоков данных в сетях / Г.А. Максименко // Системи обробки інформації: сб. науч. ст. / Харьковский Нац. ун-т Воздушных Сил им. И. Кожедуба; С.В. Кавун, Г.А. Кучук (отв. за выпуск) – Харьков, 2009. – выпуск 7 (81) – С. 33–37.
2. Шелухин, О.И. Обнаружение вторжений в компьютерные сети. Сетевые аномалии / О.И. Шелухин, Д.Ж. Сакалема, А.С. Филинова. – Горячая линия – телеком, 2013. – 220 с.
3. Шелухин, О.И. Сравнительный анализ характеристик обнаружения аномалий трафика методами кратномасштабного анализа / О.И. Шелухин, А.В. Панкрушин // Т-Comm Телекоммуникации и транспорт. – 2014. – № 6. – С. 65–70.



УДК 004.492.3

**ПРОЕКТИРОВАНИЕ СЕРВИСА ОБНАРУЖЕНИЯ РАСПРЕДЕЛЕННЫХ СЕТЕВЫХ АТАК ТИПА «ОТКАЗ В ОБСЛУЖИВАНИИ»****М.Ю. МАКАРЫЧЕВ, В.А. МАКАРЫЧЕВА**  
(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

*В данной статье рассматривается проект сервиса обнаружения DDoS-атак (ICMP-flood, UDP-flood, TCP-SYN-flood). Выделены основные подсистемы сервиса и установлено взаимодействие между ними.*

**Введение.** При обеспечении защиты сетевых ресурсов, основной задачей является своевременное обнаружение состояний сети, приводящих к частичной или полной потере ее работоспособности, искажению, уничтожению или утечке информации. Оперативное обнаружение таких состояний позволит устранить их причину и предотвратить возможные последствия угроз информационной безопасности. Актуальность темы подтверждается сложностью обнаружения атак типа «отказ в обслуживании», т.е. DoS (Denial of Service) или DDoS-атак (Distributed Denial of Service). Во-первых, они не направлены на уязвимости, которые могут быть исправлены, во-вторых, каждый отдельный пакет, отправляемый на компьютер жертвы, является вполне легитимным, в-третьих, такие атаки носят продолжительный характер – от нескольких часов до нескольких дней [1].

**Основной раздел.** Сервис обнаружения DDoS-атак представляет собой программный компонент, включающий в себя несколько подсистем:

- Подсистема настройки сервиса;
- Подсистема захвата сетевых пакетов;
- Подсистема математического анализа трафика;
- Подсистема анализа сетевых пакетов;
- Подсистема уведомлений.

Далее будет представлено подробное описание каждой подсистемы.

Подсистема настройки сервиса настраивает приложение путем считывания настроек из файла конфигурации. Подсистема следит за изменением настроек в этом файле, и если они изменены, то переконфигурирует приложение прямо во время его работы. Подсистема настройки управляет следующими атрибутами:

- список сетевых интерфейсов, с которых будет происходить захват трафика;
- максимальные пороги пакетов в секунду для фильтров анализа трафика;
- настройки уведомлений такие, как исходящий адрес электронной почты с логином и паролем, адрес SMTP-сервера и список электронных адресов, которым будут рассылаться уведомления об атаках;
- настройки расположения папок, в которых хранятся сетевые дампы и лог-файлы.

Подсистема захвата сетевых пакетов отвечает за прием сетевых пакетов и их передачу другим подсистемам. Подсистема может работать в многозадачном режиме, таким образом обслуживая несколько сетевых интерфейсов. Данная подсистема создает отдельный поток для каждого сетевого интерфейса. В этом потоке регистрируется слушатель захвата пакетов. Слушатель использует фильтр пакетов, чтобы принимать только входящий трафик, адресованный данному сетевому интерфейсу. Далее запускается цикл захвата пакетов. Принятые пакеты передаются в буфер пакетов, который используется анализаторами.

Подсистема математического анализа трафика собирает и хранит данные, которые передает подсистема захвата сетевых пакетов. Эти данные обрабатываются определенными математическими методами, которые, в свою очередь, дают представление об аномалиях входящего трафика [2]. Если подсистема математического анализа обнаружила атаку, то сразу же вступает в работу подсистема анализа сетевых пакетов.

Подсистема анализа сетевых пакетов проверяет заголовки входящих сетевых пакетов на предмет установки определенных флагов, используемых сетевыми протоколами, и подводит статистику их присутствия. Данная подсистема срабатывает только тогда, когда подсистема математического анализа обнаружила атаку. Подсистема принимает окончательное решение: была атака или нет. Чтобы выяснить причину возникновения аномалий во входящем сетевом трафике и извлечь подробную информацию об атаке, если та имела место, подсистема использует буфер сетевых пакетов. Буфер хранит сетевые пакеты, захваченные за определенный промежуток времени. Информация об атаке извлекается из заголовков сетевых пакетов. По заголовкам подсистема может определить три наиболее распространенных типа DDoS-атак на втором и третьем уровнях модели OSI:

– ICMP-flood – один из самых опасных видов DDoS-атак, использующий ICMP-сообщения для перегрузки сетевого канала атакуемого. У компьютера-жертвы после такой атаки произойдет отказ в обслуживании практически со стопроцентной вероятностью. Протокол межсетевых управляющих сообщений (ICMP) используется в первую очередь для передачи сообщений об ошибках и не используется для передачи данных. ICMP-пакеты могут сопровождать TCP-пакеты при соединении с сервером;

– UDP-flood – является аналогом ICMP-flood, где вместо ICMP-пакетов используются UDP-пакеты. Эта атака использует бессеансовый режим протокола UDP и заключается в отправке множества UDP-пакетов (как правило, большого объема) на определенные или случайные номера портов удаленного компьютера, который для каждого полученного пакета должен определить соответствующее приложение, убедиться в отсутствии его активности и отправить ответное ICMP-сообщение «адресат недоступен». В итоге атакуемая система окажется перегруженной: в протоколе UDP механизм предотвращения перегрузок отсутствует, поэтому после начала атаки паразитный трафик быстро захватит всю доступную полосу пропускания, и полезному трафику останется лишь малая ее часть;

– TCP-SYN-flood – заключается в отправке большого количества TCP-пакетов с установленным флагом SYN в достаточно короткий срок. Согласно процессу «трехкратного рукопожатия» TCP, клиент посылает TCP-пакет с установленным флагом SYN (synchronize). В ответ на него сервер должен послать клиенту TCP-пакет с установленными флагами SYN и ACK (acknowledges). После этого клиент должен ответить пакетом с флагом ACK, после чего соединение считается установленным. Принцип атаки заключается в том, что злоумышленник, посылая TCP-пакеты с установленным флагом SYN, переполняет на сервере очередь на подключения. При этом он подделывает заголовок пакета таким образом, что ответный TCP-пакет с установленными флагами SYN и ACK отправляется на несуществующий адрес. В очереди подключений появляются так называемые полуоткрытые соединения, ожидающие подтверждения от клиента. По истечении определенного тайм-аута эти подключения отбрасываются. Задача злоумышленника заключается в том, чтобы поддерживать очередь подключений заполненной таким образом, чтобы не допустить новых подключений. Из-за этого клиенты, не являющиеся злоумышленниками, не могут установить связь, либо устанавливают ее с существенными задержками.

Подсистема уведомлений рассылает по электронной почте и через мессенджеры, такие как Telegram, Viber, WhatsApp, письма с сообщением об атаке и сохраняет сетевой дамп, который можно просмотреть с помощью Wireshark [3]. Подсистема срабатывает только тогда, когда подсистема анализа пакетов подтвердила атаку. Важно понимать, что проектируемая система не защищает от DDoS-атак, а только обнаруживает и уведомляет о них администраторов безопасности или других специалистов.

Предполагается, что управление сервисом будет производиться сторонним программным обеспечением, которое не будет рассматриваться в рамках данной статьи.

**Заключение.** В результате проектирования предлагается разработать сервис обнаружения сетевых атак. Сервис может работать с несколькими сетевыми интерфейсами одновременно. С каждого сетевого интерфейса происходит захват входящих сетевых пакетов. Захваченные пакеты интерпретируются как отсчеты дискретного сигнала или временного ряда. Временной ряд подвергается какому-либо математическому методу, который определяет присутствие аномалий, т.е. сетевых атак. Если с помощью математического анализа удалось обнаружить атаку, то проводится анализ сетевых пакетов, которые были захвачены. Анализ захваченных пакетов подтверждает или отклоняет гипотезу присутствия атаки. С помощью анализа пакетов устанавливается конкретный тип атаки (если она имела место), сохраняется дамп сетевых пакетов и отправляются уведомления об атаке администраторам сети.

#### ЛИТЕРАТУРА

1. Шелухин, О.И. Обнаружение вторжений в компьютерные сети. Сетевые аномалии / О.И. Шелухин, Д.Ж. Сакалема, А.С. Филинова. – Горячая линия – телеком, 2013. – 220 с.
2. Комар, М.П. Система анализа сетевого трафика для обнаружения компьютерных атак / М.П. Комар // Вестник Брестского государственного технического университета. Серия «Физика, математика, и информатика». – 2010. – № 5. – С. 14–16.
3. Wireshark User's Guide // Wireshark [Электронный ресурс]. – Режим доступа: [https://www.wireshark.org/docs/wsug\\_html\\_chunked/](https://www.wireshark.org/docs/wsug_html_chunked/). – Дата доступа: 20.09.2019.

УДК 004.93'1

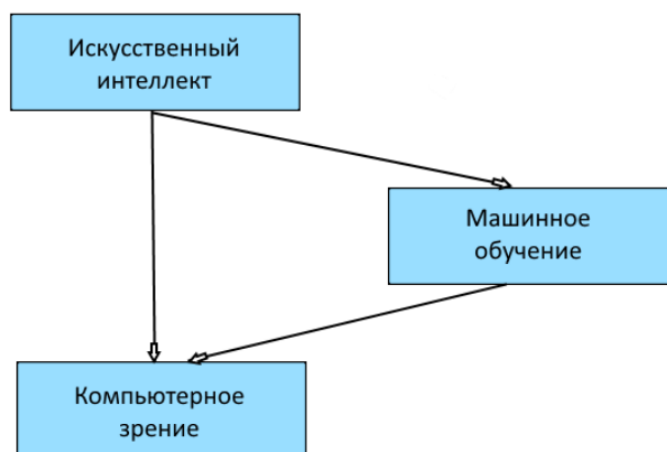
**ВВЕДЕНИЕ В КОМПЬЮТЕРНОЕ ЗРЕНИЕ****С.Н. МАЛАШОНОК***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассматривается определение «компьютерное зрение», описываются основной принцип действия, сложности реализации и основные решаемые задачи.*

Компьютерное зрение – это научное направление в области искусственного интеллекта, целью которого является разработка методов, способных находить, отслеживать, классифицировать и идентифицировать объекты, извлекая данные из изображений или видео и анализируя полученную информацию без участия (полного или частичного) человека [1].

Проблема компьютерного зрения кажется простой, потому что ее тривиально решают люди, даже самые маленькие дети. Тем не менее, это в значительной степени остается нерешенной проблемой, основанной как на ограниченном понимании биологического зрения, так и из-за сложности восприятия изображений в динамическом и почти бесконечно изменяющемся физическом мире.

Компьютерное зрение – междисциплинарная область, которая включает в себя искусственный интеллект и машинное обучение и может использовать специализированные методы и общие алгоритмы обучения. Взаимосвязь искусственного интеллекта и компьютерного зрения можно представить как показано на рисунке 1.



**Рисунок 1. – Взаимосвязь искусственного интеллекта и компьютерного зрения**

Одна конкретная проблема в компьютерном зрении может быть легко решена с помощью статистического метода, созданного вручную, тогда как другая может потребовать большого и сложного взаимодействия обобщенных алгоритмов машинного обучения.

Цель компьютерного зрения – понять содержание цифровых изображений. Как правило, это включает в себя разработку методов, которые пытаются воспроизвести способность человеческого зрения.

Понимание содержания цифровых изображений может включать извлечение описания из изображения, которое может быть объектом, текстовым описанием, трехмерной моделью и т.д. [2]

Компьютерное зрение кажется легким потому, что оно легко для человека. Однако научить компьютеры «видеть» очень сложно. Одна из причин заключается в том, что мы не совсем понимаем, как работает человеческое зрение. Изучение биологического зрения требует понимания органов восприятия, таких как глаза, а также интерпретации восприятия в мозге. Другая причина заключается в сложности визуального мира. Один и тот же объект может быть рассмотрен с любой стороны, в любых условиях освещения, с любым типом окклюзии от других объектов и так далее. Настоящая система видения должна быть способна «видеть» в любом из бесконечного числа сцен и при этом извлекать что-то значимое.

К основным задачам компьютерного зрения относятся:

– распознавание – определить содержат ли видеоданные некоторый характерный объект, особенность или активность. Эта задача может быть легко решена человеком, но до сих пор не решена удовлетворительно в компьютерном зрении в общем случае: случайные объекты в случайных ситуациях;

- идентификация – распознавание индивидуального экземпляра объекта, принадлежащего к какому-либо классу;
- обнаружение – проверка на наличие определенного условия;
- распознавание текста – нахождение изображений в наборе изображений, которые имеют определенное содержание;
- оценка движения – набор задач, связанных с оценкой движения, в которых последовательность изображений (видеоданные) обрабатываются для нахождения оценки скорости каждой точки изображения;
- восстановление изображений – удаление различного шума и размытости движущихся объектов с изображений;
- сегментация изображений – выделение на изображениях структур определенного вида;
- чтение штрих-кодов [1].

Компьютерное зрение позволяет автоматизировать процесс обработки большого объема данных с изображений или видеокамер с последующим распознаванием и классификацией объектов. Это в значительной степени упрощает работу человека и в некоторых случаях может полностью заменить его.

Вышеизложенные данные способствуют разработке мобильного приложения для учета и планирования расходов посредством сканирования чеков. Актуальность данного приложения заключается в введении статистики по различным видам и категориям товаров с целью контроля и оптимизации финансовых затрат.

#### ЛИТЕРАТУРА

1. Машинное зрение. Что это и как им пользоваться? [Электронный ресурс]. – Режим доступа <https://habr.com/ru/post/350918>. – Дата доступа 20.09.2019.
2. A Gentle Introduction to Computer Vision [«ktrnhjyysq htcehc/ – Нt;bv ljcnegf https^||machinelearningmastery/com|what-is-computer-vision/ – Lfnf ljcnegf 20/09/2019/

УДК 004.93`1

## МЕТОДЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ РАСПОЗНАНИЯ ТЕКСТА ИЗ ИЗОБРАЖЕНИЙ

**С.Н. МАЛАШОНОК**

*(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье описывается использование машинного зрения для распознавания текста, рассматриваются основные проблемы и подходы.*

Многие компании имеют тысячи документов для обработки, анализа, преобразования и выполнения различных операций. Примерами таких документов являются квитанции, счета-фактуры, формы, выписки, контракты и многие другие фрагменты неструктурированных данных, и необходимо иметь возможность быстро распознать информацию, хранящуюся в неструктурированных данных, подобные этим.

Последние достижения в области компьютерного зрения позволяют добиваться больших успехов в облегчении процесса анализа и понимания документов.

Задача извлечения текста из изображений документов называется оптическим распознаванием символов (OCR – Optical Character Recognition) и является предметом многих исследований. OCR является одной из самых ранних задач компьютерного зрения, поскольку в некоторых аспектах не требует глубокого обучения. Когда документы четко размечены и имеют определенную структуру (например, деловое письмо), существующие инструменты для OCR могут работать достаточно хорошо. Популярным инструментом для распознавания текста с открытым исходным кодом является проект Tesseract, который изначально разрабатывался Hewlett-Packard, но в последние годы находится под поддержкой Google. В последнее время провайдеры облачных услуг внедряют возможности обнаружения текста наряду с их другими предложениями, что делает методы компьютерного зрения широко доступными. Среди них GoogleVision, AWS Textract, Azure OCR и Dropbox. [1]

Однако есть много случаев использования, которые можно назвать нетрадиционным OCR, когда существующие общие решения не подходят. К таким случаям можно отнести обнаружение текста, который может находиться на дорожных знаках, номерах домов, рекламных объявлениях и т.д. Изображения могут включать в себя разделение фона, различные ориентации и длины текста, разнообразие шрифтов, отвлекающие объекты.

Другой случай, который создает аналогичные проблемы, связан с извлечением текста из изображений сложных документов. В отличие от документов с определенным макетом (таких как страница из книги, столбец из газеты), многие типы документов относительно неструктурированы по макету и имеют разбросанные по всему тексту элементы (такие как квитанции, формы, и счета). Эти изображения характеризуются сложным расположением текстовых элементов, разбросанных по всему документу и окруженных множеством «отвлекающих» объектов. В этих изображениях основная проблема заключается в правильном сегментировании объектов в изображении для идентификации нужных текстовых блоков.

На основании этих двух случаев можно выделить следующие критерии задач OCR:

- плотность текста – на печатной или письменной странице плотность текста будет велика в сравнении с текстом на изображении улицы;
- структура текста – текст на странице может располагаться в соответствии с каким-либо макетом или может располагаться в хаотичном порядке и иметь различные углы поворота;
- шрифты – печатные шрифты проще, так как они более структурированы по сравнению с рукописными символами;
- артефакты – изображения улиц имеет большее количество шумов, чем изображения страниц книг [2].

Основными этапами распознавания текста являются обнаружение текста на изображении и распознавание обнаруженного текста с помощью одного из следующих подходов: классические методы компьютерного зрения, специализированное глубокое обучение или стандартный подход глубокого обучения.

Классические методы компьютерного зрения включают в себя:

- применение различных фильтров для выделения символов из фона;
- распознавание контура каждого отдельного символа;
- применение классификации контуров символов для идентификации символов.

Обнаружение контура является довольно сложной задачей для обобщения, поэтому требует много ручной тонкой настройки. Классификация контура может быть реализована либо с помощью сопоставления с образцом, либо с помощью машинного обучения.

К специализированному глубокому обучению относится сверточная рекуррентная нейронная сеть (CRNN – Convolutional Recurrent Neural Network). Эта архитектура нейронной сети объединяет функции извлечения, моделирование последовательности и транскрипцию в единую структуру. Эта модель не нуждается в сегментации символов. Сверточная нейронная сеть извлекает элементы из входного изображения (области обнаружения текста). Глубокая двунаправленная рекуррентная нейронная сеть предсказывает последовательность меток с некоторым соотношением между символами. Уровень транскрипции преобразует кадр, созданный рекуррентной нейронной сетью, в последовательность меток. Существует два способа транскрипции: безлексиконный и основанный на лексиконах. В подходе, основанном на лексиконе, будет предсказана наиболее вероятная последовательность меток [3].

Таким образом, стандартные подходы глубокого обучения, такие как SSD, YOLO и Mask RCNN, используются для обнаружения символов и слов. Однако они часто не достигают желаемой точности, и поэтому необходимы специальные подходы.

Вышеизложенные данные поспособствуют разработке мобильного приложения для учета и планирования расходов посредством сканирования чеков. Актуальность данного приложения заключается в введении статистики по различным видам и категориям товаров с целью контроля и оптимизации финансовых затрат.

#### ЛИТЕРАТУРА

1. Learning to Read: Computer Vision Methods for Extracting Text from Images [Электронный ресурс]. – Режим доступа: <https://medium.com/capital-one-tech/learning-to-read-computer-vision-methods-for-extracting-text-from-images-2ffcdae11594>. – Дата доступа: 24.09.2019.
2. A gentle introduction to OCR [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa>. – Дата обращения: 24.09.2019.
3. Deep Learning Based OCR for Text in the Wild [«ktrnhjyysq htcehc/ – Ht;bv ljcnegf^ https^|nanonets/com|blog|deep-learning-ocr|#text-detection/ – Lfnf ljcnegf^ 24/09/2019/.

УДК 004.622

**ПОДХОДЫ К РАЗРАБОТКЕ ИНТЕРФЕЙСА  
ДЛЯ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ  
ИНСПЕКЦИИ ПО ДЕЛАМ НЕСОВЕРШЕННОЛЕТНИХ ПОЛОЦКОГО РАЙИСПОЛКОМА**

**К.В. МАРКЕВИЧ**

*(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

*В статье представлен практический способ создания интерфейса для работы с данными комиссии по делам несовершеннолетних. Проведен анализ технологий, наиболее подходящих, для разработки данной системы. Задача, реализация прототипа интерфейса на основе windows-form.*

**Введение.** Автоматизация и создание информационных систем являются на данный момент одной из самых ресурсоемких областей деятельности техногенного общества. Одной из причин активного развития данной области является то, что автоматизация служит основой коренного изменения процессов управления, играющих важную роль в деятельности человека и общества. Возникают системы управления, действие которых направлено на поддержание или улучшение работы объекта с помощью устройства управления (комплекс средств сбора, обработки, передачи информации и формирования управляющих сигналов или команд) [1].

Автоматизированная информационная система (АИС) – совокупность программно-аппаратных средств, предназначенных для автоматизации деятельности, связанной с хранением, передачей и обработкой информации.

АИС являются, с одной стороны, разновидностью информационных систем (ИС), с другой – автоматизированных систем (АС), вследствие чего их часто называют ИС или АС.

АИС может быть определена как комплекс автоматизированных информационных технологий, предназначенных для информационного обслуживания – организованного непрерывного технологического процесса подготовки и выдачи потребителям научной, управленческой и др. информации, используемой для принятия решений, в соответствии с нуждами для поддержания эффективной деятельности [2].

**Выбор технологий разработки интерфейса.** Выбор средств решения поставленной задачи исходил из того, каким программное и аппаратное обеспечение необходимо для разработки. Здесь выбор был сделан в пользу операционной системы Windows 10.

В качестве среды разработки для реализации проекта было выбрано Microsoft Visual Studio 2017. Это обусловлено следующими причинами:

1. Гибкая структура поддержки объектно-ориентированного подхода программирования.
2. Легкость и простота разработки проектов. В это свойство входит реализация всевозможных подсказок в редакторе исходного кода, широкие возможности по работе с ресурсными файлами проекта, такими как файлы изображений, иконок, курсорами, меню, акселераторами.
3. Наличие мощного отладчика для выявления ошибок в процессе реализации программы.
4. Большое количество документации по возможностям среды разработки, по применяемым типам модулей и описания основных функций.
5. При разработке проекта существует достаточно много опций по выставлению оптимизации разрабатываемого проекта, например, оптимизировать размер исполняемого модуля программы, или оптимизировать ее скорость выполнения.
6. Поддержка работы с сервисами, как средствами API – функций, таки и на уровне специальных встроенных конструкторов сервисов.
7. Производительность программы повышается за счет специальных встроенных библиотек функций – .NET Framework 7.4.2. Все зависит от версии компонентов, установленных в операционной системе.
8. Относительно небольшие системные требования, предъявляемые для установки среды и для ее запуска.

В качестве языка программирования для написания приложения был выбран объектно-ориентированный язык C#. Этот выбор обусловлен тем, что данный язык является ключевым языком в работе с платформой .NET Framework. Для реализации проекта необходимой и достаточной версией этого набора библиотек является .NET Framework 4.7.2. Данная версия необходима для поддержания работы на уровне приложения с базами данных и организации сетевого взаимодействия между отдельными модулями программы.

**Проектирование интерфейса.** При разработке интерфейса, для удобства использования была применена вкладочная структура – информация логически разделена по вкладкам, что упрощает ее поиск, не загромождает главную страницу и не приводит к необходимости создавать дополнительные формы.

Основная информация о нарушителях хранится в базе данных, над которыми, имеет возможность производить действия, после выбора соответствующих данных. Данная структура, является наиболее простой и понятной для пользователей, с различным уровнем владения персональным компьютером. Использование стандартах элементов управления, позволяет максимально снизить время обучения работе с приложением.

Для визуализации некоторых операций были созданы компоненты пользовательского интерфейса. Разработанные компоненты и их описание представлено в таблице 1.

Таблица 1. – Перечень разработанных пользовательских компонентов

Название компонента	Назначение компонента
Button_Open	Реализация кнопки для выбора документа
Button_Save	Сохранения копии документа с внесенными изменениями
FilterTextBox	Фильтрация данных из базы данных
PrintCheckBox	Реализация элемента для печати документа при сохранении
dataGridView	Реализация элемента управления для отображения текущего состояния базы данных

С помощью программы пользователь взаимодействует с подсистемой формирования документов. Программа в своем составе имеет отдельное окно, через которое пользователь получает всю необходимую информацию.

Весь функционал можно разделить на конкретные задачи, для каждой из которых реализован свой метод. Список нескольких методов и их назначение указаны в таблице 2.

Таблица 2. – Перечень методов приложения

Название метода	Назначение
Form1	Содержит в себе реализацию элемента интерфейса, в котором отображается вся информация из базы данных
FindAndReplace	Содержит реализацию части программы, для работы с документом
CreateWordDocument	Описывает метод, который осуществляет подстановку текста

**Заключение.** Правильным подходом при разработке приложения является использование современных технологий, которые позволяют решать свои задачи. Это, в первую очередь, экономит время при разработке интерфейса, а также ресурсы на обработку данных. Это является очень актуальным, при большом количестве информации, которая подлежит постоянному изменению.

Рассмотрена технология win-form позволяющая создать адаптивный интерфейс, любого уровня сложности, при этом сохраняя его функциональные возможности и привлекательный внешний вид. По итогам данной работы, была описана возможность их применения к построению интерфейса и динамической обработке запросов.

## ЛИТЕРАТУРА

1. Официальный сайт справочной информации. [Электронный ресурс]. Режим доступа <https://www.bibliofond.ru/view.aspx>. Дата доступа – 16.06.2019 г.
2. Официальный сайт справочной информации. [Электронный ресурс]. Режим доступа <https://dic.academic.ru/dic.nsf/ruwiki/334809>. Дата доступа – 16.06.2019 г.
3. Официальный сайт справочной информации. [Электронный ресурс]. Режим доступа <https://dic.academic.ru/dic.nsf/ruwiki/1338302>. Дата доступа – 16.06.2019 г.



УДК 004.622

**ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ  
АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ  
КОМИССИИ ПО ДЕЛАМ НЕСОВЕРШЕННОЛЕТНИХ ПОЛОЦКОГО РАЙИСПОЛКОМА**

**К.В. МАРКЕВИЧ**

*(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

*В статье рассматривается проектирование системы для работы с данными комиссии по делам несовершеннолетних. Проведен анализ технологий, наиболее подходящих, для разработки данной системы. Проведены исследования по актуальности разработки данной системы.*

**Введение.** Трудно представить хорошее клиент-серверное приложение без разработанной базы данных. Они позволяют удобно хранить большие объемы информации не на устройстве пользователя, а на удаленном сервере, который может эти данные предоставить по запросу от приложения.

Данная статья направлена на описание базы данных для разработанного приложения и методов ее проектирования.

**Основной раздел.** На данный момент, в Республике Беларусь, множество учреждений используют бумажный документооборот для хранения данных. Данный подход, с момента развития информационных технологий значительно уступил в эффективности, электронным системам хранения и обработки данных. В данном случае, электронные системы имеют ряд преимуществ:

- повышение качества управления административной деятельностью организации за счет комплексного внедрения автоматизированной информационной системы;
- сокращение времени оформления документации, посредством получения готовых данных из баз данных и компьютерного ввода;
- сокращение сроков предоставления отчетов и реестров услуг, на основе автоматически генерируемых шаблонов;
- оптимизация штата персонала учреждений

Данные критерии, несомненно доказывают эффективность данных систем, а также актуальность их разработки и внедрения в организацию.

**Средства решения задачи.** Основу разрабатываемой системы, составляет возможность хранить, накапливать и обрабатывать данные. Также в данном случае, данные должны храниться в строго систематизированном виде, основываясь, на нормах ведения документации, и имеют явную взаимосвязь.

Основываясь на этих данных необходимо будет использовать реляционную базу данных для хранения сущностей и язык запросов SQL для построения базы данных. В данном случае, использование именно реляционной базы данных, позволит формально и однозначно, определить ограничения целостности БД, в терминах ее концептуальной схемы.

**Достоинства реляционной модели.** Одним из важных достоинств реляционного подхода является его простота и доступность для понимания конечным пользователем. Единственной информационной конструкцией является таблица.

При проектировании реляционных баз данных применяются строгие правила, базирующиеся на математическом аппарате.

Реляционная модель обеспечивает полную независимость данных. При изменении структуры реляционной базы данных изменения, которые требуется произвести в прикладных программах, как правило, минимальны.

Манипулирование данными на уровне языка СУБД производится ненавигационно, поэтому для построения запросов и написания прикладных программ нет необходимости знания конкретной организации базы данных во внешней памяти. Конечно, при исполнении запросов на физическом уровне выполняется навигация по записям таблиц, однако эти действия производятся процедурами самой СУБД.

**Недостатки реляционной модели.** По сравнению с иерархической и сетевой моделями реляционная модель имеет более низкую скорость доступа и требует большего объема внешней памяти. В настоящее время этот фактор не является критическим вследствие многократно возросшего быстродействия компьютеров и такого же роста объема дисковой памяти.

Часто в результате логического проектирования появляется очень много таблиц, что затрудняет понимание структуры данных.

Далеко не всегда предметную область можно представить в виде совокупности таблиц. Так, в системах автоматизации проектирования и автоматизированной разработки программного обеспечения требуются гораздо более сложные структуры данных.

Для преодоления недостатков, присущих реляционной модели, в настоящее время развиваются постреляционная, многомерная и объектно-ориентированная модели. Эти модели в той или иной степени опираются на реляционную модель. Тем не менее реляционная модель и коммерческие продукты, основанные на этой модели, доминируют при построении экономических информационных систем.

Очевидно, что при реализации приложения важно сформировать некоторую архитектуру. В данном случае, для взаимодействия с базой данных, подойдет паттерн ADO.NET, представляющий собой взаимодействие 3 элементов: модель, представление и контроллер. Модель является некоторыми данными, которые описывают предметную область, и над ними могут быть выполнены некоторые операции. Также эта информация может быть предоставлена пользователю на графическом интерфейсе.

Приложение является этим самым графическим интерфейсом. Через данный интерфейс пользователи системы будут передавать данные на выполнение задачи контроллером. Контроллер, в свою очередь, реализует некоторые алгоритмы для получения данных с представления, обработки их и отправки обратно.

Цель моделирования данных состоит в обеспечении разработчика концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Для реализации приложения, необходимо установить все связи между сущностями: необходимо рассмотреть всю информационную систему в совокупности и определить взаимное влияние сущностей.

В соответствии выделенным сущностям, определен набор необходимых таблиц базы данных, представленный в таблице 1.

Таблица 1. – Перечень необходимых таблиц

Название таблицы	Описание предназначения
Parents	Таблица предназначена для хранения информации о родителях
Criminal	Таблица предназначена для хранения учетных данных о всех нарушителях
Offense	Таблица предназначена для хранения специфических данных

Важное достоинство баз данных – возможность хранить сгруппированные данные в разных таблицах и устанавливать связи между этими таблицами посредством ключей и последующее объединение этих данных в единую базу. Это позволяет снизить избыточность данных, которые хранятся, упростить организацию запросов для их получения.

Первичный ключ служит ограничителем целостности в рамках одной таблицы для идентификации, т.к. поле первичного ключа не может повторяться или быть пустым. Важное условие - каждая таблица может содержать только один первичный ключ.

Для организации связей между двумя и более таблицами, используются вторичные ключи. Они служат ограничителями целостности связей нескольких таблиц, т.к. подчиненная таблица не может ссылаться на несуществующие записи главной таблицы (что позволяет строить целостные модели данных).

Существует три вида связей между таблицами:

- «один-ко-многим»,
- «один-к-одному»,
- «многие-ко-многим».

Связь «один-ко-многим» - самая распространенная для реляционных баз данных. Для обеспечения ссылочной целостности в дочерней таблице создается внешний ключ, через который организовывается связь с дочерней таблицей. Обязательное условие состоит в том, что внешний ключ по составу полей должен совпадать с первичным ключом родительской таблицы.

**Заключение.** В ходе данного исследования была спроектирована, программная система, наиболее подходящая по архитектурным особенностям под основу для разработки приложения для работы с данными. При этом разработанная архитектура, является универсальной для любого современного языка программирования, а также оставляет возможности для реализации необходимого функционала.

УДК 004.021

**ПРОЕКТИРОВАНИЕ КРОССПЛАТФОРМЕННОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПОДБОРА СОТРУДНИКОВ ПО НАВЫКАМ****П.В. МАХАНЬКО***(Представлено: Ю.Н. КРАВЧЕНКО)*

*В данной статье рассматривается разработка мобильного приложения.*

**Введение.** React Native является самым популярный фреймворком для разработки кроссплатформенных мобильных приложений. Каждая страница приложения представляет из себя view с набором элементов языка разметки XML и React Native Elements. Стилистические особенности задаются при помощи CSS свойств, написанных на языке javascript. Такой подход называется – styled components. Логика приложения является язык программирования javascript.

**Основной раздел.** XML – расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчеркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Расширение XML – это конкретная грамматика, созданная на базе XML и представленная словарем тегов и их атрибутов, а также набором правил, определяющих какие атрибуты и элементы, могут входить в состав других элементов. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как, собственно, XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

Styled Components дает возможность написания CSS в JavaScript используя т.н. тегируемые шаблонные строки. Библиотека удаляет сопоставление между компонентами и стилями – компонент превращается в конструкцию с низкоуровневой стилизацией. При помощи технологии styled components можно инкапсулировать стили для определенных компонентов и view элементов что позволяет избежать таких проблем как конфликт имен классов.

Основой приложения служит React Native – фреймворк базирующийся на технологиях React, но позволяет разрабатывать мобильные приложения. React Native реализует технологию однонаправленной связи. Свойства передаются от родительских компонентов к дочерним. Компоненты получают свойства как множество имутабельных значений, поэтому компонент не может напрямую изменять свойства, но может вызывать изменения через функции обратного вызова. В качестве дерева DOM React использует виртуальный DOM. React создают кэш структуру в памяти что позволяет вычислять разницу между предыдущим и текущим состоянием интерфейса для оптимального обновления DOM браузера. Таким образом, программист может работать со страницей, считая, что она обновляется вся, но React самостоятельно решает, какие компоненты страницы необходимо обновить. Пример интерфейса мобильного приложения изображен на рисунках 1, 2, 3, 4

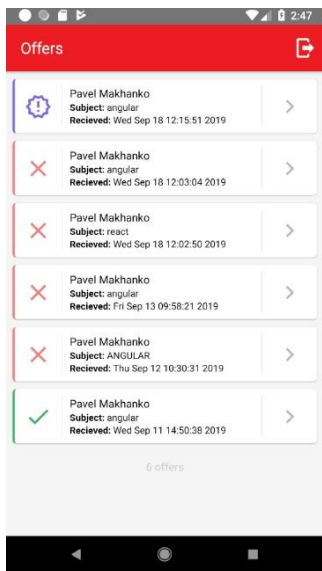


Рисунок 1. – Страница полученных предложений

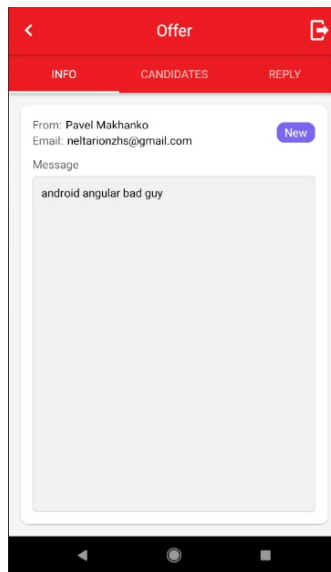


Рисунок 2. – Страница информации предложения

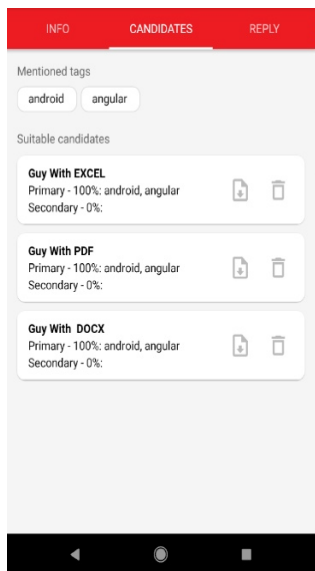


Рисунок 3. – страница предложенных кандидатов

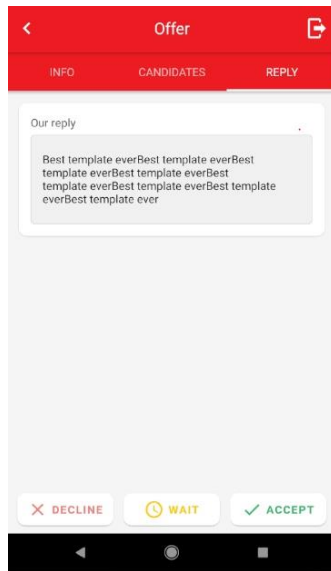


Рисунок 4. – страница ответа на предложение

**Заключение.** В данной статье рассмотрены основные элементы построения графического интерфейса для мобильного приложения. Представлены примеры и результаты.

УДК 004.021

**ПРОЕКТИРОВАНИЕ API И БАЗЫ ДАННЫХ ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ,  
РАЗРАБОТАННОГО ДЛЯ ПОДБОРА СОТРУДНИКОВ ПО НАВЫКАМ****П.В. МАХАНЬКО***(Представлено: Ю.Н. КРАВЧЕНКО)*

*В данной статье рассматриваются цели и принципы моделирования API и базы данных для веб-приложения*

**Введение.** В наше время трудно представить хорошее веб-приложение без разработанной базы данных и быстрого, оптимизированного API. Большинство современных веб приложений зачастую пользуются сторонними сервисами, которые позволяют оптимизировать и улучшить разработку. Базы данных позволяют хранить большие объемы данных, облачные хранилища позволяют доставлять контент к конечному пользователю независимо от его местоположения, а быстрые API позволяют обрабатывать запросы в кратчайшие сроки. API представляет собой промежуточное звено между клиентской частью приложения и базой данных. Так же API является отличным способом защиты пользовательских данных используя современные технологии OAuth и JSON Web Token.

**Основной раздел.** JSON Web token является открытым стандартом, создания токенов доступа основываясь на json. В момент авторизации пользователя, сервер генерирует хеш-ключи, которые отправляет клиентской части веб-приложения. В дальнейшем, каждый запрос к серверу, должен содержать в себе хеш-ключ для подтверждения прав доступа к ресурсам.

OAuth является современным решением защиты пользователей, а также удобным способом регистрировать и авторизировать пользователей в приложении. OAuth – это открытый протокол авторизации, который позволяет предоставить третьей стороне доступ к защищенным данным не передавая логин и пароль что позволяет пользователю иметь одну учетную запись для многих сервисов.

Преимущество OAuth является то, что пользователь хранит свои логин и пароль только в одном сервисе, например, Google, а для доступа к другим, он получает токен, который генерируется на стороне авторизации.

Для взаимодействия с почтовыми сервисами использовались открытые API Google и Microsoft Exchange. Данные сервисы позволяют обмениваться электронными письмами между различными адресами, а также при получении доступа к письмам третьей стороной, позволяет обрабатывать, анализировать и перенаправлять эти письма на другие адреса. Доступ получается при помощи открытых и секретных ключей, генерируемые на стороне этих сервисов.

Amazon S3 является отличным средством для хранения и получения больших объемов данных, таких как крупные файлы, документы, аудиозаписи и фильмы. Преимущество Amazon S3 является быстрая работа, простота интеграции в приложение, а также защищенность файлов при помощи шифрования.

Amazon EC2 – веб-сервис, который предоставляет вычислительные мощности в облаке, которые позволяют содержать ваше приложение, библиотеки и данные. Данный сервис отлично подходит для развертывания API и базы данных.

Amazon Cloudfront – цель данного сервиса, является доставка контента конечным пользователям с минимальными задержками и высокой скоростью передачи. При загрузке файла или приложений, Amazon самостоятельно создает реплики данных и хранит их на всех своих сервера. Это дает возможность получать данные разными пользователями из разных точек мира с одинаковой скоростью.

В качестве точек для запросов, сервер предоставляет веб-приложению конечные адреса, обращаясь к которым, веб-приложение получает определенные данные. Запросы проходят по стандартному HTTP протоколу, используя методы GET, POST. Чат и обмен файлами реализованы по технологии сокетов (протокол WebSocket). Сокет – протокол дуплексной связи между клиентом и сервером в режиме реального времени. В данном случае, веб-приложение подписывается на новые сообщения при успешном входе в приложение. Затем при получении нового сообщения оно мгновенно будет отправлено пользователю.

В качестве базы данных была выбрана документно-ориентированная база данных MongoDB. Синтаксис MongoDB схож с форматом json, а это в свою очередь не задает жестких рамок хранения данных, что позволяет удобно расширять хранимые данные. Вся база данных состоит из коллекций, а хранимые в ней данные – документы. Документ представляет из себя json-объект в виде пар ключ/значение, где в качестве значения могут храниться строки, числа, двоичные данные, массивы или другие вложенные объекты.

Реляционная модель представляет собой базу данных в виде множества взаимосвязанных отношений (таблиц), которые используются для хранения информации об объектах, представленных в базе данных.

В соответствии с требованиями программы, определены коллекции (набор) необходимых таблиц базы данных, представленный в таблице 1.

Таблица 1. – Перечень необходимых таблиц базы данных

Название коллекции	Описание
Users	Список всех пользователей
Candidates	Список всех кандидатов
Offers	Список всех предложений полученных по email
Templates	Список все шаблонов ответов на емейл
Tags	Список всех навыков

Важное достоинство баз данных – возможность хранения сгруппированных данных в разных коллекциях и в разных форматах.

**Заключение.** В данной статье были рассмотрены основные цели и принципы моделирования базы данных, построение быстрого, удобного API для веб-приложения. Были рассмотрены основные сущности базы данных. Спроектированы адреса для обращения клиентской части веб-приложения к серверной части.

УДК 004.05

**ЦИФРОВАЯ ДИСТРИБУЦИЯ ПРИЛОЖЕНИЙ.  
ВЛИЯНИЕ ОНЛАЙН-СЕРВИСОВ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ И РАЗРАБОТЧИКОВ****Н.А. МЕДВЕДЕВ***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассматривается цифровая дистрибуция и ее значение для распространения программной продукции. Также рассмотрены онлайн-сервисы, и их влияние на бизнес и возможность пользователям управлять купленным товаром.*

Распространение товара занимает важную часть истории со времен основания торговли. Благодаря ему имеется возможность эффективно находить новых клиентов и продавать нужные товары независимо от месторасположения. Одним из методов распространения товаров является цифровая дистрибуция. Значение цифровой дистрибуции играет важную роль со времен развития интернета: она позволяет быстро и безопасно распространять программное обеспечение или распространять рекламу о ней. Почему, собственно, в наше время ею пользуются все: начиная с простых пользователей и заканчивая опытными бизнесменами и распространителями. Это означает, что помимо цифрового обеспечения, можно распространять и остальные продукты, начиная с обычных предметов и заканчивая очень редкими вещами. Поэтому сейчас присутствуют множество онлайн-магазинов, где покупателю можно купить абсолютно все что угодно. Чтобы лучше понять проблему распространения продукции в интернете мы рассмотрим понятие цифровой дистрибуции и как оно влияет на распространение продукции через интернет.

Цифровой дистрибуцией называется способ распространения цифрового контента без использования материального носителя, как правило, состоящий в загрузке через Интернет на оконечное устройство пользователя [1]. Преимущество цифровой дистрибуции заключается в легком и быстром поиске и приобретении копий необходимого программного обеспечения.

Существует несколько вариантов распространения программного обеспечения через Интернет. В цифровой дистрибуции используют основные протоколы – это HTTP, P2P и FTP. Системы, распространяющие проприетарное программное обеспечение, обычно содержат технические средства защиты авторских прав, не позволяющие покупателю продавать/распространять свою копию программного обеспечения (ПО).

Существует несколько вариантов покупки прав на использование ПО:

1. PayPlay – продукция будет доступна только после ее покупки.
2. Try before you buy – пользователь скачивает программу и получает возможность какое-то время пользоваться ей. В случае, если пользователь захочет и дальше использовать полную версию программы, он должен заплатить. После этого (как правило) предоставляется специальный серийный номер, введя который, пользователь получает возможность пользоваться программой неограниченное время.
3. Рекламная – в этом случае программа абсолютно бесплатна, однако пользователю в программе показывается реклама, на показе которой и зарабатывает обладатель ПО.
4. Подписка – ежемесячная оплата права пользоваться программой или программами из определенного набора, предоставленного цифровым дистрибутором. Также существуют единичные подписки на одну определенную программу и на разный срок (в том числе неограниченный) [1].

Для осуществления цифровой дистрибуции используется онлайн-сервис, где за указанные выше методы можно получить все необходимые товары.

Основными продуктами цифровой дистрибуции являются:

Книги – основной популярностью пользуются покупка электронные книг, и основное отличие от традиционной покупки является хранение в неопределенном количестве в электронной «читалке» или любом электронном девайсе поддерживающим книжный формат.

Музыка – основными представителями являются онлайн-покупка или стриминг. В первом случае за указанную сумму покупается альбом исполнителя или несколько его песен, а во втором за подписку или под указанные условия дается свободный доступ к прослушке песен указанного исполнителя. Основное отличие от традиционной покупки физических носителей является легкий доступ к музыкальным файлам и возможность хранить в неограниченном количестве на определенном девайсе.

Фильмы – основными представителями являются также онлайн-покупка или подписка. В первом случае за указанную сумму дается неограниченный доступ к определенному фильму или сериалу, а во втором за определенную сумму на указанный срок дается доступ к просмотру через сервис, где была произведена подписка. Основным отличием от традиционной формы является легкий доступ к видео и возможность хранить в неограниченном количестве на определенном девайсе.

Видеоигры – основными представителями являются также онлайн-покупка или подписка. В первом случае за указанную сумму дается неограниченный доступ к определенному фильму или сериалу, а во втором, за определенную сумму на указанный срок дается доступ к просмотру через сервис, где была произведена подписка. Отличие от традиционной формы также является легкий доступ к видео и возможность хранить в неограниченном количестве на определенном девайсе.

Дополнительно к онлайн-сервисам может прилагаться лаунчер. Лаунчерами называются программы, позволяющие устанавливать запускать и воспроизводить купленные или активированные ПО. Ими являются такие программы как iTunes, Spotify, Steam, Netflix и т.д. Основное преимущество – хранение всего приобретенного ПО под своей учетной записью, автоматизированный процесс загрузки приложения с сервера и установки на машину пользователя.

Основное влияние цифровой дистрибуции над традиционной розничной торговлей можно указать в следующих сферах.

Книги – некоторые компании как Bookmasters Distribution потратили 4,5 млн долларов на обновление оборудования для работы с большим количеством пользователей. Дополнительно вице-президент компании Perseus Books Group отметил с переходом на электронные книги рост на 68 процентов и Independent Publishers Group заявили о росте продаж на 23 процента в первой четверти 2012 [2].

Музыка – рост цифровой дистрибуции привело к падению продаж CD дисков в 2 раза на протяжении 2000-х годов. Развитие таких систем как iTunes и позже Spotify способствовало развитию независимого пространства музыки. Дополнительно с начала цифровой продажи был отмечен рост с 160 млн в 2004 до 795 млн в 2006, что в денежном переводе составляет с 397 млн долларов до US\$2 миллиарда долларов [2].

Фильмы – за счет цифрового распространения был замечен резкий переход фильмов и сериалов на такие платформы как YouTube, Netflix, Amazon Prime и т.п. Дополнительно за счет сервисов можно отметить возрастающую популярность просмотров на Smart-TV, компьютерах, смартфонах, игровых консолях и планшетах [2].

Видеоигры – основное развитие цифрового распространения началось в 2004 году с выходом платформы Steam, который вместо продажи игр дает права на хранение игры путем принятия соглашения (в следствие нарушения права на хранение – изымаются). С конца 2000-х цифровое распространение игр охватывает все более широкую аудиторию, благодаря чему крупные издатели и производители игр, занимающиеся розничной торговлей, сосредотачивают большее внимание на цифровых продажах – в их число входят Steam, PlayStation Store, Amazon.com, GAME, GameStop и другие. И вследствие продажи физических носителей упала с 5.03 миллиардов доллара в 2007 году до 2.15 миллиардов доллара. Согласно исследованию, проведенному Superdata Research, выручка от проданных таким способом игр по всему миру достигла около 6,2 миллиардов долларов за месяц в течение февраля 2016 года. В начале 2019 года произошло развитие таких сервисов как Epic Games, Rockstar Games Launcher, battle.net, Bethesda Game Launcher что привело к конкуренции, так как компаниям выгодно владеть собственным сервисом как Ubisoft с выходом Tom Clancy's Division 2 под сервисом Uplay получила в 4.5 раза больше дохода чем под остальные сервисы, что привело к решению выпускать свою продукцию только под своим сервисом.

Таким образом, можно указать заметный рост цифровой дистрибуции и ее замену традиционный продажей физических носителей.

**Заключение.** В результате исследования были исследованы основные направления цифровой дистрибуции и в результате развития последних трендов было выведено развитие онлайн-сервисов на распространение видеоигр. Так как пользователям после обязательной регистрации на сервисе и покупки/активации дается право на хранение продукта под своим аккаунтом, что дает необходимый контроль, и позволяет разработчикам распространить и выпустить свой проект без сделки с издателем, что приводит к дополнительным продажам. Также можно добавить решение проблемы архаичности хранения купленных приложений путем создания единой системы позволяющей приобрести, загрузить и установить все необходимое.

На основании проведенного анализа планируется разработка проекта, основной задачей которого является предоставление условий распространения неигрового ПО с возможностью запустить купленные приложения с помощью специального лаунчера.

#### ЛИТЕРАТУРА

1. Цифровая дистрибуция. [Электронный ресурс] / ВМУ. – Режим доступа: [https://ru.bmstu.wiki/Цифровая\\_дистрибуция](https://ru.bmstu.wiki/Цифровая_дистрибуция). – Дата доступа: 18.09.2019.
2. Digital Distribution. [Электронный ресурс] / Wikipedia. – Режим доступа: [https://en.wikipedia.org/wiki/Digital\\_distribution](https://en.wikipedia.org/wiki/Digital_distribution). – Дата доступа: 18.09.2019.



УДК 004.05; 004.921

## ВНЕШНЯЯ ПЛАТЕЖНАЯ СИСТЕМА И ЕЕ ВЛИЯНИЕ НА БЕЗОПАСНОСТЬ СЕРВИСА

Н.А. МЕДВЕДЕВ

(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)

В данной статье рассмотрена внешняя платежная система для онлайн-сервиса и ее влияние на надежность онлайн-сервиса в целом. Рассмотрены различные платежные системы и проведен их сравнительный анализ.

Платежная система является важнейшей частью любого онлайн-сервиса, в котором происходят платежи, ведь без нее не может происходить никакая-нибудь возможность торговли приложениями. Сам платеж может происходить как во внутренней системе, так и через внешние системы, где транзакцию будет выполнять система, которую мы указали.

Итак, первоначально рассмотрим такие понятия как: электронная коммерция, что такое внутренняя и внешняя платежные системы и их особенности.

Электронная коммерция – это сфера экономики, которая включает в себя все финансовые и торговые транзакции, осуществляемые при помощи компьютерных сетей, и бизнес-процессы, связанные с проведением таких транзакций. К электронной коммерции относят:

- 1) электронный обмен информацией (Electronic Data Interchange, EDI),
- 2) электронное движение капитала (Electronic Funds Transfer, EFT),
- 3) электронную торговлю (англ. E-trade),
- 4) электронные деньги (англ. E-cash),
- 5) электронный маркетинг (англ. E-marketing),
- 6) электронный банкинг (англ. E-banking),
- 7) электронные страховые услуги (англ. E-insurance) [1].

Электронная платежная система – это система расчетов между финансовыми организациями, бизнес-организациями и интернет-пользователями при покупке-продаже товаров, а также за различные услуги через Интернет. Такие системы представляют собой электронные версии традиционных платежных систем и по схеме оплаты делятся на:

- дебетовые (работающие с электронными чеками и цифровой наличностью);
- кредитные (работающие с кредитными карточками) [2].

На рисунке 1 показана схема покупки через платежную систему.

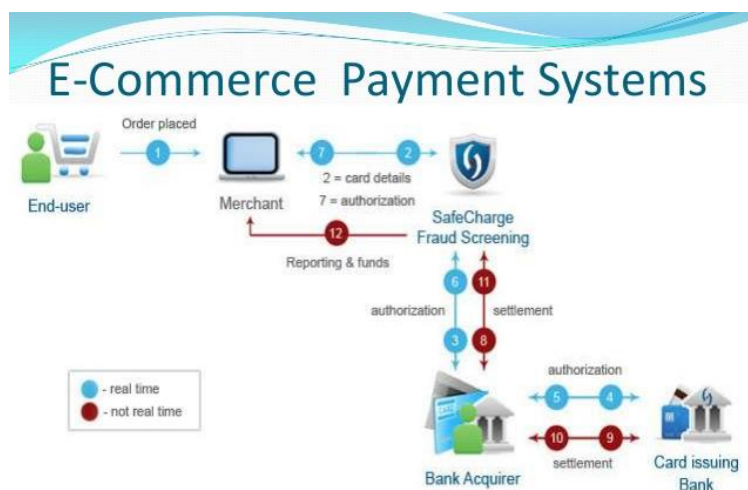


Рисунок 1. – Процесс оплаты через платежную систему

Саму платежную систему в сервисе можно разделить на два вида:

– внутренняя (или интегрированная) – называется платежная система, которая встроена в онлайн-сервис и проводит транзакцию в банк через банковскую карточку. Единственное что покупателю необходимо – данные собственной платежной карты. Основными преимуществами такой платежной системы являются полный контроль транзакции со стороны сервиса и отсутствие необходимости пользователю

предоставления дополнительных данных о других сервисах. Недостатками системы являются повышенная нагрузка из-за личного проведения необходимых транзакции, а также ее уязвимость и возможность утечки важных данных в случае взлома сервиса;

– внешняя (или сторонняя) – называется система, которая принимает данные с онлайн-сервиса и затем самостоятельно проводит необходимую транзакцию. Для платежа данная система использует данные с сервиса и/или данные от покупателя. Основными преимуществами такой системы являются снижение нагрузки сервиса и более надежная защита на случай взлома. К недостаткам такой системы можно отнести возможную дополнительную оплату за использование, требование дополнительных данных от покупателя и возможность отказа работы сторонней системы на случай нагрузки.

Далее рассмотрим некоторые из платежных систем и их особенности. Основными платежными системами являются:

**PayPal** является платежной системой, основанной в 1998 году. Она получила популярность в 2002 году с ее покупкой интернет-аукционом Ebay. На сегодняшний день она является наиболее популярной в мире. Численность ее пользователей составляет 450 млн. человек. Она имеет гарантированную транзакцию как для пользователя, так и для покупателя. Для разработчиков в платежной системе предусмотрена версия Sandbox, позволяющая при интеграции в онлайн-сервис безопасно проверить работу платежа и транзакции.

**Amazon Payments** – это платежная система, разработанная компанией Amazon. Она обладает самым популярным сервисом для оплаты через мобильные средства, а также имеет библиотеку для интеграции в собственный сервис для продажи.

**WebMoney** – система, основанная в 1997 году. Данная платежная система является самой популярной системой в постсоветском пространстве и одной из самых надежных из-за многоуровневой идентификации. От остальных отличается возможностью открывать неограниченное количество кошельков в разных валютах (WMZ – долларах, WMR – рублях, WME – евро и др.) и объединять в специальное хранилище (keeper), которому присваивается уникальный номер владельца WMID, что является очень удобным для использования.

**QIWI** – платежная система, основанная в 2007 году. Сегодня она является самой популярной системой в России из-за наличия терминалов для создания аккаунта и пополнения средств.

В мире имеются и другие, не менее популярные платежные системы, некоторые из них представлены на рисунке 2.

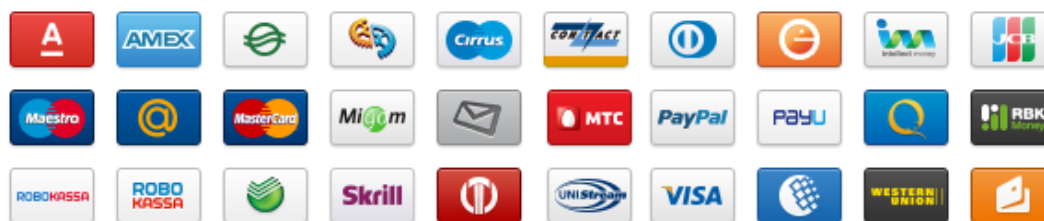


Рисунок 2. – Основные платежные системы

**Заключение.** В результате исследования были исследованы электронные платежи и их виды. Проведено сравнение нескольких наиболее распространенных платежных систем. Рассмотрены их особенности, преимущества и недостатки. На основании проведенного анализа принято решение для реализации собственного проекта использовать внешнюю платежную систему PayPal, так как она удовлетворяет по показателям защиты, удобству оплаты через сервис и библиотекам доступными для разных языков программирования.

#### ЛИТЕРАТУРА

1. Электронная коммерция. [Электронный ресурс] / Wikipedia. – Режим доступа: [https://ru.wikipedia.org/wiki/Электронная\\_коммерция](https://ru.wikipedia.org/wiki/Электронная_коммерция). – Дата доступа: 18.09.2019.
2. Система электронных платежей. [Электронный ресурс] / Wikipedia. – Режим доступа: [https://ru.wikipedia.org/wiki/Система\\_электронных\\_платежей](https://ru.wikipedia.org/wiki/Система_электронных_платежей). – Дата доступа: 18.09.2019.

УДК 004.056.5

**ОСНОВНЫЕ АСПЕКТЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ  
В МЕДИЦИНСКИХ ИНФОРМАЦИОННЫХ СИСТЕМАХ****Е.А. МЕНИЦКИЙ***(Представлено: канд. физ.-мат. наук., доц. Д.Ф. ПАСТУХОВ)*

*В статье представлены основные аспекты архитектурных особенностей по предоставлению безопасности и защищенности конфиденциальных данных в медицинских информационных системах. Рассмотрен алгоритм работы технологии распределения прав доступа к медицинской информации.*

**Ключевые слова:** *информационные технологии, медицинская информационная система, шифрование данных, защищенность данных.*

**Введение.** С увеличением объема концентрируемой информации в сфере медицины, возникает потребность в формировании стандартизированных и единообразных систем хранения данной информации. Данный вопрос помогает решить разработка специализированных медицинских информационных систем.

Медицинская информационная система отличается от других программных продуктов, прежде всего тем, что в ней хранится и обрабатывается персональная и конфиденциальная информация. В связи с этим к ним выдвигаются повышенные требования к достоверности и ограничениям доступа к информации, юридической ответственности, техническим мерам защиты данных. Любой пользователь, получающий доступ к медицинской информационной системе, несет полную ответственность за конфиденциальность информации, которую он вносит, использует или передает другим пользователям.

**Методы защиты информации в медицинских информационных системах.** На данный момент в медицинских информационных системах стоит вопрос безопасности информации в двух точках зрения:

- защита прав личности от распространения конфиденциальной информации;
- защита интересов государства и ведомств. Возможность утечки информации, злоупотребление, нарушение этики.
- исходя из указанных требований для обеспечения защиты информации и программ медицинских информационных систем должны применяться следующие средства [1]:
- правовые;
- организационно-административные;
- технические (аппаратно-программные).

Правовые средства препятствуют несанкционированному использованию информации и являются сдерживающим фактором для потенциальных нарушителей.

Сюда нужно отнести те права и обязанности по получению, обработке, ограничению в распространении информации, которые прописаны в законодательной базе государства, которому подчиняется действия по работе с медицинской информационной системой.

Для обеспечения информации, этой законодательной базе должны подчиняться как медицинские информационные системы, так и все участники принимающие участия в работе с ней.

Организационно-административные средства регламентируют процессы функционирования медицинских информационных систем, использование ее ресурсов, деятельность персонала, а также порядок взаимодействия пользователей с системой и пользователей с администраторами системы.

Исходя из этого, в медицинских информационных системах должна быть выстроена иерархия ролей, каждая из которых соотносится под различными должностями медицинских работников, сфер их деятельности, а также роли пациента. Каждая из таких ролей должна иметь ряд своих уникальных прав на взаимодействие с системой. Данное разграничение, служит для допуска к той части информации, которая соответствует компетенции пользователя. Данная особенность реализуется через технические средства защиты, и регламентируется правовыми средствами.

Технические средства выполняют следующие функции защиты: создание препятствий на возможных путях проникновения и доступа потенциальных нарушителей к медицинской информационной системе, идентификацию и авторизацию пользователей, разграничение прав доступа к ресурсам, регистрацию событий, криптографическую защиту информации [2].

Программно-технические меры системы безопасности медицинских информационных систем должны предоставлять средства распределения прав доступа, гарантируя возможность получения доступа пользователя только к той информации и программам, которые необходимы для выполнения функциональных обязанностей. Эти средства традиционно используют понятия:

- авторизация пользователя (технологии, подтверждающей, что реальный пользователь и персона, от чьего имени открывается доступ – одно и то же лицо);
- группы доступа (логического объединения пользователей в одну группу, для которой система предоставляет одинаковые права);
- права доступа (различия в возможностях работы с медицинской информационной системой, например – чтение информации и изменение данных, удаление документов или даже модификацию программного кода);
- списка контроля доступа (таблица, объединяющая группы доступа и сопоставленные им уровни прав для конкретного объекта системы).

**Алгоритм распределения прав доступа.** Алгоритм работы технологии распределения прав доступа в медицинской информационной системы выглядит следующим образом:

- При старте системы выполняется авторизация пользователя. Для авторизации, зачастую используется связка логин-пароль, где логин идентифицирует пользователя, а пароль является средством подтверждения личности. При этом пароль должен быть представлен в зашифрованном виде, при хранении в системе. Наиболее часто используемые алгоритмы AES, DES, Blowfish и другие. По мимо этого, при работе по сети, необходимо обеспечить передачу пароля по защищенным протоколам, такими как SSL – протокол передачи информации, использующий асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений [3].

– В момент первого после авторизации обращения к серверу создается сеанс связи с сервером. Во время инициализации сеанса система определяет, в какие группы данный пользователь входит и однозначно ассоциирует пользователя с этими группами. Все дальнейшие действия в системе, включая открытие БД, отображение ее элементов дизайна или других программных элементов, осуществляться только исходя из текущего списка групп доступа.

- При последовательном запросе клиентом объектов системы на стороне сервера осуществляется проверка на наличие или отсутствие необходимых прав доступа к объекту. Если ни одна из групп пользователя не включена в объект, то сервер не предоставляет клиенту информации о наличии и свойствах объекта. Учитывая высший приоритет системы безопасности, несанкционированный доступ к такому объекту (серверу, БД, представлению, программе, документу или отдельно взятому полю) становится теоретически и практически невозможным.

– При этом в системе на уровне ядра предусмотрены функции протоколирования несанкционированного доступа и программной обработки исключительных ситуаций в коде системы, которые одновременно позволяют администратору видеть все подозрительные с точки зрения безопасности запросы и, с другой стороны, обеспечивают требуемый уровень стабильности и производительности работы приложений системы.

Помимо распределения прав доступа, программно-технические меры системы безопасности должны предусматривать шифрование различной текстовой информации, а также мультимедии. Различные мультимедийные данные должны храниться в двоичном виде в базе данных с учетом алгоритмов шифрования, и представляется пользователю в виде расшифрованных данных в интерфейсе.

**Заключение.** Описанные технические, нормативные и иные средства обеспечения безопасности на сегодняшний день позволяют медицинским информационным системам обеспечивать весь необходимый комплекс мер защиты информации и программ, что является необходимым условием пригодности таких систем к их эксплуатации.

#### ЛИТЕРАТУРА

1. Информационные технологии в экономике. Методы и средства защиты информации. Авторы: Моисеенко Е.В., Лаврушина Е.Г., редактор: Л.З. Анипко, с. 25–30.
2. Cyberleninka.ru [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/algoritmy-i-tehnologii-obespecheniya-bezopasnosti-informatsii-v-meditsinskoj-informatsionnoy-sisteme-externet>. – Дата доступа: 25.09.2019.
3. SSL протокол [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/SSL>. – Дата доступа: 25.09.2019.

УДК 004.074.2

**ПОДХОДЫ К ОПТИМИЗАЦИИ СЖАТИЯ ИЗОБРАЖЕНИЙ  
БЕЗ ВИДИМОЙ ПОТЕРИ КАЧЕСТВА****Е.А. МЕНИЦКИЙ***(Представлено: канд. физ.-мат. наук., доц. Д.Ф. ПАСТУХОВ)*

*В статье представлены основные способы оптимизации изображения посредством оптимизации передаваемых параметров, поиска оптимальной логики передачи и хранения медиаданных, алгоритмы и инструменты для сжатия изображений.*

**Ключевые слова:** *информационные технологии, сжатие файлов, оптимизация размера данных, хранение данных.*

**Введение.** С увеличением объема получаемой информации, остро встает вопрос в необходимости иметь все большие вычислительные мощности по их хранению. В свою очередь, это приводит к дополнительным финансовым потерям на приобретение данных ресурсов. Конечно, всегда есть возможность по удалению неактуальной или неперспективной информации. Но данный способ не подходит для систем, которым в силу юридических, программных и других особенностей, необходимо хранения информации длительное время. Возьмем как пример для такого случая медицинскую информационную систему – как структуру с наиболее трудным взаимодействием с изображениями.

Большинство медицинских информационных систем юридически обязаны длительно хранить медицинские документы. При этом наиболее ресурсоемкая информация в данных системах – это изображения различных исследований (рентгенография, ультразвуковое исследование). В данном случае, очень важно качество получаемых изображений, для получения верного диагноза. Именно поэтому, данные системы не могут позволить себе обычное сжатие изображения с большой потерей качества. Отсюда возникает потребность по изучению подходов по оптимальному получению, хранению и сжатию изображений.

**Основные методы оптимального сжатия изображения без потерь.** Изначально необходимо решить вопрос о том, как будет происходить сжатие входного изображения: собственными средствами или сторонними сервисами. Если же все также говорить на примере медицинских информационных систем, то подойдет способ при разработке своего решения. Это даст большей гибкости при работе с программными инструментами, а также не допустит передачу конфиденциальных данных третьим лицам.

Практически любая логика по обработке изображений в приложении сводится к его сжатию. Сжатие данных – алгоритмическое преобразование данных, производимое с целью уменьшения занимаемого ими объема. Применяется для более рационального использования устройств хранения и передачи данных [1].

На данный момент можно выделить несколько основных направления по оптимизации сжатия данных:

- изменений программных параметров сжатия изображений;
- изменения в алгоритмической логике медиа приложения;
- изменения в энкодере.

**Изменений программных параметров сжатия изображений.** К данному методу относятся все возможности настройки приложений, устройств или языков программирования по изначальному получению наиболее подходящего изображения.

Примером может служить передаваемый параметр `optimize=True` при сохранении изображения посредством использования библиотеки Pillow для языка Python, которая реализует алгоритм Хаффмана – жадный алгоритм оптимального префиксного кодирования алфавита с минимальной избыточностью [2]. При наличии данного флага происходит дополнительный проход при сканировании каждого изображения. Каждый первый проход, вместо записи в файл, вычисляет статистику вхождений по каждому значению код-символ.

На рисунке 1 можно увидеть, как изменилось значения размера файла после применения флага `optimize=True`. Разница в 4657 Кб не является существенной для одного файла, но даст более ощутимый эффект при большом их количестве. При этом, разрешение самого файла остается исходным, т.е. не происходит потери качества.

Изменения в алгоритмической логике медиа приложения. Сюда необходимо отнести условия по работе только с несколькими форматами, определенными разрешениями или размерами. Зачастую эти ограничения устанавливаются программно и служат для приведения алгоритма программы к одному единственному процессу. Например, большинство пользовательского контента – JPEG и PNG форматы. JPEG отлично подходит для фотографий, но обычно не справляется с высококонтрастным дизайнерским контентом (таким как логотипы). В отличие от него, PNG сжимает изображение абсолютно без потерь, отлично

подходит для графики, но слишком громоздок для фотографий, где маленькие искажения все равно не заметны. Исходя из этого и из потребностей системы, можно сделать упор на загрузку изображений только в нужном формате. Например, в тех случаях, когда пользователи загружают фотографии в формате PNG, но при этом допустима некоторая потеря, мы можем сэкономить много места, если распознаем такие файлы и сохраним их предварительно сконвертировав в JPEG или вовсе ограничим их к загрузке.

```
>>> from PIL import Image
>>> import os
>>> foo = Image.open("image.jpg")
>>> print(os.stat('image.jpg').st_size)
12778
>>> foo.save("image.jpg",optimize=True)
>>> print(os.stat('image.jpg').st_size)
8121
>>>
```

Рисунок 1. – Пример сжатия изображения через флаг `optimize=True`

Еще одним способом оптимизации работы с изображениями можно назвать применение более современных форматов вроде JPEG2k. JPEG2k – графический формат, который вместо дискретного косинусного преобразования, применяемого в формате JPEG, использует технологию вейвлет-преобразования, основывающуюся на представлении сигнала в виде суперпозиции базовых функций – волновых пакетов. В результате такой компрессии изображение получается более гладким и четким, а размер файла по сравнению с JPEG при одинаковом качестве оказывается меньшим. JPEG2k полностью свободен от главного недостатка своего предшественника: благодаря использованию вейвлетов, изображения, сохраненные в этом формате, при высоких степенях сжатия не содержат артефактов в виде «решетки» из блоков размером 8x8 пикселей [3].

Конечно же, основным способом уменьшения размера файла будет являться изменение самого алгоритма сжатия. Рассмотрим этот способ на примере файлов JPEG.

Самый известный способ уменьшить размер файлов JPEG – настройка под названием `quality`. Многие приложения, способные сохранять в формате JPEG, определяют `quality` в виде числа от 0 до 100 и соответствуют различным таблицам квантования. Квантование – это один из шагов в процессе кодирования JPEG, когда теряется информация. Мы можем динамически изменять настройки качества, оптимизируя их для каждого отдельного изображения, чтобы достичь идеального баланса между качеством и размером, добавляя больше шумов, в том количестве, которые не будут видны человеческому глазу. Есть два способа сделать это:

- снизу вверх: Эти алгоритмы генерируют настроенные таблицы квантования, обрабатывая изображение на уровне блоков 8×8 пикселей. Они одновременно рассчитывают, сколько теоретического качества было потеряно и как эти потерянные данные усиливают или сокращают видимость искажений для человеческого глаза;

- сверху вниз: Эти алгоритмы сравнивают целое изображение с его оригинальной версией и определяют, сколько информации было потеряно. Последовательно генерируя кандидатов с различными настройками качества, мы можем выбрать того, который соответствует минимальному уровню оценки.

Данные алгоритмы позволяют оценивать изменения качества для отдельных блоков изображения или же путем создания изображений-кандидатов на разных уровнях качества, для последующей оценки падения качества каждого фрагмента изображения путем вычисления его индекса структурного сходства (SSIM) до тех пор, пока это значение находится в пределах настраиваемого, но статичного порога. Это позволяет нам выборочно понизить средний размер файла (и среднее качество) только для изображений, которые выше воспринимаемого порога.

Добавления данных подходов к алгоритмам обработки изображений, дает показатель `quality` в диапазоне от 80 до 85 (включительно). Это позволяет нам снизить средний размер файла без существенной потери в качестве изображения.

**Изменения в энкодере.** Большинство современных энкодеров изображений в своей реализации используют таблицы квантования по умолчанию данные в спецификации к формату, для сжатия файлов и конвертации в другие форматы. При этом всегда можно реализовать свои собственные улучшенные таблицы. Примером такой реализации может служить свободно распространяемый энкодер Mozjpeg разработанный компанией Mozilla [4]. Одно из отличий mozjpeg в том, что он использует альтернативную таблицу квантования. При это Mozjpeg оптимизирует как последовательные, так и прогрессивные JPEG, в среднем,

на 5% лучше, чем энкодер libjpeg-turbo, который используется во многих библиотеках по работе с изображениями. MozJPEG также сохраняя полную совместимость со стандартом JPEG и со всеми браузерами, что позволяет использовать набор инструментов этого энкодера в своих программах решениях, без существенной смены поддерживаемых форматов.

**Заключение.** Как видно из рассмотренной статьи, имеется ряд способов для оптимизации сжатия изображения популярных форматов. При этом многие из них, практически не создают значимых потерь в качестве. Такой результат достигается в основном пропорциональному добавлению потерь данных. Кажется бы, что при минимальных потерях, эффект от оптимизации будет также минимальным. Но если рассматривать данные способы для большого количества изображений, а также рассмотреть комбинации данных методов, то эффект от оптимизации может достигать значимых результатов и существенно влиять на ресурсопотребление системы.

#### ЛИТЕРАТУРА

1. Сжатие данных [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Сжатие\\_данных](https://ru.wikipedia.org/wiki/Сжатие_данных). – Дата доступа: 25.09.2019.
2. Код Хаффмана [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Код\\_Хаффмана](https://ru.wikipedia.org/wiki/Код_Хаффмана). – Дата доступа: 25.09.2019.
3. JPEG\_2000 [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/JPEG\\_2000](https://ru.wikipedia.org/wiki/JPEG_2000). – Дата доступа: 25.09.2019.
4. Mozilla Advances JPEG Encoding with mozjpeg 2.0 [Электронный ресурс]. Режим доступа: <https://research.mozilla.org/2014/07/15/mozilla-advances-jpeg-encoding-with-mozjpeg-2-0/>. – Дата доступа: 25.09.2019.

УДК 004.05

**АНАЛИЗ АРХИТЕКТУР РАЗРАБОТКИ ПРИЛОЖЕНИЙ****И.В. МИСЕВИЧ***(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)*

*В данной статье рассматриваются распространенные архитектурные подходы, при помощи которых ведется построение приложений.*

Наиболее распространенными архитектурами для построения приложений являются двух-, трех- и пятислойные архитектуры.

Двухслойная архитектура появилась в первой половине 90-х годов при проектировании приложений типа клиент-сервер. Суть заключается в том, что существует два слоя – клиент и сервер; на клиентской части располагаются средства представления данных, а также предметная логика взаимодействия с источником данных. Такие приложения еще называют толстыми клиентами. На уровне сервера располагается слой источника данных, например, СУБД. Клиент посылает серверу запросы на получение данных и получает ответ в виде данных или сообщений об ошибках. Приложения, состоящие из 2-х слоев, достаточно эффективны если необходимы данные и механизмы взаимодействия изменяются редко, а также нет необходимости в передаче данных на большие расстояния.

Трехслойная архитектура представляет собой эволюционное развитие двухслойной. В этой архитектуре предметная логика и представление разделены, что обеспечивает значительную гибкость разработки приложения, но увеличивает сложность. К преимуществам архитектуры можно отнести то, что слой представления не зависит от источника данных, а слой логики может быть адаптирован к различным видам представлений.

Однако быстрый рост количества пользователей и появившиеся вследствие этого дополнительные издержки стали снижать эффективность применения архитектуры приложения из трех слоев. Проблема состояла в том, что при добавлении дополнительного уровня представления или обновления хранилища данных приходилось вносить значительные изменения в предметную логику приложения. В некоторых случаях это вызывало ряд затруднений, и предметная логика состояла из различных частей, которые плохо взаимодействовали друг с другом. Единственным выходом из сложившейся ситуации было добавление новых уровней абстракции. Так появилась пятислойная архитектура распределенного приложения. В трехслойную архитектуру было введено два дополнительных слоя: контроллер/медиатор и слой сопоставления данных. Предназначение слоя контроллера заключается в том, что уровень представления обращается к слою контроллера за необходимыми порциями данных и полностью изолируется от логики предметной области. Таким образом, предметная логика могла быть сохранена для любого из представлений в неизменном виде. При добавлении нового представления достаточно было реализовать слой представления и реализовать дополнительный контроллер.

Если сравнивать по поддержанию бизнес-логики в актуальном состоянии для множества приложений и написанию дополнительного контроллера, то становится очевидно, что новое решение является более эффективным. Таким образом, слой контроллера надежно вписался в архитектуру распределенного приложения.

С серверами БД ситуация обстояла несколько хуже – промышленные сервера стоят дорого, поэтому с экономической точки зрения покупать для каждого приложения свой сервер БД невыгодно. При этом, используя трехслойную архитектуру, необходимо еще обеспечить соответствие предметной логики принципам хранилища данных, что в большинстве случаев делало приложения непереносимыми между различными серверами БД.

Решение проблемы взаимодействия слоев предметной логики и сервера БД заключается во введении дополнительного слоя – слоя сопоставления. Такой слой включает в свой состав механизмы отображения данных из БД, ориентированные на конкретную реализацию БД с одной стороны, а с другой стороны – обеспечение унифицированного интерфейса доступа со стороны слоя предметной области.

Приложения, построенные с использованием пятиуровневой архитектуры, показали значительно лучшую расширяемость при добавлении новых представлений или миграции на другие сервера БД. Предметная логика оставалась неизменной или дорабатывалась, но все изменения на этом уровне касались только уровней контроллера и сопоставления, но, как правило, эти изменения были достаточно простыми.

К недостаткам пятиуровневой архитектуры можно отнести еще большее увеличение сложности по сравнению с трехуровневой архитектурой.



Таким образом мы рассмотрели многоуровневые архитектуры разработки программ. Их достоинства и недостатки, и причины появления.

#### ЛИТЕРАТУРА

1. Блог программиста [Электронный ресурс] Многослойная архитектура. Определение и задачи. Режим доступа: <https://pro-prof.com/forums/topic/multilayered-architecture-definition-and-tasks>. Дата доступа: 26.09.2019.
2. Академик [Электронный ресурс] Трехуровневая архитектура. Режим доступа: <https://dic.academic.ru/dic.nsf/ruwiki/198141>. Дата доступа: 26.09.19.
3. Электронный университет ВГУ [Электронный ресурс] Трехзвенная архитектура с сервером приложений. Режим доступа: <https://edu.vsu.ru/mod/book/view.php?id=12520&chapterid=237>. Дата доступа: 26.09.2019.

УДК 004.05

## ЭФФЕКТИВНОСТЬ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ

*И.В. МИСЕВИЧ**(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)*

*В данной статье, рассматриваются причины эффективности автоматизации тестирования, а также факторы мешающие автоматизации быть эффективной на проекте.*

В данной статье, я хотел бы выразить свои мысли по эффективности автоматизации тестирования. Так все же, почему автоматизация выполняет эффективную роль на проекте? Существует ряд следующих причин:

1. **Возможность использовать автоматические тесты повторно.** Данный фактор, несомненно, экономит временные и финансовые ресурсы компании, ведь запустить сборку автоматических тестов гораздо быстрее чем выполнять их вручную. На данный момент машинные ресурсы в большинстве своем проведут проверку сравнительно быстрее человека.

2. **Отсутствие влияние человеческого фактора.** О данном аспекте можно сказать то, что любой человек, не зависимо от своего опыта в тестировании, может допустить ошибку, пропустить некое условие, упустить малозаметное расхождение. Автоматический же тест, первоначально имеет готовые сценарий, от которого он не может отойти, что исключает факт чего-то не учесть, а также проверяет равно те полученные данные с ожидаемыми, учитывая эквивалентность, исключение схожей альтернативы, форматы и т.д., а также может проверить все это с учетом больших объемов данных, в которых человек может просто запутаться. По мимо этого, сюда можно отнести фактор опыта человека при работе с отдельными подсистемами. Ведь при работе с новой для себя системой, человеку нужно время для обучения, при котором скорость его работы будет ниже. Автоматические тесты при этом знают правильный сценарий работы с системой изначально.

3. **Возможность генерации и проверки на большом объеме тестовых данных.** К примеру, основной процесс миграции данных изменяет тестовые данные, что не позволяет их использовать повторно. А так как телеком проекты – это множество различных подсистем, для формирования входных тестовых данных вручную инженеру приходилось обращаться к каждой из таких систем по отдельности, тратя на это много времени. Автоматические тесты при сами обращаются к подсистемам и прогоняют данные для их формирования в нужный вид.

4. **Оптимизация объема тестирования.** Данный фактор положительно влияет не только из-за возможности покрытия части функциональностью автоматическими тестами и уменьшить работу ручным тестировщикам, но и помогает на ранних этапах выявить не работающие подсистемы, при запуске автоматических тестов. Это довольно важно, так как зачастую невозможность проверить какой-то процесс, только после подготовки теста данных, проверки части условий и т.д., только расходует время.

5. **Регулярность запуска автоматических тестов.** Когда проект имеет большое количество разных подсистем, это требует их частого обновления. Именно поэтому автоматические тесты помогают ускорить проведение регрессионного тестирования, путем частых запусков.

Все названные аспекты влияют на повышении скорости тестирования при этом сохранении компетентности команды, как и при ручном тестировании, а это в свою очередь положительно влияет на экономическую составляющую.

По мимо этого, я выделил ряд наиболее существенных факторов, мешающих автоматизации быть эффективной на проекте:

1. **Нерабочее тестовое окружение.** Данный аспект не только зачастую полностью блокирует работу автоматизированных тестов, но и провоцирует частую смену окружения. Это в свою очередь ведет к изменению конфигураций тестов и трате дополнительного времени. Также при неработающем процессе блокирующим автоматический тест, приходится переключаться на другие задачи, а это в свою очередь влечет накопление задач у инженера по автоматизации.

2. **Отсутствие нужных тестовых данных.** Как уже говорилось ранее, многие процессы изменяют тесты данные, делая их не пригодными для повторного использования. При запуске множества автоматических тестов тестовые данные могут быстро израсходоваться. Также сюда следует отнести вероятность изменение одних данных, влияющих на другие.

3. **Частое изменение структуры подсистем.** Изменения в работе процессов, методов, возвращаемых и получаемых данных и т.д. заставляет постоянно прибегать к дополнительной изменению тестов, и в целом дополнительной их поддержке.

4. **Малая компетентность инженера по автоматизации тестирования в проекте.** На телекоммуникационных проектах число таких подсистем довольно большое и временные затраты на разбор их работы зачастую бывают довольно большие. Данный аспект часто заставляет прибегать к дополнительной помощи со стороны ручных инженеров, для разбора про корректности работы подсистемы. Поэтому для устранения данного негативного фактора, перед вводом в проект, инженеру по автоматическому тестированию, следует минимально изучить проект и его подсистемы.

Как мы можем видеть, наличие автоматизации тестирования на проекте, довольно положительно сказывается на процессе тестирования в целом. Однако следует учитывать, возможное наличие рисков. В большинстве случаев, большинство рисков помогает избежать предварительная подготовка инженеров и тестового окружения. При этом, по моему мнению, временные и финансовые затраты будут гораздо меньше, чем устранение рисков в дальнейшем.

#### ЛИТЕРАТУРА

1. Википедия [Электронный ресурс] / Автоматизированное тестирование – Режим доступа: [https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5\\_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). – Дата доступа: 26.09.2019.
2. habr [Электронный ресурс] / Автоматизация тестирования программных систем – Режим доступа: <https://habr.com/ru/post/160257>. – Дата доступа: 26.09.2019.
3. QA Academy [Электронный ресурс] / Автоматизация тестирования – Режим доступа: <https://qa-academy.by/qaacademy/news/avtomatizaciya-testirovaniya/>. – Дата доступа: 26.09.2019.

УДК 004.422.324

## ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ

Д.Д. МОРОЗОВ

(Представлено: канд. физ.-мат. наук., доц. О.В. ГОЛУБЕВА)

Статья вводит понятия генератора случайных чисел и генератора псевдослучайных чисел. Здесь приведены их принципиальные различия и требования, которым они должны удовлетворять. Приведены примеры их использования, в том числе комбинированного.

**Ключевые слова:** генератор псевдослучайных чисел, генератор случайных чисел.

**Генератор псевдослучайных чисел (ГСПЧ)** – алгоритм, порождающий последовательность чисел, элементы которого почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному). При этом от качества используемых ГСПЧ напрямую зависит качество получаемых результатов.

Наравне с существующей необходимостью генерировать легко воспроизводимые последовательности случайных чисел, также существует необходимость генерировать совершенно непредсказуемые или попросту абсолютно случайные числа. Такие генераторы называются **генераторами случайных чисел** (ГСЧ – англ. *random number generator, RNG*). Так как такие генераторы чаще всего применяются для генерации уникальных симметричных и асимметричных ключей для шифрования, они чаще всего строятся из комбинации криптостойкого ГПСЧ и внешнего источника энтропии (и именно такую комбинацию теперь и принято понимать под ГСЧ).

Простой ГСЧ с источником энтропии получится, если в качестве источника энтропии использовать текущее время, то для получения целого числа от 0 до N достаточно вычислить остаток от деления текущего времени в миллисекундах на число N+1. Недостатком этого ГСЧ является то, что в течение одной миллисекунды он выдает одно и то же число.

Источниками настоящих случайных чисел могут быть: физические шумы, такие, как детекторы событий ионизирующей радиации; дробовой шум в резисторе или космическое излучение. Устройства базирующиеся на таких источниках случайных чисел не находят большого применения в приложениях сетевой безопасности потому что имеют ряд недостатков:

- время и трудозатраты при установке и настройке по сравнению с программными ГПСЧ;
- дороговизна;
- генерация случайных чисел происходит медленнее, чем при программной реализации ГПСЧ;
- невозможность воспроизведения ранее сгенерированной последовательности случайных чисел.

В то же время случайные числа, получаемые из физического источника, могут использоваться в качестве порождающего элемента для программных ГПСЧ. Такие комбинированные генераторы применяются в криптографии, лотереях, игровых автоматах.

Поэтому, качественные требования, предъявляемые к ГПСЧ:

- достаточно длинный период, гарантирующий отсутствие заикливания последовательности в пределах решаемой задачи. длина периода должна быть математически доказана;
- эффективность – быстрота работы алгоритма и малые затраты памяти;
- воспроизводимость – возможность заново воспроизвести ранее сгенерированную последовательность чисел любое количество раз;
- портируемость – одинаковое функционирование на различном оборудовании и операционных системах.

Недостатком ГСПЧ является то, что никакой детерминированный алгоритм не может генерировать полностью случайные числа, он может только аппроксимировать некоторые их свойства. Любой ГПСЧ с ограниченными ресурсами рано или поздно заикливается – начинает повторять одну и ту же последовательность чисел.

Длина циклов ГПСЧ зависит от самого генератора. Если порождаемая последовательность ГПСЧ сходится к слишком коротким циклам, то такой ГПСЧ становится предсказуемым и непригодным для практических приложений.

Большинство простых арифметических генераторов хотя и обладают большой скоростью, но страдают от многих серьезных недостатков:

- слишком короткий период/периоды.
- последовательные значения не являются независимыми.

- некоторые биты «менее случайны», чем другие.
- неравномерное одномерное распределение.
- обратимость.

В частности, алгоритм RANDU, десятилетиями использовавшийся на мейнфреймах, оказался очень плохим, что вызвало сомнения в достоверности результатов многих исследований, использовавших этот алгоритм.

Почти все крупные производители микрочипов поставляют аппаратные ГСЧ с различными источниками энтропии, используя различные методы для их очистки от неизбежной предсказуемости. Однако на данный момент скорость сбора случайных чисел всеми существующими микрочипами (несколько тысяч бит в секунду) не соответствует быстродействию современных процессоров.

В современных исследованиях осуществляются попытки использования измерения физических свойств объектов (например, температуры) или даже квантовых флуктуаций вакуума в качестве источника энтропии для ГСЧ.

В персональных компьютерах авторы программных ГСЧ используют гораздо более быстрые источники энтропии, такие, как шум звуковой карты или счетчик тактов процессора. Сбор энтропии являлся наиболее уязвимым местом ГСЧ. Эта проблема до сих пор полностью не разрешена во многих устройствах (например, смарт-картах), которые таким образом остаются уязвимыми. Многие ГСЧ используют традиционные испытанные, хотя и медленные, методы сбора энтропии вроде измерения реакции пользователя (движение мыши и т. п.), как, например, в PGP и Yagrow, или взаимодействия между потоками, как, например, в Java SecureRandom.

Криптографические приложения используют для генерации случайных чисел детерминированные алгоритмы, следовательно, генерируют последовательность чисел, которая теоретически не может быть статистически случайной. В то же время, если выбрать хороший алгоритм, полученная численная последовательность – **псевдослучайных чисел** – будет проходить большинство тестов на случайность. Одной из характеристик такой последовательности является большой период повторения.

#### ЛИТЕРАТУРА

1. Материал из Википедии – свободной энциклопедии. ГСПЧ [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Генератор\\_псевдослучайных\\_чисел](https://ru.wikipedia.org/wiki/Генератор_псевдослучайных_чисел). – Дата доступа: 23.09.2019.
2. Материал из Википедии – свободной энциклопедии. ГПЧ [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D1%8B%D0%B9\\_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80\\_%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85\\_%D1%87%D0%B8%D1%81%D0%B5%D0%BB](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D1%8B%D0%B9_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB). – Дата доступа: 23.09.2019.

УДК 004.056.57

**ОСНОВНЫЕ РАЗНОВИДНОСТИ ВРЕДНОСНЫХ ПРОГРАММ  
И МЕРЫ ПРЕДОСТОРОЖНОСТИ ПРОТИВ НИХ****Д.Д. МОРОЗОВ***(Представлено: канд. физ.-мат. наук., доц. О.В. ГОЛУБЕВА)*

*В данной статье мы рассмотрим основные виды вредоносных программ и методы избежать заражения ими, потому что это проблема с которой сталкивался любой пользователь компьютера. Чем больше используется компьютер, тем больше риск заражения вредоносными программами.*

Давайте рассмотрим некоторые из наиболее распространенных типов вредоносных программ, которые можно встретить в Интернете.

– Вирусы – получили свое название за способность «заражать» множество файлов на компьютере. Они распространяются и на другие машины, когда зараженные файлы отправляются по электронной почте или переносятся пользователями на физических носителях, например, на USB-накопителях или (раньше) на дискетах.

– Черви – в отличие от вирусов, для распространения не требуются вмешательства человека: они заражают один компьютер, а затем через компьютерные сети распространяются на другие машины без участия их владельцев. Используя уязвимости сети, например, недостатки в почтовых программах, черви могут отправлять тысячи своих копий и заражать все новые системы, и затем процесс начинается снова. Помимо того, что многие черви просто «съедают» системные ресурсы, снижая тем самым производительность компьютера, большинство из них теперь содержит вредоносные «составляющие», предназначенные для кражи или удаления файлов.

– Рекламное ПО – одним из наиболее распространенных типов вредоносных программ является рекламное ПО. Программы автоматически доставляют рекламные объявления на хост-компьютеры. Среди разновидностей Adware - всплывающие рекламные объявления на веб-страницах и реклама, входящая в состав «бесплатного» ПО.

– Шпионское ПО – делает то, что предполагает его название - следит за вашими действиями на компьютере. Оно собирает информацию (например, регистрирует нажатия клавиш на клавиатуре вашего компьютера, отслеживает, какие сайты вы посещаете и даже перехватывает ваши регистрационные данные), которая затем отправляется третьим лицам, как правило, киберпреступникам.

– Программы-вымогатели – заражают ваш компьютер, затем шифруют конфиденциальные данные, например, личные документы или фотографии, и требуют выкуп за их расшифровку. Если вы отказываетесь платить, данные удаляются.

– Боты – это программы, предназначенные для автоматического выполнения определенных операций. Они могут использоваться для легитимных целей, но злоумышленники приспособили их для своих вредоносных целей. Проникнув в компьютер, боты могут заставить его выполнять определенные команды без одобрения или вообще без ведома пользователя.

– Руткиты – позволяют третьей стороне получать удаленный доступ к компьютеру и управлять им. Эти программы используются IT-специалистами для дистанционного устранения сетевых проблем. Но в руках злоумышленников они превращаются в инструмент мошенничества: проникнув в ваш компьютер, руткиты обеспечивают киберпреступникам возможность получить контроль над ним и похитить ваши данные или установить другие вредоносные программы.

– Троянские программы – более известные как троянцы, эти программы маскируются под легитимные файлы или ПО. После скачивания и установки они вносят изменения в систему и осуществляют вредоносную деятельность без ведома или согласия жертвы.

Между тем, рост количества устройств, взаимодействующих друг с другом в Интернете Вещей (IoT), открывает дополнительные возможности: зараженный автомобиль съедет с дороги, или зараженная «умная» печь продолжит нагреваться, пока не случится превышение нормальной нагрузки. Вредоносное ПО будущего может сделать такой физический ущерб реальностью.

У пользователей есть ряд неправильных представлений о вредоносных программах: например, многие считают, что признаки заражения всегда заметны и поэтому они смогут определить, что их компьютер заражен. Однако, как правило, вредоносное ПО не оставляет следов, и ваша система не будет показывать каких-либо признаков заражения.

Так же не стоит верить, что все сайты с хорошей репутацией безопасны. Они также могут быть взломаны киберпреступниками. А посещение зараженного вредоносным кодом легитимного сайта – еще большая вероятность для пользователя потерять свою личную информацию. Также многие пользователи

считают, что их личные данные - фотографии, документы и файлы - не представляют интереса для создателей вредоносных программ. Киберпреступники же используют общедоступные данные для того, чтобы атаковать отдельных пользователей, или собрать информацию, которая поможет им создать фишинговые письма, чтобы проникнуть во внутренние сети организаций.

**Стандартные методы заражения.** Существует несколько стандартных способов заражения компьютерными вирусами или вредоносными программами. Это ссылки на вредоносные сайты в электронной почте или сообщениях в социальных сетях, посещение зараженного сайта (известного как drive-by загрузка) и использование зараженного USB-накопителя на вашем компьютере. Уязвимости операционной системы и приложений позволяют злоумышленникам устанавливать вредоносное ПО на компьютеры. Поэтому для снижения риска заражения очень важно устанавливать обновления для систем безопасности, как только они становятся доступными.

Киберпреступники часто используют методы социальной инженерии, чтобы обманом заставить пользователя делать что-то, что угрожает его безопасности или безопасности его компании. Фишинговые сообщения являются одним из наиболее распространенных методов. Цель хакеров - написать сообщение так, чтобы оно было убедительным. Это может быть, например, предупреждение о возможном вирусном заражении или уведомление из банка или сообщение от старого друга.

Конфиденциальные данные, такие как пароли, являются главной целью киберпреступников. Помимо использования вредоносных программ для перехвата паролей в момент их ввода, злоумышленники также могут собирать пароли с веб-сайтов и других компьютеров, которые они взломали. Вот почему так важно использовать уникальный и сложный пароль для каждой учетной записи. Он должен состоять из 15 и более символов, включающих буквы, цифры и специальные символы. Таким образом, если киберпреступникам удастся взломать один аккаунт, они не получат доступ ко всем вашим учетным записям. К сожалению, большинство пользователей имеют очень слабые пароли: вместо того, чтобы придумать труднодоступную комбинацию, они обращаются к standby-паролям типа «123456» или «Password123», которые преступники легко подбирают. Даже контрольные вопросы не всегда могут служить эффективной защитой, потому что многие люди дают один и тот же ответ на вопрос.

**Признаки заражения.** Хотя большинство вредоносных программ не оставляет никаких явных следов, и ваш компьютер работает нормально, иногда все же можно заметить признаки возможного заражения. Самый первый из них - снижение производительности, т.е. процессы происходят медленные, загрузка окон занимает больше времени, в фоновом режиме работают какие-то случайные программы. Еще одним нарастающим признаком может считаться измененных домашних интернет-страниц в вашем браузере или более частое, чем обычно, появление всплывающих объявлений. В некоторых случаях вредоносное ПО даже может влиять на базовые функции компьютера: не открывается Windows, нет подключения к Интернету или доступа к более высокоуровневым функциям управления системой более высокого уровня. Если вы подозреваете, что ваш компьютер может быть заражен, немедленно произведите проверку системы.

## ЛИТЕРАТУРА

1. Материал из Википедии – свободной энциклопедии. Компьютерные вирусы [Электронный ресурс] Режим доступа: [https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9\\_%D0%B2%D0%B8%D1%80%D1%83%D1%81](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B2%D0%B8%D1%80%D1%83%D1%81). – Дата доступа: 23.09.2019.
2. Сибирикс [Электронный ресурс] – Режим доступа: <https://blog.sibirix.ru/2017/08/16/malware/> – Дата доступа: 23.09.2019.
3. Home-engineer [Электронный ресурс] – Режим доступа: <http://www.home-engineer.ru/interesnoe-i-poleznoe-pro-computers/vidy-kompyuternyx-virusov/> – Дата доступа: 23.09.2019.

УДК 004.35

**НАЗНАЧЕНИЕ УСТРОЙСТВА УПРАВЛЕНИЯ ПРИБОРАМИ GPIB.  
АКТУАЛЬНОСТЬ РАЗРАБОТКИ УСТРОЙСТВА УПРАВЛЕНИЯ ПРИБОРАМИ ПЗШИ  
В СОВРЕМЕННОМ МИРЕ****Е.Н. ПАЦАНКОВ**  
(Представлено: И.С. РУСЕЦКИЙ)

*В статье рассматривается назначения устройства управления приборами GPIB. Проведен анализ технологий, наиболее подходящих, для разработки данного устройства. Проведены исследования по актуальности разработки данного устройства.*

**Введение.** Бурное развитие электронных устройств и расширение возможностей измерительных приборов практически сразу поставили вопрос о взаимодействии между контрольно-управляющей и измерительной аппаратурой с компьютерной техникой. Компания Hewlett-Packard в середине 60-х годов представила интерфейс Hewlett-Packard Interface Bus (HP-IB) как многоцелевой контроллер [1]. Бурно растущий рынок и значительное расширение сферы применений контрольно-измерительной аппаратуры потребовали возможностей комбинирования вычислительной мощности компьютера и конечного измерительного прибора. Интерфейс HP-IB стал удачным и гармоничным решением этой задачи.

В 1970-х стандарт HP-IB был преобразован в более общий GPIB (General Purpose Interface Bus – интерфейсная шина общего назначения), а также был принят как стандарт IEEE-488.

Во многих технических лабораториях все еще имеются «устаревшие» измерительные приборы, примерно 1980-х годов, которые обладают превосходными измерительными свойствами, но в современном мире не интегрируемые с ПК из-за отсутствия современных интерфейсов передачи данных. Эти интерфейсы обеспечивают дистанционное управление и сбор данных, обычно через USB или Ethernet.

Многие современные измерительные приборы оснащены интерфейсом IEEE488, который на Западе известен как GPIB, а в России как КОП (канал общего пользования по ГОСТ 26.003-80). Он позволяет объединять приборы в автоматизированные измерительные комплексы. Но для управления таким комплексом нужен компьютер, оборудованный адаптером этого интерфейса. В типовой комплектации большинства персональных компьютеров и других устройств он отсутствует, а как самостоятельное изделие стоит недешево.

Разрабатываемая система позволит использовать одноплатный микрокомпьютер Raspberry Pi для управления каналом общего пользования и обменом по нему информацией с измерительными приборами, оснащенными данным интерфейсом.

**Основной раздел. Приборный интерфейс КОП (GPIB)**

Простота использования, постоянное развитие аппаратной поддержки GPIB, разработка новых GPIB-совместимых приборов приводит к росту числа пользователей шины, несмотря на мощную конкуренцию со стороны архитектур VMEbus и др. Поскольку шина GPIB полностью стандартизована и протестирована, большинство производителей автоматизированных измерительных систем и приборов встраивают в свои изделия интерфейсы GPIB в качестве основного канала передачи данных.

GPIB-магистраль состоит из 16 сигнальных линий и 8 линий заземления или возврата сигнала представлена на рисунке 1.

Магистраль разбита на три шины. Сигнальные линии группируются следующим образом: 8 линий входят в шину данных, 3 линии – в шину согласования (протокола) и 5 линий – в шину управления интерфейсом. Шина данных (Data Lines) имеет 8 линий (DIO1...DIO8) и предназначена для передачи байта информации (данные и команды интерфейса). Шина управления (Bus Control Lines) содержит 5 линий (ATN, IFC, SRQ, REN, EOI), назначение которых будет разъяснено ниже. Шина синхронизации (Handshake Lines) содержит 3 линии (DAV, NRFD, NDAC), используемых для квитирования передачи байта информации. Сигналы на линиях имеют отрицательную TTL логику (нулю соответствует напряжение >2,5 В, единице – 0,8 В). Передача информации идет последовательно побайтно по линиям DI0-DI7 в асинхронном режиме.

Стандарт GPIB определяет три различных типа устройств, которые могут быть подключены к шине: «listener» (слушатель, приемник), «talker» (передатчик, источник) и/или контроллер. Устройства, подключенные к системе, могут менять свое состояние («listener» либо «talker») по командам с модуля, имеющего состояние «контроллер». Устройство в состоянии «listener» считывает сообщения с шины; устройство в состоянии «talker» посылает сообщения на шину. В каждый момент времени в состоянии «talker» может быть одно и только одно устройство, в то время как в состоянии «listener» может быть произвольное количество устройств. Магистраль одновременно может обслуживать до 15 устройств с адресами от 0 до 30 включительно. Контроллер выполняет функции арбитра и определяет, какие из устройств в данный момент находятся в состоянии «talker» и «listener».



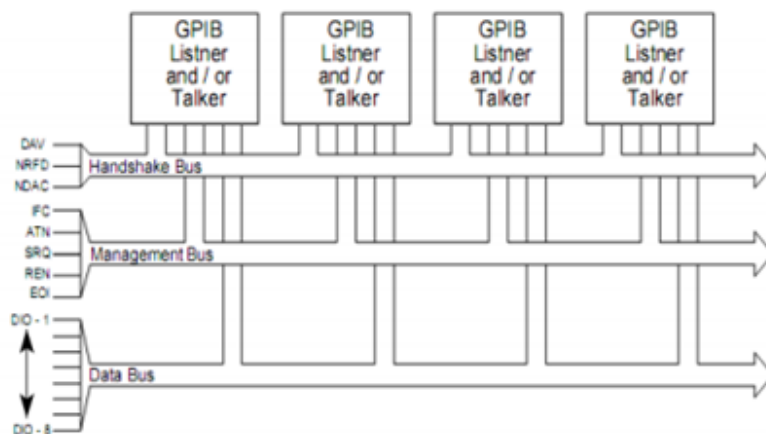


Рисунок 1. – GPIB-магистраль

Обозначение контактов разъема GPIB приведено в таблице 1.

Таблица 1. – Обозначение контактов разъема GPIB

№ п/п	Мнемоника	Название	Функция
1	DIO1	DIO1	Данные (младший бит)
2	DIO2	DIO2	Данные
3	DIO3	DIO3	Данные
4	DIO4	DIO4	Данные
5	EOI	End of Identify (конец идентификации)	Управление интерфейсом
6	DAV	Data Valid (данные готовы)	Рукопожатие
7	NRFD	Not Ready for Data (не готов к приему)	Рукопожатие
8	NDAC	Not Data Accepted (данные не приняты)	Рукопожатие
9	IFC	Interface Clear (очистить интерфейс)	Управление интерфейсом
10	SRQ	Service Request (запрос обслуживания)	Управление интерфейсом
12	–	Shield (экран)	Шасси
13	DIO5	DIO5	Данные
14	DIO6	DIO6	Данные
15	DIO7	DIO7	Данные
16	DIO8	DIO8	Данные (старший бит)
17	REN	Remote Enable (разрешить работу в удаленном режиме)	Управление интерфейсом
18	–	DAV Return (возврат DAV)	Шасси
19	–	NRFD Return (возврат NRFD)	Шасси
20	–	NDAC Return (возврат NDAC)	Шасси
21	–	IFC Return (возврат IFC)	Шасси
22	–	SRQ Return (возврат SRQ)	Шасси
23	–	ATN Return (возврат ATN)	Шасси
24	–	Signal Ground (земля данных)	Шасси

Назначение линий шины управления интерфейса:

1. ATN – Attention, (УП – управление) применяется для кодирования типа передаваемой информации. Контроллер устанавливает линию ATN в логическую 1 при посылке команд, и в логический 0, когда источник посылает данные.

2. IFC – Interface Clear (ОИ – очистка интерфейса) используется системным контроллером для инициализации или реинициализации шины. При этом производится установка всех интерфейсных карт в исходное состояние.

3. REN – Remote Enable (РУ – дистанционное управление) переводит приборы, подключенные к шине, в режим выполнения команд с шины (а не с передней панели прибора) и обратно. Используется в реализации интерфейсной функции RL.

4. SRQ – Service Request (ЗО – запрос на обслуживание) используется в реализации функции SR, рассмотренной ниже. Любой прибор может установить линию SRQ для асинхронного запроса на обслуживание контроллером.

5. EOI – End of Identify (КП –конец передачи) – использует источник для указания на последний байт сложного сообщения. Контроллер выставляет этот сигнал для инициации параллельного опроса подключенных к шине приборов при реализации функции PP. Назначение линий шины синхронизации:

- DAV – Data Valid (СД – сопровождение данных) – выдается передатчиком при передаче данных;
- NRFD – Not Ready for Data (ГП – готовность к приему) выдается приемником при его неготовности;
- NDAC – Not Data Accepted (ДП – данные приняты) – устанавливается приемником при отсутствии приема данных. Эти сигналы используются для синхронизации передачи данных по шине данных, изображенном на рисунке 2.

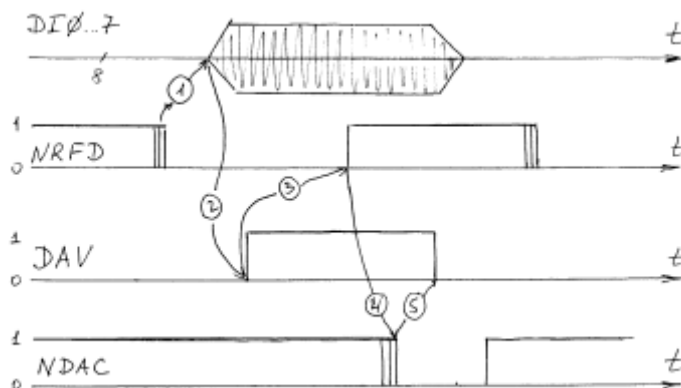


Рисунок 2. – Передача байта по шине GPIB

Исходное положение при передаче данных – все приемники выставляют нулевой сигнал NRFD (1). После этого передатчик устанавливает байт на шине данных и подтверждает его достоверность сигналом DAV=1 (2). Некоторое время идет процесс приема данных, приемник устанавливает при этом сигнал неготовности к новому приему (3). Конец приема квитируется установкой сигнала NDAC=0 (4). После получения этого сигнала от всех приемников передатчик сбрасывает сигнал достоверности данных DAV = 0 (5) и снимает байт данных. После этого приемник восстанавливает сигнал NDAC=1. Система приходит в исходное состояние и процесс повторяется [1].

Одноплатный микрокомпьютер Raspberry Pi. Основным преимуществом использования одноплатного микрокомпьютера Raspberry Pi является его габаритные размеры. Данный микрокомпьютер в полной мере подходит по системным характеристикам для обработки и обмена данными, через устройство управления с измерительными приборами через GPIB интерфейс. Для написания программного обеспечения используется язык программирования Python. Данный язык достаточно простой и в полной мере подходит для реализации поставленной задачи [2].

Актуальность разработки устройства управления приборами GPIB. Данные «устаревшие» измерительные приборы обладают превосходными измерительными свойствами, которые не уступают, а в чем-то и лучше современных измерительных устройств. Однако существующие на рынке GPIB адаптеры имеют очень большую цену. Однако существует возможность разработать устройство, связующее ЭВМ с измерительными устройствами через интерфейс GPIB, гораздо дешевле и не хуже по качеству зарубежных адаптеров.

**Заключение.** В данной статье были рассмотрены назначение устройства управления приборами GPIB, а также актуальность данного устройства в современном мире. Рассмотрен интерфейс GPIB и обмен данными GPIB прибора с ЭВМ.

#### ЛИТЕРАТУРА

1. GPIB интерфейс [Электронный ресурс] – режим доступа: <http://kepstr.eltech.ru/tor/ptri/Literatura/Interf.pdf>. – Дата доступа: 24.09.2019.
2. RaspberryPi model 3 [Электронный ресурс] – режим доступа: <https://micro-pi.ru/raspberry-pi-3-model-b-pi-bcm2837/>. – Дата доступа: 24.09.2019.

УДК 004.35

## РАЗРАБОТКА УСТРОЙСТВА УПРАВЛЕНИЯ ПРИБОРАМИ GPIB

**Е.Н. ПАЦАНКОВ***(Представлено: И.С. РУСЕЦКИЙ)*

*В статье представлено разработка устройства управления приборами GPIB. Проведен анализ технологий, наиболее подходящих, для разработки данного устройства. Представлен алгоритм работы устройства, функциональная схема устройства управления приборами GPIB. Проведен анализ технологий, наиболее подходящих, для разработки данного устройства.*

**Введение.** Бурное развитие электронных устройств и расширение возможностей измерительных приборов практически сразу поставили вопрос о взаимодействии между контрольно-управляющей и измерительной аппаратурой с компьютерной техникой. Компания Hewlett-Packard в середине 60-х годов представила интерфейс Hewlett-Packard Interface Bus (HP-IB) как многоцелевой контроллер [1]. Бурно растущий рынок и значительное расширение сферы применений контрольно-измерительной аппаратуры потребовали возможностей комбинирования вычислительной мощности компьютера и конечного измерительного прибора. Интерфейс HP-IB стал удачным и гармоничным решением этой задачи.

В 1970-х стандарт HP-IB был преобразован в более общий GPIB (General Purpose Interface Bus – интерфейсная шина общего назначения), а также был принят как стандарт IEEE-488.

Во многих технических лабораториях все еще имеются «устаревшие» измерительные приборы, примерно 1980-х годов, которые обладают превосходными измерительными свойствами, но в современном мире не интегрируемые с ПК из-за отсутствия современных интерфейсов передачи данных. Эти интерфейсы обеспечивают дистанционное управление и сбор данных, обычно через USB или Ethernet.

Многие современные измерительные приборы оснащены интерфейсом IEEE488, который на Западе известен как GPIB, а в России как КОП (канал общего пользования по ГОСТ 26.003-80). Он позволяет объединять приборы в автоматизированные измерительные комплексы. Но для управления таким комплексом нужен компьютер, оборудованный адаптером этого интерфейса. В типовой комплектации большинства персональных компьютеров и других устройств он отсутствует, а как самостоятельное изделие стоит недешево.

Разрабатываемая система позволит использовать одноплатный микрокомпьютер Raspberry Pi для управления каналом общего пользования и обменом по нему информацией с измерительными приборами, оснащенными данным интерфейсом.

**Основной раздел***Алгоритм работы устройства.*

На основании технического задания разработана блок-схема алгоритма работы устройства приведенная на рисунке 1.

Выход из бесконечного цикла. Данное условие проверяет завершен ли бесконечный цикл. Если бесконечный цикл завершен, то программа прекращает работу. Если не завершен, программа продолжает работать согласно алгоритму.

Принят запрос на обработку данных от устройства Slave. Данное условие проверяет готово ли устройство Slave обрабатывать данные. Если да, то осуществляется выбор исполняемой функции и ее исполнение. Если нет, то запускается условие на проверку «Готовы ли данные для передачи от устройства Master».

Готовы ли данные для передачи от устройства Master. Если данные готовы для передачи, то осуществляется выбор исполняемой функции и исполнение данной функции. Если нет, то снова запускается проверка на готовность устройства Slave для обработки данных.

Выбор исполняемой функции. В данном блоке осуществляется выбор исполняемой функции.

*Функциональная схема устройства*

Функциональная схема данного устройства разрабатывалась на основе технического задания и анализа функций и задач, которые должно выполнять и реализовывать проектируемое устройство. Спроектированная функциональная схема представлена на рисунке 2.

Функциональная схема состоит из следующих блоков: блок питания, Raspberry Pi, разъем GPIO, буферы данных, буферы управления, индикатор состояния, индикатор питания, разъем GPIB и GPIB устройства.

Блок питания предназначен для питания Raspberry Pi и имеет характеристики напряжения 5 В, ток более 2 А.

Raspderry PI предназначен для управления и получения данных с GPIB устройства. С Raspderry PI питается вся схема: буферы данных и управления, индикаторы состояния и питания.

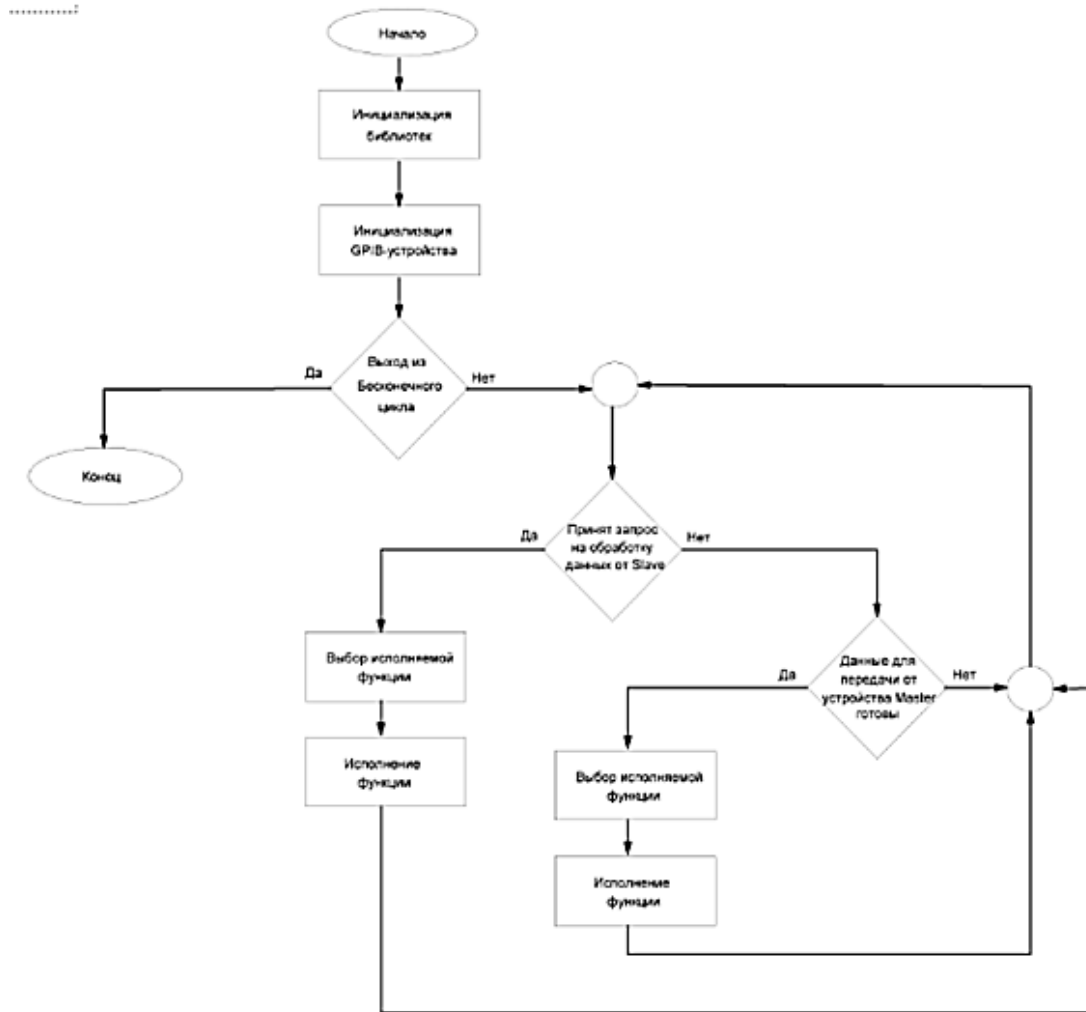


Рисунок 1. – Блок-схемы алгоритма работы устройства

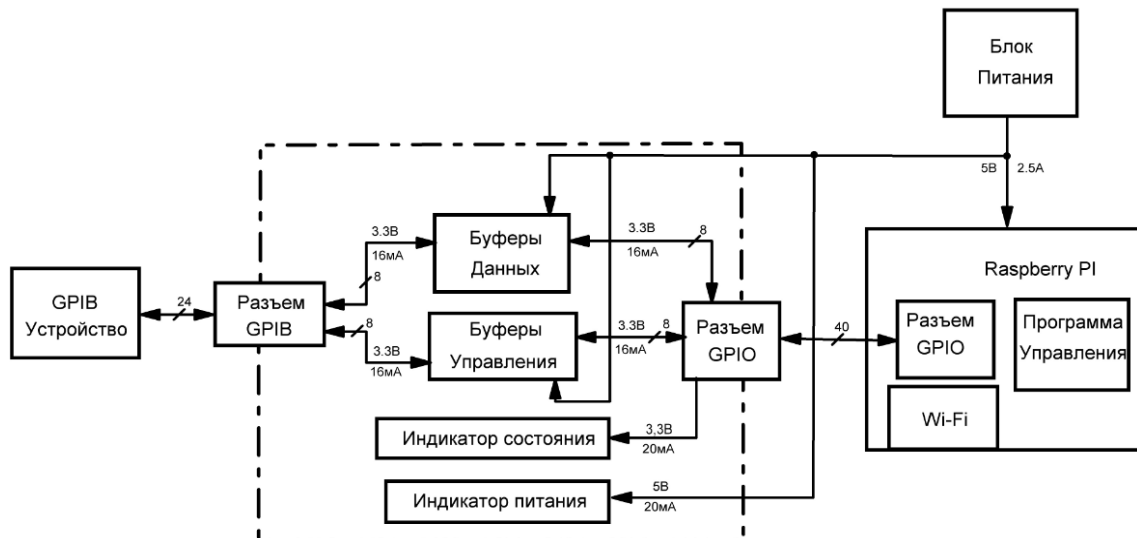


Рисунок 2. – Схема электрическая функциональная

Буферы управления питаются от 5 В с током потребления до 15 мА. С Raspberry PI через разъем GPIO и буфер управления в GPIB устройство поступает сигнал о том, что он готов принять информацию. После чего ожидается готовность информации с GPIB устройства для ее передачи. Сигналы приходят с платы Raspberry PI через разъем GPIO напряжением 3,3 В и током 16 мА.

Буферы данных питаются от 5 В. С Raspberry PI через разъем GPIO и буфер управления в GPIB устройство поступает сигнал о том, что он готов принять информацию. После чего ожидается готовность информации с GPIB устройства для ее передачи. С GPIB устройства через буфер данных начинается передача данных. Сигналы приходят с платы Raspberry PI через разъем GPIO напряжением 3,3 В и током 16 мА.

Индикатор питания указывает, что на плату приходит питающее напряжение 5 В. Горящий или мигающий индикатор состояния обозначает, что идет передача данных с GPIB устройства [2].

**Заключение.** В данной статье были рассмотрены устройства управления приборами по интерфейсу GPIB. Разработаны алгоритм работы и функциональная схема устройства управления приборами GPIB.

#### ЛИТЕРАТУРА

1. GPIB интерфейс [Электронный ресурс] – режим доступа: <http://kepstr.eltech.ru/tor/ptri/Literatura/Interf.pdf>. – Дата доступа: 24.09.2019.
2. Проектирование функциональной схемы устройства [Электронный ресурс] – режим доступа: [https://github.com/elektronomikon/raspi\\_gpib\\_shield](https://github.com/elektronomikon/raspi_gpib_shield). – Дата доступа: 25.09.2019.

УДК 004.04

## РЕАЛИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ОНЛАЙН-ИГРОЙ

С.Д. ПАШКЕВИЧ

(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

Предлагается подход к реализации системы управления многопользовательской онлайн-игрой, как распределенного веб-приложения, взаимодействие компонентов которого осуществляется согласно принципам архитектурного стиля REST. Описываются функциональные возможности системы управления и ее структурные компоненты. Приводится диаграмма компонентов веб-приложения.

**Введение.** Несмотря на популярность игр, ориентированных на компьютеры, мобильные многопользовательские онлайн-игры в последние годы набирают все большую популярность. Рост аппаратных характеристик мобильных устройств за последние пять лет расширил возможности разработчиков игр, позволяя создавать намного более крупные и амбициозные проекты [1]. Зачастую онлайн игры требуют наличие специальной функциональности, которая выносится в отдельные приложения, позволяющие осуществлять администрирование игрового процесса.

Системы управления предназначены не только для осуществления контроля за игровым процессом, но и для получения статистической информации, необходимой для анализа данных, а также изучения интересов игроков.

Администратор приложения получает возможность самостоятельно определять права пользователей, а также редактировать, формировать либо изменять структуру отображаемой информации, осуществлять техническую поддержку пользователей и рассылку новостей. Пользователь, в свою очередь, получает возможность редактировать свою личную информацию, следить за своей игровой статистикой и статистикой других игроков, обратиться за помощью к администраторам игры и так далее. Перечисленные возможности значительно облегчают поддержку и дальнейшее развитие игры в целом, а также обуславливают необходимость в наличии системы управления, так как дополнительная функциональность для администрирования не относится к игровому процессу напрямую и ее наличие не всегда удобно в рамках игрового приложения, особенно мобильного.

Игровой мир представляет собой галактику, которая состоит из множества секторов, которые содержат определенное количество планет. На планетах игрок может брать квесты, покупать/продавать товары, улучшать оборудование и производить починку космического корабля, с помощью которого осуществляется перемещение по игровому миру. Каждому игроку предоставляется персонаж, имеющий следующие характеристики: точность пилота, маневренность пилота, знание техники, навык торговли, лидерство, обаяние пилота. После выполнения заданий, либо участия в сражениях, игроку начисляется опыт, который может быть использован для улучшения характеристик своего персонажа. Галактику населяют три расы: зеленые, красные и синие. Персонаж каждой расы обладает характерными для нее свойствами. Игроку также предлагается возможность выбора расы.

**Система управления многопользовательской онлайн-игрой.** Функциональность, предоставляемая системой управления, обуславливается описанием игры. Страница *регистрации и авторизации* является входной точкой в приложение, позволяет пользователю зарегистрироваться, либо авторизоваться через популярные социальные сети: Facebook, Google+, Twitter. После успешного прохождения процедуры авторизации/регистрации пользователю будут присвоены специальные права доступа, которые могут ограничивать либо предоставлять доступ к различным ресурсам системы управления, например, таким как *админпанель*, также пользователю станет доступна страница *профиль пользователя*.

Страница *профиль пользователя* отображает раздел *игровой статистики*, раздел *личной информации*, раздел *управления игровым процессом* и *меню*, которое предоставляет возможность перемещаться по страницам системы управления, доступным пользователю.

Раздел *личной информации* предназначен для отображения информации о пользователе: имя, e-mail, возраст, пол.

Раздел *игровой статистики* предназначен для отображения информации об игровом прогрессе пользователя: точность пилота, маневренность пилота, знание техники, навык торговли, лидерство, *обаяние пилота*. Кроме этого игровая статистика включает в себя отображение сведений о космическом корабле. Данные, отображающиеся в разделах *личной информации* и *игровой статистики*, принимаются с приложения-сервер при загрузке страницы и динамически обновляются.

Раздел *управления игровым процессом* предоставляет возможность управлять функциональностью, которая обеспечивает взаимодействие с игровым процессом, таким образом пользователю предоставлены следующие возможности:

– возможность модифицировать компоненты космического корабля, такие как: *корпус, двигатель, топливный бак, радар, сканер, ремонтный дроид, генератор защитного поля, вооружение;*

- возможность изменить расу, за которую он играет, среди доступных: *зеленые, красные и синие*;
- возможность просматривать информацию о планетах, которые доступны пользователю, а также выбрать определенную планету в качестве базы.

Раздел *меню* позволяет перейти на страницы: *админпанель* (при наличии прав доступа пользователя), *галактика, игроки, техническая поддержка*.

Страница *админпанель* позволяет: управлять учетными записями пользователей, отвечать на вопросы пользователей из раздела технической поддержки, управлять игровым сервером.

Страница *галактика* отображает список доступных звездных систем, пользователь может открыть любую из них. После выбора звездной системы будет отображен список планет, пользователь может открыть подробную информацию о планете, кроме этого будет выведен список игроков, которые находятся в этой системе, либо на планете в данный момент.

Страница *игроки* выводит список игроков в зависимости от выбранного фильтра: зарегистрированные, сейчас в игре. Выбрав игрока из списка можно ознакомиться с его игровой статистикой.

Страница *техническая поддержка* позволяет осуществлять обратную связь пользователей с администраторами.

Веб-клиент должен предоставлять следующие возможности, соответствующие концепции:

- динамическое обновление данных, которые будут изменяться при взаимодействии игроков друг с другом в рамках игрового процесса: статистика, текущий уровень игрока, урон во время сражения, количество побед и поражений, информация о пользователях, взаимодействующих друг с другом;

- осуществлять управление игровым сервером и игровым процессом;

- обмен данными с игровым сервером с помощью WebSocket, для обновления информации, которая отображается на страницах;

- взаимодействие с приложением-сервер с помощью HTTP запросов.

**Структурные компоненты системы управления.** Наличие веб-браузера на всех современных компьютерах и мобильных устройствах обуславливает разработку веб-клиента, с помощью которого будет происходить взаимодействие пользователя и администратора с сервером и игровой средой.

Для реализации клиентской части веб-приложения выбран компонентно-ориентированный архитектурный стиль, позволяющий декомпозировать приложение на более маленькие логические составляющие, которые могут быть ориентированы на работу с определенной частью REST API. Благодаря этому, любой компонент может быть заменен с минимальными изменениями в остальных частях системы, так как взаимосвязанность между компонентами приложения минимальна, что ускоряет разработку и дальнейшее сопровождение приложения.

Динамическое обновление данных осуществляется с помощью технологии WebSocket, которая позволяет подключиться к игровому серверу и соответственно принимать необходимые данные.

Наличие пересекающихся возможностей у веб-клиента и мобильного игрового приложения, таких как регистрация, авторизация, управление профилем пользователя и так далее, обуславливают разработку приложения-сервер, которое будет представлять из себя REST API приложение и наличие централизованной базы данных. Архитектура REST требует использование большинства возможностей протокола HTTP [2]. REST-приложение обладает рядом преимуществ в сравнении с JSON-RPC, приложение-клиент не зависит от имен и аргументов процедур, имеет стандартные коды состояния, обладает поддержкой кеширования, также позволяет направлять запрос от приложения-клиента путем добавления дополнительной информации в URL, либо заголовок HTTP запроса [3].

Для ограничения доступа сторонних приложений, приложение-сервер позволяет производить аутентификацию на время выполнения запроса с помощью RBAC. За контролем доступа приложения-клиента к REST API отвечает Cross-origin resource sharing (CORS) – технология современных браузеров, которая позволяет предоставлять веб-странице доступ к ресурсам другого домена [4].

Исходя из концепции системы управления, приложение сервер должно позволять взаимодействовать с базой данных посредством специальных запросов к определенной узлам. Каждый узел должен соответствовать объекту предметной области, например: пользователь, настройки, сообщения, космический корабль, галактика, компонент корабля, статистика. Запрос отправленный от клиента будет обработан контроллером, который также соответствует объекту предметной области, данный контроллер расположен в компоненте REST Controller Component, который взаимодействует с Model Component. Каждый контроллер должен предоставлять функциональность по считыванию, редактированию, удалению и добавлению новых записей.

Приложение-сервер включает следующие структурные компоненты:

- Database Component – реализует взаимодействие с базой данных;

- Active Record Component – предоставляет интерфейс для преобразования переданных параметров в SQL, взаимодействует с компонентом базы данных;

- Model Component – представляет собой модель объекта реального мира, реализующий интерфейс Active Record;
- CORS Component – компонент, который отвечает за предоставление доступа к ресурсам REST API;
- Service Component – предоставляет интерфейс для обработки данных, которые принадлежат различным моделям;
- RBAC Component – предоставляет интерфейс, позволяющий производить авторизацию на время выполнения запроса и определяет права доступа;
- REST Controller Component – предоставляет интерфейс для взаимодействия с REST API.

Веб-клиент состоит из следующих структурных компонентов:

- Firebase SDK (library) – предоставляет API для работы с сервисами от Google, а также API для реализации регистрации и авторизации;
- Controller Component – осуществляет взаимодействие с View и Http компонентом, предоставляет данные для отображение на пользовательском интерфейсе;
- View Component – формирует пользовательский интерфейс;
- HTTP Component – осуществляет взаимодействие с REST API, отвечает за авторизацию и регистрацию пользователей.

На рисунке изображена диаграмма компонентов, которая показывает разбиение программной системы на структурные компоненты.

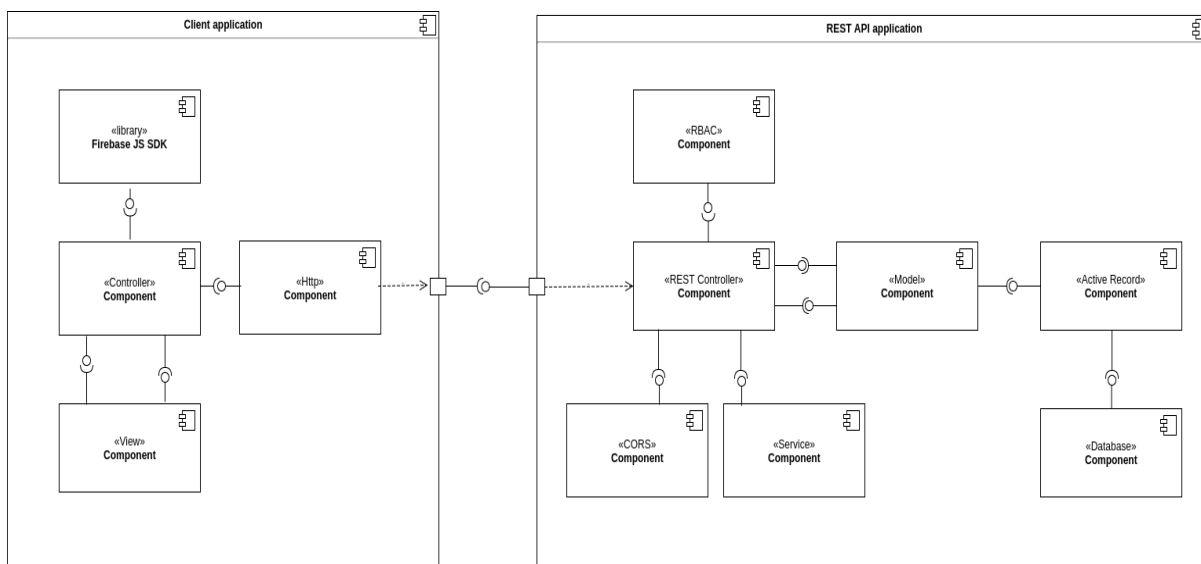


Рисунок. – Диаграмма компонентов веб-приложения

**Закключение.** Предложен подход к реализации системы управления многопользовательской онлайн игрой с использованием архитектурного стиля REST. Определены функциональные возможности системы управления и ее структурные компоненты.

#### ЛИТЕРАТУРА

1. Games-inn [Электронный ресурс] / Games-inn.ru © 2018. – Режим доступа: <https://games-inn.ru/post/5-luchshih-igr-mmo-na-android-i-ios/>. – Дата доступа: 15.09.2019.
2. Smartum [Электронный ресурс] / Smartum.pro © 2018. – Режим доступа: <http://smartum.pro/ru/blog-ru/gid-dlya-nachinayuschih-po-rest-api/>. – Дата доступа: 16.09.2019.
3. Qaru [Электронный ресурс] / Qaru © 2018. – Режим доступа: <http://qaru.site/questions/2533/rest-vs-json-grc>. – Дата доступа: 17.09.2019.
4. Wikipedia [Электронный ресурс] / Wikimedia Foundation, Inc © 2018. – Режим доступа: [https://ru.wikipedia.org/wiki/Cross-origin\\_resource\\_sharing](https://ru.wikipedia.org/wiki/Cross-origin_resource_sharing). – Дата доступа: 20.09.2019.



УДК 004.04

## ПРИМЕНЕНИЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ПРИ СОЗДАНИИ РАСПРЕДЕЛЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ

*С.Д. ПАШКЕВИЧ*

*(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

*Рассматриваются основные виды веб-приложений. Предлагается подход к проектированию сложных веб-приложений, основанный на использовании микросервисной архитектуры. Описываются функциональные возможности такого рода приложений, а также возможные средства взаимодействия между микросервисами.*

**Введение.** Развитие интернет-технологий, сервисов, а также интернет-услуг, ставит перед разработчиками и веб-приложениями новые, все более сложные задачи. Возрастающие требования к веб-приложениям, необходимость интеграции со сторонними сервисами, обеспечение необходимой скорости доступа к информации, все эти критерии обуславливают создание сложных приложений, которые будут соответствовать современным стандартам.

Существует два основных вида веб-приложений: монолитные и приложения с микросервисной архитектурой.

Монолитное приложение – это приложение, построенное, как единое целое. Такие приложения часто состоят из трех основных частей: пользовательский интерфейс, база данных и сервер. Серверная часть должна быть способна обрабатывать HTTP запросы в соответствии с базовой логикой, а также возвращать результат выполнения, который может быть использован другими приложениями для дальнейшего отображения конечному пользователю. Все операции, относящиеся к обработке данных, выполняются в единственном процессе, то есть запуск самой простой операции требует инициализации всего приложения, что делает большие монолитные программы ресурсозатратными. Кроме этого, если приложение существует уже достаточно долгое время, в процессе разработки и внедрения новых функциональных возможностей, оно увеличивается до огромных размеров, что в свою очередь требует затрачивать больше времени на разработку. Масштабировать приложение горизонтально приходится целиком, путем его запуска на нескольких физических серверах, чтобы иметь возможность распределять нагрузку.

Эти неудобства привели к архитектурному стилю микросервисов: построению приложений в виде набора сервисов. В дополнение к возможности независимого развертывания и масштабирования каждый сервис также получает четкую физическую границу, которая позволяет разным сервисам быть написанными на разных языках программирования. Они также могут разрабатываться разными командами.

Архитектурный стиль микросервисов – это подход, при котором единое приложение строится как набор небольших сервисов, каждый из которых работает в собственном процессе и взаимодействует с остальными используя легковесные механизмы. Эти сервисы построены вокруг бизнес-потребностей и инициализируются независимо, с использованием полностью автоматизированной среды. Существует абсолютный минимум централизованного управления этими сервисами. Сами по себе сервисы могут быть написаны на разных языках и использовать разные технологии хранения данных [1].

Одним из основных преимуществ микросервисов является возможность изменения отдельно взятого микросервиса, не меняя при этом связанных с ним компонентов.

Таким образом, следует выделить основные особенности:

- упрощается процесс масштабирования разрабатываемого приложения;
- каждый микросервис не зависит от других микросервисов;
- микросервисы могут быть взаимозаменяемы.

Однако, не смотря на все преимущества, существуют некоторые недостатки: требуется учитывать тот факт, что любой из микросервисов может выйти из строя, кроме этого, необходимо предусмотреть реализацию поведения микросервисов и их взаимодействие.

**Основная часть.** Взаимодействие микросервисов в рамках одного приложения для осуществления поставленной задачи следует выполнять, опираясь на потребности пользователей либо бизнес-логику предприятия, для которого разрабатывается приложение.

Таким образом, при проектировании и разработке приложения, следует учитывать основные термины, с которыми приходится сталкиваться пользователям конечного продукта, разрабатывать компоненты системы, которые обязаны соответствовать бизнес логике, строить приложение опираясь на процессы, с которыми сталкивается пользователь в повседневности, в рамках своей рабочей деятельности, а не на данных.

Предметно-ориентированное проектирование (DDD) – это набор принципов и схем, направленных на создание оптимальных систем объектов. Сводится к созданию программных абстракций, которые называются моделями предметных областей. В эти модели входит бизнес-логика, устанавливающая связь между реальными условиями области применения продукта и кодом.

Предметно-ориентированное проектирование не является какой-либо конкретной технологией или методологией. DDD – это набор правил, которые позволяют принимать правильные проектные решения. Данный подход позволяет значительно ускорить процесс проектирования программного обеспечения в незнакомой предметной области [2].

Исходя из этого, можно сделать вывод, что DDD отлично подходит для непростых предметных областей.

Большинство корпоративных приложений со сложной бизнес-логикой и технической реализацией разделяются на уровни. Уровни являются логическим артефактом и не относятся к развертыванию службы. Они нужны, чтобы помочь разработчикам управлять сложными процессами в коде. Различные уровни (например, уровень модели предметной области, уровень представления данных и т. д.) могут иметь различные типы.

Например, сущность может загружаться из базы данных. Затем часть этих сведений, или объединенные данные, включающие данные из других сущностей, могут отправляться в пользовательский интерфейс клиента. Суть в том, что объект содержится в уровне модели предметной области и не должен передаваться в другие области, к которым он не принадлежит, такие как уровень представления данных [3].

Таким образом, используя DDD и микросервисную архитектуру, можно проектировать достаточно сложные и большие корпоративные приложения. Каждый микросервис обязан реализовывать задачи определенной части из области определения, что гарантирует то, что микросервисы будут независимы не только физически, но и логически. Кроме этого, такой подход позволит одновременно работать разным командам в рамках одного приложения, используя разные языки программирования. У разработчиков одного из микросервисов нет необходимости вникать в функциональность другого микросервиса, что также ускоряет разработку.

Следующий важный шаг, который стоит учитывать при проектировании такого рода систем – взаимодействие между ними.

Существуют следующие подходы:

- взаимодействие посредством HTTP запросов;
- взаимодействия с помощью очереди событий.

Так как приложение распределенное, то есть, некоторые операции будут выполняться различными сервисами, то следует учитывать тот факт, что не все микросервисы будут доступны в момент выполнения этой операции. Таким образом, если один из микросервисов делает HTTP запрос на обновление данных в другом микросервисе и тот не будет доступен в данный момент времени, то операция будет прервана, что в свою очередь может стать причиной неактуальных данных. Решить эту проблему позволяют специальные службы, которые способны строить очереди из событий, которые происходят в одном приложении, затем другие приложения отслеживают события из этой очереди, в результате чего, при появлении определенного события, выполняют ту или иную операцию. Такая очередь событий должна обладать способностью накапливать события в том случае, если приложение, для которого событие было предназначено, недоступно. RabbitMQ соответствует этим требованиям.

RabbitMQ – это мультипротокольный брокер сообщений, позволяющий организовать отказоустойчивый кластер с полной репликацией данных на несколько узлов, где каждый узел может обслуживать запросы на чтение и запись [4]. То есть основная цель RabbitMQ – принимать и отдавать сообщения.

RabbitMQ позволяет взаимодействовать различным программам при помощи протокола AMQP, является отличным решением для построения SOA (сервис-ориентированной архитектуры) и распределением отложенных ресурсоемких задач [5].

Таким образом, использовать данную службу, при построении распределенного веб-приложения, является хорошим решением. Кроме этого, такой подход позволит отправить выполнение ресурсоемкой операции в фоновый процесс, в следствии чего, у пользователя нет необходимости ожидать завершения выполнения данной операции, так как результаты будут вычислены, не блокируя пользовательский интерфейс.

Следовательно, RabbitMQ станет посредником при взаимодействии микросервисов в рамках распределенного приложения. Каждый микросервис, в случае необходимости, будет иметь возможность сообщить другим микросервисам о том, что завершена та или иная операция, путем публикации сообщения в очередь. После этого, любой микросервис будет иметь возможность отслеживать необходимые сообщения и реагировать на них заданным образом.

На рисунке представлен пример взаимодействия микросервисов.

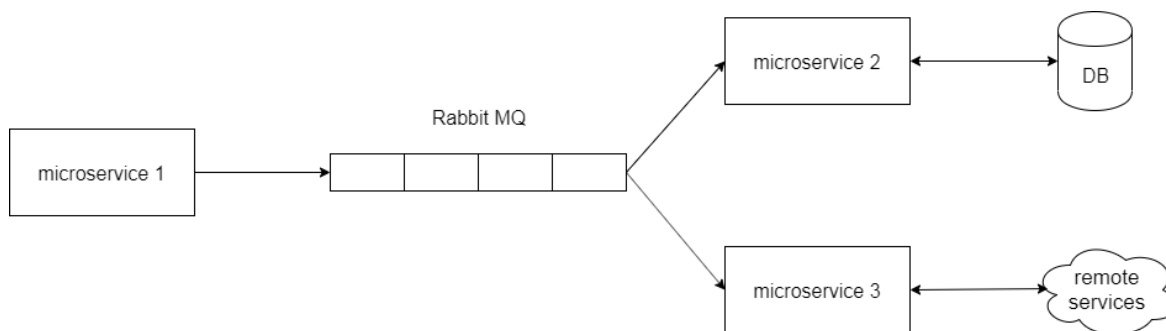


Рисунок. – Пример взаимодействия микросервисов

**Заключение.** Предложен подход к реализации распределенных веб-приложений. Рассмотрены основные виды веб-приложений, а также средства взаимодействия между микросервисами.

#### ЛИТЕРАТУРА

1. Habr [Электронный ресурс] / habr.com © 2019. – Режим доступа: <https://habr.com/ru/post/249183/>. – Дата доступа: 10.09.2019.
2. The Web Land [Электронный ресурс] / thewebland.net © 2019. – Режим доступа: <https://thewebland.net/domain-driven-design/>. – Дата доступа: 16.09.2019.
3. Microsoft [Электронный ресурс] / Microsoft © 2019. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>. – Дата доступа: 17.09.2019.
4. Habr [Электронный ресурс] / habr.com © 2019. – Режим доступа: <https://habr.com/ru/company/tensor/blog/341068/>. – Дата доступа: 25.09.2019.
5. Habr [Электронный ресурс] / habr.com © 2019. – Режим доступа: <https://habr.com/ru/post/149694/>. – Дата доступа: 27.09.2019.

УДК 004.932;

**ВВЕДЕНИЕ В WEBP ФОРМАТ СЖАТИЯ ИЗОБРАЖЕНИЙ****А.И. ПИРОВИЧ***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассматривается формат WebP изображения и особенности его реализации. Описываются преимущества исследуемого формата над другими форматами изображений.*

Для масштабируемых веб-сайтов с большим количеством контента и изображений добавление изображений на страницу приведет к решению двух важных задач:

- выбор правильного формата для изображения;
- оптимизация изображений.

Существуют различные форматы изображений, такие как JPG, JPEG, PNG, Gif, SVG и другие. Рассмотрим их альтернативу с более высокой сжимаемостью – Webp.

Webp – это формат изображения, разработанный компанией Google. Он служит альтернативой PNG и JPG форматов с гораздо меньшим размером при том же качестве изображения. WebP так же поддерживает прозрачность и анимацию. В зависимости от сложности и размера JPG изображения он сжимает изображение сильнее на 25–35%. WebP берет лучшие части как формата JPG с потерями, так и формата PNG без потерь и делает их доступными в компактном файле.

Сравнительный анализ размеров изображения от процента сжатия демонстрируются на графиках, представленных на рисунке 1. Безусловно видно, что WebP обеспечивает лучшее сжатие, чем другие форматы, особенно для небольших изображений [1].

В 2019 году WebP поддерживается в 80% браузеров как показано на рисунке 2 [2]. Что делает WebP более популярным в использовании. Но все же в Safari и IE картинки будут иметь искажения. Для этого существует простое решение без JavaScript и бэкенд-сервисов. Возможно с помощью HTML-5 элемента <picture> добавить альтернативные форматы для одной картинки.

Для добавления альтернативных форматов, используется элемент <source> вместе с <picture>.

У элемента <source> есть несколько атрибутов, которые мы можем использовать, чтобы определить изображение и когда оно будет использовано:

- sizes для перечня размеров каждого файла;
- srcset для пути к файлу изображения. Несколько файлов могут быть использованы для изображений разных размеров и плотности пикселей;
- type для MIME-типа формата;
- media для медиа-выражения – определяет, какое изображение будет использовано.

Листинг 1. – Пример использования элемента <picture>

```
<?
<picture>
  <source type="image/webp" srcset="image.webp">
  <source type="image/jpeg" srcset="image.jpg">
  
</picture>
```

Таким образом, WebP является настолько универсальным форматом, что он может использоваться вместо всех существующих форматов изображений, чтобы получать изображения меньшего размера и хорошего качества. Он помогает загружать изображения в Интернете намного быстрее, сокращая время рендеринга веб-страниц, что значительно упрощает работу пользователей с быстро загружающимися веб-сайтами с красивыми изображениями. Использование изображений в формате WebP в мобильных браузерах открывает широкие возможности для оптимизации работы пользователей.

Вышеизложенные данные поспособствуют разработке мобильного приложения для распознавания одежды на фото при организации ее поиска в интернет-магазинах.

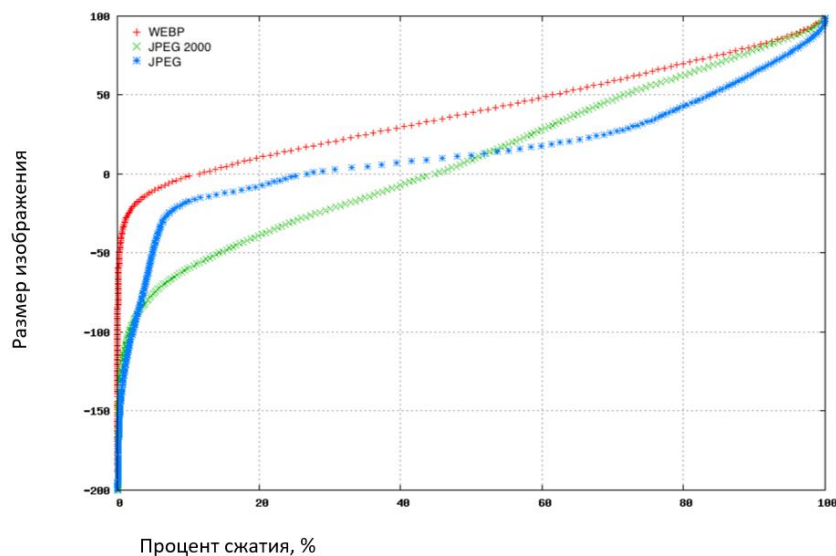


Рисунок 1. – Сравнение размеров изображения от процента сжатия

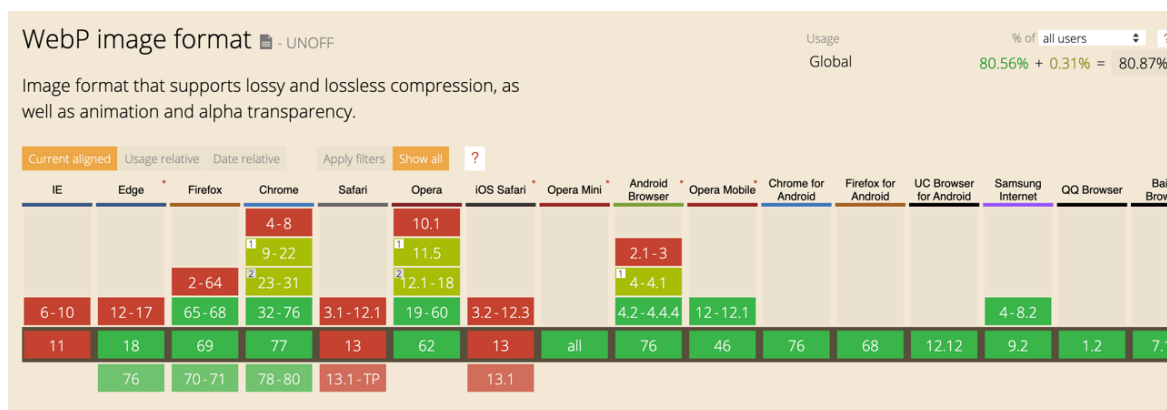


Рисунок 2. – Поддержка WebP в разных браузерах

ЛИТЕРАТУРА

1. Comparative study of WebP, JPEG and JPEG 2000 [Электронный ресурс]. – Режим доступа: [https://developers.google.com/speed/webp/docs/c\\_study#results](https://developers.google.com/speed/webp/docs/c_study#results). – Дата доступа: 20.09.2019.
2. WebP image format [Электронный ресурс]. – Режим доступа: <https://caniuse.com>. – Дата доступа: 20.09.2019.

УДК 004.8.

**ВВЕДЕНИЕ В ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ****А.И. ПИРОВИЧ***(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЕНОК)*

*В данной статье рассматривается интеллектуальный анализ данных, описываются области его применения.*

Интеллектуальный анализ данных – это изучение большого количества исходных данных для выделения из них полезной информации, что дает компаниям преимущество перед их конкурентами. Эта информация состоит из значимых формул, цифр, статистики и других необходимых данных, которые уже присутствуют в исходной информации, но ранее были не очевидны для человека.

Наиболее популярный инструмент, который используется при анализе – искусственный интеллект (ИИ). Технологии искусственного интеллекта пытаются работать по примеру человеческого мозга, а именно – путем разумного предположения, например, обучения, и использование дедуктивного мышления. Некоторые из наиболее распространенных методов искусственного интеллекта используются в интеллектуальном анализе данных, например, нейронные сети, кластеризация и деревья решений [1].

Остановимся более подробно на каждом из них.

Правило работы нейронной сети основано на связях между данными или на использовании образца исходных данных. В результате, программное обеспечение непрерывно анализирует значения и сравнивает его с другими значениями, и программное обеспечение сравнивает эти значения до тех пор, пока не найдет новый шаблон. Эти шаблоны могут быть названы правилами. Далее программное обеспечение ищет другие шаблоны, которые основаны на этих правилах, или отправляет оповещение, когда находит необходимое значение.

Кластеризация делит данные на группы, основываясь на схожих чертах или на ограниченных диапазонах данных. Кластеры используются, когда данные не помечены как наилучшими для использования интеллектуального анализа данных. В качестве примера можно рассмотреть страховую компанию, которая пытается выявить случаи мошенничества. В документах компании изначально такие записи не будут храниться помеченными как мошеннические или фальсифицированные. Однако, после анализа модели в рамках кластеров, программное обеспечение, которое проведет анализ, может начать выявлять правила, которые указывают на необходимые записи, а именно на записи с ошибками, либо записи мошеннического характера. Задачу кластеризации формулировали в различных направлениях, например, статистика, распознавание образов, оптимизация процессов, машинное обучение и т.п. Поэтому существует много синонимов кластеру – класс, таксон, сгущение [2].

Деревья решений, так же, как и кластеры, разделяют данные на подмножества, а затем анализируют подмножества, разделяя их еще на более мелкие подмножества, и так далее (еще на несколько уровней при необходимости). Финальные подмножества так малы, что анализирующий процесс может найти неочевидные модели и взаимосвязи внутри данных.

Когда данные для анализа выявлены, они должны быть очищены. Очистка освобождает данные от дубликатов информации и от ошибок. Далее, данные должны быть сохранены в едином формате в необходимых категориях или полях. Инструменты для анализа могут работать со всеми типами хранения данных, от больших хранилищ данных (баз данных) до простых файлов. Хранилища данных и витрины данных это методы хранения, которые включают архивирование больших объемов данных таким образом, что позволяет легко получить доступ в случае необходимости.

Когда процесс завершается, анализирующее программное обеспечение генерирует отчет. Аналитик читает отчет, для того чтобы понять, необходимо ли выполнять дальнейшую обработку данных, такую как очистка параметров, использование других инструментов анализа данных для изучения данных, или даже переработка данных, если они не пригодны для использования. Если не требуется проведение дальнейших работ, отчет передается ответственным людям для принятия соответствующих мер.

Возможности интеллектуального анализа данных могут использоваться для многих областей, таких как здравоохранение, сбор статистики из СМИ, решение сложных задач в производственных процессах, а также анализа последовательностей в человеческих генах. Действительно нет ограничений для бизнеса и областей изучения, где интеллектуальный анализ данных может быть полезным.

В целом, преимущества интеллектуального анализа данных заключаются в способности выявлять скрытые шаблоны и взаимосвязи в данных, которые можно использовать для прогнозирования, влияющего на бизнес. Конкретные преимущества интеллектуального анализа данных зависят от цели и отрасли. Отделы продаж и маркетинга могут собирать данные о клиентах, чтобы повысить коэффициент конверсии

потенциальных клиентов или создать индивидуальные маркетинговые кампании. Данные интеллектуального анализа данных об исторических моделях продаж и поведении клиентов могут использоваться для построения моделей прогнозирования будущих продаж, новых продуктов и услуг.

Компании в финансовой отрасли используют инструменты интеллектуального анализа данных для построения моделей риска и выявления случаев мошенничества. В обрабатывающей промышленности используются инструменты интеллектуального анализа данных для повышения безопасности продукции, выявления проблем с качеством, управления цепочкой поставок и улучшения операций.

Вышеизложенные данные используются для изучения принципов работы интеллектуального анализа данных, которые способствуют разработке мобильного приложения для распознавания одежды из фото и поиска данной одежды в интернет-магазинах.

#### ЛИТЕРАТУРА

1. Интеллектуальный анализ данных [Электронный ресурс]. – Режим доступа: <http://translated-by.com/you/data-mining/into-ru/trans>. – Дата доступа: 20.09.2019.
2. Алгоритмы кластеризации на службе Data Mining [Электронный ресурс]. – Режим доступа: <https://basegroup.ru/community/articles/datamining>. – Дата доступа: 20.09.2019.

УДК 004

**АЛГОРИТМ РАСПОЗНАВАНИЯ ДОРОГИ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ.  
ОБНАРУЖЕНИЕ СТАТИЧЕСКИХ И ДИНАМИЧЕСКИХ ОБЪЕКТОВ  
НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ С ИСПОЛЬЗОВАНИЕМ MACHINE LEARNING  
И СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЯ****В.А. ПЛЯСОВ***(Представлено: О.В. МИХНОВИЧ)*

*В данной статье будет рассмотрен алгоритм для распознавания различных объектов на видеопоследовательности с использованием машинного обучения и компьютерного зрения.*

В настоящий момент компьютерное зрение широко используется для решения различного рода задач: беспилотный транспорт, идентификация объектов на фото/видео, распознавание лиц с камер видеонаблюдения и прочих.

Компьютерное зрение – теория и технология создания машин, которые могут производить обнаружение, отслеживание и классификацию объектов.

Как научная дисциплина, компьютерное зрение относится к теории и технологии создания искусственных систем, которые получают информацию из изображений. Видеоданные могут быть представлены множеством форм, таких как видеопоследовательность, изображения с различных камер или трехмерными данными [1].

Машинное обучение – обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться. Различают два типа обучения. Обучение по прецедентам, или индуктивное обучение, основано на выявлении общих закономерностей по частным эмпирическим данным. Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний. Дедуктивное обучение принято относить к области экспертных систем, поэтому термины машинное обучение и обучение по прецедентам можно считать синонимами [2].

Для реализации системы распознавания статических и динамических объектов на видеопоследовательности необходимо обозначить следующие шаги алгоритма:

1. Предобработка видео. На данном этапе необходимо провести стабилизацию кадра видеопоследовательности.

2. Обнаружение дороги на видеопоследовательности:

– Выделение краев на дороге. Выделить края на изображении можно с помощью оператора Собеля, который хорошо подходит для поставленной задачи [3]. Выделения краев необходимо для того, чтобы определить пороговые значения для перспективной трансформации кадра.

– Перспективная трансформация кадра видеопоследовательности. Здесь появляется необходимость трансформации кадра к виду «птичьего полета». На рисунке 1 изображена перспективная трансформация кадра.

– Определение линий на дороге. Для этого так же можно воспользоваться оператором Собеля, чтобы выделить все края.

Определение кривизны дороги. Для нахождения кривизны дороге необходимо воспользоваться полиномом второго порядка, формула которого описывается в выражении (1).



а)



б)

**Рисунок 1. – Перспективная трансформация кадра видеопоследовательности изображения:  
а) до применения трансформации, б) результирующий кадр**



$$f(y) = Ay^2 + By + C, \quad (1)$$

где  $A, B, C$  – изменяемые аргументы функции  $y$ .

Для нахождения радиуса кривизны необходимо решить дифференциальное уравнение 2-го порядка, формула которого представлена в выражении (2):

$$R_{curve} = \frac{[1 + (\frac{dx}{dy})^2]^{3/2}}{|\frac{d^2x}{dy^2}|} \quad (2)$$

Решив данное уравнение, получаем радиус кривизны дороги, после чего данное значение можем применить к исходному кадру видео. На рисунке 2 показано выделение дороги на кадре видеопоследовательности.



Рисунок 2. – Получение кривизны дороги и ее выделение на исходном кадре видеопоследовательности

3. Идентификация объектов на дороге. Данную часть алгоритма можно разделить на несколько этапов:

– Сегментация изображения для выделения объектов на видеопоследовательности. Для сегментации изображения используется WaterShed алгоритм, который уже реализован с помощью OpenCV библиотеки. Результат работы сегментации проиллюстрирован на рисунке 3.

– Использование машинного обучения для идентификации типа объекта. Для того чтобы точно определить тип объекта на видеопоследовательности, необходимо большое количество эталонных данных для системы машинного обучения. На данный момент в интернете имеется большое количество видео с автомагистралей, что позволяет наполнить базу эталонными данными.

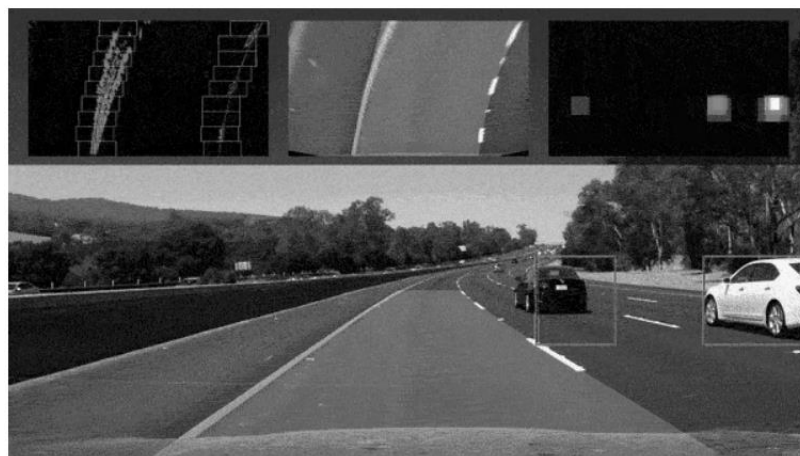


Рисунок 3. – Выделение объектов на дороге с помощью сегментации изображения

Из вышесказанного можно сделать вывод, что алгоритм распознавания различного рода объектов на дороге является комбинированным и требует больших вычислительных затрат, в связи с большим количеством действий на этапе обработки кадра видеопоследовательности, однако нынешние облачные сервисы позволяют производить подобные манипуляции с видео «на лету».

#### ЛИТЕРАТУРА

1. Компьютерное зрение [Электронный ресурс]. – 2019. – Режим доступа: [https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5\\_%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D0%B5](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5_%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D0%B5) – Дата доступа: 20.09.2019.
2. Машинное обучение [Электронный ресурс] – 2019г. – Режим доступа: [http://www.machinelearning.ru/wiki/index.php?title=Машинное\\_обучение](http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение). – Дата доступа: 20.09.2019.
3. Оператор Собеля [Электронный ресурс] – 2019 г. – Режим доступа: [https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80\\_%D0%A1%D0%BE%D0%B1%D0%B5%D0%BB%D1%8F](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%A1%D0%BE%D0%B1%D0%B5%D0%BB%D1%8F). Дата доступа: 20.09.2019.

УДК 004

## АРХИТЕКТУРА СИСТЕМЫ РАСПОЗНАВАНИЯ СТАТИЧЕСКИХ ОБЪЕКТОВ НА ВИДЕОИЗОБРАЖЕНИЯХ АВТОМОБИЛЬНЫХ ДОРОГ С РАЗЛИЧНЫМ ТИПОМ ПОКРЫТИЯ С ИСПОЛЬЗОВАНИЕМ ОБЛАЧНЫХ СЕРВИСОВ AWS

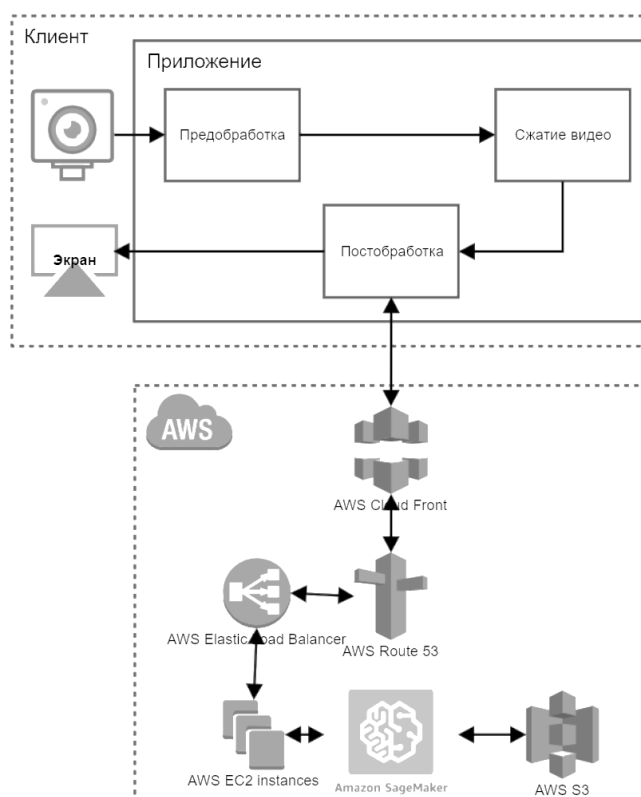
**В.А. ПЛЯСОВ***(Представлено: О.В. МИХНОВИЧ)*

В данной статье будут рассмотрены основные моменты в архитектуре данной системы с использованием облачных сервисов Amazon Web Services (AWS). Из каких основных частей будет построена система, как будет реализован обмен информацией между машиной и облачными сервисами. На основе какого протокола будет организован обмен данными и как они будут представлены.

На данный момент для решения различного рода нетривиальных задач, для которых необходимы большие вычислительные мощности, к примеру, машинное обучение, распределенные системы и т.д., набирают популярность облачные сервисы, которые дают широкий спектр возможностей для своих клиентов.

Облачные сервисы – это сервисы, работающие на облачных хранилищах. То есть, их не нужно устанавливать на компьютер и получать доступ с любой точки выхода. В онлайн (облачных) хранилищах данные хранятся на многочисленных распределенных в сети серверах, предоставляемых в пользование клиентам, в основном, третьей стороной [1].

Системы распознавания статических объектов на видеоизображениях автомобильных дорог с различным типом покрытия с использованием облачных сервисов AWS будут представлять распределенную систему, состоящую из нескольких приложений: клиентская часть системы (программа, которая производит базовую обработку видеопотока с устройства), серверная часть (представлена сервисами AWS) и система машинного обучения, построенная на базе сервиса AWS SageMaker. На рисунке 1 представлена вся архитектура системы.

**Рисунок 1. – Архитектура системы**

Клиентская часть системы представлена приложением, которое прodelывает следующие действия над входящим видеопотоком:

1. Предобработка:
  - Выравнивание цветового баланса.
  - Баланс белого.
  - Уменьшение шума.
2. Сжатие видео. Данная операция необходима для того, чтобы снизить передаваемый трафик между облачными сервисами и клиентской частью системы.
3. Постобработка.
  - Простая сегментация изображения.
  - Выделения контуров, в результате чего упростить поиск объектов на видео.
  - Получение результирующего кадра видео с сервисов.
  - Вывод полученного кадра на экран.

Сервер представляется в виде облачных сервисов AWS, которые служат для того, чтобы выполнять всю обработку видео и обнаружение объектов на нем, полностью исключив нагрузку с клиенткой части системы, что освобождает клиента от необходимости устанавливать дополнительные вычислительные мощности внутри машины (компьютер). Серверная часть представлена следующими сервисами:

1. AWS Cloud Front. Данный сервис необходим для получения всех входящих запросов со стороны клиента. Представляет собой шаблон проектирования «Фасад», который закрывает всю работу системы от внешних источников воздействия.

2. AWS Route 53. Данный сервис необходим для того, чтобы перенаправить запросы со стороны клиента на соответствующую подсеть внутри сети AWS. Работает вместе с AWS VPC network для создания внутренних сетей и повышение безопасности системы.

3. AWS EC2 instance. Представляет собой виртуальную машину с предустановленной операционной системой для выполнения различного рода действия. Данный тип машин используется для основной обработки видео, его сегментации и разбиения на участки (дорога, обочина, объекта). А также он необходим для того, чтобы выполнить идентификацию статического объекта на дороге при использовании AWS SageMaker.

4. AWS Elastic Load Balancer. Так как система подразумевает, что ей могут пользоваться несколько клиентов одновременно, в связи с этим появляется необходимость в распределении нагрузки между несколькими AWS EC2 instance, чтобы процесс обработки видеопотоков занимал минимальное количество времени. Так же данный сервис имеет возможность запускать дополнительные AWS EC2 instance в случае пиковых нагрузок.

5. Amazon SageMaker. Сервис, который имеет широкий спектр для машинного обучения. Позволяет полностью управлять процессом обучения, начиная от создания модели для системы и заканчивая выбором эталонных данных для обучения системы.

6. Amazon S3. Облачное хранилище данных AWS. Сервис необходим для хранения эталонных данных, которые используются для обучения системы Amazon SageMaker.

Так же система подразумевает процесс аутентификации, который построен на базе Amazon Cognito, который поддерживает аутентификацию на уровне заголовков запросов, что в конечном итоге делает процесс передачи данных более простым в реализации.

Процесс передачи видеопотока между клиентом и сервером будет реализован на протоколе RTSP, который служит для передачи потокового видео на сервер.

Исходя из вышеперечисленной информации, можно сделать вывод, что архитектура данной системы является сложной, но, при этом, позволяет пользоваться ей сразу нескольким клиентам. При анализе системы были выявлены простые пути реализации распределения нагрузки между несколькими AWS EC2 instance. Основную сложность при реализации составляет передача видеопотока клиент-сервер-клиент, потому что серверу необходимо сделать полный анализ видеопотока, что требует дополнительных время затрат. Основываясь на этом, можно сделать заключение, что результирующий видеопоток, на котором будут выведены все объекты, будет приходиться с задержкой.

#### ЛИТЕРАТУРА

1. Облачные сервисы – что это такое [Электронный ресурс]. – 2019. – Режим доступа: <https://hi-news.ru/tag/oblachnye-servisy> – Дата доступа: 17.09.2019.

УДК 004.021

## РАССМОТРЕНИЕ РАБОТЫ НЕЙРОСЕТЕЙ В ОБЩЕМ ВИДЕ

**В.С. РАДЧЕНКО**  
(Представлено: Д.В. ПЯТКИН)

В статье представлены теоретические данные по работе нейросетей и основные данные о составлении сетей нейронов.

**1 Структура искусственного нейрона**

Искусственный нейрон является элементарной структурной единицей искусственной нейронной сети и представляющей из себя упрощенную модель биологического нейрона. На рисунке 1 изображен нейрон, входами которого могут быть либо входные данные, либо выходы от такого же нейрона. Входы соединены с ячейкой нейрона S с помощью синаптических связей. Каждый синапс имеет свой вес, при передаче в ячейку нейрона входного параметра, он соответственно умножается на вес, т.е.  $x_i * w_i$ . Состояние нейрона S, определяется в виде формулы:

$$S = \sum_{i=1}^N x_i * w_i \quad (1)$$

Выход нейрона, есть функция его состояния  $y=f(S)$ .

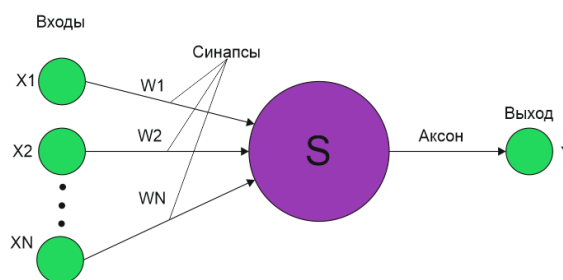


Рисунок 1. – Визуализация искусственного нейрона

Функция  $f$  называется функцией активации. Такие функции не могут быть линейными, поскольку нейронные сети с линейной функцией активации эффективны только на одном уровне, независимо от того, насколько сложна их структура. Одной из наиболее распространенных функций является нелинейная функция с насыщением, называемая логистическая функция или сигмоида:

$$f(x) = \frac{1}{1 + e^{-ax}} \quad (2)$$

Параметр  $a$  отвечает за пологость функции. Чем он меньше, тем более пологим становится сигмоида. Следует отметить, что данная функция дифференцируема на всей оси абсцисс, что является необходимым условием в некоторых алгоритмах обучения нейронных сетей. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем большие, а также предотвращает насыщение от больших сигналов так как они соответствуют областям аргументов, где сигмоида имеет пологий наклон (рисунок 2) [1].

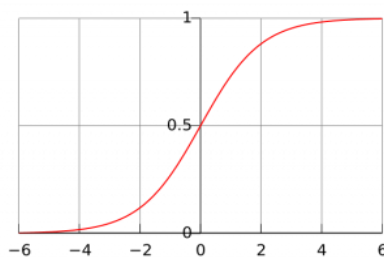


Рисунок 2. – Сигмоида

## 2 Архитектура нейронной сети

Можно выделить несколько основных типов нейронных сетей: Многослойные сети. В многослойных сетях один или несколько нейронов объединяются в слои (рисунок 3). Слой – совокупность нейронов, на вход которых подается один и тот же общий сигнал. В сетях такого типа, внешние входные данные подаются на входы нейронов первого слоя, а выходные данные являются результатом последнего выходного слоя. Кроме входного и выходного слоев, в многослойных сетях так же присутствует один или несколько скрытых слоев. Связи выходов нейронов от некоторого слоя  $i$  к некоторому слою  $i+1$  называют последовательным.

Полносвязные нейронные сети. В полносвязной нейронной сети каждый нейрон передает свой сигнал остальным нейронам (рисунок 4).

Выходными сигналами, могут быть все или некоторые сигналы нейронов после нескольких тактов функционирования. Все входные сигналы, подаются на вход всем нейронам [6].

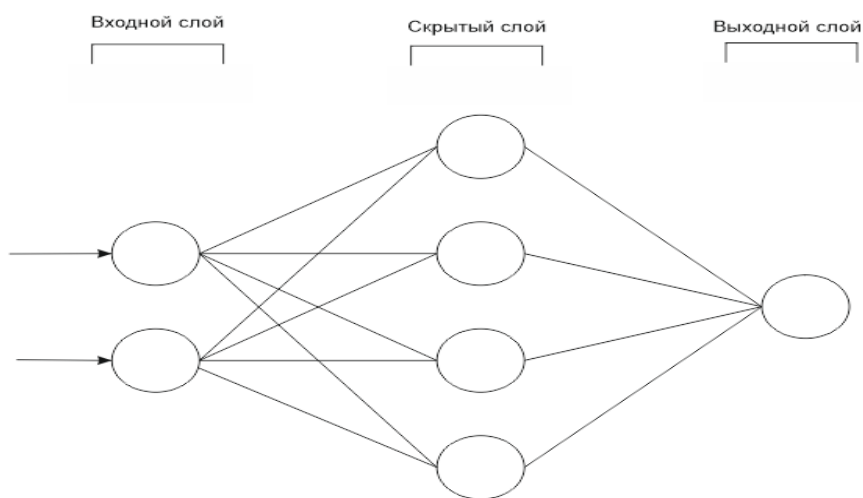


Рисунок 3. – Многослойная нейронная сеть

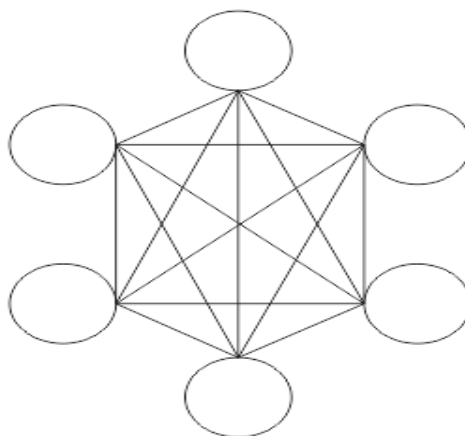


Рисунок 4. – Полносвязная нейронная сеть

Каждая нейронная сеть имеет не только свою архитектуру, но и тип, который лучше подходит для решения конкретной задачи. К примеру, сверточные нейронные сети (convolutional neural network CNN) гораздо лучше справляется с распознаванием образов и проблемами компьютерного зрения. Ее отличие от других типов ИНС заключается в том, что каждый фрагмент изображения умножается на матрицу (ядро) свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения (рисунок 5). Если каждый пиксель изображения рассматривался бы отдельно, это привело бы сеть к быстрому переобучению и ее способность распознавания образов была бы точна только на обучающей выборке [3].

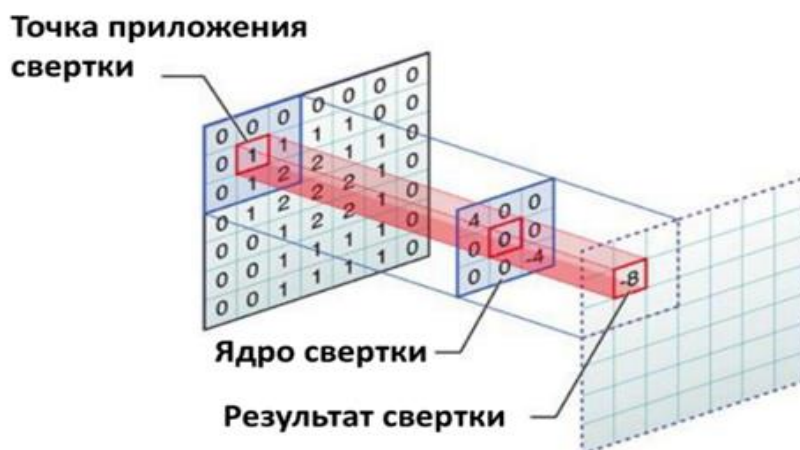


Рисунок 5 – Сверточная нейронная сеть

Существуют так же развертывающие нейронные сети (deconvolutional networks DN), так же называемые обратными графическими сетями и являющиеся обратными к CNN (рисунок 6). Их задача генерировать изображения по заданным признакам. К примеру, при передаче сети слова «кот» она должна будет сгенерировать изображения похожие на котов.

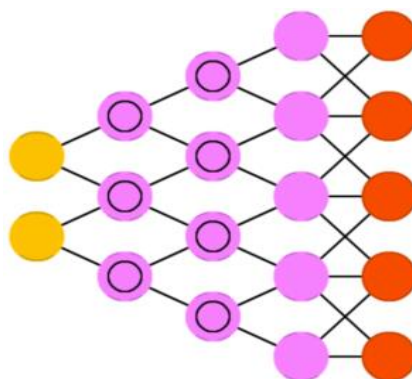


Рисунок 6. – Развертывающаяся нейронная сеть

Для предсказания слов в предложении, предсказании следующего числа в последовательности, выделения главной мысли текста, генерации новой информации похожей на данную используются рекуррентные нейронные сети (recurrent neural network RNN). В рекуррентных нейронных сетях нейроны обмениваются информацией между собой, к примеру, в добавок к новому кусочку входящих данных нейрон так же получает информацию о предыдущем состоянии сети (рисунок 7). Таким образом, в сети реализуется так называемая «память», что принципиально меняет характер ее работы и позволяет анализировать любые последовательности данных.

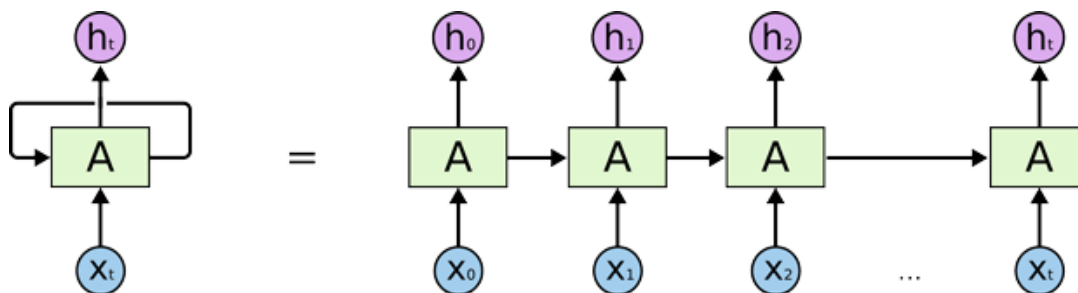


Рисунок 7. – Рекуррентная нейронная сеть

Самыми простыми нейронными сетями, являются сети прямого распространения (feedforward neural network FFN), которые взяты за основу для создания многих других сетей. Данные сети прямолинейны и передают информацию от входа к выходу. Нейроны каждого слоя не связаны между собой, а соседние слои обычно полностью связаны. Вид FFN можно рассматривать как многослойную сеть (рисунок 3) [3].

#### ЛИТЕРАТУРА

1. Нейронные сети: полный курс. С. Хайкин. 2-е издание, Издательский дом Вильямс, 2016 – 1104 с.
2. Neural Networks: A Systematic Introduction. R. Rojas. Springer Science & Business Media – 502 с.
3. Нейронные сети для обработки информации. С. Осовский. Пер. с польского И.Д. Рудинского. – М. : Финансы и статистика, 2013. – 344 с.



УДК 004.021

## ОБУЧЕНИЕ НЕЙРОСЕТЕЙ

**В.С. РАДЧЕНКО**

*(Представлено: Д.В. ПЯТКИН)*

*В статье представлены методы для обучения нейронов в нейросетях и функции, которые помогут в понимании данного процесса.*

### 1 Обучение нейронных сетей

Процесс обучения нейронной сети является необходимым, для ее способности выполнять поставленные задачи. Этот процесс может быть рассмотрен как настройка архитектуры сети и весов связей посредством моделирования среды, в которую эта сеть встроена. Свойство нейронной сети обучаться на примерах позволяет упростить задание условий для решения конкретной задачи, по сравнению с системами, которые следуют набору определенных правил, составленных экспертами. Разделяют алгоритмы обучение с учителем и без учителя [1].

Процесс обучения с учителем подразумевает предъявление сети выборки обучающих примеров. Каждый образец подается на вход нейронной сети, затем проходит процедуру обработки внутри ИНС. После вычисления выходного сигнала ИНС сравнивает полученный результат с соответствующим значением целевого вектора, представляющего собой требуемый выход сети. Вычислив ошибку, происходит изменение весовых коэффициентов связей внутри сети по выбранному алгоритму. Веса подстраиваются под каждый вектор до тех пор, пока ошибка по всему массиву входных данных не достигнет заданного уровня.

Обучение без учителя не требует знания правильного ответа на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или корреляция в системе данных, что позволяет распределить образы по категориям.

В зависимости от решаемой задачи в обучающей выборке используются те или иные типы данных и различные размерности входных/выходных сигналов. Входные данные примеров обучающей выборки – изображения, таблицы чисел, распределения. Типы входных данных – бинарные (0 и 1), биполярные (-1 и 1) числа, целые или действительные числа из некоторого диапазона. Выходные сигналы сети – вектора целых или действительных чисел. Для решения практических задач часто требуются обучающие выборки большого объема. Из-за жестко ограниченного объема оперативной памяти компьютера разместить в ней большие обучающие выборки невозможно. Поэтому выборка делится на страницы – группы примеров. В каждый момент времени лишь одна страница примеров располагается в памяти компьютера, остальные – на жестком диске. Страницы последовательно загружаются в память компьютера. Обучение сети происходит по всей совокупности страниц примеров, по всей обучающей выборке.

В настоящее время отсутствует универсальная методика построения обучающих выборок. Набор обучающих примеров формируется по усмотрению пользователя программы моделирования нейронных сетей индивидуально для каждой конкретной решаемой задачи [2].

Если в необученную нейронную сеть ввести входной сигнал одного из примеров обучающей выборки, то выходной сигнал сети будет существенно отличаться от желаемого выходного сигнала, определенного в обучающей выборке. Функция ошибки численно определяет сходство всех текущих выходных сигналов сети и соответствующих желаемых выходных сигналов обучающей выборки. Наиболее распространенной функцией ошибки является среднеквадратичное отклонение. Однако предложены и другие функции ошибки.

Для обучения нейронных сетей могут быть использованы различные алгоритмы. Можно выделить две большие группы алгоритмов – градиентные и стохастические. Градиентные алгоритмы обучения сетей основаны на вычислении частных производных функции ошибки по параметрам сети. Среди градиентных различают алгоритмы первого и второго порядков. В стохастических алгоритмах поиск минимума функции ошибки ведется случайным образом [2].

При обучении сетей, как правило, используется один из двух следующих критериев останова: останов при достижении некоторого малого значения функции ошибки, останов в случае успешного решения всех примеров обучающей выборки.

Перед обучением выполняется инициализация нейронной сети, то есть присваивание параметрам сети некоторых начальных значений. Как правило, эти начальные значения – некоторые малые случайные числа.

Для формирования обучающих выборок, инициализации и обучения в программах моделирования нейронных сетей используются специальные процедуры. Возможность использования многостраничного обучения является очень важной при решении практических задач с помощью нейронных сетей, моделируемых на обычных компьютерах.

Обучение – это итерационная процедура, которая при реализации на обычных компьютерах, требует значительного времени. Алгоритмы обучения существенно различаются по скорости сходимости. Одной из самых важных характеристик программ для моделирования нейронных сетей является скорость сходимости алгоритма (или алгоритмов) обучения, которые реализованы в программе [2].

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: емкость, сложность образцов и вычислительная сложность. Под емкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут быть на ней сформированы. Сложность образцов определяет число обучающих примеров, необходимых для достижения способности сети к обобщению. Слишком малое число примеров может вызвать «переобученность» сети, когда она хорошо функционирует на примерах обучающей выборки, но плохо – на тестовых примерах, подчиненных тому же статистическому распределению. Известны 3 основных типа правил обучения: коррекция по ошибке, машина Больцмана и правило Хебба.

Правило коррекции по ошибке. При обучении с учителем для каждого входного примера задан желаемый выход  $d$ . Реальный выход сети  $y$  может не совпадать с желаемым. Принцип коррекции по ошибке при обучении состоит в использовании сигнала  $(d-y)$  для модификации весов, обеспечивающей постепенное уменьшение ошибки. Обучение имеет место только в случае, когда персептрон ошибается. Известны различные модификации этого алгоритма обучения [2].

Обучение Больцмана. Представляет собой стохастическое правило обучения, которое следует из информационных теоретических и термодинамических принципов. Целью обучения Больцмана является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Обучение Больцмана может рассматриваться как специальный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

Правило Хебба. Самым старым обучающим правилом является постулат обучения Хебба. Хебб опирался на следующие нейрофизиологические наблюдения: если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью этого правила является то, что изменение синаптического веса зависит только от активности нейронов, которые связаны данным синапсом. Это существенно упрощает цепи обучения.

Для того что бы проверить навыки, приобретённые нейронной сетью в процессе обучения, используется имитация функционирования сети. В сеть вводится некоторый сигнал, который, как правило, не совпадает ни с одним из входных сигналов примеров обучающей выборки. Далее анализируется полученный выходной сигнал сети. Тестирование обученной сети может проводиться на одиночных входных сигналах, либо на контрольной выборке, которая имеет структуру, аналогичную обучающей выборке [2].

### 1.1 Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки (back propagation) относится к методу обучения с коррекцией по ошибке и, как правило, применяется к многослойным персептронам. Это метод обучения «с учителем», при котором «наставник» обучает сеть, также как ребенка обучают читать и писать. При обучении на входной слой многократно подаются образы сигналов, распознаванию которых нейронная сеть должна быть обучена, и корректируются веса нейронов для достижения желаемого выходного сигнала. Для улучшения качества распознавания образы, подаваемые на входной слой, могут быть слегка изменены (добавлен шум и т.п.). Детально процедура обучения выглядит так:

1) Выборка входных данных (множество образов, классифицированных учителем) разбивается на две: обучающую и контролирующую последовательности. Обычно обучающая последовательность содержит больше образов, чем проверочная.

2) Производится инициализация всех весов, включая пороговые, небольшими случайными величинами (обычно в диапазоне  $[-1; +1]$ ). Это определяет начальную точку на поверхности ошибок для метода градиентов, позиция может оказаться решающей для сходимости сети.

3) Производится прямой проход сети для первого образа из обучающей выборки от входного слоя через скрытые слои к выходному слою: каждый нейрон суммирует произведения входов на веса и выдает результат функции активации, примененной к этой сумме, на нейроны следующего уровня.

4) Вычисляется разность между действительным и желаемым выходным значением каждого нейрона выходного слоя. При их несовпадении имеет место ошибка в распознавании (классификации) образов.

5) Производится процедура обратного распространения этих ошибок по связям от выходных нейронов к входным и определяются ошибки для каждого нейрона. Рассмотрим эту процедуру подробнее. Предположим, выходное значение нейронной сети было 0.5, а желаемое 0. Пусть ошибка определяется по формуле

$$E = y_i - d_i,$$

где  $E$  – ошибка сети,

$y_i$  – желаемое значение на  $i$ -ом выходе нейронной сети,

$d_i$  – выходное значение сети на  $i$ -ом выходе.

В данном случае, на выходе ошибка будет равна  $E = 0.5 - 0 = 0.5$ . Если рассматривать в качестве функции активации сигмоид, то расчет  $\Delta w$ , на который необходимо сдвинуть веса синаптических связей предпоследнего последнего слоя, будет производиться по формуле

$$\Delta w = E * \text{sigmoid}(x)dx$$

Здесь  $\text{sigmoid}(x)dx$  – производная функции активации, равная  $\text{sigmoid}(x)(1-\text{sigmoid}(x))$ . Под  $x$  подразумевается значение, которое было получено путем суммирования всех выходных значений нейронов предпоследнего слоя, соответственно умноженных на  $w_i$ . Завершающий шаг коррекции весов синаптических связей между последним и предпоследним слоем будет производиться по следующей формуле

$$w_i = w_i - m_i * \Delta w * l,$$

где  $m_i$  – выходные значения  $i$  нейрона предпоследнего слоя,

$l$  – параметр, отвечающий за скорость обучения.

После того как веса посчитаны, мы можем посчитать ошибку выходного значения  $m_i$  по формуле  $E = w_i * \Delta w$ . Зная ошибку каждого нейрона на предыдущем слое, применяя формулы, которые использовались выше, можем найти отклонения весов и значения нейронов на предыдущих слоях.

б) Вновь производится прямой проход сети уже для очередного образа обучающей выборки...

Шаги 3–6 повторяются до достижения некоторого критерия, например, достижения ошибкой установленного предела.

По завершению обучающей фазы сеть проверяется при помощи контролирующей последовательности, содержащей образы, не предъявленные ранее. (При этом не производится коррекция весов, а лишь вычисляется ошибка). В случае, если качество работы найдено удовлетворительным, сеть считается готовой к работе. В противном случае сеть подвергается повторному обучению, при котором возможно изменение некоторых параметров (начальные веса, количество нейронов в скрытых слоях, дополнительные обучающие образы и т.п.) [3].

#### **Заключение**

При обучении нейросетей используются можно использовать несколько способов основные из которых перечислены в данной статье.

#### ЛИТЕРАТУРА

1. Нейронные сети для обработки информации. С. Осовский. Пер. с польского И.Д. Рудинского. – М. : Финансы и статистика, 2012. – 344 с.
2. Neural Networks: A Systematic Introduction. R. Rojas. Springer Science & Business Media – 502 с.
3. Лекции по нейронным сетям [Электронный ресурс] – <https://www.intuit.ru/studies/courses/6/6/lecture/178> (15.09.2019).

УДК 004.021

## ПРОЕКТИРОВАНИЕ ВЕБ-ИНТЕРФЕЙСА ДЛЯ ВВЕДЕНИЯ ДИАЛОГА С ЧАТ-БОТОМ

**К.С. САМСОНОВ**  
(Представлено: Ю.Н. КРАВЧЕНКО)

В данной статье рассматривается построение интерфейса пользователя для чат-бота.

**Введение.** Long Polling – это технология, которая позволяет получать данные о новых событиях с помощью «длинных запросов». Сервер получает запрос, но отправляет ответ на него не сразу, а лишь тогда, когда произойдет какое-либо событие (например, придет новое сообщение), либо истечет заданное время ожидания. Используя этот подход, Вы можете мгновенно отображать в своем приложении важные события. С помощью User Long Poll API Вы не сможете отправить сообщение, для этого используйте метод.

Callback API – это инструмент для отслеживания активности пользователей в Вашем сообществе ВКонтакте. С его помощью Вы можете реализовать, например, Бота для отправки мгновенных ответов на поступающие сообщения. Систему автоматической модерации контента.

**Основной раздел.** Callback API присылает уведомления на Ваш сервер, как только в сообществе происходит нужное событие. Событие может быть каким угодно: комментарий к фотографии, новая запись на стене, вступление в сообщество, отправка сообщения, и многое другое.

Для подключения Callback API в сообществе нужно указать адрес скрипта на Вашем сервере и выбрать события, которые Вы хотите получать. Например, если Ваш бот должен распознавать текстовые команды, отметьте событие «Входящее сообщение»

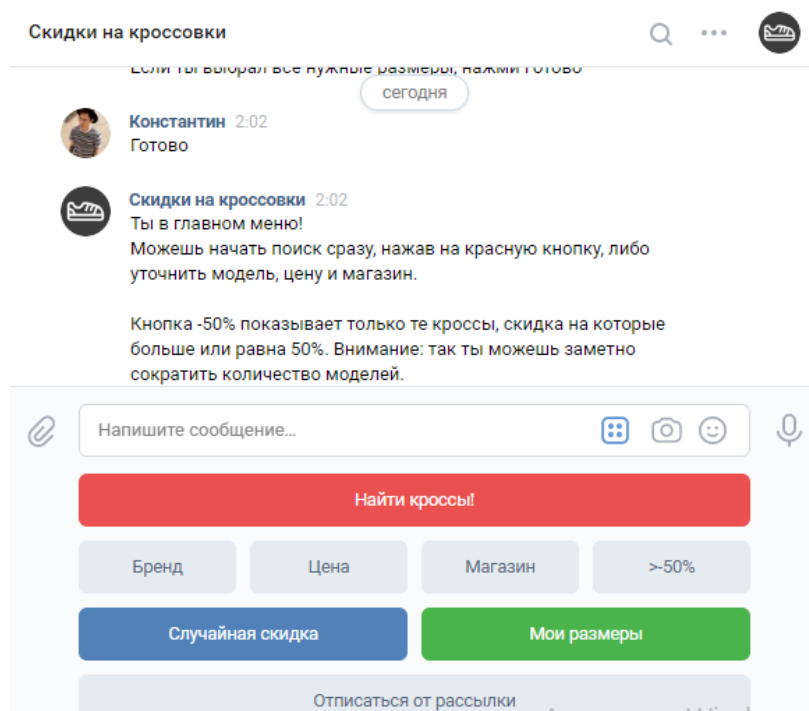


Рисунок 1. – Общение с чат ботом

Long Polling – это технология, которая позволяет получать данные о новых событиях с помощью «длинных запросов». Сервер получает запрос, но отправляет ответ на него не сразу, а лишь тогда, когда произойдет какое-либо событие (например, придет новое сообщение), либо истечет заданное время ожидания. Используя этот подход, Вы можете мгновенно отображать в своем приложении важные события. С помощью User Long Poll API Вы не сможете отправить сообщение, для этого используйте метод. Перед подключением к Long Poll серверу необходимо получить данные сессии (server, key, ts) методом messages.getLongPollServer. Мы рекомендуем передавать актуальный номер версии Long Poll в параметре lp\_version.

Затем составьте запрос такого вида:

```
https://{server}?act=a_check&key={key}&ts={ts}&wait=25&mode=2&version=2
```

В нем используются следующие параметры:

- key – секретный ключ сессии;
- server – адрес сервера;
- ts – номер последнего события, начиная с которого нужно получать данные;
- wait – время ожидания (так как некоторые прокси-серверы обрывают соединение после 30 секунд, мы рекомендуем указывать wait=25). Максимальное значение – 90.
- mode – дополнительные опции ответа. Сумма кодов опций из списка:
  - 2 – получать вложения;
  - 8 – возвращать расширенный набор событий;
  - 32 – возвращать pts (это требуется для работы метода messages.getLongPollHistory без ограничения в 256 последних событий);
  - 64 – в событии с кодом 8 (друг стал онлайн) возвращать дополнительные данные в поле \$extra (см. Структура событий);
  - 128 – возвращать поле random\_id (random\_id может быть передан при отправке сообщения методом messages.send).
- version – версия. Актуальная версия: 3. Для версии 0 (по умолчанию) идентификаторы сообществ будут приходить в формате group\_id + 1000000000 для сохранения обратной совместимости. Мы рекомендуем использовать актуальную версию.

Для первого запроса в рамках сессии значения для параметров server, key и ts необходимо получить методом messages.getLongPollServer. В последующих запросах используйте те же server и key и новое значение ts, которое придет Вам в ответе от Long Poll сервера.

**Заключение.** В данной статье рассмотрены основные элементы построения интерфейса для чат-бота. Представлены примеры и результаты.

УДК 004.021

**ПРОЕКТИРОВАНИЕ API ДЛЯ ЧАТ-БОТА,  
РАЗРАБОТАННОГО ДЛЯ ПОИСКА ТОВАРА ВНУТРИ МАГАЗИНА****К.С. САМСОНОВ***(Представлено: Ю.Н. КРАВЧЕНКО)*

*В данной статье рассматриваются цели и принципы моделирования API для чат-бота*

**Введение.** Чат-бот – это скрипт на Вашем сервере, который получает уведомления о новых событиях из ВКонтакте и обрабатывает их. Например, определяет текстовую команду в сообщении от пользователя и присылает в ответ какое-нибудь изображение.

Для создания бота Вам потребуются:

- Сообщество, от имени которого Ваш бот будет общаться с пользователями ВКонтакте.
- Сервер, который будет принимать уведомления о событиях.
- Логика самого бота – скрипт, который определяет, как бот реагирует на то или иное событие.

В первую очередь Вам нужно продумать функциональность чат-бота. Составьте список возможных текстовых команд или событий, на которые бот должен реагировать, и соответствующие им ответные действия бота. Стоит помнить, что люди могут ошибаться, набирая команды, и присылать боту текст, который отличается от всех учтенных Вами вариантов – важно предусмотреть действие по умолчанию (можно в таком случае присылать пользователю список всех поддерживаемых команд).

**Основной раздел.** Для подключения чат-бота Вы можете использовать любое свое сообщество ВКонтакте – группу, встречу или публичную страницу.

Первым делом нужно указать, что в вашем сообществе будет работать бот. Для этого перейдите в «Управление сообществом» → «Сообщения» → «Настройки для бота» и включите пункт «Возможности ботов».

Сообщество – это точка входа для пользователей, которые будут общаться с Вашим ботом. Стоит позаботиться о том, чтобы оно выглядело привлекательно и содержало описание Вашего бота, иначе его действия могут стать сюрпризом для пользователей. Не забудьте включить сообщения в Вашем сообществе («Управление сообществом» → «Сообщения»), когда бот будет готов к использованию, чтобы ему можно было написать.

**1.1. Получение ключа доступа.** Ключ доступа потребуется Вам для получения обновлений (в Long Poll), а также для работы с API. Откройте раздел «Управление сообществом» («Управление страницей», если у Вас публичная страница), выберите вкладку «Работа с API» и нажмите «Создать ключ доступа». Вы можете создать несколько ключей с разными правами доступа. Ключи нельзя размещать публично – узнав его, третье лицо может обращаться к API ВКонтакте от имени Вашего сообщества. Если ключ был скомпрометирован, необходимо удалить его из списка – после этого он станет недействителен. Также Вы можете получить ключ доступа с помощью OAuth. Используйте схему авторизации Authorization Code Flow.

**1.2. Добавление ботов в беседы.** Чтобы пользователи смогли добавлять бота в беседы, необходимо включить эту возможность в настройках. Перейдите в «Управление сообществом» → «Сообщения» → «Настройки для бота» и поставьте галочку в пункте «Разрешать добавлять сообщество в беседы». После этого в сообществе появится кнопка «Пригласить в беседу», открывающая модальное окно со списком бесед. Сейчас у бота в беседе есть три уровня доступа:

Администратор

- Выдать его может только создатель беседы. Бот может:
- видеть все сообщения в беседе
- отправлять сообщения в беседу
- Получать объект беседы
- Получать участников беседы

```
public static function init($config)
{
    $capsule = new Capsule;
    $capsule->addConnection([
        'driver' => 'mysql',
        'host' => $con-
fig['db_host'],
```

```
        'database' => $con-  
fig['db_name'],  
        'username' => $config['db_us-  
er'],  
        'password' => $config['db_pass-  
word'],  
        'charset' => 'utf8mb4',  
        'collation' =>  
'utf8mb4_unicode_ci',  
        'prefix' => '',  
    ];  
};
```

**Заключение.** В данной статье были рассмотрены основные цели и принципы моделирования API, для веб-приложения. Спроектированы общая архитектура разработки чат-бота.

УДК 004.415.25

**РЕАЛИЗАЦИЯ ГОРИЗОНТАЛЬНОГО ПЕРЕМЕЩЕНИЯ В ПРИЛОЖЕНИИ  
ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ****М.А. СЕРГЕЕВ***(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

*В статье представлен практический способ реализации механизма горизонтального перемещения в приложениях виртуальной реальности разрабатываемых на Unreal Engine 4. Решена задача реализации горизонтального перемещения игрока с учетом тренажерной болезни.*

**Ключевые слова:** информационные технологии, виртуальная реальность, тренажерная болезнь, Unreal Engine 4.

**Введение.** Первоочередная задача при создании игрового приложения виртуальной реальности – это определение типа перемещения игрока внутри игрового мира. Всего можно выделить пять вариантов перемещения: естественное перемещение, телепортация, линейное перемещение, физическое перемещение, искусственное перемещение и транспортные средства. Телепортация, по сути, переносит все фиксированное игровое пространство в место, указанное игроком, в остальном игрок, как и при естественном перемещении, ограничен физическим пространством, регистрирующимся датчиками. Транспортные средства помогают ограничить тренажерную болезнь, позволяя игроку иметь визуальный якорь в виде виртуального объекта, внутри которого игрок движется в виртуальном мире. Физическое перемещение позволяет игроку двигать себя при помощи некоторых физических действий, например, взмахи крыльями, что приводит к ожидаемому для игрока результату и снижает эффект «vr motion sickness». Искусственное же перемещение дает игроку большую свободу, чем остальные варианты и подходит для создания экшн-игр и лабиринтов, оно контролируется при помощи минимальных для игрока движений (нажатие кнопки на контроллере), что привычно для большинства пользователей и позволяет проще переносить уже существующие игры в VR. [1]

Однако, искусственное перемещение сильнее, чем остальные виды перемещения приводит к тренажерной болезни («vr motion sickness», «simulator sickness»), которое сильно уменьшает время одного игрового сеанса. Эффект тренажерной болезни представляет собой сбой в ориентации мозга в пространстве, связанный с тем, что ощущения вестибулярного аппарата не соответствуют изображению, поступающему по визуальным и тактильным каналам [2].

В данной статье будет представлена надежная реализация линейного перемещения.

**Реализация линейного перемещения.** Игровым движком для реализации приложения был выбран Unreal Engine 4. Таким образом, разработка будет вестись при помощи Blueprints, графического языка программирования.

Приложение будет состоять из нескольких функциональных частей, которые взаимодействуют между собой. Компонент контроллера, хранящий в себе 3d-модель контроллера, его анимации и часть логики взаимодействия BP\_MotionController [3], компонент игрока MotionControllerPawn [4], хранящий в себе весь основной функционал пользователя.

Для реализации передвижения игрока в горизонтальной плоскости с озвучкой шагов было разработано следующее решение (рисунок 1).

С джойстика (для Oculus) или тачпада (для HTC Vive) считывается значение положения, которое изменится от -1 до 1 по осям X и Y. Это значение передается в функции Moving Forward для оси Y и Moving Right для оси X. Также эти значения сравниваются с нулем, так как считываются постоянно, и, если хотя бы одно из них не равно нулю, меняет значение булевой переменной Is Moving на положительное, если же оба значения равны нулю, Is Moving становится отрицательной.

В функциях же значения осей умножаются на константу равную 0.6, выявленную экспериментальным путем как наиболее комфортную для такого типа перемещения. Далее эта константа, путем встроенной функции Add Movement Input, суммируется с направлением взгляда камеры вперед для Move Forward и вправо для Move Right.

При такой скорости перемещения, в ходе тестирования было проверено среднее время игровой сессии до наступления дезориентации. Согласно данным тестирования, при первой игровой сессии среднее значение ее длительности равняется 6 минут. При повторных время может увеличиться до 30 минут без возникновения тренажерной болезни.



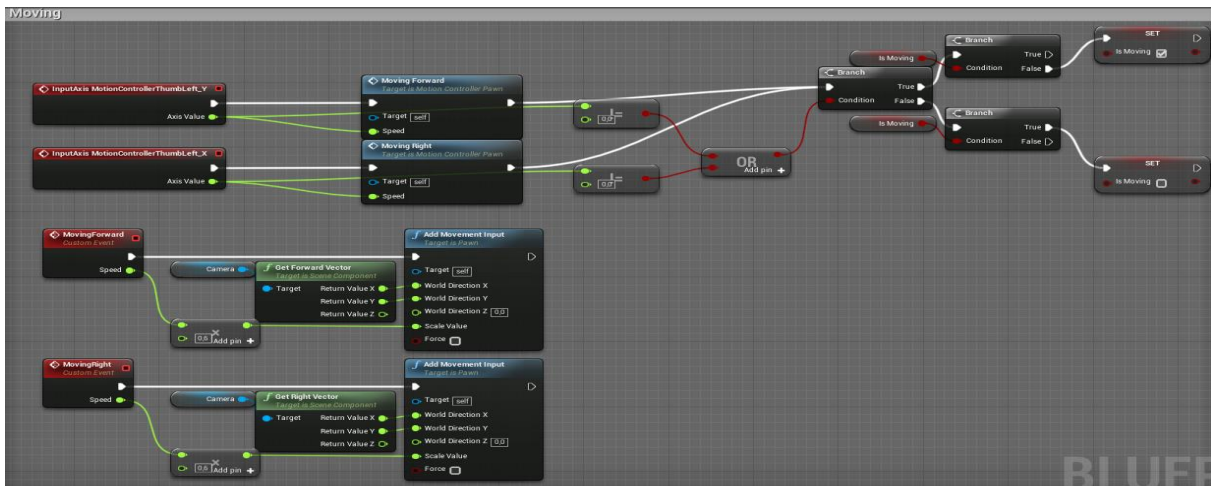


Рисунок 1. – Реализация линейного передвижения при помощи

**Реализация звуковой составляющей перемещения.** Переменная *Is Moving* является флагом, инициирующим запуск звука шагов во время перемещения по горизонтальной плоскости. Проверка, запускающая или отключающая звук шагов находится в корневом событии компонента *MotionControllerPawn*, а именно в *EventBeginPlay* (рисунок 2).

Цикл воспроизведения звуков шагов воспроизводит не один повторяющийся звуковой файл с заданным количеством шагов, а случайный из семнадцати отдельных семплов шага, которые выбираются в компоненте *Random Footstep Sound Cue*. Именно запуск этого компонента и инициируется переменной *Is Moving*, когда она получает значение *true*.

Звуковая составляющая перемещения также ускоряет привыкание к этому типу передвижения и акклиматизацию игрока в игровом пространстве.

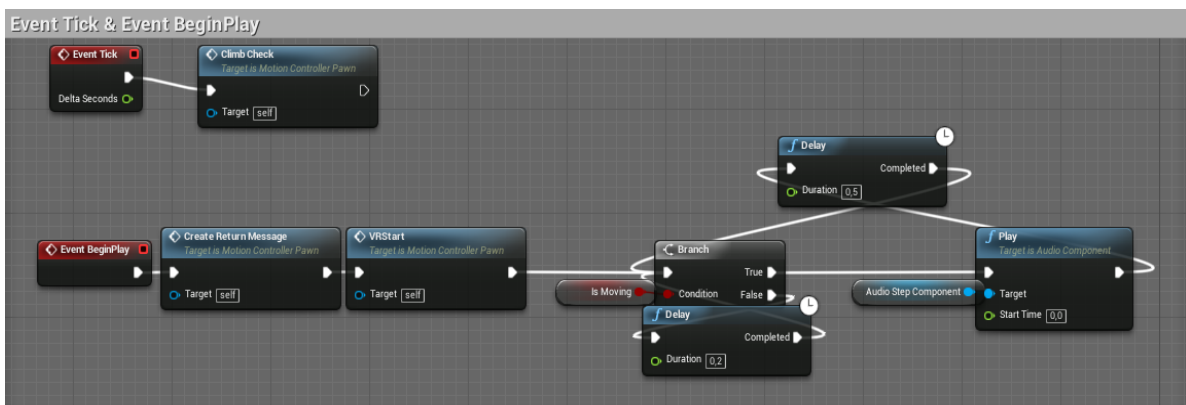


Рисунок 2. – Реализация запуска аудиосоставляющей перемещения

**Заключение.** Тип перемещения, подходящего под задачи VR-приложения, является одним из основополагающих факторов в восприятии виртуального мира игроком. Соответственно, при разработке приложений виртуальной реальности не следует преуменьшать значимость эффекта «vr motion sickness», обращая внимание не только на момент его наступления в первой игровой сессии, но и на общую динамику повторных игровых сессий.

Рассмотрены инструменты игрового движка Unreal Engine 4, позволяющего создавать приложения практически любого уровня сложности, а также удобного в создании приложений виртуальной реальности. По итогам данной работы, был описан метод создания модуля перемещения игрока в приложении виртуальной реальности.

## ЛИТЕРАТУРА

1. Unreal Engine VR для разработчиков / Митч Маккеффри; [пер. с англ. Н.И. Веселко, О.В. Максименковой, А.А. Незнанова]. – Москва : Эксмо, 2019. – 256 с.

2. UXplanet «Motion sickness in VR» by Anastasiia Ku [Электронный ресурс]. – Режим доступа: <https://uxplanet.org/motion-sickness-in-vr-3fa8a78216e2> – Дата доступа: 11.09.2019.
3. Unreal Engine 4 Documentation [Электронный ресурс]. – Режим доступа: <https://docs.unrealengine.com/en-us/> – Дата доступа: 10.09.2019.
4. Unreal Engine 4 Документация [Электронный ресурс]. – Режим доступа: <https://uengine.ru/docs> – Дата доступа: 10.09.2019.

УДК 004.415.25

РЕАЛИЗАЦИЯ ВЕРТИКАЛЬНОГО ПЕРЕМЕЩЕНИЯ В ПРИЛОЖЕНИИ  
ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

М.А. СЕРГЕЕВ

(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

В статье представлен практический способ реализации механизма вертикального перемещения в приложениях виртуальной реальности разрабатываемых на Unreal Engine 4. Решена задача реализации перемещения игрока по вертикальным поверхностям.

**Ключевые слова:** информационные технологии, виртуальная реальность, Unreal Engine 4.

**Введение.** Игровые приложения виртуальной реальности часто базируются на взаимодействии с мелкими деталями и объектами виртуальных миров. Примеров взаимодействия игрока с игровым пространством в более глобальном смысле, имеется не так много. В основном это связано с ограничениями, накладываемыми реализацией vr-приложений. Кроме того, необходимо снизить риск дезориентации игрока в пространстве, что чревато возникновением эффекта «укачивания» игрока.

Однако, возможность взаимодействия игрока с таким глобальным объектом игрового мира, как он сам, вполне возможна и без укачивания.

В данной статье будет представлена надежная реализация перемещения игрока в виртуальной реальности по вертикальным поверхностям.

**Реализация вертикального перемещения.** Игровым движком для реализации приложения был выбран Unreal Engine 4. Таким образом, разработка приложения будет вестись при помощи Blueprints, графического языка программирования.

Приложение будет состоять из нескольких функциональных частей, которые взаимодействуют между собой. Компонент контроллера, хранящий в себе 3d-модель контроллера и его анимации BP\_MotionController [1], компонент игрока MotionControllerPawn [2], хранящий в себе весь основной функционал пользователя, интерфейс, связывающий игрока и компоненты, с которыми он взаимодействует, и компонент поверхностей для вертикального перемещения BP\_ClimbObj, хранящий в себе модель примитива поверхности для вертикального перемещения, а также специально созданный материал для упрощенного визуального восприятия объектов этого типа. Помимо этого, компонент BP\_ClimbObj хранит в себе логику перемещения по вертикальным поверхностям (рисунки 1–3).

Логика подобного перемещения следующая: у игрока имеется два контроллера, каждый из которых взаимодействует с объектом определенного типа (специальной стеной для перемещения). В момент начала взаимодействия, вся модель игрока начинает двигаться относительно точки, в которой началось взаимодействие контроллера и объекта, данное движение происходит с помощью контроллера, модель которого зафиксирована в вышеуказанной точке.

На рисунке 1 представлено событие, определяющее точку отсчета, относительно которой двигается игрок. Помимо этого, в данном событии также обновляется переменная, в которой ведется счет количества контроллеров, который взаимодействуют с объектом.

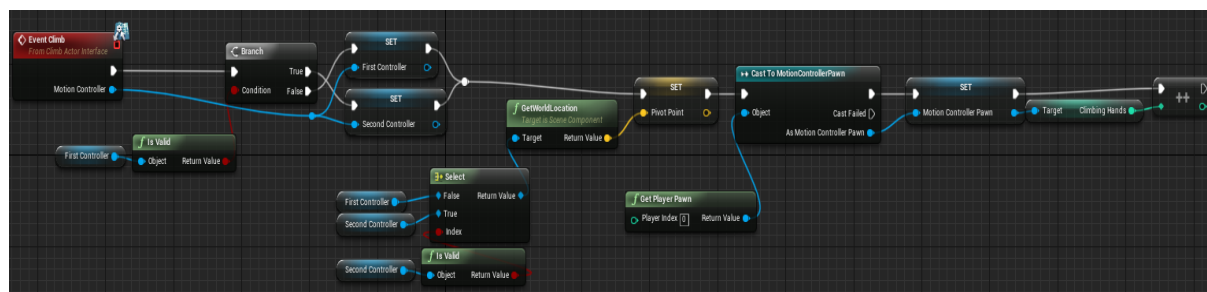


Рисунок 1. – Событие, определяющее точку отсчета

На рисунке 2 представлено событие, определяющее момент, когда взаимодействие прекращается. В нем также предусмотрен вариант развития событий, при котором во время взаимодействия одного контроллера с поверхностью, происходит взаимодействие другого контроллера. Соответственно, точка отсчета меняется, и переменная-счетчик взаимодействующих контроллеров обновляется снова.

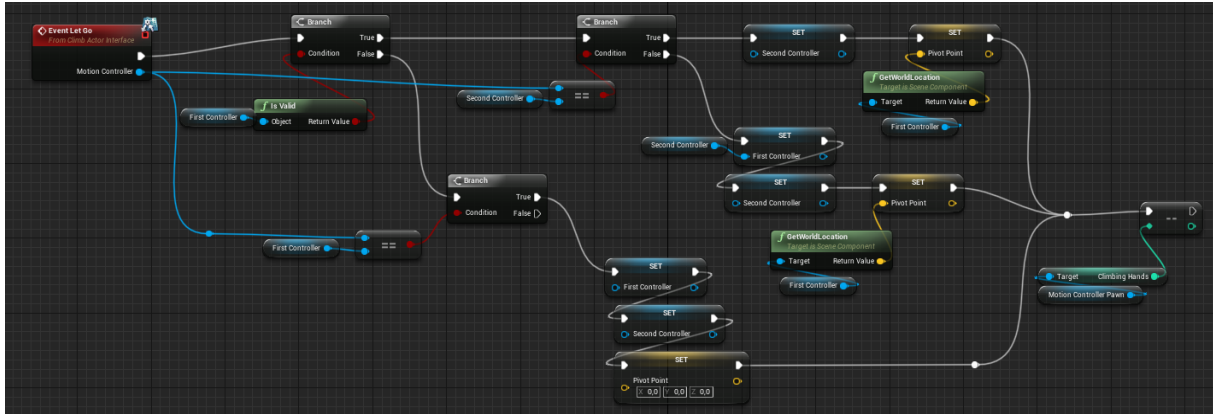


Рисунок 2. – Событие прекращения взаимодействия

На рисунке 3 продемонстрировано обновление позиции игрока в реальном времени. Оно проверяет переменную Is Climbing, принимающую значение true, в момент взаимодействия контроллера с поверхностью, и хранящуюся в компоненте MotionControllerPawn и валидность контроллера и самого игрока. После данной проверки инициируется перемещение модели игрока при помощи узла AddActorLocalOffset, который за точку отсчета берет точку, в которой началось взаимодействие контроллера с поверхностью. Узел же, который следует за ним, возвращает контроллер игрока в эту точку.

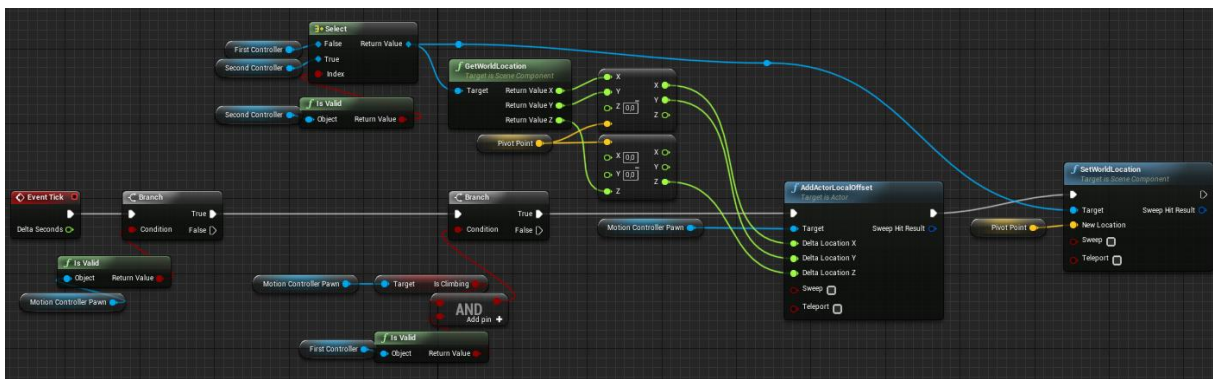


Рисунок 3. – Логика обновления позиции игрока в 3d-пространстве

На рисунках 4-6 продемонстрирован код падения игрока, после окончания взаимодействия с поверхностью перемещения. Располагается данная часть кода в компоненте MotionControllerPawn.

На рисунке 4 продемонстрированы проверки, определяющие взаимодействует ли хоть один контроллер с поверхностью, либо же нет. В случае, если взаимодействующих контроллеров больше нуля, переменная IsClimbing присваивает значение true, а IsFalling – false. В ином случае, идет дополнительная проверка вышеуказанных переменных, приводящая либо к присваиванию переменной взаимодействующих контроллеров значение 0, либо присваивание переменной IsClimbing значение false, а IsFalling – true. При повторной проверке значение переменной IsFalling true проиницирует начало падения.

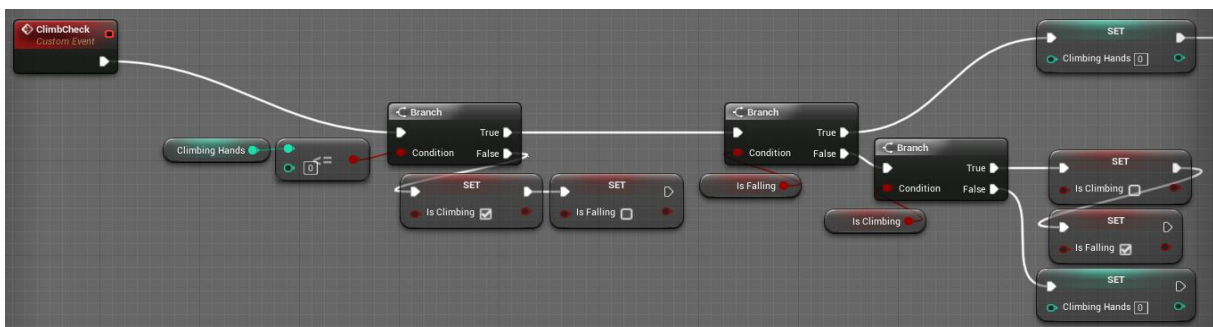


Рисунок 4. – Событие падения ч. 1

На рисунке 5 продемонстрирована проверка, определяющая, находится ли игрок уже на поверхности при помощи узла LineTraceByChannel. В случае, если игрок уже находится на такой поверхности, IsClimbing и IsFalling принимают значение false, а скорость падения (falling speed) становится равной нулю.

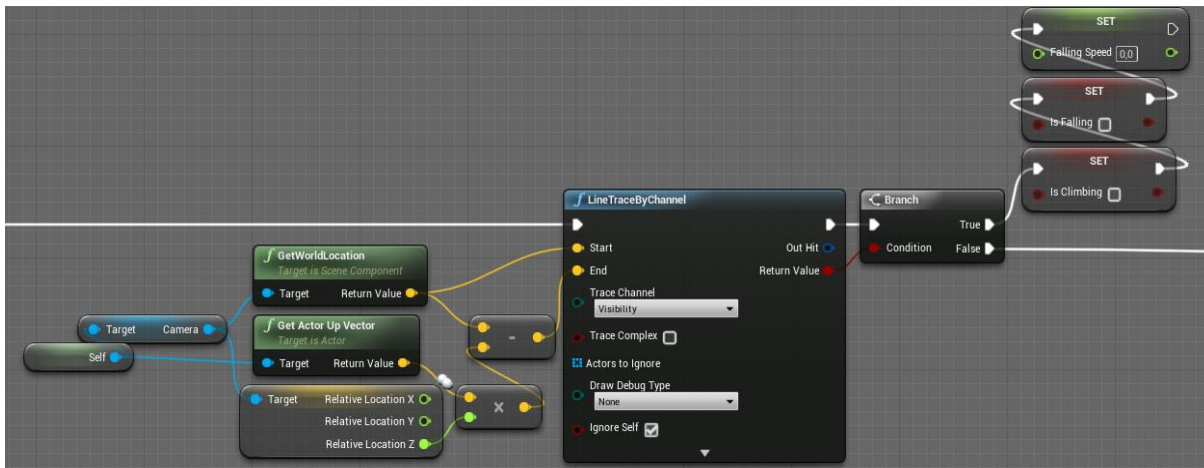


Рисунок 5. – Событие падения ч. 2

На рисунке 6 продемонстрирована математическая часть реализации падения виртуальной модели игрока. Переменной Falling Speed, определяющей значение скорости падения присваивается ускорение, вычисляемое по значениям земной гравитации. Данная переменная ограничивается значением 300,0 и умножается на значение времени. Полученное число перемножается с инвентированным верхним вектором игрока, что переходит в узел AddLocalOffset, который, в свою очередь, приводит в движение модель игрока.

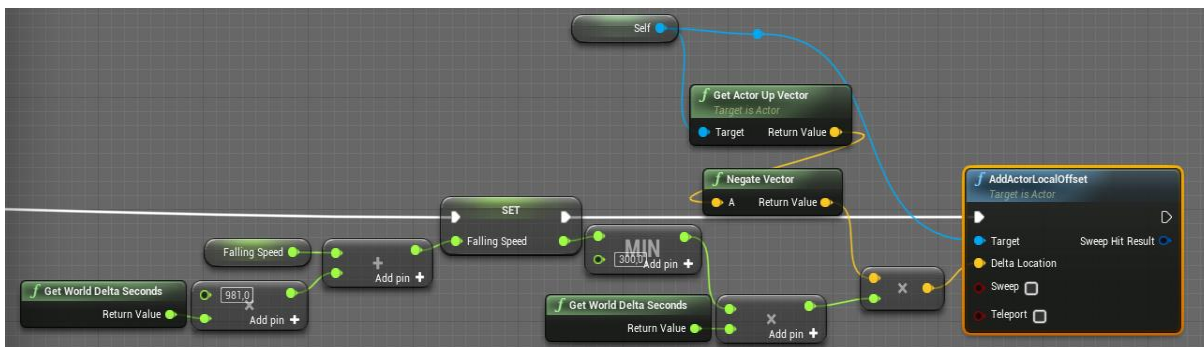


Рисунок 6. – Событие падения ч. 3

**Заключение.** Приложения виртуальной реальности могут базироваться на обширном количестве геймплейных концепций, влияющих на восприятие пользователем игрового мира, а также, затрагивающих, в том числе, столь базовую вещь, как тип передвижения пользователя в виртуальном мире. В данной статье была продемонстрирована реализация самой новаторской концепции перемещения модели игрока, а именно, физический тип перемещения, согласно которому игрок перемещает свою модель не только при помощи нажатий клавиш контроллера, но и при помощи реальных физических действий, совершаемых его телом [3].

Рассмотрены инструменты игрового движка Unreal Engine 4, позволяющего создавать приложения практически любого уровня сложности, а также удобного в создании приложений виртуальной реальности. По итогам данной работы, был описан метод создания модуля физического вертикального перемещения игрока.

ЛИТЕРАТУРА

1. Unreal Engine 4 Documentation [Электронный ресурс]. – Режим доступа: <https://docs.unrealengine.com/en-us/> – Дата доступа: 10.09.2019.

2. Unreal Engine 4 Документация [Электронный ресурс]. – Режим доступа: <https://uengine.ru/docs> – Дата доступа: 10.09.2019.
3. Unreal Engine VR для разработчиков / Митч Маккефри; [пер. с англ. Н.И. Веселко, О.В. Максименковой, А.А. Незнанова]. – Москва : Эксмо, 2019. – 256 с.

УДК 003.26

## ЦИФРОВЫЕ ВОДЯНЫЕ ЗНАКИ КАК МЕТОД ЗАЩИТЫ ИЗОБРАЖЕНИЯ

А.А. СИВОГРАКОВ

(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)

В данной статье представлена классификация нанесения цифровых водяных знаков. Разобраны виды алгоритмов встраивания по направлениям, выделены их основные преимущества и недостатки.

**Введение.** В эпоху компьютеров, наиболее распространенными нарушениями интеллектуальной собственности являются: плагиат, «пиратство», изменение информации, подделка информации, недобросовестная конкуренция.

При хранении, распространении и передачи интеллектуальной собственности, зачастую в качестве защиты, используют цифровые водяные знаки. Цифровой водяной знак (ЦВЗ) – это специальная метка, встраиваемая в цифровой сегмент, с целью тем или иным образом отслеживать распространение информации по сетям связи, обеспечивать поиск информации в мультимедийных базах данных. Цифровой водяной знак, применительно к изображению как правило не видим, то есть изначальное изображение, и изображение со встроенным ЦВЗ, визуально неотличимы зрительной системой человека.

**Классификация ЦВЗ.** Впервые термин «Digital watermarking» был введен в работе [1]. Он получил свое название от способа защиты от подделки ценных бумаг. В настоящее время разработаны различные методы нанесения ЦВЗ, классификация которых представлена на рисунке 1.



Рисунок 1. – Классификация методов ЦВЗ

Встраиваемый знак может быть как видимый глазу, так и не видимый. Второй вариант более распространен и делится на хрупкие, стойкие (робастные) и полухрупкие ЦВЗ. В случае хрупкой системы ЦВЗ, водяной знак разрушается после любых незначительных изменений контейнера. Такие знаки необходимы для аутентификации сигнала (цифровые отпечатки пальцев). Робастные знаки наоборот должны переносить многие виды атак: аффинные преобразования (повороты, обрезание), сжатие и другие. Как раз такие знаки и используются для определения авторства, так как их сложно разрушить. Полухрупкий ЦВЗ – это знак с избирательной сложностью. Такой знак может допускать определенные преобразования контейнера, разрушаясь от других [3].

Алгоритмы встраивания делятся на обратимые и необратимые. Обратимые алгоритмы позволяют извлечь водяной знак и полностью восстановить контейнер для дальнейшей работы. Такие алгоритмы применяются для медицинских и военных целей, где любые искажения изображений категорически запрещены. Необратимые алгоритмы при извлечении ЦВЗ вносят изменения в первоначальный контейнер, поэтому при разработке таких алгоритмов цель разработчика заключается в снижении уровня искажений до минимума.

В более сложных обратимых методах можно выделить алгоритмы, основанные на модификациях гистограмм изображения и алгоритмы, основанные на преднамеренной регулировке значения разности между соседними парами пикселей. Первая группа является простой в реализации и использует минимум информации для декодера, недостатками же является ограничение объема встраивания, который зависит

от количества вхождений точек максимума яркости. Вторая группа алгоритмов позволяет встраивать в сообщение большие объемы информации, но при этом ухудшается качество маркированного изображения.

Различают линейные и, соответственно, нелинейные методы нанесения ЦВЗ, а также методы использующие фрактальное кодирование, основанное на предположении, что изображение самоподобно [2]. Линейные алгоритмы делятся на алгоритмы встраивания (аддитивные), когда цифровое изображение добавляется в цифровое сообщение, и алгоритмы слияния (fusion), когда в одно изображение встраивается другое, например, логотип.

Так же множеством разработчиков было предложено использование корреляционных алгоритмов. Но использование таких алгоритмов оправдано, если пользователю необходимо извлечь скрытое сообщение, и основной контейнер воспринимается как шум (необратимый метод). Главным преимуществом алгоритмов слияния перед алгоритмами встраивания является допущение легкое искажение ЦВЗ при извлечении.

Алгоритмы пространственной области внедряют ЦВЗ в исходное изображение. Их преимуществом является то, что нет необходимости выполнять преобразования изображений. ЦВЗ в таких методах обычно внедряется за счет манипуляций яркостью или цветовыми составляющими. Недостатком таких алгоритмов является довольно слабая устойчивость к различным операциям обработки изображений.

Частотные алгоритмы, основанные на преобразованиях изображения, реализуются сложнее, т.к. перед внедрением ЦВЗ необходимо «перераспределить энергию» контейнера, чтобы встроить сообщение в специальные спектральные области. За счет подобной декомпозиции изображения ЦВЗ становится робастным к атакам.

Наибольшую сложность представляет внедрение ЦВЗ в низкочастотную область, содержащую большую часть энергии изображения, потому что неоптимальное внедрение может привести к значительному искажению контейнера. Данная сложность является одновременно и преимуществом, поскольку любая попытка злоумышленника извлечь ЦВЗ из низкочастотной области также приведет к значительному искажению изображения. Таким образом, при встраивании ЦВЗ в частотную область изображения необходимо соблюдение компромисса между объемом встраиваемого ЦВЗ и качеством стеганоcontainers [4].

Методы, основанные на моментах изображений, применяются для защиты ЦВЗ от геометрических преобразований контейнера. Однако они имеют узкую направленность, а их основным недостатком является низкий уровень безопасности от других видов атак.

**Заключение.** Цифровые водяные знаки на данный момент являются наиболее эффективным средством защиты авторских прав на произведения мультимедиа. Это один из основных способов предотвращения нарушений авторских прав в интернете. Это сфера быстро развивается, поэтому существует множество разных видов и типов встраивания. На сегодняшний день существует большое количество методов внедрения ЦВЗ, каждый из которых обладает своими достоинствами и недостатками, которые надо учитывать при использовании того или иного метода для защиты мультимедийных данных от незаконного распространения и модификации.

#### ЛИТЕРАТУРА

1. Osborne C., van Schyndel R., Tirkel A. A Digital Watermark // IEEE Intern. Conf. on Image Processing, 1994. P. 86-90.
2. Грибунин В.Г., Оков И.Н., Туринцев В.И. Цифровая стеганография. – М. : СОЛОН-Пресс 2002. 272 с.
3. Балакин А.В., Елисеев А.С., Гуфан А.Ю. / Использование стеганографических методов для защиты текстовой информации. – М. : Т-comm Телекоммуникации и транспорт, 2009. С 42–50.
4. Elshoura, S.M. A secure high capacity full-gray-scale-level multi-image information hiding and secret image authentication scheme via Tchebichef moments / S.M. Elshoura, D.B. Megherbi // Signal Processing: Image Communication. 2013. – Vol. 28. P. 531–552.



УДК 004.51

## РАЗРАБОТКА ИНТЕРФЕЙСА ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

А.А. СИВОГРАКОВ

*(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)*

*В данной статье рассматриваются ключевые особенности создания дизайна пользовательского интерфейса для мобильного приложения. На что стоит обратить внимание и какие ошибки допускать не стоит.*

**Введение.** В настоящее время, когда прогресс никого не обошел стороной, а смартфоны и планшетные компьютеры стали неотъемлемой частью нашей жизни, компаний-разработчиков мобильных приложений становится все больше и больше. И это не удивительно, учитывая стремительные темпы роста рынка мобильных приложений, как следствие роста рынка самих смартфонов [1].

Учитывая огромное количество приложений на мобильном рынке нужно четко понимать с чего же начинается сама разработка. Опуская все можно сказать, что все будет начинаться с разработки с разработки дизайна пользовательского интерфейса (UI – англ. User Interface), который крайне важен, чтобы мобильное приложение было удобно и понятно в использовании для пользователя [2].

**Особенности.** На этапе разработки UI нужно уделить внимание многим деталям, ведь в отличие от разработки веб-приложения, когда на поздних этапах перемены в изначальном дизайне не будут включать в себя больших затрат сил и времени, то в случае разработки мобильного приложения, перемены в дизайне на поздних стадиях могут серьезно пошатнуть темпы разработки, занимая значительное количество ресурсов и времени.

Так на что же стоит обратить внимание во время разработки дизайна? В первую очередь следует обратить внимание на следующую проблему. Учитывая, что на данный момент плоский дизайн стал преобладать на обеих наиболее популярных платформах для мобильных приложений, а именно Android и iOS, правильно распределить элементы управления становится сложнее. Ведь в плоском дизайне невозможно сделать визуальное разделение на несколько уровней, чтобы один элемент был «поверх» другого. В данном случае следует сделать акцент на визуальную иерархию, то есть располагать элементы управления правильным образом, а именно так, чтобы наиболее важные из них были выделены сильнее, находясь при этом с остальными элементами на одном уровне [3].

Далее следует учесть линию взгляда пользователя. Как правило, большинство пользователей будут просматривать экран приложения слева-направо и сверху-вниз. Учитывая данный фактор, стоит тщательно разработать сценарий использования приложения, чтобы пользователь смог интуитивно находить элементы управления по мере надобности. [3, 4].

Но полностью скомпоновать дизайн отнюдь не означает закончить его разработку. Ведь, как бы не расположив элементы управления на экранах приложения, дизайн пользовательского интерфейса должен оставаться «гибким» [5]. Под гибкостью дизайна подразумевается следующее: мобильное приложение должно смотреться гармонично на любом устройстве. Ведь при обилии различных разрешений и соотношении сторон экранов очень сложно сохранить один и тот же формат. Но чтобы не сохранять его, а изменять в зависимости от характеристик конкретного устройства, стоит использовать метод модульного программирования во время разработки. Суть данного метода заключается в том, чтобы при непосредственном программировании дизайна большинство наиболее значимых пунктов разбивались на отдельные самостоятельные части - модули. Ведь в данном случае, чтобы изменить ту или иную часть интерфейса, не придется исправлять весь код, потребуется всего лишь изменить модуль, отвечающий за элемент, который требуется изменить.

Также стоит соблюдать единство дизайна при переносе приложения с одной платформы на другую. Это нужно, чтобы пользователь не терял способность к быстрой адаптации при переходе, например, с мобильной версии приложения к ее веб-версии.

Важным этапом при разработке дизайн также является масштабирование элементов управления, ведь от того, как их расположить, во многом зависит степень удобства при использовании приложения. Как показывает статистика, 49% пользователей смартфонов, держат свои устройство одной рукой, используя для управления свой большой палец [6]. Отталкиваясь от этого, можно примерно представить площадь экрана, с которой пользователь сможет комфортно работать.

И наконец, стоит отличать упрощение UI от уменьшения количества экранов и элементов управления на них. Ведь при чересчур малом количестве рабочих экранов интерфейс может получиться перегруженным, из-за чего с ним будет работать не проще, а наоборот – сложнее.

Как уже было отмечено, многие пользователи предпочитают пользоваться своим смартфоном, используя только одну руку. Но с увеличением диагоналей самих устройств, данный способ эксплуатации станет все более и более неудобным. Так, средняя диагональ смартфона увеличилась на 0,8 дюйма за последние два года [2]. Из этого следует, что в ближайшем будущем пользоваться смартфоном одной рукой будет попросту неудобно, а от способа использования смартфона напрямую зависит расположение элементов управления для наиболее эффективного способа навигации внутри приложения.

Также на данный момент лучшие производители смартфонов развивают новый сегмент рынка носимой электроники – умные часы. В этом случае при разработке дизайна приложения для столь маленького устройства, диагональ экрана которого в среднем не превышает и двух дюймов, будут учитываться несколько другие правила, по сравнению с разработкой дизайна приложений для смартфонов.

**Заключение.** Результатом исследования стал ряд описанных особенностей, которые следует соблюдать при разработке дизайна UI для мобильно приложения.

#### ЛИТЕРАТУРА

1. J'son & Partners Consulting. Российский рынок смартфонов. Итоги 2014 года. URL: [http://www.json.ru/poleznye\\_materialy/free\\_market\\_watches/analytics/rynok\\_razrabotki\\_mobilnuh\\_prilozhenij/](http://www.json.ru/poleznye_materialy/free_market_watches/analytics/rynok_razrabotki_mobilnuh_prilozhenij/) (дата обращения 25.09.2019 г.).
2. J'son & Partners Consulting. Российский рынок смартфонов. URL: [http://json.tv/ict\\_telecom\\_analytics\\_view/rossiyskiy-rynok-smartfonov-itogi-2015012002050](http://json.tv/ict_telecom_analytics_view/rossiyskiy-rynok-smartfonov-itogi-2015012002050) (дата обращения 25.09.2019 г.).
3. Peter Eckert. Dollars And Sense: The Business Case For Investing In UI Design. URL: <http://www.fasteodesign.com/1669283/dollars-and-sense-the-business-case-for-investing-in-ui-design> (дата обращения 25.09.2019 г.).
4. Студия дизайна Genue. Что следует знать о плоском дизайне. URL: <http://habrahabr.ru/company/genue/blog/204194/> (дата обращения 25.09.2019 г.).
5. Modern UI Design for Web and Mobility Delivers a Better Overall User Experience. URL: <http://www.oracle.com/us/corporate/features/ui-design/index/> (дата обращения 25.09.2019 г.).
6. Thinking Beyond the PC – Why UI Design Will Play an Important Role in Application Interaction Across Multiple Platforms. URL: <http://www.boostlabs.com/thinking-beyond-the-pc-why-ui-design-will-play-an-important-role-in-application-interaction-across-multiple-Platforms/> (дата обращения 25.09.2019 г.).

УДК 004.223.2

**СЛОЖЕНИЕ ДВУХ ЭЛЛИПТИЧЕСКИХ ТОЧЕК И ГРУППОВЫЕ ОПЕРАЦИИ.  
СУЖЕНИЕ МНОЖЕСТВА ЭЛЛИПТИЧЕСКИХ ТОЧЕК НА МНОЖЕСТВО ОСТАТКОВ  
ЭЛЛИПТИЧЕСКОЙ КРИВОЙ. ГРАФИЧЕСКОЕ ТЕСТИРОВАНИЕ С ИНТЕРФЕЙСОМ.**

**П.Р. СИНИЦА**

*(Представлено: канд. физ.-мат. наук., доц. Д.Ф. ПАСТУХОВ)*

*В статье представлена определение сложения двух эллиптических точек и групповые операции. Сужение множества рациональных бесконечного числа эллиптические точки на конечное целочисленное множество остатков эллиптической кривой. Применение циклической абелевой группы. Программная реализация с графическим интерфейсом.*

**Ключевые слова:** эллиптическая кривая, шифрование, дешифрование, групповые операции, множество остатков, абелева группа, интерфейс.

**Введение.** Эллиптическая криптография – раздел криптографии, который изучает асимметричные криптосистемы, основанные на эллиптических кривых над конечными полями.

Шифрование данных методом эллиптических кривых преследует цели выработать метод быстрого и эффективного шифрования на базе эллиптической криптографии и в то же время повысить устойчивость шифрования (стойкость шифра) и целостность передаваемой информации в процессе протоколе обмена.

Роль основной криптографической операции выполняет операция скалярного умножения точки на эллиптической кривой на данное целое число, определяемое через операции сложения и удвоения точек эллиптической кривой. Последние, в свою очередь, выполняются на основе операции сложения, умножения и инвертирования в конечном поле, над которыми рассматривается кривая. Особый интерес к криптографии эллиптических кривых обусловлен теми преимуществами, которые дают ее применение в беспроводных коммуникациях – высокое быстродействие и небольшая длина ключа.

**Сложение двух эллиптических точек и групповые операции. Сужение множества.** Циклическую группу образуют из множества точек эллиптической кривой, связанных геометрической групповой структурой, дополняют полевой целочисленной структурой по модулю простого числа  $p$ :

$$y^2 = x^3 + ax + b \pmod{p} \quad (1)$$

В конечном итоге мы пользуемся формулами (3), (4) и (6), получая последовательно все точки эллиптической кривой циклической абелевой группы.

Формулы (1), (2), (3), (4) получены в первой части статьи:

$$\begin{cases} \left( \frac{y_2 - y_1}{x_2 - x_1} \right) \equiv (y_2 - y_1) \pmod{p} * (x_2 - x_1)^{-1} \pmod{p}, (x_2 - x_1)(x_2 - x_1)^{-1} \equiv 1 \pmod{p} \\ \frac{3x_1^2 + a}{2y_1} \equiv (3x_1^2 + a) \pmod{p} * (2y_1)^{-1} \pmod{p}, (2y_1) * (2y_1)^{-1} \equiv 1 \pmod{p} \end{cases} \quad (2)$$

Краткое описание алгоритма построения последовательности точек:

1. Находим обратный элемент к  $2y_1$ , либо к  $x_2 - x_1$ .
2. Находим числа  $k_1 = (y_2 - y_1) \pmod{p} * (x_2 - x_1)^{-1} \pmod{p}$ , либо  $k_1 = (3x_1^2 + a) \pmod{p} * (2y_1)^{-1} \pmod{p}$ .
3. Находим числа

$$\begin{cases} x = (k_1^2 - x_1 - x_2) \pmod{p} \\ y = (-y_1 + k_1(2x_1 + x_2 - k_1^2)) \pmod{p} \end{cases} \quad (3)$$

Либо по формулам:

$$\begin{cases} x = (k_1^2 - 2x_1) \pmod{p} \\ y = (-y_1 + k_1(3x_1 - k_1^2)) \pmod{p} \end{cases} \quad (4)$$

Если говорить вкратце, то формулу шифрования и дешифрования можно записать в виде:

$(kG, Pm + k * Pb)$  (шифрование)  $\rightarrow Pm + k * nb * G - nb * k * G = Pm$  (дешифрование), где  $nb$  – закрытый ключ абонента  $b$ , а  $Pb$  открытый ключ абонента  $b$ .

Программная реализация с графическим интерфейсом. В качестве интерфейса программы ввода данных можно рассмотреть, например, следующий интерфейс с вводом длины сообщения в символах  $nn$ , параметры эллиптической кривой  $a$ ,  $b$ ,  $p$ , открытый ключ  $(kx, ky)$ , случайное число  $k$ .

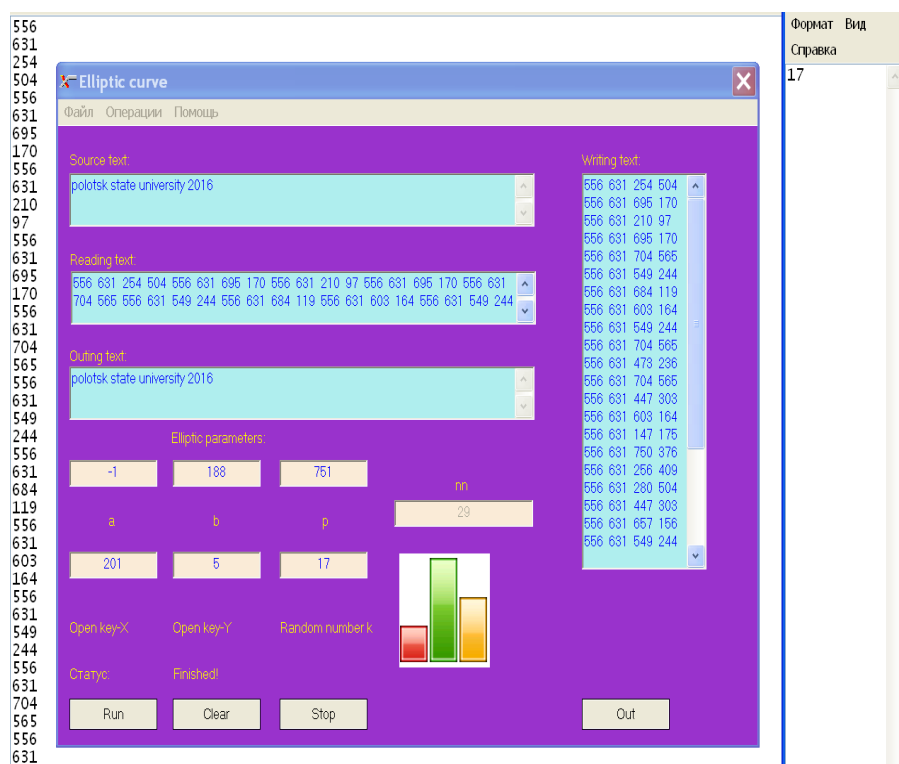


Рисунок 1. – Шифрование интерфейсом исходного текста

Интерфейс включает следующие окна: окна ввода чисел  $a$ ,  $b$ ,  $p$ , окна ввода открытого ключ Open Key – X, Open Key – Y, окно ввода случайного числа Random number  $k$ .

Функциональные элементы запуска, очистки и остановки интерфейса выхода из программы Run, Clear, Stop, Out.

Окно ввода первичной информации Source text, куда вводится исходный текст с помощью клавиатуры и установки курсора в окно ввода.

Окно шифрованного текста Writing text непосредственно перед записью шифра в текстовый файл *balk1.txt*.

Длина записи исходного текста в окне  $nn$  не является активной, т.е. в это окно ничего не заносится. Длина строки вводимого текста  $nn$  автоматически изменяется в данном окне при удалении или добавлении каждого нового символа.

Окно прочитанного шифра Reading text из файла *1.txt* должно совпадать с шифром в окне Writing text.

Окно расшифрованного текста Outing text должно выводить текст, посимвольно совпадающий с исходным текстом.

Данный интерфейс не только шифрует исходный текст, но и является программой – контролем на всех этапах шифрования и дешифрования и создания текстового файла для шифра.

Интерфейс содержит также функциональные окна с падающим списком опций, с помощью которых также можно шифровать исходный текст (например, операции – кодировать, закрыть; файл – просмотреть; помощь – о программе).

На рисунке 1 показана введенная фраза с латинским шрифтом Полоцкий государственный университет 2019: «*polotsk state university 2019*» с длиной строки  $nn = 29$ . Результат ввода текста, параметры эллиптической кривой  $a = -1$ ,  $b = 188$ ,  $p = 751$ , открытый ключ  $(kx, ky) = (201, 5)$ .

На рисунке 1 мы видим результат шифрования в поле Writing text. Каждому исходному символу текста соответствует четыре целые координаты двух точек эллиптической кривой, расположенных в одной строке. Для удобства ввода и построчного считывания тот же шифр в один столбец записывается в текстовый файл *1.txt*.

На рисунке 2 представлен интерфейс дешифрования, входными параметрами для которого являются данные из файла *1.txt* и *2.txt*.

Кроме того, интерфейс программы содержит пассивный логический оператор «Статус», который в случае успешного запуска возвращает состояние Finished. В случае ошибки данный логический оператор указывает состояние (род произошедшей ошибки).

Справа на рисунке 2 указано случайное число  $k = 17$ , записанное программой в текстовый файл 2.txt. Оно совпадает с числом, введенным в интерфейс в окно Random number  $k$ . Однако по каналу связи в открытом виде число  $k$  не передается, так как оно скрыто в шифре вида  $(k * G, Pm + k * Pb)$ . Число  $k$  в текстовый файл выводится только на этапе тестирования программы.

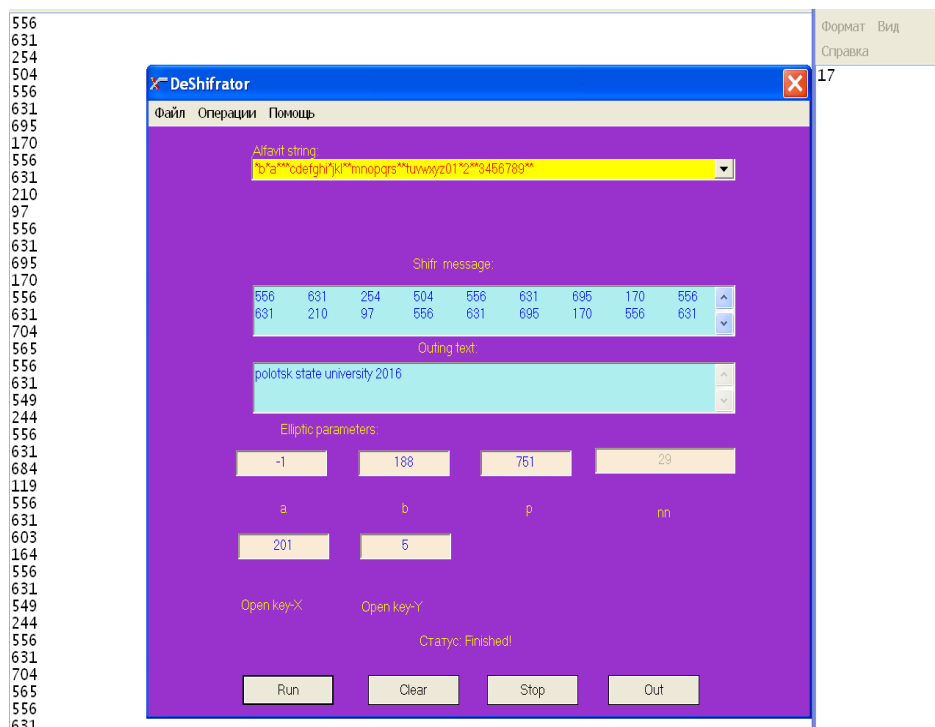


Рисунок 2. – Интерфейс дешифрования

Стоит отметить, что интерфейс дешифрования имеет 2 возможности:

- 1) загружать шифр из текстового файла и дешифровать отдельно;
- 2) вставляя шифр в окно шифра в режиме онлайн и дешифровать его в окно дешифрованного текста.

**Заключение.** Правильным подходом при разработке веб-интерфейса является использование современных технологий, которые позволяют решать свои задачи во всех браузерах. Это, в первую очередь, экономит время при разработке интерфейса, а также ресурсы на обработку данных. Это является очень актуальным, при большом количестве информации, которая подлежит постоянному изменению.

В данной статье был разработан дружелюбный интерфейс пользователя для шифрования и дешифрования данных на основе эллиптических кривых. Так же описано удобство использования.

#### ЛИТЕРАТУРА

1. Пастухов Д.Ф., Пастухов Ю.Ф., Сеница П.Р., Шифрование данных на базе эллиптических кривых: учебно-методическое пособие для студентов спец. 1-98 01 01; Электронная библиотека Полоцкого государственного университета. – Полоцк, 2016. – С. 1–183.
2. Neal Koblitz. Random Curves: Journeys of a Mathematician, – Springer, 2009. – С. 312–313. – 402 с. – ISBN 9783540740780.
3. Koblitz N.A. Course in number theory and cryptography. – USA: Springer – Verlag?1994. – 235 с.
4. Ишмухаметов Ш.Т. Методы факторизации натуральных чисел. – Казань: КФУ, 2011. – 1990 с.
5. Koblitz N.A. Course in number theory and cryptography. – USA: Springer – Verlag?1994. – 235 с.

УДК 003.26

**МОДИФИЦИРОВАННОЙ МЕТОД ШИФРОВАНИЯ ТЕСТОВЫХ ДАННЫХ  
ЭЛЛИПТИЧЕСКИМИ КРИВЫМИ С ИСПОЛЬЗОВАНИЕМ  
УНИКАЛЬНОЙ АЛФАВИТНОЙ СТРОКИ-КЛЮЧА.**

**П.Р. СИНИЦА**

*(Представлено: канд. физ.-мат. наук., доц. Д.Ф. ПАСТУХОВ)*

*В статье представлена математика эллиптических кривых. А также вывод формул сложения и удвоение эллиптические точек. Представлено описание алгоритма шифрования и дешифрования. Показана работа разработанной программы на базе математики эллиптических кривых в консоли.*

**Ключевые слова:** эллиптическая кривая, шифрование, дешифрование, консоль, абелева группа, тестирование, формула сложения, формула удвоения, ключ.

**Введение.** Эллиптическая криптография – раздел криптографии, который изучает асимметричные криптосистемы, основанные на эллиптических кривых над конечными полями. Основное преимущество эллиптической криптографии заключается в том, что на сегодняшний день неизвестно существование суб-экспоненциальных алгоритмов решения задачи дискретного логарифмирования. Роль основной криптографической операции выполняет операция скалярного умножения точки на эллиптической кривой на данное целое число, определяемое через операции сложения и удвоения точек эллиптической кривой. Последние, в свою очередь, выполняются на основе операции сложения, умножения и инвертирования в конечном поле, над которыми рассматривается кривая. Особый интерес к криптографии эллиптических кривых обусловлен теми преимуществами, которые дают ее применение в беспроводных коммуникациях – высокое быстродействие и небольшая длина ключа.

**Математика эллиптических кривых.** В криптографических методах используют эллиптические кривые над полем целых чисел с характеристикой поля  $r = 2$  либо более  $r > 3$ . В дальнейшем мы будем рассматривать поле целых чисел с характеристикой  $r > 3$ .

Криптографические кривые с характеристикой поля  $r > 3$  имеют канонический вид

$$y^2 = x^3 + ax + b, \quad (1)$$

где  $a, b$  – целочисленные коэффициенты кривой;

$p$  – простое достаточно большое число.

Как видно из формулы (1), если точка с координатами  $(x, y)$  удовлетворяет уравнению (1), то уравнению (1) удовлетворяет также и точка с  $(x, -y)$ . Под эллиптической кривой понимают геометрическое множество точек (1) дополненное бесконечно – удаленной точкой.

Следующее число, называемое дискриминантом кривой:  $\Delta = -16(4a^3 + 27b^2)$ , не должно быть равным нулю (в этом случае отсутствуют точки самопересечения и точки возврата). Если дискриминант положителен  $\Delta > 0$ , то график имеет 2 части, если  $\Delta < 0$ , то одну часть.

На множестве точек эллиптической кривой определяют группу по сложению точек эллиптической кривой (раздел математики называется алгебраической геометрией). Суммой двух точек э. к.  $P, Q$  называется третья точка  $R$ , лежащая на прямой  $PQ$  и эллиптической кривой одновременно, и обозначается как  $R = P + Q$ , т.е.  $-R + P + Q = 0$ .

Операцией группового сложения называют 3 точки э. к., удовлетворяющих уравнению

$$R' + P + Q = 0. \quad (2)$$

Откуда видно, что  $R' = -R$  ( $R', R$  – элементы взаимно обратные по групповой операции). С другой стороны, прямая параллельная координатной оси  $y$ , пересекает ровно 2 точки эллиптические кривые (зеркально симметричные относительно оси  $x$ ) и бесконечно удаленную точку (в противоположных направлениях), следовательно, взаимно обратные точки эллиптические кривые  $R', R$  – имеют координаты  $(x, y)$  и  $(x, -y)$  соответственно. Единицей по групповому сложению определяют геометрически бесконечно удаленную точку и обозначают  $0$ . Итак, для групповой операции по сложению необходимо провести секущую через точки  $P, Q$  и зеркально отобразить точку  $R, R' = -R$ .

Возможны частные случаи:

1)  $P = Q$  – секущая прямая вырождается в касательную  $R' + 2P = 0$

2)  $P + Q + 0 = 0 \Leftrightarrow P = -Q$  точки  $P, Q$  (зеркально симметричные) имеют одинаковые абсциссы.

Следующей точкой по сложению выбирают точку  $Q + 0 = Q$  (образующий элемент абелевой группы).

3)  $P + P + 0 = 0 \Leftrightarrow P = 0$  – секущая прямая одновременно является вертикальной прямой и касательной.

Криптография использует конечные циклические абелевы группы с порождающим элементом  $G$ . При этом любую точку эллиптической кривой циклической группы  $1 \leq k \leq n_0$  получают по формуле:  $P_k = (GG \dots G)$ . Порядком группы точек эллиптической кривой называется число  $n_0$ , такое что  $P_{n_0} = 0$  – нулевой элемент группы. Зная порождающий элемент группы  $G$ , можно составить таблицу всех точек эллиптической кривой, при сложении точек с порядком  $k > n_0$  все точки периодически повторяются:  $P_k = P_{k-n_0*s}$ , где  $1 \leq k - n_0 * s \leq n_0 - 1, s \in N$ . В зависимости от общей ситуации частных случаев 1), 2), 3) координаты точек эллиптической кривой вычисляются по формулам (индексы 1 и 2 соответствуют точкам  $P, Q$  соответственно):

$$\begin{cases} x = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 = k^2 - x_1 - x_2 \\ y = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x) = -y_1 + k(2x_1 - x_2 - k^2) \end{cases} \quad (3)$$

$$\begin{cases} x = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 = k^2 - 2x_1 \\ y = -y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x) = -y_1 + k(3x_1 - k^2) \end{cases} \quad (4)$$

Вывод формул (3) и (4)

Угловым коэффициентом прямой проходящей через 2 точки равен:  $k = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1}$ , где точка прямой  $(x, y)$  является скользящей по прямой. Для точек  $1(x_1, y_1), 2(x_2, y_2), (x, y)$  получим:

$$\begin{aligned} y^2 &= x^2 + ax + b, \\ y_1^2 &= x_1^2 + ax + b, \\ y_2^2 &= x_2^2 + ax + b. \end{aligned}$$

Вычтем  $(y_2 - y_1)(y_2 + y_1) = (x_2 - x_1)(x_2^2 + x_1x_2 + x_1^2) + a(x_2 - x_1)$ , откуда

$$k = \frac{y_2 - y_1}{x_2 - x_1}, k(y_2 + y_1) = (x_2^2 + x_1x_2 + x_1^2) + a, \text{ аналогично,}$$

$$k = \frac{y - y_1}{x - x_1}, k(y + y_1) = (x^2 + x_1x + x_1^2) + a, \text{ и последняя формула}$$

$$k = \frac{y - y_2}{x - x_2}, k(y + y_2) = (x^2 + x_2x + x_2^2) + a.$$

Вычтем из третьей формулы вторую, получим  $k(y_2 - y_1) = x(x_2 - x_1) + (x_2 - x_1)(x_2 + x_1)$

Откуда  $k^2 = x + x_2 + x_1 \Leftrightarrow x = k^2 - x_2 - x_1$ .

Для координаты  $y = y_1 + k(x - x_1) = y_1 + k(k^2 - 2x_1 - x_2)$ . Остается вспомнить, что для групповой операции нужно выбрать зеркальную точку:

$$(x, -y) = (k^2 - x_2 - x_1, -y_1 + k(-k^2 + 2x_1 + x_2)) \quad (5)$$

Таким образом формула (3) доказана.

В случае перехода секущей в касательную получим  $x_2 = x_1, x = k^2 - 2x_1$

Далее дифференцируем уравнение 1) по  $x$ :

$$2yy' = 3x^2 + a, \Leftrightarrow k = y' = \frac{3x^2 + a}{2y} = \frac{3x_1^2 + a}{2y_1},$$

Из формулы (5) получим  $(x, -y) = (k^2 - 2x_1, -y_1 + k(-k^2 + 3x_1)), k = \frac{3x_1 + a}{2y_1}$ . Таким образом, доказана формула (4).

Циклическую группу образуют из множества точек эллиптической кривой (уравнение (1), связанных геометрической групповой структурой (формулы (3), (4), дополняют полевой целочисленной структурой по модулю простого числа  $p$ , т.е. вместо (1) решают сравнения

$$y^2 = x^3 + ax + b \pmod{p} \quad (6)$$

В конечном итоге мы пользуемся формулами (3), (4) и (6), получая последовательно все точки эллиптической кривой циклической абелевой группы.

Как видно из формул 3) и 4) координаты точек эллиптической кривой являются рациональными числами, если первые 2 точки кривой также рациональные, т.е. геометрическая групповая операция оставляет координаты точек рациональными и дальше. Анализ формул (3) и (4) показывает, что если угловой коэффициент прямой принимает целочисленные значения, то координаты  $x$ ,  $y$  будут и дальше целыми. Таким образом, необходимо решить сравнение:

$$\begin{cases} \left(\frac{y_2 - y_1}{x_2 - x_1}\right) \equiv (y_2 - y_1) \pmod{p} * (x_2 - x_1)^{-1} \pmod{p}, (x_2 - x_1)(x_2 - x_1)^{-1} \equiv 1 \pmod{p} \\ \frac{3x_1^2 + a}{2y_1} \equiv (3x_1^2 + a) \pmod{p} * (2y_1)^{-1} \pmod{p}, (2y_1) * (2y_1)^{-1} \equiv 1 \pmod{p} \end{cases} \quad (7)$$

Краткое описание алгоритма построения последовательности точек:

- 1) Находим обратный элемент в (7) к  $2y_1$ , либо к  $x_2 - x_1$ .
- 2) Находим числа  $k_1 = (y_2 - y_1) \pmod{p} * (x_2 - x_1)^{-1} \pmod{p}$ , либо  $k_1 = (3x_1^2 + a) \pmod{p} * (2y_1)^{-1} \pmod{p}$ .
- 3) Находим числа

$$\begin{cases} x = (k_1^2 - x_1 - x_2) \pmod{p} \\ y = (-y_1 + k_1(2x_1 + x_2 - k_1^2)) \pmod{p} \end{cases} \quad (8)$$

Либо по формулам:

$$\begin{cases} x = (k_1^2 - 2x_1) \pmod{p} \\ y = (-y_1 + k_1(3x_1 - k_1^2)) \pmod{p} \end{cases} \quad (9)$$

**Описание алгоритма шифрования и дешифрования.** Формулу шифрования и дешифрования выглядит следующим образом:  $(kG, Pm + k * Pb)$  (шифрование)  $\rightarrow Pm + k * nb * G - nb * k * G = Pm$  (дешифрование), где  $nb$  – закрытый ключ абонента  $b$ , а  $Pb$  открытый ключ абонента  $b$ .

Сообщение(число) должно быть равно разности  $x$  координаты и  $y$  координаты точек эллиптической кривой. Поскольку это возможно не для всех остатков по модулю простого числа  $p$ . То мы создаем свою собственную алфавитную строку, которая одновременно является и дополнительным ключом шифрования, в котором есть все буквы латинского алфавита и все цифры, расположенные по порядку как в английском алфавите или не по порядку. Кроме того, приходится делать пробелы, чтобы заполнить их символом звездочка. Идея заключается в следующем. Нужно расположить все буквы и цифры в алфавитной строке на тех позициях (порядковый номера), для которых существует точка эллиптической кривой, разность координат которой равна номеру позиции буквы в алфавитной строке. Таким образом мы отображаем номера всех букв в алфавитной строке в точки эллиптической кривой. Пример алфавитной строки:  $*b*a***cdefghi * jkl **mnopqrs**fuvwxyz01*2**3456789**$ . Далее из другой строки-слова по символам считаются буквы и записываются в свой массив. Далее сравниваются очередной символ слова и символы алфавитной строки. Как только произошло совпадение букв, символу слова сопоставляется номер позиции в алфавитной строке расположения этого символа и соответствующая ему точка эллиптической кривой, разность координат которой равна данной позиции символа в алфавитной строке. При дешифровании по найденной точке нужно вычесть ее  $x$  и  $y$  координаты и считать в алфавитной строке символ с данной позицией и записать эту букву алфавита на выходе. Алфавитные строки определяются экспериментально, чтобы все буквы и знаки препинания, числа английского алфавита шифровались и дешифровались однозначно с помощью эллиптической криптографии.

**Тестирование программы в консоли.** Для тестирования шифрования данных введем фразу с латинским шрифтом Полоцкий государственный университет 2019: «polotsk state university 2019» с длиной



строки  $mn = 29$ . Результат ввода текста, параметры эллиптической кривой  $a = -1, b = 188, p = 751$ , открытый ключ  $(kx, ky) = (201, 5)$  приведены на рисунке 1.

На рисунке 2 мы видим результат шифрования и дешифрования. Каждому исходному символу текста соответствует четыре целые координаты двух точек эллиптической кривой, расположенных в одной строке. Для удобства ввода и построчного считывания тот же шифр в один столбец записывается в текстовый файл *balka1.txt*. Мы видим полное совпадение шифра на двух этапах.

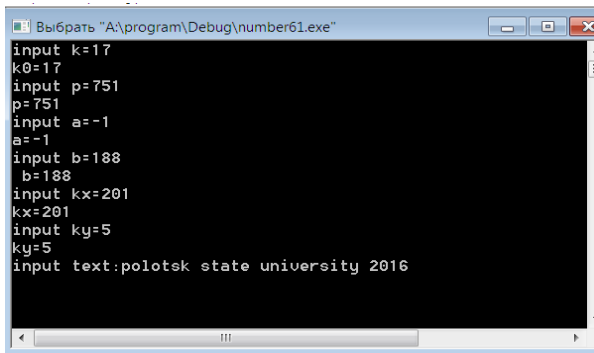


Рисунок 1. – Ввод данных в программу

Кроме того, видно также, что вводимое в программу случайное число  $k = 17$  на рисунке 1 и записанное программой в текстовый файл *balka2.txt* также совпадают.

В ходе работы программы составляется протокол шифрования по каждому шифруемому входному символу. Функция проверки  $prov(x_1, y_1, a, b, p)$  осуществляет принадлежность каждой проверяемой точки с координатами  $(x_1, y_1)$  эллиптической кривой с параметрами  $a, b, p$ . Если точка принадлежит эллиптической кривой, то программа проверки возвращает число 0, в противном случае возвращается другое целое число. Принадлежность точек данной эллиптической кривой проводится во многих частях программы и отсутствие принадлежности точки кривой сообщается в протоколе сообщений немедленно. В протоколе шифрования указываются координаты точки эллиптической кривой, соответствующей каждому входному символу и разность координат точки  $x - y$ , которая равна положению исходного символа в алфавитной строке. Например, по точке эллиптической кривой с координатами  $(xm, ym) = (680, 657)$ ,  $des = xm - ym = 680 - 657 = 23$  исходный символ, используя алфавитную строку (нумерация символов в алфавитной строке начинается с нуля, поэтому  $des = 23$  соответствует 24 символу, т.е. латинской букве *p* (рисунок 2 снизу).

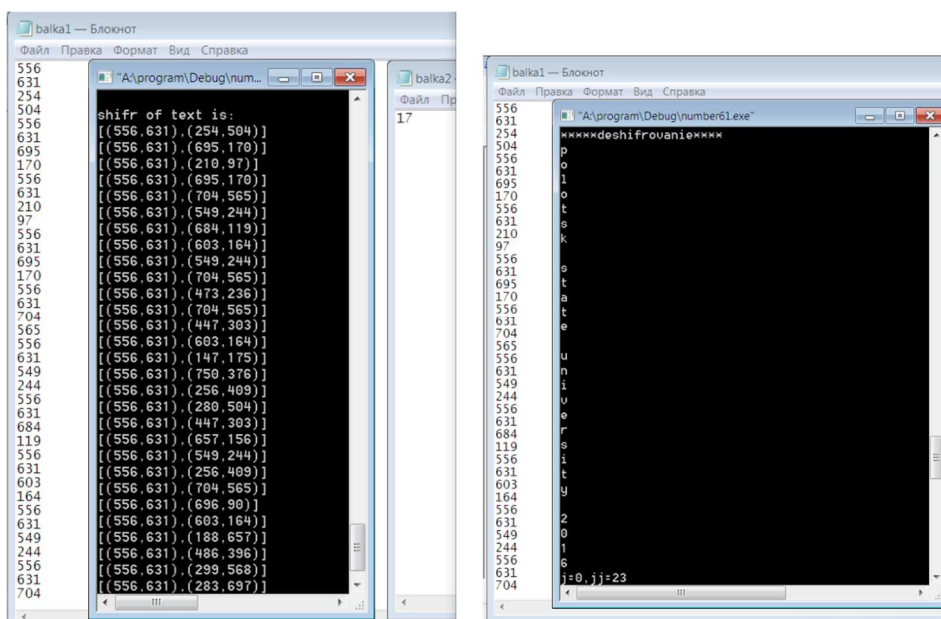


Рисунок 2. – Результат выполнения операции шифрования(слева) и дешифрования(справа)

Действительно, фраза «polotsk state university 2019» начинается с буквы  $p$ .

Второй символ  $(xm, ym) = (266, 244)$ ,  $des = xm - ym = 266 - 244 = 22$  соответствует 23 по счету символу в алфавитной строке, т.е. латинской букве  $o$ , что соответствует второй букве в слове polotsk. Из консоли видно, что шифр имеет другие координаты, чем координаты точки эллиптической кривой, разность координат которой есть позиция символа, то есть в открытом виде шифр не содержит координат точек сообщения. Так же мы видим, что все точки текста дают функцией проверки значение ноль, то есть все точки являются точками эллиптической кривой (рисунок 2 снизу).

**Заключение.** Развитие шифрования и его методов привело к их широчайшей распространенности. Сейчас для конечного пользователя не составляет труда зашифровать раздел на жестком диске или переписку и установить защищенное соединение в интернет. В связи с тем, что шифрование и другие информационные технологии проникают в наш быт, растет число компьютерных преступлений. Зашифрованная информация так или иначе представляет собой объект защиты, который, в свою очередь, должен подвергаться правовому регулированию.

Имея длинную кодовую строку ключ можно увеличить пространство ключей до такой степени, что их перебор даже при известных параметрах кривой  $a$ ,  $b$ ,  $p$  и образующего элемента группы  $G$ , становится невозможным для злоумышленника. Даже для алфавита из 36 символов – 26 букв английского алфавита и 10 цифр, минимальная кодовая строка имеет число переборов  $36! = 3.7 \times 1041$ . Пусть суперкомпьютер может анализировать шифры со скоростью миллиард шифров в секунду. При этом понадобится времени порядка  $3,7 \times 1032 \text{ с} = 10^{25}$  лет. Что реально невозможно для разгадывания кодовой строки методом перебора даже при известных параметрах эллиптической кривой и образующем элементе группы  $G$ . Основываясь на этих фактах данный алгоритм обеспечит надежное шифрование данных.

#### ЛИТЕРАТУРА

1. Березин Б.В., Дорошкевич П.В. Цифровая подпись на основе традиционной криптографии: вып. 2 – М : МП Ирбис – П, 1992 – 202 с.
2. Бутакова Н.Г., Семенов В.А., Федоров Н.В. Криптографическая защита информации: учеб. пособие для вузов. – М. : Изд-во МГИУ, 2011. – 316 с.
3. Жданов О.Н., Чалкин В.А. Эллиптические кривые: Основы теории и криптографические приложения. – М. : Книжный дом ЛИБРИКОМ, 2013. – 200 с.
4. Жданов О.Н., Золотарев В.В. Методы и средства криптографической защиты информации. – 167 с.
5. Болотов А.А., Гашков С.Б., Фролов А.Б., Часовских А.А. Элементарное введение в эллиптическую криптографию.
6. Recommended Elliptic Curves for Government Use.
7. SEC 2/ Recommended Elliptic Curves Domain Parametres.

УДК 004.223.2

**ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ДЛЯ ПРИЛОЖЕНИЯ  
«УМНАЯ КНИГА РЕЦЕПТОВ»****А.А. СКУКОВСКАЯ**  
(Представлено: Т.С. РУДЬКОВА)

В данной статье рассматриваются цели и принципы моделирования базы данных для приложения «Умная книга рецептов».

**Введение.** Приложение «Умная книга рецептов» в первую очередь должно представлять собой удобное хранилище данных, а также инструмент по составлению полезного и вкусного меню, подбору блюд исходя из физических характеристик пользователей.

Данная статья направлена на описание базы данных для разработанного приложения и методов ее проектирования.

**Основной раздел.** Разработка базы данных выполняется с помощью моделирования данных. Цель моделирования данных состоит в разработке концептуальной схемы базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных. Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD). С помощью ERD осуществляется детализация накопителей данных DFD-диаграммы, а также документируются информационные аспекты бизнес-системы, включая идентификацию объектов – сущностей, свойств этих объектов – атрибутов и их связей с другими объектами (отношений).

Сущность (Entity) – множество экземпляров реальных или абстрактных объектов (людей, событий, состояний, идей, предметов и др.), обладающих общими атрибутами или характеристиками. Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности. Каждая сущность должна обладать некоторыми свойствами:

- иметь уникальное имя;
- иметь один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь;
- иметь один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности.

Каждая сущность может обладать любым количеством связей с другими сущностями модели. Связь (Relationship) – это ассоциация между сущностями, при которой каждый экземпляр одной сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, и наоборот.

Атрибут (Attribute) представляет тип характеристики или свойств, ассоциированных с множеством реальных или абстрактных сущностей – объектов (людей, мест, событий, состояний, идей, предметов и т.д.). Экземпляр атрибута – это определенная характеристика отдельного элемента множества. Экземпляр атрибута определяется типом характеристики и ее значением, называемым значением атрибута. Экземпляр сущности должен обладать единственным определенным значением для ассоциированного атрибута [1].

В ходе анализа знаний и проектирования базы данных выявлены основные сущности:

- Сущность *Блюдо* описывает блюда, у которых может быть большое количество вариаций рецептов, характеризуется названием, типом (например, суп, салат), принадлежностью к национальной кухне какой-либо страны, типом события, к которому готовится данное блюдо (например, завтрак, Новый год).
- Сущность *Рецепт* описывает один из возможных рецептов конкретного блюда, характеризуется названием, количеством порций, калорийностью, привязкой к конкретному блюду.
- Сущность *Продукты* описывает продукты, характеризуется названием, калорийность и показателями БЖУ.
- Сущность *Ингредиенты* описывает ингредиенты рецепта, характеризуется количеством грамм и имеет привязку к рецепту и продукту.
- Сущность *Технология приготовления* описывает действия для приготовления рецепта, характеризуется описанием, продолжительностью, номером действия.
- Сущность *Пользователь* содержит в себе логины и пароли пользователей, зарегистрированных в приложении.
- Сущность *Характеристики пользователя* описывает характеристики пользователя, характеризуется параметрами пользователя (вес, рост, возраст), показателями БЖУ для пользователя, необходимым количеством калорий и датой добавления характеристики.

– Сущность *Любимые рецепты* описывает рецепты, понравившиеся пользователю, характеризуется приоритетом.

На основе выделенных основных сущностей необходимо выделить подсущности к основным сущностям:

- Для сущности Блюдо выделим подсущности *тип, тип по событию и национальная кухня*.
- Для сущности Рецепт выделим подсущность *тип рецепта*.
- Для сущности Технология приготовления выделим подсущность *действие*.

Реляционная модель представляет базу данных в виде множества взаимосвязанных отношений (таблиц), которые используются для хранения информации об объектах, представленных в базе данных.

Одно из важнейших достоинств реляционных баз данных состоит в том, что можно хранить логически сгруппированные данные в разных таблицах и задавать связи между ними, объединяя их в единую базу. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов.

В каждой таблице базы данных может существовать первичный ключ. Под первичным ключом понимают поле или набор полей, однозначно (уникально) идентифицирующих запись. Первичный ключ должен быть минимально достаточным: в нем не должно быть полей, удаление которых из первичного ключа не отразится на его уникальности.

Для организации связей между двумя и более таблицами, используются вторичные ключи. Они служат ограничителями целостности связей нескольких таблиц, т.к. подчиненная таблица не может ссылаться на несуществующие записи главной таблицы (что позволяет строить целостные модели данных).

Существует три вида связей между таблицами:

- «один-ко-многим»;
- «один-к-одному»;
- «многие-ко-многим».

Связь «один-ко-многим» – самая распространенная для реляционных баз данных. Для обеспечения ссылочной целостности в дочерней таблице создается внешний ключ, через который организовывается связь с дочерней таблицей. Обязательное условие состоит в том, что внешний ключ по составу полей должен совпадать с первичным ключом родительской таблицы [2].

Таблица. – Первичные и внешние ключи отношений

№ п/п	Название таблицы	Первичный ключ	Внешние ключи
1	Блюдо	IDБлюда	IDТипБлюда, IDНациональности, IDТипСобытия
2	ТипБлюда	IDТипБлюда	–
3	НациональнаяКухня	IDНациональности	–
4	ТипБлюдаПоСобытию	IDТипСобытия	–
5	Рецепт	IDРецепта	IDБлюда
6	Продукты	IDПродукта	–
7	Ингредиенты	IDИнгредиента	IDПродукта, IDРецепта
8	ТипРецепта	IDТипРецепта	–
9	СвязьРецептТип	IDСвязи	IDРецепта, IDТипРецепта
10	ТехнологияПриготовления	IDТехнологии	IDРецепта, IDДействия
11	Действие	IDДействия	–
12	Пользователь	IDПользователя	–
13	ХарактеристикиПользователя	IDХарактеристики	IDПользователя
14	ЛюбимыеРецепты	IDЛюбимогоРецепта	IDРецепта, IDПользователя

Поэтому, для реализации приложения, необходимо рассмотреть всю информационную систему в совокупности, определить взаимное влияние сущностей, установить все связи между сущностями и объектами (рисунок 1).

Также, важным является выбор системы управления базой данных. Для разработки представленной базы данных удобно использовать MS SQL Server 2014, т.к. данный сервер поддерживает множество типов данных необходимых при разработке и является удобным в использовании.

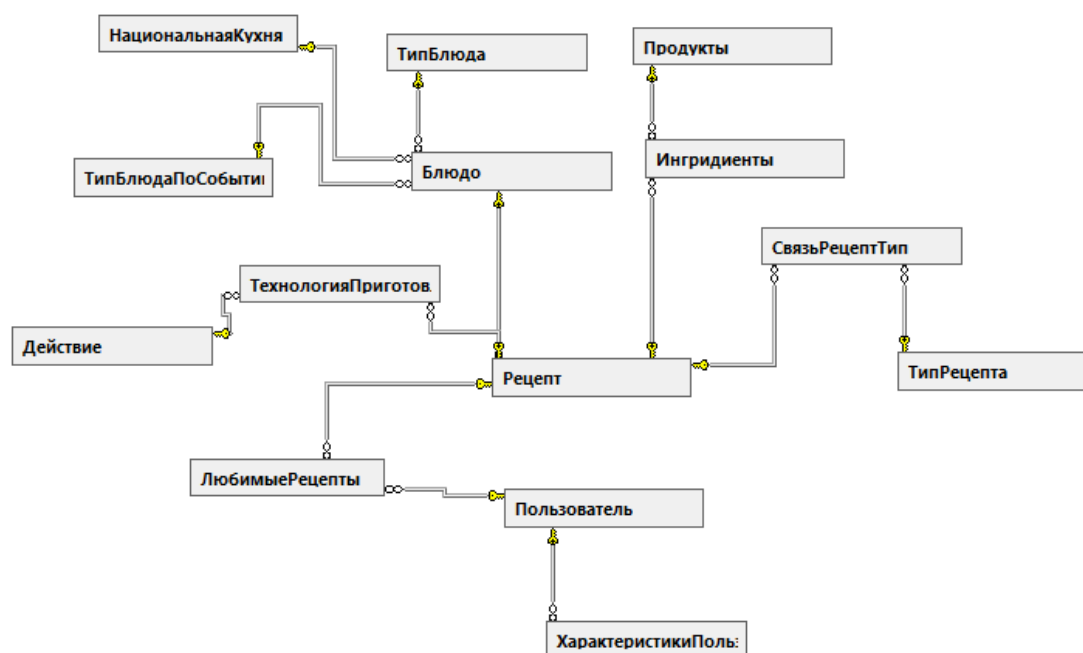


Рисунок. – Связи сущностей

**Заключение.** В данной статье были рассмотрены основные цели и принципы моделирования базы данных для приложения «Умная книга рецептов». Также были рассмотрены основные сущности спроектированной базы данных.

#### ЛИТЕРАТУРА

1. Studfiles [Электронный ресурс]. Режим доступа: <https://studfiles.net/preview/2915208/page:21/>. – Дата доступа: 18.09.2019.
2. Информационный ресурс кафедры информатики УГАТУ [Электронный ресурс]. Режим доступа: <http://informatic.ugatu.ac.ru/lib/office/Proekt.htm>. – Дата доступа: 20.09.2019.

УДК 004.223.2

## ПРИЛОЖЕНИЕ «УМНАЯ КНИГА РЕЦЕПТОВ»

**А.А. СКУКОВСКАЯ**  
(Представлено: Т.С. РУДЬКОВА)

*В статье рассматривается разработка приложения для хранения кулинарных рецептов с функцией подсчета калорий и определения оптимальных блюд. Проведен анализ технологий, использованных для разработки данного приложения. Проведены исследования по актуальности разработки данного приложения.*

Введение. В настоящее время, большое количество людей переходят с бумажных носителей информации на электронные, в связи с быстротой и удобством использования последних. Популярность использования электронных носителей обуславливается не только удобством добавления и хранения информации, а также наличием дополнительных функций для обработки большого количества данных. Использование электронных хранилищ не ограничивается профессиональным использованием, они так же популярны и в повседневной жизни, например, в качестве записных книжек.

Наряду с информационными технологиями, активно популяризируются спорт и здоровое питание, что делает актуальным внедрение информационных технологий в сферу здорового образа жизни. В настоящий момент, создается большое количество приложений для занятий спортом, подсчета калорий и составления меню, что подтверждает актуальность созданного приложения.

Основной раздел. Проектирование такого программного продукта начинается с базы данных, необходимой для хранения всей информации приложения. В соответствии с предметной областью и назначением приложения, база данных хранит сведения о пользователях, рецептах, продуктах, технологиях приготовления блюд и значениях белков, жиров и углеводов (БЖУ), как для рецептов, так и для отдельных пользователей, исходя из их физических характеристик.

В связи с этим, для разработки информационной базы данных необходимо использовать технологию хранения данных – СУДБ MS SQL 2014.

Для разработки всего приложения – среду разработки Microsoft Visual Studio 2017 и язык программирования C#, т.к. это язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции, имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML [1].

При реализации базы данных необходимо использовать MS SQL Server в связи с хорошей совместимостью с выбранным языком программирования и простотой его использования.

Подключение к базе данных осуществить с помощью Entity Framework, т.к. данный фреймворк обеспечивает быстроту и простоту подключения к базе данных и удобную работу с ней с помощью встроенных функций. Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне программист оперирует таблицами, индексами, первичными и внешними ключами, то на концептуальном уровне, который предлагает Entity Framework, он уже работает с объектами [2].

Для обеспечения функционала, а также удобства использования информационной системой необходимо разработать такое приложение, которое позволит добавлять новые рецепты, добавлять, удалять, редактировать информацию о себе, о своих любимых рецептах, отображать информацию о рецептах и подбирать оптимальные блюда для каждого пользователя приложения.

Таким образом, при создании базы данных одной из важных задач является обеспечение целостности данных. Целостность (от англ. integrity – нетронутость, неприкосновенность, сохранность, целостность) – понимается как правильность данных в любой момент времени. Но эта цель может быть достигнута лишь в определенных пределах: СУБД не может контролировать правильность каждого отдельного значения, вводимого в базу данных. Для этого существует ряд средств, помогающих разработчику минимизировать возможность нарушения целостности данных базы: триггеры, проверки («check»), уникальность («unique») и др. Триггер – это хранимая процедура специального типа, автоматически выполняющаяся при наступлении заданного события [3].

В данном приложении триггеры также необходимо использовать для автоматических подсчетов калорийности блюд и характеристик пользователей. В таблице представлены основные случаи использования ограничений и триггеров.

Таблица. – Ограничения и триггеры таблиц

№ п/п	Название таблицы	Описание
1	Ингредиенты	При добавлении нового ингредиента, пересчитывается калорийность всего блюда
		При добавлении нового ингредиента действует ограничение UNIQUE для комбинации рецепта и продукта
2	Рецепт	При удалении рецепта, удаляются все связанные с ним таблицы
3	ТехнологияПриготовления	При добавлении новой технологии, пересчитывается калорийность всего блюда в зависимости от действия
4	ХарактеристикиПользователя	При изменении веса, возраста и роста пересчитываются значения БЖУ и калорий
5	СвязьРецептТип	При добавлении нового типа рецепту, действует ограничение UNIQUE для комбинации рецепта и типа

Кроме того, для обеспечения целостности данных, также необходимо присутствие триггеров на добавление, удаление и изменение таблиц.

**Заключение.** Разработанное приложение отвечает всем требованиям предметной области, таблицы созданной базы данных отвечают требованиям нормализации, что позволяет обеспечить целостность и непротиворечивость информации. Для создания приложения были использованы наиболее подходящие и современные языки программирования и среды разработки.

#### ЛИТЕРАТУРА

1. Progopedia [Электронный ресурс]. Режим доступа: <http://progopedia.ru/language/csharp/>. – Дата доступа: 17.09.2019.
2. METANIT.COM. Сайт о программировании [Электронный ресурс]. Режим доступа: <https://metanit.com/sharp/entityframework/1.1.php>. – Дата доступа: 17.09.2019.
3. Studme.org [Электронный ресурс]. Режим доступа: <https://studme.org/93824/informatika/triggery>. – Дата доступа: 17.09.2019.

УДК 004.021

РАЗРАБОТКА СИСТЕМЫ ДЛЯ СКРЫТИЯ ИНФОРМАЦИИ  
С ИСПОЛЬЗОВАНИЕМ ШИФРА ПЕРЕСТАНОВКИ

А.И. СМОЛЯК

(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)

В данной статье рассматриваются: алгоритм скрытия данных с использованием шифра перестановки и основные функции разрабатываемого программного продукта.

**Ключевые слова:** информационные технологии, скрытие информации, криптография, шифр перестановки, C#.

**Введение.** Проблема защиты информации путем ее преобразования, исключаяющего ее прочтение посторонним лицом, волновала человеческий ум с давних времен. История криптографии – ровесница истории человеческого языка. Более того, первоначально письменность сама по себе была криптографической системой, так как в древних обществах ею владели только избранные. Священные книги Древнего Египта, Древней Индии тому примеры.

**Основной раздел.** Для разработки программного средства организации и функционирования программы необходимо выбрать среду разработки, с помощью которой будет производиться проектирование.

Для разработки этого приложения был выбран язык C#.

Основными функциями разрабатываемого программного продукта являются шифрование и дешифрование данных.

Шифр перестановки – это метод симметричного шифрования, в котором элементы исходного открытого текста меняют местами. Элементами текста могут быть отдельные символы (самый распространенный случай), пары букв, тройки букв, комбинирование этих случаев и так далее.

*Алгоритм шифра перестановки*

Исходное сообщение разбивается на блоки длины  $m$ , где  $m$  – это длина ключа.

Ключ в шифре перестановки имеет следующий вид:

1	2	3	4
2	4	1	3

Шифр перестановки: ключ

В первой строке таблицы указаны номера символов блока по порядку, а во второй строке указаны номера позиций, которые должны занимать указанные символы в зашифрованном блоке текста.

Кодирование осуществляется перестановкой букв. Таким образом первый символ из исходного блока должен быть переставлен на второе место, второй на четвертое, третий на первое, четвертый на третье.

Если данным ключом зашифровать слово кофе, то получится слово фкео.

Дешифрование производится в обратном порядке. На примере указанного ключа: второй символ из зашифрованного блока ставим на первое место, четвертый на второе, первый на третье, третий на четвертое.

При использовании любого блочного шифра (шифр перестановки не исключение), может возникнуть ситуация, когда текст не делится на равные блоки длины  $m$ . То есть остаток от деления длины текста  $n$  на длину ключа  $m$  не равен нулю.

В таких случаях длину исходного сообщения увеличивают на  $m - (n \% m)$  символов, чтобы оно делилось на равные блоки длины  $m$ .

Листинг 1 – Шифр перестановки

```

1:: class Transposition
2:: {
3::     private int[] key = null;
4::     public void SetKey(int[] _key)
5::     {
6::         key = new int[_key.Length];
7::         for (int i = 0; i < _key.Length; i++)
8::             key[i] = _key[i];
9::     }
10::    public void SetKey(string[] _key)

```



```
11:: {
12:: key = new int[_key.Length];
13:: for (int i = 0; i < _key.Length; i++)
14:: key[i] = Convert.ToInt32(_key[i]);
15:: }
16:: public void SetKey(string _key)
17:: {
18:: SetKey(_key.Split(' '));
19:: }
20:: public string Encrypt(string input)
21:: {
22:: for (int i = 0; i < input.Length % key.Length; i++)
23:: input += input[i];
24:: string result = "";
25:: for (int i = 0; i < input.Length; i += key.Length)
26:: {
27:: char[] transposition = new char[key.Length];
28:: for (int j = 0; j < key.Length; j++)
29:: transposition[key[j] - 1] = input[i + j];
30:: for (int j = 0; j < key.Length; j++)
31:: result += transposition[j];
32:: }
33:: return result;
34:: }
35:: public string Decrypt(string input)
36:: {
37:: string result = "";
38:: for (int i = 0; i < input.Length; i += key.Length)
39:: {
40:: char[] transposition = new char[key.Length];
41:: for (int j = 0; j < key.Length; j++)
42:: transposition[j] = input[i + key[j] - 1];
43:: for (int j = 0; j < key.Length; j++)
44:: result += transposition[j];
45:: }
46:: return result;
47:: }
48:: }
```

**Заключение.** В данной статье был рассмотрен алгоритм сокрытия информации с использованием шифра перестановки, а также основные функции разрабатываемого программного продукта.

#### ЛИТЕРАТУРА

1. А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. Основы криптографии. – Гелиос АРВ, 2002. – ISBN 5-85438-137-0.
2. А. В. Бабаш, Г. П. Шанкин. Криптография. – М. СОЛОН-ПРЕСС, 2007. – ISBN 5-93455-135-3.
3. Фред Б. Риксон. Коды, шифры, сигналы и тайная передача информации. – Астрель, 2011. – ISBN 978-5-17-074391-9.
4. Дориченко С. А., Яценко В. В. 25 этюдов о шифрах: Популярно о современной криптографии. – Теис, 1994. – ISBN 5-7218-0014-3.

УДК 004.021

**ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА СИСТЕМЫ  
ДЛЯ СКРЫТИЯ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ШИФРА ПЕРЕСТАНОВКИ****А.И. СМОЛЯК***(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)*

В данной статье рассматривается проектирование графического интерфейса системы для скрытия информации с использованием шифра перестановки.

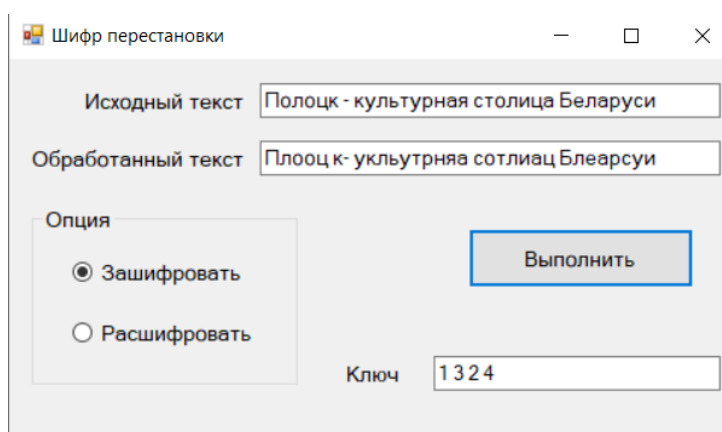
**Ключевые слова:** информационные технологии, скрытие информации, криптография, шифр перестановки, C#.

**Введение.** Интерфейс программы должен обладать целым рядом свойств: естественность, согласованность, дружелюбность, простота, гибкость, эстетическая привлекательность.

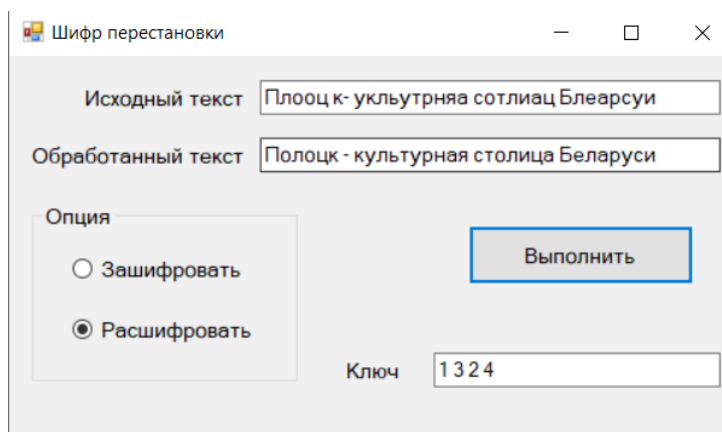
**Основной раздел.** Приложение «Шифр перестановки» представляет собой приложение для скрытия информации.

Для работы приложения необходимо ввести текст в поле Исходный текст, а также ввести ключ в поле Ключ (рисунок 1), далее нажать кнопку Выполнить.

После нажатия кнопки выполнить происходит кодирование текста, закодированный текст выводится в поле Обработанный текст.

**Рисунок 1. – Интерфейс шифрования**

Для расшифровки сообщения необходимо ввести зашифрованное сообщение в поле Исходное сообщение, выбрать радиокнопку Расшифровать (рисунок 2) и нажать кнопку Выполнить.

**Рисунок 2. – Интерфейс дешифрования**

Однако если ввести ключ, отличный от того, что был введен при шифровании, текст расшифрованного сообщения будет отличаться от зашифрованного.

**Заключение.** В данной статье рассмотрен способ построения графического интерфейса системы скрытия информации на основе шифра перестановки.

#### ЛИТЕРАТУРА

1. А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. Основы криптографии. – Гелиос АРВ, 2002. – ISBN 5-85438-137-0.
2. А. В. Бабаш, Г. П. Шанкин. Криптография. – М. СОЛОН-ПРЕСС, 2007. – ISBN 5-93455-135-3.
3. Фред Б. Риксон. Коды, шифры, сигналы и тайная передача информации. – Астрель, 2011. – ISBN 978-5-17-074391-9.
4. Дориченко С. А., Яценко В. В. 25 этюдов о шифрах: Популярно о современной криптографии. – Теис, 1994. – ISBN 5-7218-0014-3.

УДК 004.223.2

## МЕТАДАННЫЕ И ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

К.В. СТАНКЕВИЧ

*(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)*

*В статье рассматривается состав и применение метаданных в прикладном программировании на различных этапах разработки программного обеспечения. Метаданные рассмотрены, как инструмент для отделения уровня работы с данными.*

**Ключевые слова:** *информационные технологии, метаданные, информационные системы, информация, прикладное программирование.*

Самое простое для понимания определение: метаданные – это данные о данных. Чаще всего к метаданным относят информацию, описывающую структуры используемых программой данных, а также, способы их представления, методы их обработки, права доступа к ним и прочие настройки. Большинство более-менее качественных ресурсов в интернете созданы на базе некоторой системы управления содержанием сайта, которая, в свою очередь, формирует «контент» веб-страниц на основе определенной информации, извлекая ее, к примеру, из базы данных. Эта информация и является метаданными по отношению к содержимому управляемого сайта, потому как содержит перечень страниц, панелей, меню, профилей пользователей и прочих структурных элементов сайта – является данными, описывающими данные.

В настоящее время само по себе использование метаданных при разработке программ и ресурсов в интернете уже не является откровением, но то, как и на сколько широко и глубоко эти метаданные используются, существенно влияет не только на «читаемость» программного кода или html разметки, но и на возможность декомпозиции программы на отдельные функциональные компоненты, пригодные, в том числе, для повторного использования.

Многим известна трехуровневая архитектура программных продуктов, представленная уровнем работы с данными, уровнем бизнес – логики, который определяется назначением программы и третьим уровнем – уровнем пользовательского интерфейса. В этой статье рассмотрим метаданные, как инструмент для отделения уровня работы с данными от уровня бизнес – логики и, как средство «разгрузки» уровня пользовательского интерфейса от ненужных ему деталей.

**Состав и применение метаданных в прикладном программировании.** Большинство программных продуктов, выполняющих что-то осмысленное и полезное, используют для хранения своих данных реляционные базы данных – этот путь естественный и весьма хорошо «протопан». По этой причине выделить описание используемых программой таблиц в элемент метаданных кажется логичным и первым, что приходит в голову. Хранение в специальных структурах данных информации о таблицах (имя, перечень полей с указанием типа и прочих атрибутов) позволит инкапсулировать логику работы с отдельной таблицей (например, чтение, обновление, вставка и удаление записей) в специальном сервисе. Этот сервис мог бы предоставлять удобный программный интерфейс, а при инициализации получать на вход метаданные, описывающие структуру нужной вам таблицы – метаописание таблицы. Естественно, сразу приходит мысль в раздел метаданных включить метаописание соединения с базой данных. Такой подход поможет изолировать особенности работы с той или иной СУБД (Oracle, MS Access, MS SQL Server, MySQL и т.п.) от общего подхода к работе с данными.

Если вы мыслите категориями объектно-ориентированного подхода к проектированию программных продуктов, то согласитесь с аргументом, что работы с базой данных в терминах таблиц, пусть даже посредством сервисов, работающих на основе метаданных, - недостаточно для полноценного отделения уровня работы с данными от уровня бизнес-логики. Объекты, с которыми имеет дело программа, могут не описываться исключительно отдельными записями конкретной таблицы: элементы информационной модели могут быть представлены в базе данных совокупностью связанных записей, хранящихся в разных таблицах. Да и логика интерпретации данных одной и той же таблицы может отличаться в каждом конкретном случае, и определяться контекстом использования, например, текущим бизнес-сценарием или профилем пользователя.

Одним из следующих шагов в расширении структур метаданных может стать создание метаописания интерфейса таблицы, а именно, описание интерпретации каждого отдельного поля таблицы, например: простое значение, значение из справочника, ссылка на другой объект, автоматически вычисляемое значение и т.д. Метаописание интерфейса таблицы должно характеризоваться именем интерфейса, именем таблицы и набором характеристик для каждого поля, в состав которых входит описанный выше способ интерпретации, модификатор доступа «только для чтения», подпись и прочие полезные атрибуты.

Следующим шагом, логично завершающим первый этап разработки набора метаданных, является введение метаописания информационного объекта, которое, помимо уникального имени, будет иметь коллекцию ссылок на интерфейсы таблиц с указанием различных характеристик этих ссылок – компонентов объекта. Смысл такого подхода в том, что моделируемые программой объекты зачастую имеют множество функциональных аспектов, которые создаются по различным шаблонам проектирования и предоставляют информацию о своем состоянии через отдельные интерфейсы. Для каждого объекта, в таком случае, должен быть, как минимум, определен тип связи с экземплярами его компонентов, например: «один ко многим», «один к одному», «один к одному или нулю» (наличие информации необязательно) и т.д. Разработка и реализация сервиса работы с данными программы в терминах экземпляров информационных объектов – брокером информационных объектов – существенно поможет в решении поставленной выше задачи: отделении реализации уровня работы с данными от реализации уровня бизнес-логики. Уровень пользовательского интерфейса также будет упрощен, потому как User Interface будет иметь дело с экземпляром или коллекцией экземпляров информационных объектов и их брокером. Методы этого брокера будут позволять создавать, читать, удалять и сохранять изменения в информационном объекте в знакомом многим контексте транзакций. Также, проверки целостности и допустимости значений атрибутов компонентов (свойств интерфейсов) уместно частично возложить на сервисы метаданных.

**Заключение.** Расширений наборов метаданных придумано и разработано достаточно много с большими различиями в названиях, назначениях, структурах и т.д. В состав метаданных включают метаопределения запросов, метаопределения бизнес - сценариев и процедур контроля целостности данных, метаопределения профилей пользователя и другие. Большинство серьезных современных систем автоматизации построены на основе метаданных не только ради того, чтобы их код был прозрачным и разделяемым на уровни.

#### ЛИТЕРАТУРА

1. Метаданные и ваша безопасность [Электронный ресурс] – Режим доступа: <https://www.opengsm.com/blog/metadannie-i-bezopasnost/> – Дата доступа: 23.09.2019.
2. Доступные метаданные файла [Электронный ресурс] – Режим доступа: <https://suip.biz/ru/?act=mat> – Дата доступа: 23.09.2019.
3. Метаданные [Электронный ресурс] – Режим доступа: <https://ru.bmstu.wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5>. – Дата доступа: 23.09.2019.

УДК 004.223.2

## ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

К.В. СТАНКЕВИЧ

(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)

В статье рассматриваются различные виды тестирования. Проведены исследования по использованию различных видов тестирования на разных этапах разработки программного обеспечения.

**Ключевые слова:** информационные технологии, тестирование ПО, информационные системы, информация, функциональное тестирование, регрессионное тестирование, функциональное тестирование, нефункциональное тестирование.

Тестирование – ряд мероприятий, связанных с различного рода испытаниями объекта тестирования с целью установления соответствия или несоответствия его характеристик определенным требованиям и выявления дефектов. Дефектами, в свою очередь, могут быть как ошибки в работе, так и неприемлемое качество функционирования в определенных условиях эксплуатации.

Результаты тестирования - это список выявленных несоответствий и дефектов, но, как правило, без указания их причин, хотя, опытный специалист по качеству продукции в состоянии провести первичную диагностику некорректного поведения объекта и сообщить диагноз его разработчикам.

Тестирование программного продукта проводят как на этапах создания, так и на этапах технического сопровождения и дальнейшего развития.

**Тестирование на этапе создания программного продукта.** Виды тестирования на этапе создания программного продукта классифицируют по преследуемым этим процессом целям. Функциональное тестирование – проверка корректного выполнения программой заложенных в нее функций. Нефункциональное тестирование – все прочие виды испытаний, такие как тестирование производительности в различных режимах эксплуатации, тестирование эргономики пользовательского интерфейса, тестирование отказоустойчивости и т.д.

**Функциональное тестирование.** Для проведения функционального тестирования персоналом отдела технического контроля разрабатывается документ программа и методика испытаний функционала приложения (ПМИ). Документ ПМИ содержит перечень сценариев тестирования программного продукта (test cases) с подробным описанием шагов. Каждый шаг сценария тестирования характеризуется действиями пользователя (специалиста по тестированию) и ожидаемыми результатами – ответной реакцией программы на эти действия. Программа и методика испытаний обязана имитировать эксплуатацию программного продукта в реальном режиме. Это означает, что сценарий тестирования должен быть построен на основе анализа операций, которые будут выполнять будущие пользователи системы, а не быть искусственно составленной последовательностью понятных только разработчику манипуляций. Функциональное тестирование может проводиться на различных уровнях тестирования, перечень которых зависит от сложности приложения:

- компонентное (модульное) тестирование,
- интеграционное тестирование.

Каждый компонент системы может быть рассмотрен, как отдельная подсистема, и внутри этой подсистемы могут быть выделены ее компоненты. В этом случае для таких подсистем отдельно проводятся компонентное и интеграционное тестирование. Таким образом, для сложных программных продуктов получаем иерархическую структуру процесса тестирования, на каждом уровне которой объектом тестирования является определенная часть программного комплекса (рисунок).

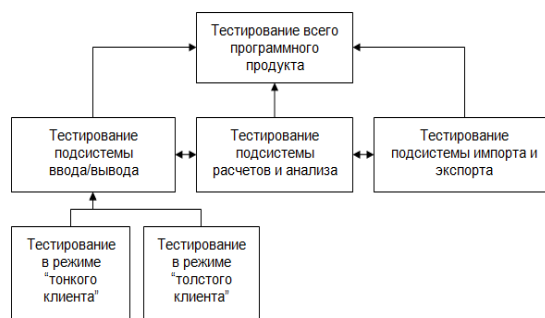


Рисунок. – Пример иерархической структуры процесса тестирования программного продукта

Выделение уровней может происходить по принципу общей функциональности (подсистема ввода/вывода данных, подсистема расчетов и аналитики, подсистема хранения данных и т.п.), по принадлежности к конкретной части проектного решения (сервер, клиент, посредник), по используемым технологиям, либо по всему сразу. В этом случае тестирование может проводиться снизу-вверх, и при каждом переходе на более высокий уровень, протестированные ранее части программы уже выступают в качестве компонентов подсистемы более высокого порядка. Бывает, что проводят тестирование сверху вниз, начиная от испытаний, связанных с общим поведением программы и заканчивая деталями. В этой ситуации может оказаться так, что на высоких уровнях абстракции реализация деталей еще отсутствует и заменяется различного рода имитаторами – функциональными заглушками, которые только со временем, при переходе на более низкие уровни заменяются реальными функциональными компонентами. Такой процесс на ранних этапах является ничем иным как тестированием прототипа программного продукта.

**Нефункциональное тестирование.** Нефункциональное тестирование, как следует из самого названия, в первую очередь оценивает второстепенные качества функционирования программного продукта. Второстепенные не в том смысле, что эти качества программного продукта менее важны, чем корректное выполнение им своих основных функций, а в том, что они никому не интересны, если продукт, в принципе, не выполняет то, что должен. Кому, например, интересны результаты замеров производительности подсистемы расчетов, если сами результаты этих расчетов не верны? Для кого имеет значение удобство пользовательского интерфейса программы, если его работа не приводит к ожидаемым от него результатам? Ну а насчет важности нефункционального тестирования можно сказать так: иногда его результаты имеют более серьезный вес в общей оценке работы программного продукта, чем результаты функционального тестирования. Например, для системы, работающей в реальном времени куда как более важно успевать взаимодействовать со всеми наблюдаемыми внешними объектами, чем корректно, с точки зрения функциональных требований, обрабатывать события от конкретного источника. Повторюсь, что это вовсе не означает, что можно некорректно взаимодействовать с отдельно-взятым объектом в угоду производительности.

**Тестирование производительности.** Достаточно интересные виды тестирования с точки зрения условий их проведения и оценки результатов. В ходе этапа тестирования производительности в первую очередь проводят нагрузочное тестирование, целью которого является проверка, будет ли система адекватно реагировать на внешние воздействия в режиме, близком к режиму реальной эксплуатации. Адекватная реакция – это время отклика в пределах, установленных требованиями к программному продукту. Внешние воздействия (нагрузка) – это, чаще всего, запросы пользователей. Ну а режим реальной эксплуатации подразумевает то, что система развернута на соответствующем требованиям аппаратном обеспечении, а самих пользователей примерно столько, сколько и планировалось. Кроме нагрузочного тестирования проводят испытания в условиях минимальных аппаратных средств и максимальной нагрузки – стрессовое тестирование, а также, испытания в условиях предельных объемов обрабатываемой информации – объемное тестирование. Выделяют еще один вид тестирования: тестирование стабильности и надежности, которое включает в себя не только длительное испытание программного продукта в нормальных условиях, но и способность его возвращаться в нормальный режим функционирования после непродолжительных периодов стрессовых нагрузок.

**Прочие виды нефункционального тестирования.** К прочим видам тестирования относят тестирование процесса установки или развертывания программного продукта, тестирование эргономики пользовательского интерфейса (не корректной его работы, а именно удобства использования), тестирование способности системы к восстановлению нормальной работы после серьезных сбоев, вызванных отказами аппаратного или системного программного обеспечения. Отдельно проводят испытания системы на различных конфигурациях, если требованиями они предусмотрены. Конфигурации могут отличаться вплоть до операционных систем серверной и клиентской частей программного комплекса. Тестирование отдельной конфигурации может сводиться к проведению для нее всего комплекса испытаний программного продукта.

#### **Тестирование на этапе сопровождения программного продукта.**

**Регрессионное тестирование.** Регрессионное тестирование проводят по результатам исправления выявленных на этапе эксплуатации программного продукта ошибок и дефектов. Также, к этому виду тестирования относят испытания программного продукта после внесения в него незначительных изменений, которые не должны влиять на общую функциональность, а вызваны такими обстоятельствами, как необходимость перехода на новую версию операционной системы или новый набор прикладных библиотек стороннего производителя. Цель регрессионного тестирования проста: доказать, что «ничего не сломалось», и что программный продукт по-прежнему соответствует всем заявленным ранее требованиям.

**Предварительное тестирование новой версии программного продукта.** Комплекс предварительных мероприятий, направленный на то, чтобы быстро показать жизнеспособность новой версии программного продукта или отправить ее на доработку, выявив в максимально короткие сроки наиболее серьезные дефекты. Ключевое слово в определении – быстро. Подобный вид тестирования еще называют дымовым тестированием по аналогии с испытаниями печниками новых печей: если дым не повалил оттуда, откуда

не положено, значит все в порядке. Если предварительное тестирование заканчивается успешно, новую версию программного продукта отправляют на более детальное «обследование», которое включает функциональное и нефункциональное тестирование.

**Заключение.** Виды и структура процесса тестирования определяются сложностью самого объекта тестирования. То же самое с теми, кто тестирует программный продукт. Ими могут быть и сами программисты (предварительное тестирование), могут быть сотрудники отдела качества компании разработчика, а могут быть и привлеченные ресурсы. Например, если программа является серийным продуктом, то все ее основные испытания заканчиваются системным тестированием, а потом уже начинаются рекламные кампании и продажи. Если программный продукт – частный заказ, то после системного тестирования проводят еще и приемочное тестирование. И системное тестирование, и приемочное тестирование направлены на комплексное испытание системы (все виды предусмотренного для нее функционального и нефункционального тестирования), но проводятся на разных площадках и с участием разного персонала. Системное тестирование проводится на стороне разработчика, а приемочное на стороне заказчика и на его же аппаратном и системном программном обеспечении. Что касается процесса тестирования на этапе сопровождения, то он может длиться намного дольше, чем сам процесс создания программного решения и прерываться только выпуском абсолютно новой версии программного продукта, после чего все начинается заново.

#### ЛИТЕРАТУРА

1. Тестирование и качество ПО [Электронный ресурс] – Режим доступа: <https://software-testing.ru/> – Дата доступа: 23.09.2019.
2. Виды Тестирования ПО [Электронный ресурс] – Режим доступа: <https://www.it-courses.by/all-software-testing-types/> – Дата доступа: 23.09.2019.



УДК 004.223.2

## ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVA ДЛЯ РАЗРАБОТКИ ГИБРИДНОЙ КРИПТОСИСТЕМЫ

**А.В. СУББОТИН**

(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)

В статье рассматриваются криптографические возможности в языке JAVA. В ходе анализа были рассмотрены основные классы и интерфейсы расширения. Проведены исследования по актуальности разработки с использованием данного расширения.

**Ключевые слова:** информационные технологии, криптографические методы, безопасность, язык программирования Java.

Для реализации криптоустойчивых приложений, написанных на языке программирования Java, создано расширение Java Cryptography Extension (JCE), которое является частью платформы Java.

Java Cryptography Architecture (JCA) – название внутреннего дизайна API криптографии в Java. JCA структурирован вокруг нескольких основных классов и интерфейсов общего назначения. Реальная функциональность этих интерфейсов обеспечивается поставщиками. Таким образом, можно использовать класс Cipher (Шифр) для шифрования и расшифровки некоторых данных, но конкретная реализация шифра (алгоритм шифрования) зависит от конкретного используемого поставщика.

**Основные классы и интерфейсы.** API криптографии Java состоит из следующих пакетов Java:

- java.security
- java.security.cert
- java.security.spec
- java.security.interfaces
- javax.crypto
- javax.crypto.spec
- javax.crypto.interfaces

Рассмотрим основные классы и интерфейсы этих пакетов.

### 1. Provider (Поставщик криптографии)

Класс Provider (java.security.Provider) является центральным классом в Java crypto API. Для того чтобы использовать Java crypto API, вам нужно установить поставщика криптографии. Java SDK поставляется с собственным поставщиком криптографии. Один из самых популярных поставщиков криптографии для Java crypto API называется Bouncy Castle.

```
import org.bouncycastle.jce.provider.BouncyCastleProvider;

import java.security.Security;

public class ProviderExample {
    public static void main(String[] args) {

        Security.addProvider(new BouncyCastleProvider());

    }
}
```

Рисунок 1. – Добавление Provider

### 2. Cipher (Шифр)

Класс Cipher (javax.crypto.Cipher) представляет криптографический алгоритм. Шифр может использоваться как для шифрования, так и для расшифровки данных. Создание экземпляра класса шифр, который использует алгоритм шифрования AES для внутреннего использования:

```
Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
```

Рисунок 2. – Инициализация шифра

### Инициализация шифра

Перед использованием экземпляра шифра его необходимо инициализировать. Экземпляр шифра инициализируется вызовом метода `init()`. Метод `init()` принимает два параметра:

- Режим – Шифрование / Расшифровка
- Ключ

Первый параметр указывает, режим работы экземпляра шифр: шифровать или расшифровывать данные. Второй параметр указывает, какой ключ они используют для шифрования или расшифровки данных.

### 3. Keys (Ключи)

Для шифрования или дешифрования данных вам нужен ключ. Существует два типа ключей в зависимости от того, какой тип алгоритма шифрования используется:

- Симметричные ключи
- Асимметричные ключи

Симметричные ключи используются для симметричных алгоритмов шифрования. Алгоритмы симметричного шифрования используют один и тот же ключ для шифрования и расшифровки. Асимметричные ключи используются для алгоритмов асимметричного шифрования. Алгоритмы асимметричного шифрования используют один ключ для шифрования, а другой для дешифрования. Алгоритмы шифрования с открытым и закрытым ключом – примеры асимметричных алгоритмов шифрования.

Каким-то образом сторона, которая должна расшифровать данные, должна знать ключ, необходимый для дешифрования данных. Если дешифрующая сторона не является стороной шифрующей данные, эти две стороны должны договориться о ключе или обменяться ключом. Это называется обменом ключами.

### 4. Подпись (Signature)

Класс `Signature` (`java.security.Signature`) используется для цифровой подписи данных. Когда данные подписаны, цифровая подпись создается из этих данных. Таким образом, подпись отделена от данных.

Цифровая подпись создается путем создания дайджеста сообщения (хеша) из данных и шифрования этого дайджеста сообщения с помощью закрытого ключа устройства, лица или организации, которая должна подписать данные. Дайджест зашифрованного сообщения называется цифровой подписью.

Чтобы подписать данные, вы должны инициализировать экземпляр подписи в режиме подписи вызывая метод `initSign(...)`, передавая закрытый ключ для подписи данных.

После инициализации экземпляра подписи, его можно использовать для подписи данных. Это делается вызовом метода `update()`, передавая данные для подписи в качестве параметра. Можно вызывать метод `update()` несколько раз, чтобы дополнить данные для создания подписи. После передачи всех данных в метод `update()`, вызывается метод `sign()` для получения цифровой подписи.

Чтобы проверить подпись, нужно инициализировать экземпляр подписи в режиме проверки путем вызова метода `initVerify(...)`, передавая в качестве параметра открытый ключ, который используется для проверки подписи. После инициализации в режиме проверки в метод `update()` передаются данные, которые подписаны подписью. Вызов метода `verify()`, возвращает значение `true` или `false` в зависимости от того, можно ли проверить подпись или нет.

```
SecureRandom secureRandom = new SecureRandom();
KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("DSA");
KeyPair keyPair = keyPairGenerator.generateKeyPair();

Signature signature = Signature.getInstance("SHA256WithDSA");

signature.initSign(keyPair.getPrivate(), secureRandom);

byte[] data = "abcdefghijklmnopqrstuvxyz".getBytes("UTF-8");
signature.update(data);

byte[] digitalSignature = signature.sign();

Signature signature2 = Signature.getInstance("SHA256WithDSA");
signature2.initVerify(keyPair.getPublic());

signature2.update(data);

boolean verified = signature2.verify(digitalSignature);
System.out.println("verified = " + verified);
```

Рисунок 3. – Процесс подписи

**Заключение.** В ходе данного исследования были сделаны выводы об актуальности используемой криптографической платформы. Возможности данного расширения позволяют спроектировать криптоустойчивую систему без установки дополнительных расширений.

#### ЛИТЕРАТУРА

1. Java Cryptography [Электронный ресурс] – Режим доступа: <http://tutorials.jenkov.com/java-cryptography/index.html> – Дата доступа: 23.09.2018.
2. Java Docs [Электронный ресурс] – Режим доступа: <https://docs.oracle.com/javase/7/docs/api/> – Дата доступа: 23.09.2018.

УДК 004.223.2

**ИСПОЛЬЗОВАНИЕ ПРОТОКОЛА АВТОРИЗАЦИИ OAuth 2.0  
В ПОЛЬЗОВАТЕЛЬСКОМ ПРИЛОЖЕНИИ****А.В. СУББОТИН***(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)*

*В статье рассматривается возможность использования в качестве протокола авторизации протокол OAuth 2.0.*

**Ключевые слова:** информационные технологии, авторизация, шифрование, аутентификация.

OAuth 2.0 – протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу.

**Принцип работы.** Как и первая версия, OAuth 2.0 основан на использовании базовых веб-технологий: HTTP-запросах, редиректах и т. п. Поэтому использование OAuth возможно на любой платформе с доступом к интернету и браузеру: на сайтах, в мобильных и desktop-приложениях, плагинах для браузеров.

Ключевое отличие от OAuth 1.0 – простота. В новой версии нет громоздких схем подписи, сокращено количество запросов, необходимых для авторизации.

Общая схема работы приложения, использующего OAuth, такова:

- получение авторизации;
- обращение к защищенным ресурсам.

Результатом авторизации является access token – некий ключ (обычно просто набор символов), предъявление которого является пропуском к защищенным ресурсам. Обращение к ним в самом простом случае происходит по HTTPS с указанием в заголовках или в качестве одного из параметров полученного access token'a.

В протоколе описано несколько вариантов авторизации, подходящих для различных ситуаций:

- авторизация для приложений, имеющих серверную часть (чаще всего, это сайты и веб-приложения);
- авторизация для полностью клиентских приложений (мобильные и desktop-приложения);
- авторизация по логину и паролю;
- восстановление предыдущей авторизации.

**Авторизация для приложений, имеющих серверную часть.** Редирект на страницу авторизации:

- На странице авторизации у пользователя запрашивается подтверждение выдачи прав.
- В случае согласия пользователя, браузер редиректится на URL, указанный при открытии страницы авторизации, с добавлением в GET-параметры специального ключа – authorization code.
- Сервер приложения выполняет POST-запрос с полученным authorization code в качестве параметра. В результате этого запроса возвращается access token (рисунок 1).

Это самый сложный вариант авторизации, но только он позволяет сервису однозначно установить приложение, обращающееся за авторизацией (это происходит при коммуникации между серверами на последнем шаге). Во всех остальных вариантах авторизация происходит полностью на клиенте и по понятным причинам возможна маскировка одного приложения под другое. Это стоит учитывать при внедрении OAuth-аутентификации в API сервисов.

**Авторизация полностью клиентских приложений.**

- Открытие встроенного браузера со страницей авторизации.
- У пользователя запрашивается подтверждение выдачи прав.
- В случае согласия пользователя, браузер редиректится на страницу-заглушку во фрагменте (после #) URL которой добавляется access token.
- Приложение перехватывает редирект и получает access token из адреса страницы (рисунок 2).

Этот вариант требует поднятия в приложении окна браузера, но не требует серверной части и дополнительного вызова сервер-сервер для обмена authorization code на access token.

**Авторизация по логину и паролю.** Авторизация по логину и паролю представляет простой POST-запрос, в результате которого возвращается access token. Такая схема не представляет из себя ничего нового, но вставлена в стандарт для общности и рекомендуется к применению только, когда другие варианты авторизации не доступны.

**Восстановление предыдущей авторизации.** Обычно, access token имеет ограниченный срок годности. Это может быть полезно, например, если он передается по открытым каналам. Чтобы не заставлять

пользователя проходить авторизацию после истечения срока действия *access token*'а, во всех перечисленных выше вариантах, в дополнение к *access token*'у может возвращаться еще *refresh token*. По нему можно получить *access token* с помощью HTTP-запроса, аналогично авторизации по логину и паролю.

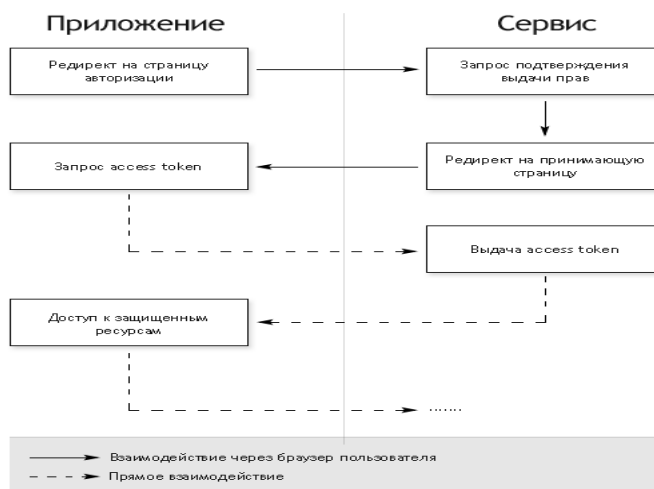


Рисунок 1. – Авторизация для приложений, имеющих серверную часть



Рисунок 2. – Авторизация полностью клиентских приложений

**Заключение.** OAuth – простой стандарт авторизации, основанный на базовых принципах интернета, что делает возможным применение авторизации практически на любой платформе. Стандарт имеет поддержку крупнейших площадок и очевидно, что его популярность будет только расти.

ЛИТЕРАТУРА

1. Материал из Википедии – свободной энциклопедии. OAuth [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/OAuth> – Дата доступа: 23.09.2018.
2. OAuth 2.0 [Электронный ресурс]. Режим доступа: <https://oauth.net/2/> – Дата доступа: 23.09.2018.

**МАШИНОСТРОЕНИЕ**

УДК 621.893

**ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОЙ КОНЦЕНТРАЦИИ НАНОАЛМАЗОВ  
В МАСЛЕ В ЗАВИСИМОСТИ ОТ РЕЖИМА РАБОТЫ ТРИБОПАРЫ****Ю.А. ГАСС, Д.В. ЛОПАТИН***(Представлено: А.А. ГУЩА)*

**Введение.** Ресурс узлов трения машин в значительной степени определяется работоспособностью используемых смазочных материалов. Современное развитие техники связано с повышением скоростей и нагрузок в узлах трения, что в свою очередь обуславливает сложность решения проблемы обеспечения заданной долговечности трибосопряжения, заключающаяся в необходимости учета взаимозависимых параметров комплекса «технология – трибоузел – эксплуатация» [1].

Для улучшения эксплуатационных свойств поверхности трения необходимо дальнейшее повышение износостойкости сопряжённых поверхностей, что требует разработки новых смазочных материалов, обладающих повышенной нагрузочной способностью и обеспечивающих более низкую интенсивность изнашивания [2].

В последнее время получают все большее распространение попытки направленного улучшения свойств масел введением в их состав различных добавок. Использование добавок в маслах, и, прежде всего, совместное введение присадок и наполнителей, позволяет наиболее гибко регулировать структуру масел. Все большее научное и практическое значение приобретают работы по использованию в качестве твердых добавок различного рода наноразмерных компонентов. Их использование дает результаты нового качественного уровня по сравнению с применяемыми ранее добавками. И одним из перспективных направлений является трибомодифицирование наноразмерными алмазосодержащими добавками [3].

**Цель работы:** изучение влияния концентрации наноразмерных компонентов на степень износа трибопары при различных режимах работ.

**Основная часть.** В качестве основного масла использовалось базовое масло И20 (минеральное масло промышленного типа с кинематической вязкостью при 40 °С – 29–35 мм<sup>2</sup>/с).

Остальные виды масел получались добавлением нанокompозитов в состав масла И20 в количестве от 0,09% до 0,9%.

Как компонент масла нанокompозиты способствуют значительному уменьшению коэффициента трения и износа трущихся поверхностей. Нанокompозиты представляют собой детонационные наноалмазы размером 4–5 нм. Их свойства в суспензии позволяют образовывать множество микроскопических трибопленок на поверхностях трения. Микропленки из нанокompозитов значительно увеличивают срок действия и полезные свойства используемых смазочных материалов.

Исследования по выявлению зависимости концентрации наноалмазов от режима работы проводились по схеме трения «палец-диск». Индентор изготовлен из стали ШХ – 15 в форме стержня Ø3 x 15 мм, материал находился в отожжённом состоянии (режим термообработки: закалка – отжиг при температуре 700–720 °С; твердость образцов 25–27 HRC). В качестве контртела выбран диск Ø70 x 6 мм из закаленной стали ШХ – 15 (твердость 57–61 HRC).

Триботехнические испытания проводились на универсальной машине трения MODEL: MMW – 1А вертикального типа с компьютерным управлением. Данная модель позволяет поддерживать силу нагрузки постоянной с отклонением ±2Н. Относительная погрешность измерения силы трения не превышала ±2% при жидкостном режиме смазывания.

Перед проведением исследования для уменьшения шероховатости поверхности и, как следствие, повышения точности получаемых данных образцы подвергались шлифовке на наждачной бумаге с зернистостью Р600.

Основным режимом работы был принят режим со следующими параметрами:

- сила нагружения: 212 Н;
- удельная нагрузка: 10 МПа;
- скорость скольжения: 0,2 м/с;
- расстояние, пройденное образцами: 3000 м.

В процессе испытаний фиксировались значения величин силы трения и коэффициента трения с частотой один раз в 1 с в режиме реального времени с возможностью сохранения в файл. Полученные данные аккумулировались в графическом и текстовом виде и после аппроксимации подвергались анализу.

Для измерения величины износа был использован метод отпечатков. Суть данного метода заключается в измерении размера отпечатка до проведения испытания и после. После этого разность размеров пересчитывается на величину абсолютного износа. Размер отпечатков снимался каждую 1000 м.

Отпечатки (рис. 1) выполнялись на микротвердомере BUEHLER Model No 1105D для определения твёрдости по Викерсу при помощи индентора пирамидальной формы с противолежащим углом  $136^\circ$ .

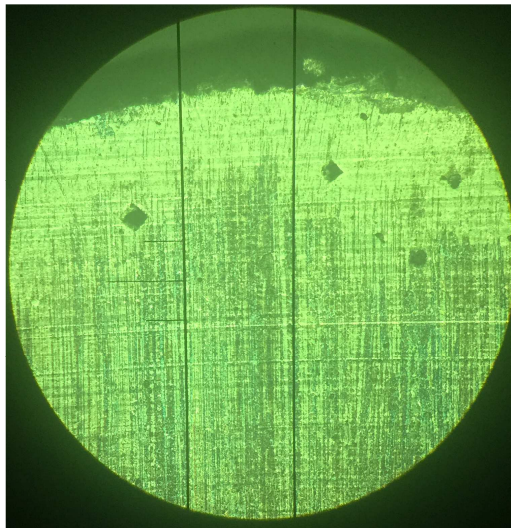


Рисунок 1. – Отпечатки на микротвердомере

Анализ полученных данных показал, что наиболее высокие смазывающие свойства при данном режиме работы проявляются у масел с содержанием наноалмазов 0,36–0,45%. При меньшей концентрации не достигается максимальный эффект, а увеличение процентного содержания не приводит к значительному снижению коэффициента трения.

В ходе испытаний также были рассмотрены следующие режимы работы:

1-ый:

- сила нагружения 636 Н;
- удельная нагрузка: 30 МПА;
- скорость скольжения: 0,5 м/с;
- расстояние, пройденное образцами: 3000 м.

2-ый:

- сила нагружения 636 Н;
- удельная нагрузка: 30 МПА;
- скорость скольжения: 0,1 м/с;
- расстояние, пройденное образцами: 3000 м.

При данных режимах работы невозможно было определить износ, т.к. после прохождения 2000 м отпечатки полностью стирались.

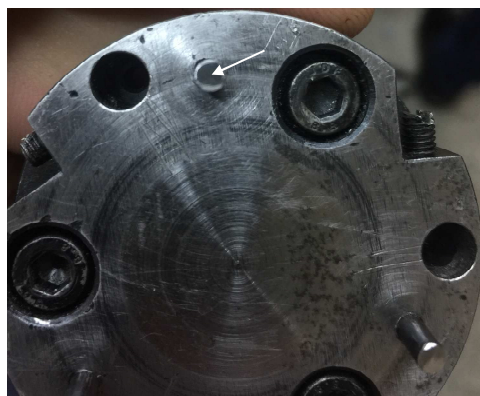


Рисунок 2. – Тёмные полосы износа на образцах

Кроме того, при основном режиме работы была выявлена зависимость характера трения от толщины масляного слоя. При недостаточной толщине на отпечатках образовывались тёмные полосы износа.

В зоне этих тёмных полос невозможно было определить размеры отпечатков, и как следствие узнать величину износа. Это, вероятно, связано с ухудшением подтекания смазочного состава в зону трибоконтакта и нарушением условий смазывания пары трения за счет вязкости смазочного материала, вследствие чего в центре контакта происходило сухое трение. Также такое явление наблюдалось при испытании масла с содержанием наноалмазов 0,09%. Причём, увеличение толщины масляного слоя не дало положительных результатов.

При основном исследуемом режиме работы диапазон оптимальной концентрации наноалмазов совпал с данными, полученными в [4] с удельным давлением 1 МПа. Это объясняется тем, что для исследования использовался ряд масел с разностью концентрации наноалмазов между последующими образцами на 0,09%. Полученные данные свидетельствуют о том, что оптимальная концентрация наноалмазов в жидкой смазке практически не зависит от режима работы трибопары.

**Заключение.** Испытание трибопары при различных режимах работы и в присутствии масла с различным содержанием наноалмазов показало, что оптимальная концентрация наноразмерных компонентов в масле практически не зависит от режима работы трибопары. Однако добавление наноалмазов в масла ограничено тем фактором, что при хранении они оседают на дне ёмкости из-за более низкой вязкости масла по сравнению с пластичными смазками, состав которых остаётся однородным на протяжении всего срока хранения.

#### ЛИТЕРАТУРА

1. Леонтьев, Л.Б. Системный анализ технологии формирования износостойких покрытий на поверхностях трения деталей / Л.Б. Леонтьев, А.Л. Леонтьев, В.Н. Макаров // *Фундаментальные исследования*. – 2014. – № 12 (часть 4). – 729–734 с.
2. Pešić, R. Methods of Tribological Improves and Testing of Piston Engines, Compressors and Pumps / R. Pešić, A. Davinić, S. Veinović // *Tribology in industry*. – 2005. – Vol. 27, № 1&2. – P. 38–47.
3. Масла и пластичные смазки на основе наноалмазов для снижения трения тяжело нагруженных узлов // Южно-Уральский инновационно-технологический центр [Электронный ресурс]. – 2009–2011. – Режим доступа: [http://suitc.ru/ru/inno/org/tech\\_profile/tech\\_offer/detail.php?ShowIBlock=21&ElementID=2629&ORG\\_ID=2627&CHILD\\_ID=2628](http://suitc.ru/ru/inno/org/tech_profile/tech_offer/detail.php?ShowIBlock=21&ElementID=2629&ORG_ID=2627&CHILD_ID=2628). – Дата доступа: 20.02.2019.
4. Дудан, А.В. Влияние концентрации наноразмерного компонента смазочного материала на свойства поверхности трения / А.В. Дудан, А.А. Гуца, Э.В. Колесов // *Инновационные технологии в машиностроении: материалы междунар. науч.-тех. конф., Новополоцк, 19-20 апр. 2018 г.* / Полоц. гос. ун-т; под. ред. чл.-корр., д-ра техн. наук, проф. В. К. Шелега; д-ра техн. наук, проф. Н. Н. Попок. – Новополоцк, 2018. – 192–194 с.



УДК 621.893

**ОПРЕДЕЛЕНИЕ ВЕЛИЧИНЫ ИЗНОСА ПОВЕРХНОСТИ ТРЕНИЯ ПРИ РАЗЛИЧНЫХ РЕЖИМАХ ЕЕ ТЕРМООБРАБОТКИ В СРЕДЕ СМАЗОЧНОГО МАТЕРИАЛА С НАНОРАЗМЕРНЫМИ АЛМАЗОСОДЕРЖАЩИМИ ДОБАВКАМИ**

*Д.В. ЛОПАТИН, Ю.А. ГАСС  
(Представлено: А.А. ГУЩА)*

*В статье представлены исследования влияния величины износа поверхности трения в среде смазочного материала с наноразмерными алмазосодержащими добавками в зависимости от ее режима термообработки.*

**Введение.** На сегодняшний день для изменения свойств материала и, как следствие, изменения характера трения применяют различные виды термообработки. Наиболее распространенными являются закалка, отпуск и отжиг [1].

*Закалка* – термическая обработка стали, сплавов, основанная на перекристаллизации стали (сплавов) при нагреве до температуры выше критической; после достаточной выдержки при критической температуре для завершения термической обработки следует быстрое охлаждение. Основная цель закалки – повышение твердости и прочности стали [2].

*Отпуск* – термическая обработка (термообработка) стали, сплавов, проводимая после закалки для уменьшения или снятия остаточных напряжений в стали и сплавах, повышающая вязкость, уменьшающая твердость и хрупкость металла.

*Отжиг* – это нагрев стали до температуры, определяемой целью отжига, выдержка при этой температуре и последующее медленное охлаждение вместе с печью. Время выдержки должно быть достаточным для нагрева изделия по всему сечению. Цель отжига – устранение химической неоднородности сталей, понижение твердости для облегчения механической обработки и др [3].

Сложность решения проблемы обеспечения заданной долговечности трибосоприятия заключается в необходимости учета взаимозависимых параметров комплекса «технология – трибузел – эксплуатация». Высокие показатели износостойкости поверхностей трения обеспечиваются за счёт применения современных смазочных материалов, однако во многих случаях этого недостаточно.

Для улучшения эксплуатационных свойств поверхности трения необходимо дальнейшее повышение износостойкости сопряжённых поверхностей. Одним из перспективных направлений является трибомодифицирование наноразмерными алмазосодержащими добавками [4].

**Цель работы:** определение степени износа поверхности трения в зависимости от применяемого вида термообработки материала в среде смазки, модифицированной добавками наноразмерных алмазных частиц.

**Методы исследований.** Для выявления зависимости степени износа от вида термообработки материала были испытаны образцы, находившиеся в закалённом, отпущенном и отожжённом состоянии. Все образцы изготовлены из стали ШХ-15 и представляют собой стержень размером Ø3 x 15 мм. В качестве контртела был выбран диск Ø70x6 мм из закаленной стали ШХ – 15.

Испытания на износ проводились на универсальной машине трения MODEL: MMW – 1A вертикального типа с компьютерным управлением. Машина поддерживает силу нагрузки в течении эксперимента постоянной с отклонением ±2 Н. Относительная погрешность измерения силы трения не превышала ±2% при жидкостном режиме смазывания.

Перед проведением исследования для повышения точности эксперимента производилась подготовка поверхности образцов. Для уменьшения шероховатости производилась притирка с использованием наждачной бумаги зернистостью Р600.

Все образцы испытывались при одинаковом режиме трения, со следующими параметрами: – сила нагружения: 212 Н; – удельная нагрузка: 10 МПа; – скорость скольжения: 0,2 м/с; – расстояние, пройденное образцами: 3000 м; – масло с содержанием наноалмазов концентрацией 0,36%.

В процессе испытаний фиксировались значения величин силы трения и коэффициента трения с частотой один раз в 1 с в режиме реального времени с возможностью сохранения в файл. Полученные данные аккумулировались в графическом и текстовом виде и после аппроксимации подвергались анализу.

Для оценки величины износа использовался метод отпечатков. Отпечатки наносились на образцы на микротвердомере BUEHLER Model No 1105D, они имеют пирамидальную форму, оставляемую индентором для определения твердости по Викерсу с противолежащим углом 136°. В сечении отпечатки представляют собой ромб, диагонали которого измерялись до проведения эксперимента и после. После этого разность размеров пересчитывается в величину абсолютного износа. Размер отпечатков определялся после каждой 1000 м.

**Результаты и обсуждение.** После прохождения 3000 м на образцах наблюдались темные полосы. Это, вероятно, вызвано ухудшением подтекания масла в зону трибоконтакта и нарушением условий смазывания пары трения за счет вязкости смазочного материала, вследствие чего в центре контакта происходило сухое трение. Причём, у закаленных образцов темные полосы были узкими (рис. 1), в то время как на отожжённых ширина полос доходила до половины диаметра стержня, т.е. наблюдается обратная зависимость между твердостью образцов и шириной темных полос.

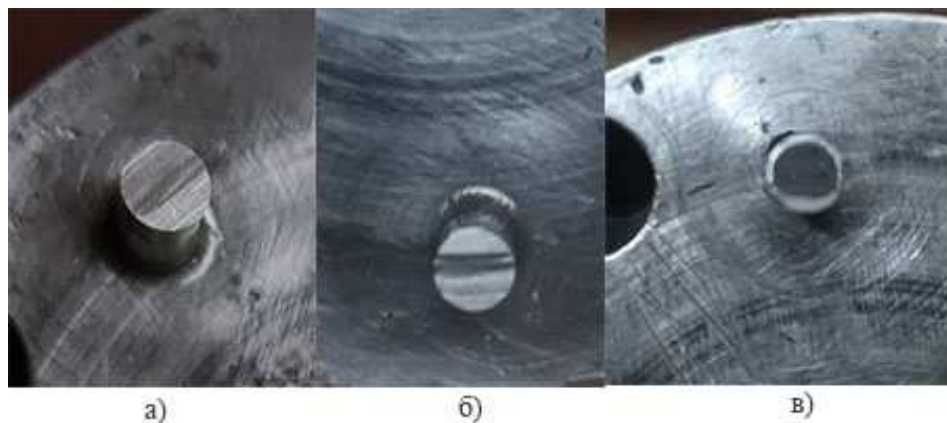


Рисунок 1. – Тёмные полосы на образцах: а – закалка, б – отпуск, в – отжиг

Данные полученные в результате исследований представлены на рисунке 2.

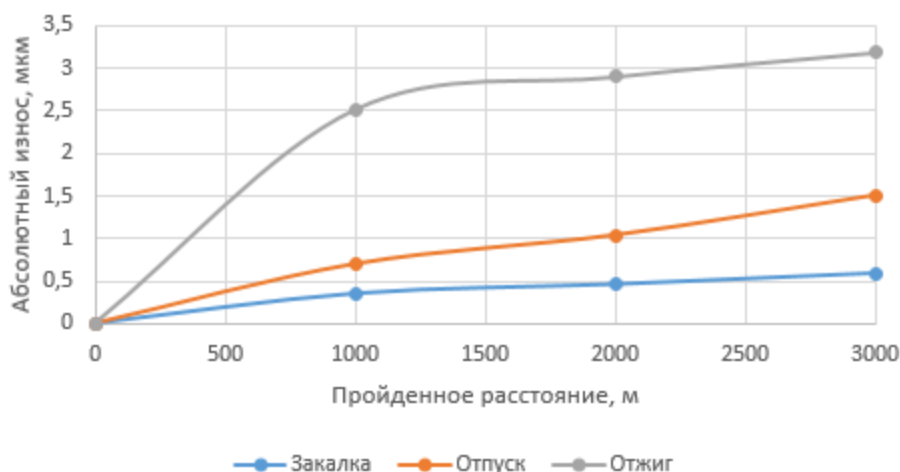


Рисунок 2. – График зависимости износа поверхности трения от режима ее термообработки

Из представленных данных можно видеть, что в случае испытаний отожженной стали в среде смазки, модифицированной наноразмерными добавками, регистрируется ускоренная приработка трибосопряжения. Это обусловлено присутствием в смазочном материале наноразмерных алмазно-графитовых частиц, обеспечивающих формирование в поверхностном слое элемента пары трения ультрадисперсной субзерненной структуры, обладающей повышенной стойкостью к зарождению и распространению трещин при фрикционном взаимодействии.

В случае триботехнических испытаний образцов отпущенной стали ШХ 15 с твердостью *HRC* 50 величина абсолютного износа стали снижается по сравнению с испытаниями образцов отожженной стали. На начальных стадиях испытаний регистрируется интенсификация процесса приработки трибосопряжения, и, связанный с этим относительно пониженный износ отпущенного образца. Интенсивность изнашивания образцов отпущенной стали, определенная на стадиях установившегося износа, примерно в 2 раза ниже, чем интенсивность изнашивания отожженной стали.

Сопоставляя данные по интенсивности изнашивания закаленной стали ШХ 15 с аналогичными данными для стали в отпущенном состоянии можно отметить, что присутствие в смазочном материале

наноразмерных добавок приводит к интенсификации процесса приработки трибосопряжений, увеличению микротвердости поверхностных слоев стали и снижению интенсивности изнашивания в случае твердости стали  $HV \leq 5000$  МПа. Для закаленной стали ШХ 15 ( $HV = 7900$  МПа) вследствие ее низкой трещиностойкости характерно снижение износостойкости при трении в среде смазки, модифицированной добавками наноразмерных алмазных частиц.

**Заключение.** Испытание образцов при трении в среде смазки, модифицированной добавками наноразмерных алмазных частиц с различными режимами термообработки показало, что твердые образцы меньше подвержены износу, о чем свидетельствует кривая износа закаленной стали в сравнении с отожженной и отпущенной. Вид термообработки также влияет на характер трения: чем более пластичный материал, тем хуже подтекание смазочного материала в зону контакта.

#### ЛИТЕРАТУРА

1. Леонтьев, Л.Б. Системный анализ технологии формирования износостойких покрытий на поверхностях трения деталей / Л.Б. Леонтьев, А.Л. Леонтьев, В.Н. Макаров // *Фундаментальные исследования*. – 2014. – № 12 (часть 4). – 729–734 с.
2. Новиков, И. И. Теория термической обработки металлов: учебник / И. И. Новиков. – 3-е изд. – М. : *Металлургия*, 1978. – 392 с.
3. Глухова, А. Термическая обработка стали / А. Глухова // *Станки для обработки металла и дерева. Технологии. Инструмент и расходные материалы*. [Электронный ресурс]. – Режим доступа: <https://stankiexpert.ru/spravochnik/materialovedenie/termicheskaya-obrabotka-stali.html>. – Дата доступа: 25.09.2019.
4. Pešić, R. Methods of Tribological Improves and Testing of Piston Engines, Compressors and Pumps / R. Pešić, A. Davinić, S. Veinović // *Tribology in industry*. – 2005. – Vol. 27, № 1&2. – P. 38–47.

УДК 621.9.04

## АНАЛИЗ УСЛОВИЙ ЭКСПЛУАТАЦИИ УДАРНОЙ ГОЛОВКИ ГОМОГЕНИЗАТОРА FBF 075 И ВОЗМОЖНЫХ ПУТЕЙ ЕЕ ИЗГОТОВЛЕНИЯ С УПРОЧНЕНИЕМ

А.А. ПОТАПОВ

(Представлено: канд. техн. наук, доц. О.П. ШТЕМПЕЛЬ)

*В исследовании описывается принцип действия гомогенизирующей ступени и конструктивные особенности её деталей, благодаря которым возникает диспергирование продукта. Раскрыта сущность работы ударной головки и её износа. Проведены исследования химического состава, микроструктуры, а также микрометрический анализ материала наконечника ударной головки гомогенизатора FBF 075.*

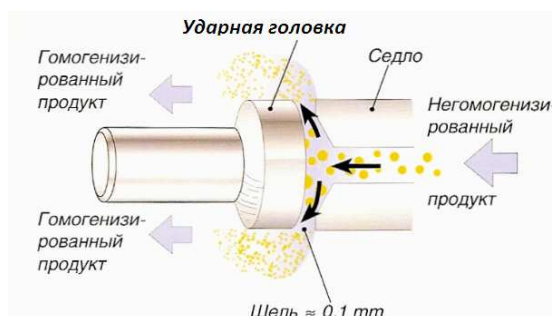
Введение. В настоящее время молочная промышленность превратилась в высокоразвитую отрасль сельского хозяйства и является одной из ведущих в пищевой промышленности страны. Предприятия молочной промышленности располагают современным, высокопроизводительным оборудованием, в том числе поточно-механизированными и автоматизированными линиями. Научно-технический прогресс в молочной промышленности способствует внедрению новых способов обработки и переработки молока на основе применения прогрессивного, наиболее высокопроизводительного оборудования. [1, 2]

Одним из основных видов оборудования в поточно-механизированных линиях по переработке молока является гомогенизатор FBF 075. Гомогенизатор предназначен для дробления жира в молоке, смесях мороженого и сливках, при этом жир не отстаивается, изменяется лишь некоторые физические свойства продуктов (повышается вязкость) и улучшается вкус продукта [1]. Главной и в тоже время быстро изнашивающейся деталью гомогенизатора, благодаря которой достигается дисперсность - является ударная головка (рис. 1).



**Рисунок 1. – Внешний вид ударной головки**

Принцип действия гомогенизирующей ступени (рис. 2) основан на том, что гомогенизируемая жидкость нагнетается под большим давлением в канал, которое составляет 15–20 МПа, периодически поднимает прижатую ударную головку и с большой скоростью движется через узкую щель между головкой и седлом. Высота клапанной щели при работе гомогенизатора не превышает 0,1 мм, а скорость молока при движении его в щели обычно достигает 150–200 м/сек. При этом молоко подвергается в зоне клапана сильному механическому воздействию, которое и приводит к раздроблению жировых шариков, т.е. к гомогенизации [1].



**Рисунок 2. – Схема гомогенизирующей ступени**

Ударная головка и седло непосредственно обеспечивают дробление и равномерное распределение в молочном продукте жировых шариков. По своей природе многие компоненты молочных продуктов и моющих растворов являются поверхностно-активными веществами (ПАВ) которые, как известно из фундаментальных исследований, интенсифицируют процессы пластического деформирования и изнашивания металлов. Каждая из рабочих поверхностей этих деталей имеет свои особенности изнашивания. Они изнашиваются от перемещающегося по ним продукта, поэтому на поверхностях формируются раковины, которые ориентированы вдоль направления перемещения продукта и могут быть отнесены к химико-механическому изнашиванию, которое соответствует окислительному изнашиванию, протекающему при высоких давлениях и наличии ПАВ. Наиболее интенсивному изнашиванию подвержена запорная фаска ударной головки, на которой четко просматривается деформация материала за пределы фаски. Более того, процессы пластического деформирования настолько разрушительны, что приводят к отслоению материала. Последнее обуславливает нарушение режима гомогенизации (рис. 3) [3].

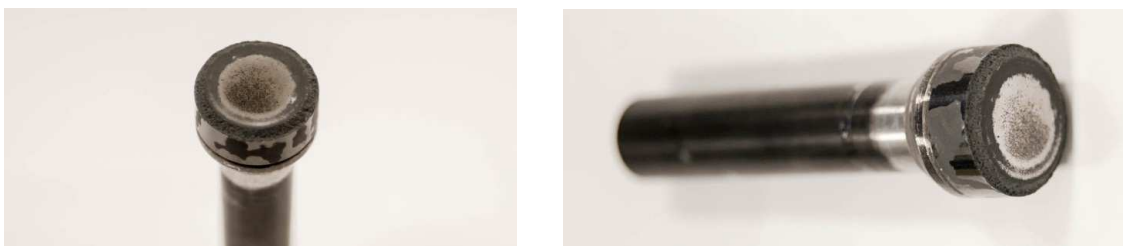


Рисунок 3. – Внешний вид изношенной ударной головки, бывшей в эксплуатации

Изнашивание запорной фаски ударной головки происходит главным образом из-за явления под названием кавитация. Обычно это явление происходит в тех случаях, когда абсолютное давление в какой-нибудь области жидкости упадет ниже некоторого критического значения давления, при котором возникают разрывы сплошности жидкости и образуются каверны, заполненные парами или газами. Часто давление совпадает с давлением насыщенных паров жидкости при данной температуре, и возникновение кавитации может трактоваться как явление вскипания жидкости, обусловленное повышением температуры или понижением давления. После вступает в силу эрозия- это механическое разрушение поверхности материала, вызываемое гидравлическими ударами, возникающими при быстром сжатии, а за тем взрывом кавитационных каверн, именно они вызывают в материале разрушения, причиной которых, возможно, являются усталостные процессы. [4, 5]

В работе исследовалась изношенная ударная головка гомогенизатора FBF 075 были выполнены:

- исследование особенностей конструкции ударной головки,
- определен химический состав материалов ее различных частей,
- исследование микростроения материала,
- микродюрOMETрический анализ материала.



Анализ конструкции ударной головки гомогенизатора FBF 075 после продольно-поперечного разреза видим, что ударная головка состоит из двух частей наконечника и корпуса (рис. 4).



Рисунок 4. – Разрез изношенной ударной головки, бывшей в эксплуатации:  
а – наконечник; б – корпус

Химический состав частей ударной головки гомогенизатора FBF 075 определяли методом спектрального анализа, который был проведен при помощи спектрометров двух марок: Pmi-masterplus и ThermoNitonXL2. Общий вид и краткое описание приборов приведено в таблице 1.

Таблица 1. – Общий вид и краткое описание спектрометров

Общий вид	Краткое описание
<p>Thermo Scientific Niton XL2</p> 	Рентгенофлуоресцентные спектрометры серии XL2 фирмы Thermo-Niton используются для подсчета неразрушающим способом показателей массы химических элементов в твердых, жидких, порошкообразных материалах и веществах. РФА спектрометры используются в горнодобывающей, металлургической, геологической и химической отраслях промышленности, связанных с добычей и переработкой металла, лома и горных пород.
<p>PMI-master plus</p> 	Для измерения массовой доли химических элементов в металлах и сплавах и применяются в аналитических лабораториях промышленных предприятий, в цехах для быстрой сортировки и идентификации металлов и сплавов и для анализа больших конструкций без разрушения.

Результаты спектрального анализа представлены в таблице 2.

Таблица 2. – Химический состав сплавов ударной головки гомогенизатора FBF 075

Часть ударной головки	Химический элемент	Содержание, %
<p>Наконечник</p> 	Cr	6,52
	Mo	0,365
	Fe	2,22
	Ni	3,49
	Ta	0,535
	W	86,57
<p>Корпус</p> 	Fe	62,0
	Nn	0,982
	Cr	23,1
	Ni	12,2
	Cu	0,400
	Ti	0,0316
	Si	0,324

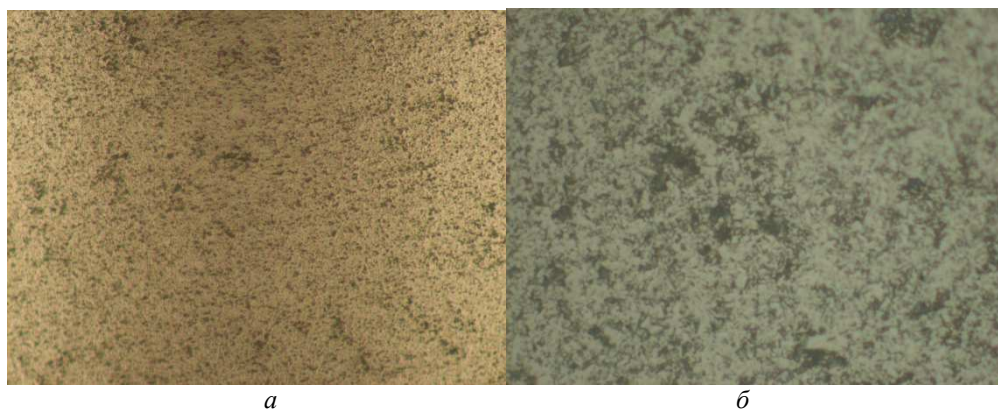
Анализ полученных данных по химическому составу (таб. 2.) свидетельствуют, что наконечник ударной головки изготовлен из твердого сплава на основе вольфрама, а корпус из хромоникелевой коррозионностойкой стали.

Анализ микроструктуры материала наконечника ударной головки проводили на инвертированном металлографическом микроскопе Nikon "Epihot200" (рис. 5).



**Рисунок 5. – Инвертированный металлографический микроскоп Nikon "Epihot200"**

Анализ микроструктуры позволил установить, что исследованный материал имеет характерное для сплавов на основе вольфрама полученных методами порошковой металлургии строение (Рис. 6). Подобные сплавы применяются для обеспечения высоких механических свойств для ответственных деталей работающих в тяжелых условиях [6].



**Рисунок 6. – Микроструктура материала наконечника:  
а – при кратном увеличении x200; б – при кратном увеличении x1000**

Измерения микротвердости материала наконечника ударной головки на основе вольфрама проводили по стандартной методике на полуавтоматическом микротвердомере BUEHLER Model No 1105D (рис. 7).



**Рисунок 7. – Твердомер BUEHLER Model No 1105D**

Твердость измеряли по Виккерсу (HV) путем вдавливания правильной четырёхгранной алмазной пирамиды с углом  $136^\circ$  между противоположными гранями. Среднее значение твердости получили 1100 (HV), что подтверждает предположение об применении в данном случае сплава на основе вольфрама полученного методами порошковой металлургии.

По известным данным в условиях кавитационного износа можно применить и другие кавитационно-стойкие материалы такие как: мартенситные нержавеющие стали, алюминиевая бронза, сплавы карбида титана, карбида вольфрама, титановольфрамовые твердые сплавы, монель-металл, мартенситно-старяющиеся стали с низким содержанием углерода и т.д. [7], а для повышения прочностных характеристик поверхностного слоя используют следующие виды упрочнения, такие как ХТО, а точнее борирование, диффузионное хромирование, азотирование, цементация и нитроцементация. Кроме ХТО возможно использование обработки высококонцентрированными потоками энергии, такими как обработка лазером, плазмой, электроискровым и электронно-лучевым упрочнением [8].

В результате проведенной работы установлен механизм изнашивания ударной головки гомогенизатора FBF075. Установлено, что оригинальная импортная, дорогостоящая головка состоит из 2-х частей корпуса и наконечника. Корпус изготовлен из высоколегированной коррозионностойкой хромоникелевой стали. Наконечник являющейся более ответственной частью детали обеспечивающей её работоспособность, изготавливается из сплава на основе вольфрама легированного хромом, никелем, железом и другими химическими элементами, методом порошковой металлургии. Сделано обоснованное предположение о возможности изготовления ударной головки на предприятиях Р.Б. с применением современных материалов и способов их упрочнения.

#### ЛИТЕРАТУРА

1. Сурков В.Д. Липатов Н.Н., Барановский Н.В. Технологическое оборудование предприятий молочной промышленности – 2-е издание, М.: Легкая и пищевая промышленность. 1970. – 547 с.: ил.
2. Крусь Г.Н. Тиняков В.Г. Фофанов Ю.Ф. Технология молока и оборудование предприятий молочной промышленности. М.: Агропромиздат. 1986. – 280 с.
3. Удовкин А.И., Назаров И.В., Толстоухова Т.Н. Гомогенизаторы для молока и молочных продуктов: монография / – Волгоград: АЧИИ ФГБОУ ВО Донской ГАУ, 2016. – 187 с.
4. Арзуматов Э.С. Кавитация в местных гидравлических сопротивлениях. – М.: Энергия, 1978. – 304 с.: ил
5. Пылаев Н.И., Эдель Ю.У. Кавитация в гидротурбинах. Л., “Машиностроение” (Ленинградское отделение), 1974. – 256 с.
6. Чувильдеев В.Н., Нохрин А.В., Москвичева А.В., и др. Описание к патенту № RU 2442834 С2. Способ улучшения механических свойств порошковых изделий из тяжелых сплавов на основе вольфрама и порошковое изделие с механическими свойствами, улучшенными этим способом. 2009 г. 9 с.: ил.
7. Пирсол И., Кавитация. Пер. с англ., Журавлева Ю.Ф. Ред., предисл. И дополн. Эпштейна. – М.: Мир, 1975. – 95 с.: ил.
8. Белевитин, В.А. Упрочнение и восстановление деталей машин: справочное пособие / В.А. Белевитин, А.В. Суворов. – Челябинск: Изд-во Челябинск. Гос. пед. ун-та, 2015. – 263 с.: ил.



УДК 621.9.04

**ИННОВАЦИОННЫЙ ПОДХОД К ВЫБОРУ МЕТОДОВ УПРОЧНЕНИЯ  
УДАРНЫХ ГОЛОВОК ГОМОГЕНИЗАТОРА FBF 075****А.А. ПОТАПОВ***(Представлено: канд. техн. наук, доц. О.П. ШТЕМПЕЛЬ)*

*В статье описывается принцип работы ударной головки гомогенизатора FBF 075, теории объясняющие природу кавитационной эрозии, и её способы уменьшения воздействия на деталь. Рассмотрены материалы и нанесение защитных покрытий методами химико-термической обработки, а также способы упрочнения высококонцентрированными потоками энергии повышающие стойкость к эрозии.*

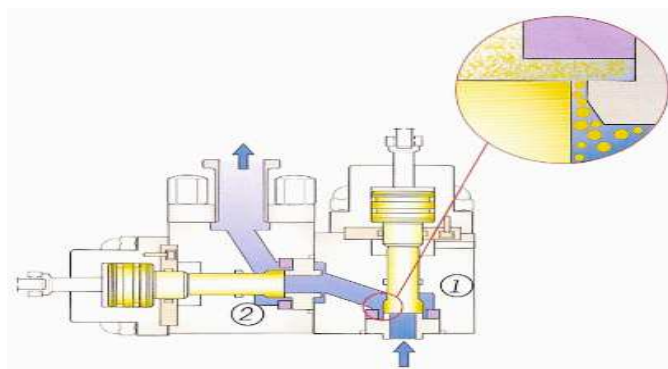
**Введение.** Молочная промышленность – одно из важнейших направлений сельского хозяйства. Чтобы молочный продукт сохранял полезные качества, необходимо обеспечивать технологию его правильной обработки и переработки. В которую входит не только охлаждение и транспортировка на молочный завод, но и определение кислотности и жирности, контроль качества, санитарные пробы.

На (рис. 1), представлен гомогенизатор - это аппарат, служащий для дробления шариков жира на более мелкую фракцию. В процессе гомогенизации достигается равномерное распределение жира, однородность продукта и повышенная вязкость. Продукт приобретает стойкость к отстаиванию сливок и развитию процессов окисления. Основными характеристиками процесса гомогенизации являются давление в процессе работы и производительность [1].



**Рисунок 1. – Общий вид гомогенизатора FBF 075**

Основной процесс гомогенизатора является диспергирование, оно достигается благодаря гомогенизирующим ступеням. Гомогенизирующая первая и вторая ступени (рис. 2) представляет собой две ударные (гомогенизирующие) головки аналогичной конструкции, связанные каналом, позволяющим продукту переходить последовательно от первой ступени ко второй.



**Рисунок 2. – Схема гомогенизирующей головки:  
1 – Первая гомогенизирующая ступень; 2 – Вторая гомогенизирующая ступень**

**Анализ работы гомогенизатора и ударной головки.** Основными рабочими органами гомогенизирующей головки являются седло и клапан (рис. 3), от конструкции, которых в известной мере зависит степень дисперсности молока при гомогенизации. Разнообразие конструктивных исполнений гомогенизирующих устройств обусловлено стремлением повысить гомогенизирующий эффект за счет повышения турбулентности потока гомогенизируемой жидкости, усиления явлений кавитации, повышения скорости движения жидкости на входе в клапанную щель [1].

При прохождении потока молока по узкому каналу его скорость возрастает (рис. 2 и рис. 3). Скорость будет расти до тех пор, пока статическое давление не снизится до такого уровня, при котором жидкость закипает. Максимальная скорость главным образом зависит от давления на входе. Когда жидкость покидает щель, скорость снижается, а давление начинает расти. Кипение жидкости прекращается, и паровые пузырьки взрываются – такой феномен называется кавитация.

Из-за кавитации, а так же высокой скорости потока, жидкость оказывает сильное механическое действие на седло и клапан (рис. 3.), что вызывает быстрый их износ. Клапан и седло изготавливают из стали и сплавов высокой твердости. [2, 3]

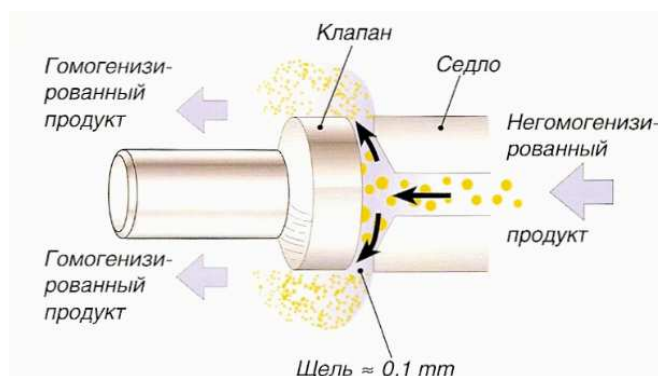


Рисунок 3. – Гомогенизирующая ступень в детальном виде

Гомогенизатор представляет собой поршневой (плунжерный) насос, продавливающий продукт под высоким давлением сквозь регулируемый зазор. Основными показателями работы гомогенизаторов являются универсальная рабочая и кавитационная характеристики. Универсальная характеристика гомогенизатора представляет зависимость между его производительностью, затрачиваемой мощностью и КПД. Она дает представление об уровне совершенства конструкции гомогенизатора и его техническом состоянии. На (рис. 4) представлено устройство гомогенизатора [2].

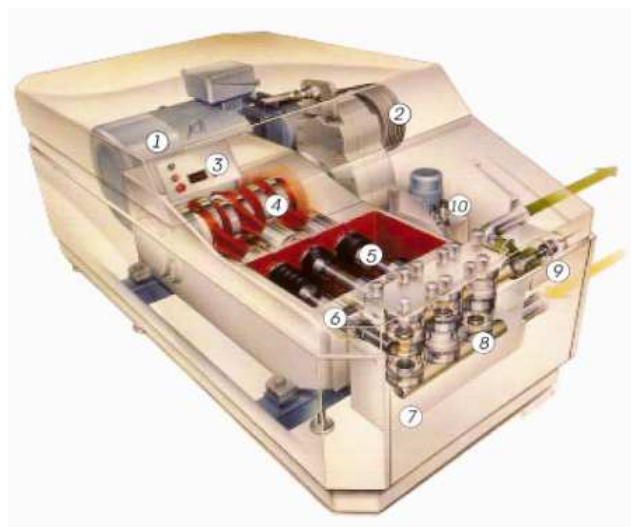


Рисунок 4. – Устройство гомогенизатора:

1 – Главный двигатель привода; 2 – Клиноремная передача; 3 – Указатель давления; 4 – Кривошипно-шатунный механизм; 5 – Поршень; 6 – Уплотнение поршня; 7 – Литой насосный блок из нержавеющей стали; 8 – Клапаны; 9 – Гомогенизирующая головка; 10 – Гидравлическая система

**Основные виды кавитационной эрозии.** С проблемами кавитации сталкиваются при рассмотрении широкого круга вопросов, связанных с течениями жидкостей, от исследования тока крови в сосудах до проектирования турбин и корабельных винтов. Ее возникновение зависит от физических свойств жидкости и параметров течения (давление, температура, скорость). В технике кавитация приводит к значительному снижению эффективности работы машин и к эрозии их деталей, а кавитация в крови может вызвать заболевание сердца и артерий. Кавитация уменьшает подъемную силу подводных крыльев, ухудшает характеристики насосов, турбин и других механизмов. Чтобы по возможности свести к минимуму вредное действие кавитации, обычно для каждой машины определяют ее кавитационные характеристики. Проектирование многих машин и установок осуществляется обязательно с учетом их кавитационной эрозии. Основным методом борьбы с эрозией состоит в соответствующем подборе материалов при изготовлении машин и механизмов; этот подбор производится путем сравнительных испытаний.

В действительности давление, при котором начинается кавитация, существенно зависит от физического состояния жидкости. Если жидкость содержит большое количество растворенного воздуха, то уменьшение давления приводит к выделению воздуха из жидкости и образованию газовых полостей (каверн), в которых давление выше, чем давление насыщенных паров жидкости. При наличии в жидкости микроскопических, не видимых глазом пузырьков кавитация может возникать при давлениях, превышающих давление насыщенного пара. Каждый кавитационный пузырек, формируясь из ядра, растет до конечных размеров, после чего он схлопывается. Весь процесс происходит в течение нескольких миллисекунд. Пузырьки могут появляться друг за другом настолько быстро, что кажутся одной каверной. Схлопывание пузырька происходит в течение миллисекунд или даже микросекунд. Доказано, что возникающие ударные волны могут привести к высоким перепадам давления (до 400 МПа.) в окружающей пузырька жидкости.

Основной проблемой в кавитации является эрозия. Высокие быстро меняющиеся давления и тепловые ударные волны, вызывают в материале вблизи сжимающегося пузырька разрушения, причиной которых, возможно, являются усталостные процессы. Насосы, турбины, винты, а также такие элементы, как клапаны и вентили, нередко подвергаются существенной эрозии. Процесс эрозии материала может происходить настолько быстро, что винты корабля после одного рейса, а рабочее колесо насоса после нескольких недель работы выходят из строя. Поэтому довольно часто, через регулярные промежутки времени, приходится заваривать эрозионные раковины, созданные кавитацией на лопатках турбин [4].

Существуют следующие наиболее распространенные теории, объясняющие природу кавитационной эрозии.

1. Гидромеханические теории, из которых следует, что эрозия материала происходит за счет возникновения больших давлений при разрушениях газовых пузырьков, при этом разрушение материала происходит за счет гидравлических ударов жидкости о поверхность омываемого кавитационным потоком тела.

2. Химическая теория объясняет эрозию материала как следствие химического воздействия агрессивных реагентов активизирующихся в кавитационном потоке, а механическое воздействие его лишь как создающее повышенную температуру и другие благоприятные условия для протекания химических процессов.

3. По электрической теории эрозия материала происходит вследствие электрических разрядов (Манометрический эффект) возникающих при разрушении газовых пузырьков.

4. Электрохимическая теория утверждает, что эрозия в металле происходит вследствие ионизации газов в полости каверны и активного в этих условиях электрического и химического воздействия на поверхности находящихся в потоке деталей. При этом преимущественное влияние может иметь как электрическое, так и химическое воздействие в зависимости от параметров и рода кавитационных потоков физических свойств материалов и др.

5. Термоэлектрические теории считают, что эрозия материала происходит вследствие термического и электрического воздействия [5].

**Способы уменьшения кавитационной эрозии.** Существуют различные способы уменьшения кавитационной эрозии, например такие как:

1. Изменение гидравлических режимов работы местного сопротивления;
2. Выбор соответствующего типа и геометрических характеристик местного сопротивления, обеспечивающих бескавитационную работу в заданных пределах изменения расхода и давления потока жидкости;
3. Многоступенчатое дросселирование потока жидкости, при котором общий перепад давлений распределяется между отдельными дроссельными элементами;
4. Выбор материалов с повышенной кавитационной стойкостью;
5. Подвод воздуха в зону кавитации;
6. Добавление в рабочую жидкость различных присадок.

В нашем случае мы не можем использовать пункты 1, 2, 3, 5 и 6 т.к. они нарушат кавитационную характеристику потока, вследствие чего изменится дисперсность и санитарно-гигиенические нормы продукта. [5, 6]

**Выбор материалов для увеличения кавитационной стойкости.** Большинство материалов, подвергающихся действию кавитации, обладает начальным (инкубационным) периодом, в течение которого эрозия отсутствует. За инкубационным следует период быстрого увеличения эрозионных повреждений, а затем период относительно стабильного состояния, когда скорость эрозии почти постоянна. Наконец, когда поверхность уже сильно изъедена и покрыта раковинами, скорость эрозии уменьшается.

На практике наибольший интерес представляют инкубационный период и период наибольшей скорости эрозии. Если длительность инкубационного периода будет превышать срок службы какого-либо компонента машины, то опасность эрозии этого компонента отсутствует. К сожалению, до сих пор мы еще не в состоянии предсказывать продолжительность инкубационного периода для конкретных частей машины. Однако, определяя скорость эрозии различных материалов, можно оценить относительное сопротивление этих материалов эрозии. Такой вид испытаний обычно используется в качестве ускоренного метода определения сопротивления стойкости материалов к эрозии.

Результаты лабораторных испытаний в таблице 1 показывают, что наиболее стойким к кавитационной эрозии материалом, как в пресной, так и в морской воде является алюминиевая бронза. К тому же этот материал отличается очень хорошими антикоррозионными свойствами. Однако алюминиевая бронза трудно поддается литью, поэтому в настоящее время разрабатываются другие сплавы, свободные от этого недостатка. Коррозионные и эрозионные свойства таких сплавов почти такие же, как у алюминиевой бронзы. Высоким сопротивлением эрозии обладают нержавеющие стали, лучшей из них считается мартенситная сталь.

Повреждения от кавитации могут быть уменьшены путем выбора подходящих сплавов. Часто рекомендуют применять твердые, стойкие к кавитации сплавы, в частности нержавеющие стали [4].

Таблица 1. – Эрозионные и прочностные характеристики металлов[4]

Материал	Модуль юнга, кг/см <sup>2</sup>	Прочность на разрыв, кг/см <sup>2</sup>	Относительное удлинение, %	Твердость HV	Скорость эрозии мм <sup>3</sup> /ч
Низкоуглеродистая сталь	2,11×10 <sup>6</sup>	5 970	16	200	5,5
Нержавеющая сталь	2,12×10 <sup>6</sup>	7 960	26	253	1,3
Высокопрочная латунь	0,915×10 <sup>6</sup>	4 020	17	120	4,7
Орудийный металл	0,88×10 <sup>6</sup>	2 900	22	96	4,9
Алюминиевая бронза	1,2×10 <sup>6</sup>	4 730	12	155	0,8
Бронза «Новостон»	1,24×10 <sup>6</sup>	7 150	21	203	1,5
Монель К-500	1,66×10 <sup>6</sup>	8 400	19	289	2,8
Монель К-500 (старейший)	1,46×10 <sup>6</sup>	11 300	13	396	1,2
Титан	1,18×10 <sup>6</sup>	13 200	19	425	0,8
Никель	1,24×10 <sup>6</sup>	4 870	35	152	4,4
Железо	1,72×10 <sup>6</sup>	10 200	–	345	1,3

Так же используют твердые сплавы (ТС) – керамико-металлические материалы, состоящие из карбидов тугоплавких металлов и пластичного связующего (металла или сплава) при содержании тугоплавкой фазы 50% (объемн.). Современные ТС получают из высших карбидов тугоплавких металлов и металлов железной группы (железа, никеля, кобальта) спеканием в присутствии жидкой фазы.

Различают карбида-вольфрамовые (WC-Co), титано-вольфрамовые (WC-TiC-Co), титано-тантало-вольфрамовые (WC-TiTaC-Co) и безвольфрамовые твердые сплавы. В государственных стандартах СССР вольфрамовые (вольфрамокобальтовые) сплавы обозначаются буквами ВК с прибавлением цифры, обозначающей содержание кобальта, например, ВК6, значит 6% кобальта. Титано-вольфрамовые твердые сплавы обозначаются буквами ТК. Марки сплавов в пределах группы также характеризуют содержание компонентов.

Следует отметить, что имеющиеся в литературе данные о физико-механических и теплофизических свойствах карбидов и других тугоплавких соединений зачастую противоречивы вследствие исключительно сильного влияния дефектов кристаллического строения данных соединений на их свойства. Карбиды переходных металлов отличаются высокой химической стойкостью и не разлагаются большинством минеральных кислот, их смесей и растворов щелочей. Стойкость карбидов против окисления кислородом воздуха убывает в следующей последовательности: TiC, ZrC, VC, TaC, NbC, MoC, WC.

Микротвердость карбидов высокая, и уменьшается при переходе от карбидов металлов IV-A группы к карбидам металлов групп V-A и VI-A, а модули упругости увеличиваются. В таблице 2 приведены значения микротвердости и модули упругости некоторых карбидов.

Таблица 2. – Механические характеристики карбидов [7]

Карбид	Микротвердость, ГПа	$E$ , ГПа	Карбид	Микротвердость, ГПа	$E$ , ГПа
TiC	28,5	320	TaC	16,0	290
ZrC	29,3	355	Cr <sub>3</sub> C <sub>2</sub>	13,3	380
HfC	29,1	359	MoC	15,0	544
VC	20,9	276	WC	17,3	720
NbC	19,6	345			

Области применения карбидов значительно расширились. Их используют в качестве огнеупорных материалов, коррозионностойких материалов химической промышленности; в авиационной и ракетной технике, для электротехнического и радиотехнического назначения [7].

**Повышение кавитационной стойкости деталей машин нанесением защитных покрытий и обработкой высококонцентрированными потоками энергии.** Одним из признанных методов повышения эксплуатационных характеристик деталей машин и механизмов является химико-термическая обработка (ХТО), которая радикальным образом изменяет состав и физико-химические свойства поверхностных слоев деталей, что позволяет значительно повысить их эксплуатационные свойства, в том числе и кавитационную стойкость. Это увеличивает надежность и долговечность машин, работающих в условиях кавитационной эрозии. Приведем примеры ХТО подходящие для кавитационной стойкости.

Азотирование – наиболее эффективный способ повышения износостойкости аустенитных нержавеющих сталей. Азотированию можно подвергать любые стали перлитного, ферритного и аустенитного классов, а также чугуны. Азотированные стали имеют высокую поверхностную прочность, не изменяющуюся при нагреве до 400–450 °С; высокую износостойкость и низкую склонность к задирам; высокое сопротивление кавитации и коррозии в атмосфере, пресной воде и паре. Деформация изделий при азотировании минимальна, азотированный слой хорошо шлифуется и полируется.

Цементацией – (науглероживанием) называют ХТО, заключающуюся в диффузионном насыщении поверхности материала углеродом при высоких температурах в углеродосодержащей среде. Процесс цементации широко применяют в промышленности благодаря его высокой эффективности и доступности. Цементации подвергаются в основном стали и в некоторых случаях тугоплавкие металлы и сплавы. Цементацию тугоплавких металлов и сплавов (Mo, W, Ti, Nb, Ta, жаропрочных сплавов на основе Co и Ni) применяют для повышения твердости и износостойкости поверхности, а также с целью повышения коррозионной стойкости и достижения особых электрохимических и электрофизических свойств.

Борирование – химико-термическая обработка насыщением поверхностных слоев стальных изделий и некоторых других металлов бором при температурах 900–950 °С. Цель борирования – это повышение твердости, износостойкости (особенно абразивной) и некоторых других свойств стальных изделий. Диффузионный слой толщиной 0,05–0,15 мм, состоящий из боридов FeB и Fe<sub>2</sub>B, обладает весьма высокой твердостью, стойкостью к абразивному изнашиванию и коррозионной стойкостью.

Диффузионное хромирование – это процесс насыщения поверхностных слоев изделия хромом при высоких температурах. Хромирование придает поверхностному слою изделий особые физико-химические свойства, позволяющие повысить ресурс изделий, работающих в условиях:

- коррозии под напряжением в средах азотной и уксусной кислот, перекиси водорода, растворе поваренной соли, морской воды, топлив, содержащих серу и окислы ванадия;
- износа при различных видах трения;
- кавитационной эрозии;
- знакопеременных нагрузок;
- ползучести при повышенных температурах;
- термических ударов.

В ряде случаев, например, для повышения жаростойкости, эрозионной стойкости или кислотоустойчивости, хромирование применяют к никелю (никелевым сплавам), тугоплавким металлам (их сплавам) и неметаллическим материалам (графиту и металлокерамике). В таблице 3 приведем показатели твердости покрытий, полученных различными видами ХТО [7, 8].

Таблица 3. – Сравнительные показатели твердости [8]

№ п/п	Процесс ХТО	Твёрдость (HV)
1	Борирование	1800–2000
2	Диффузионное хромирование	1350–1500
3	Цементация	500–700
4	Азотирование	300–350
5	Нитроцементация	570–690

Среди методов поверхностного упрочнения высококонцентрированными потоками энергии наибольшее распространение получили, обработка лазером, плазмой, электроискровым и электронно-лучевым упрочнением. К наиболее перспективным и прогрессивно развивающимся способам поверхностной закалки следует отнести эти способы, основанные на применении высококонцентрированных источников нагрева. Их применение позволяет получить более высокие эксплуатационные свойства изделий и качество упрочнения. Внедрение высококонцентрированных источников нагрева позволяет резко сократить энергозатраты, уменьшить коробление деталей, исключить необходимость использования различных сред и при этом полностью автоматизировать процесс. В соответствии с применяемым источником нагрева различают следующие способы поверхностной термической закалки:

Лазерное упрочнение – термическое упрочнение лазерным излучением металлов и сплавов лазерным излучением, основанное на локальном нагреве участка поверхности под воздействием излучения и последующем охлаждении этого поверхностного участка со сверхкритической скоростью в результате теплоотвода теплоты во внутренние слои металла. Основной целью лазерного термоупрочнения сталей, чугунов и цветных сплавов является повышение износостойкости деталей, работающих в условиях трения, прежде всего – при трении качения и скольжения. В результате лазерной закалки достигаются высокая твердость поверхности, высокая дисперсность структуры, уменьшение коэффициента трения, увеличение несущей способности поверхностных слоев и другие параметры. В ряде случаев лазерное термоупрочнение приводит к повышению теплостойкости. В качестве основного недостатка лазерного термоупрочнения следует отметить высокую отражающую способность многих реальных металлических изделий, требующих упрочнения, что в итоге негативно отражается на коэффициенте полезного действия лазеров при реализации поверхностной упрочняющей обработки.

Плазменное упрочнение состоит в высокоскоростном нагреве потоком плазмы поверхностного слоя металла и быстром его охлаждении в результате передачи тепла в глубинные слои материала детали. Цель плазменной закалки – упрочнение поверхностного слоя деталей и инструмента с обеспечением его толщины до нескольких миллиметров при неизменном общем химическом составе материала и сохранении во внутренних слоях первоначальных свойств исходного металла деталей и инструмента. Плазменная закалка не дает деформаций, благодаря чему закаленным деталям во многих случаях не требуется финишная шлифовка, что существенно снижает трудоемкость и себестоимость производства. Материалы, подверженные плазменной закалке, – инструментальные стали, чугуны, твердые сплавы, цементированные и нитроцементированные стали, цветные сплавы и другие материалы. Основные преимущества плазменного упрочнения заключаются в: большой глубине упрочненного слоя по сравнению, например, с лазерной закалкой; высокий эффективный КПД нагрева плазменной дугой (до 85%), для сравнения при лазерном упрочнении – 5%; возможность ведения процесса без применения охлаждающих сред, вакуума, специальных покрытий для повышения поглотительной способности упрочняемых поверхностей.

Электроискровое упрочнение основано на использовании электрических разрядов для обработки рабочей поверхности изделия и создания поверхностного слоя (покрытия) с требуемыми эксплуатационными свойствами. Осуществляется воздействие на металлические поверхности в газовой среде короткими (до 1000 мкс) электрическими разрядами энергией от сотых долей до десятка и более джоулей и частотой не более 1000 Гц. В ходе процесса электроискрового упрочнения происходит следующее: идут процессы преимущественного разрушения материала электрода (анода) и образования вторичных структур в рабочей его части, осуществляется перенос продуктов эрозии электрода на деталь (катод), после чего на поверхности обрабатываемого изделия протекают микрометаллургические процессы, элементы материала электрода диффундируют в поверхностный слой изделия и поверхность изделия приобретает новый специфичный рельеф. Образуется на поверхности изделия измененный слой, включающий белый слой, диффузионную зону и зону термического влияния, при этом изменяются свойства поверхностного слоя, формируется поверхностный слой мелкодисперсного состава, вплоть до наноуровня. Эти свойства управляются в широких пределах и обеспечивают требуемые качества: повышение микротвердости, износостойкости, жаростойкости и другие. Достоинства метода электроискрового упрочнения в возможности локального формирования покрытий в строго указанных местах радиусом от долей миллиметра и более (без защиты остальной поверхности); высокой адгезии электроискрового покрытия с основным материалом; отсутствии нагрева и деформации изделия в процессе обработки [8].

Электронно-лучевое упрочнение представляет собой мощное эффективное средство термического воздействия на материал. В качестве энергоносителя при реализации электронно-лучевых технологий выступает электронный луч. Генератором электронного луча служит электронная пушка, обеспечивающая эмиссию свободных электронов, их ускорение в электростатическом поле, фокусировку и отклонение пучка с посредством электрического и магнитного полей. Сформированный пушкой электронный пучок выводится в рабочую камеру, из которой предварительно насосами откачивают воздух. При взаимодействии электронного пучка с поверхностью обрабатываемой детали кинетическая энергия электронов превращается в другие формы энергии (главным образом в тепловую). Часть электронов отражается

и рассеивается поверхностью объекта, появляется рентгеновское излучение, кроме того, возникают вторичные и тепловые электроны. Закалка производится электронным лучом. Достоинством процесса является: минимальное коробление изделия; независимость степени поглощения энергии от оптических свойств и шероховатости поверхности, отсутствие необходимости нанесения покрытия на поверхность для повышения поглощающей способности; возможность использования одного оборудования для проведения различных технологических процессов - сварки, поверхностной закалки, плавления; отсутствие окисления поверхности [9].

**Заключение.** В результате исследования были проанализированы методики уменьшения воздействия кавитационной эрозии на ударную головку путем изменения материалов и нанесения различных видов упрочнения. Актуальность темы исследования в работе определяется тем, что вопросам импортозамещения в настоящее время в Республике Беларусь уделяется весьма повышенное внимание, а ударная головка гомогенизатора FBF 075 представляется в оригинальном виде импортного производства.

#### ЛИТЕРАТУРА

1. Крусъ Г.Н., Тиняков В.Г. Фофанов Ю.Ф. Технология молока и оборудование предприятий молочной промышленности. М.: Агропромиздат. 1986. – 280 с.
2. Мегедь В. Справочник – Технология производства молочных продуктов – Раздел Гомогенизаторы 2005. – 7с.
3. Сурков В.Д. Липатов Н.Н., Барановский Н.В. Технологическое оборудование предприятий молочной промышленности – 2-е издание, М.: Легкая и пищевая промышленность. 1970.-547 с. Ил.
4. Пирсол И., Кавитация. Пер. с англ., Журавлева Ю.Ф. Ред., предисл. И дополн. Эпштейна. М., “Мир”, 1975. 95с. с илл.
5. Арзуматов Э.С. Кавитация в местных гидравлических сопротивлениях. – М.: Энергия, 1978. – 304 с., ил
6. Пылаев Н.И., Эдель Ю.У. Кавитация в гидротурбинах. Л., “Машиностроение” (Ленинградское отделение), 1974. – 256 с.
7. Анциферов В.Н., Бобров Г.В. и др. Порошковая металлургия и напыление покрытия. Учебник для вузов. М.; Металлургия, 1987. – 792 с.
8. Белевитин, В.А. Упрочнение и восстановление деталей машин: справочное пособие / В.А. Белевитин, А.В. Суворов. – Челябинск: Изд-во Челябинск. Гос. пед. ун-та, 2015. – 263 с.: ил.
9. Рыкалин Н.Н., Углов А.А., Зуев И.В., Кокора А.Н.. Лазерная и электронно-лучевая обработка материалов: Справочник/ – М.: Машиностроение, 1985. – 496 с.

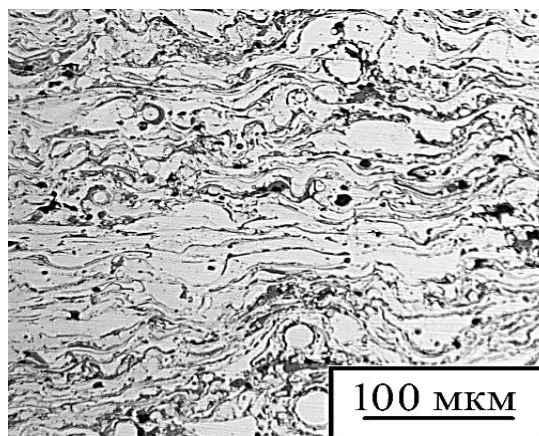
УДК 621.924.1

**ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ИСПОЛЬЗОВАНИЯ ПОКРЫТИЙ  
ИЗ ИНСТРУМЕНТАЛЬНЫХ СТАЛЕЙ ДЛЯ ПОВЫШЕНИЯ СТОЙКОСТИ  
РЕЖУЩИХ ИНСТРУМЕНТОВ****Д.А. БАШЛАЧЁВ, Е.В. ПУЙМАН***(Представлено: д-р техн. наук, проф. Н.Н. ПОПОК)*

*Изучены структуры и фазовый состав упрочняющих покрытий из высоколегированных инструментальных сталей, определено распределение модифицирующей примеси по глубине упрочненных слоев инструментальных материалов, исследована износостойкость покрытия, выработаны технологические рекомендации по режимам ионно-плазменной обработки высоколегированных инструментальных сталей.*

**Введение.** Проблема повышения износостойкости режущих инструментов, функционирующего при повышенных температурах и давлениях, становится все более значимой в связи с постоянным увеличением уровня действующих нагрузок, скоростей и рабочих температур, а также применением агрессивных сред. Одним из наиболее перспективных способов существенного повышения физико-механических, триботехнических характеристик и антикоррозионных свойств, а также выносливости сталей и сплавов является модифицирование поверхностных слоев этих материалов концентрированными потоками ионов металлов и газов [1, 2], интенсивно развиваемый в Полоцком государственном университете [3]. Важной особенностью метода является сохранение прочностных свойств подложки за счет пониженной температуры ионно-лучевой обработки, что весьма существенно для повышения ресурса работоспособности материалов с упрочненными поверхностными слоями. Весьма перспективно использование для покрытий высоколегированных инструментальных сталей мартенситного типа. Дополнительные возможности для повышения прочности и износостойкости газотермических покрытий с метастабильными фазами открывает использование технологии ионно-плазменной обработки их поверхностных слоев.

**Основная часть.** На рисунке 1 представлена характерная микроструктура напыленного покрытия из инструментальной стали 95X18, которая содержит перемеживающиеся прослойки металлов и оксидов, остаточная пористость покрытия не превышает ~3%, а содержание оксидов составляет ~25 об. %.



**Рисунок 1. – Микроструктура газотермического покрытия  
из проволоочной стали 95X18**

Газотермическое покрытие из инструментальной стали 95X18 подвергается ионно-плазменной обработке с нанесением нитрида титана. Образцы с покрытием 95X18 перед нанесением ионно-плазменного покрытия из TiN находились в нешлифованном и шлифованном состояниях (рис. 2, а, б). Нанесение покрытия ионно-плазменным методом проводилось на установке PVM-0,5 FN при следующих режимах: вакуум в камере установки –  $3-6 \cdot 10^{-3}$  МПа, температура – 150–250 °С, ток – 85–95 А, давление азота 0,097 Па и пропана – 100 Па, время нанесения покрытия 20–30 мин.

Ответный образец изготовлен из быстрорежущей стали марки P6M5. Износ образцов определяется путём взвешивания на аналитических весах модели ВК-1500,1.

Рентгеновская дифрактограмма от образца покрытия из стали 95X18 с осажденным слоем TiN на нешлифованную и шлифованную поверхности представлена на рисунке 3



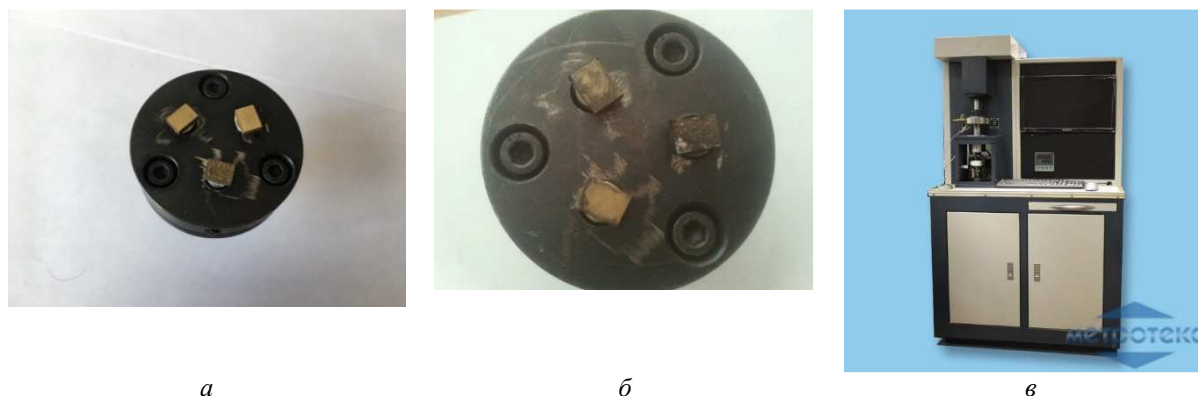


Рисунок 2. – Внешний вид образцов до (а), после (б) испытания и машины трения (в)

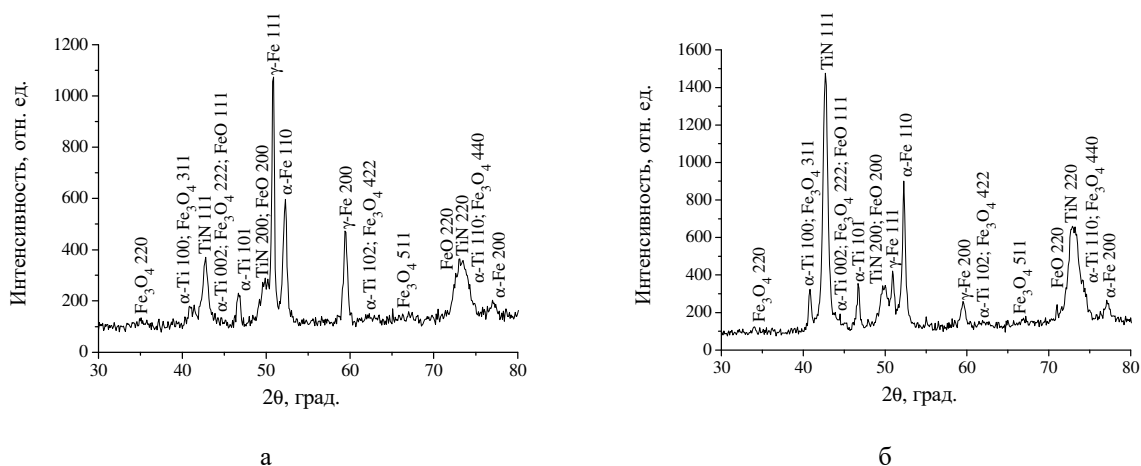


Рисунок 3. – Фрагмент рентгеновской дифрактограммы (СоKα) покрытия из стали 95X18 с покрытием нитридом титана на нешлифованных (а) и шлифованных (б) образцах

Исследование износостойкости образцов с комбинированным (газотермическим и ионно-плазменным) покрытием из инструментальной стали 95X18 и нитрида титана (TiN) производилась на машине трения модели МТ-393 (рис. 2, в) с частотой вращения образцов 200 мин<sup>-1</sup> и усилием прижима 20–40 Н. Результаты исследования износа образцов представлены на рисунке 4.

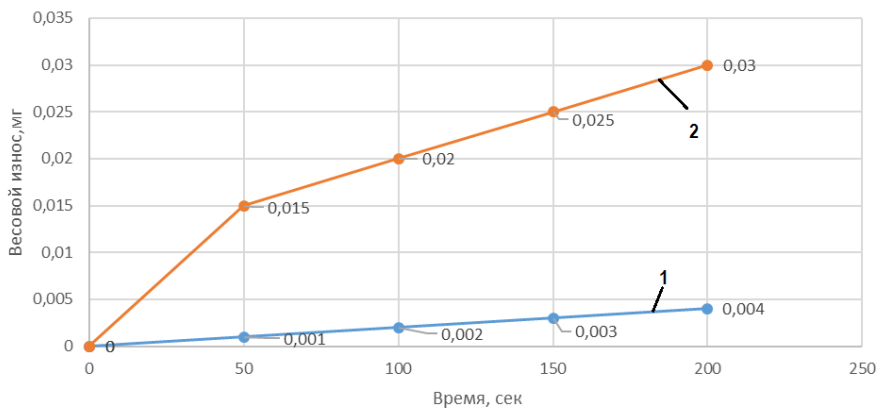


Рисунок 4. – Зависимости износостойкости образцов от времени испытания: 1 – шлифованный, 2 – нешлифованный

Как видно из графика, износостойкость образцов с нанесённым покрытием из нитрида титана на шлифованную поверхность газотермического покрытия 95X18 выше в 2 раза, чем нешлифованного.

**Заключение:**

1. Микроструктура покрытия из инструментальной стали 95X18 по пористости (менее 3%) и содержанию оксидов (~25%) соответствует требованиям, предъявляемым к материалам для режущих инструментов.

2. Нанесение ионно-плазменным методом тонкоплёночного покрытия (3–5 мкм) на шлифованную поверхность образца из стали 95X18 способствует увеличению содержания TiN и снижению содержания  $\gamma$ -фазы. Микротвёрдость поверхности при этом существенно увеличивается (до 4 раз).

3. Износ образцов с комбинированным покрытием из стали 95X18 и TiN снижается до 2 раз в сравнении с нешлифованными образцами при условии нагружения образцов в пределах 20–40Н и их линейной скорости, не превышающей 20–40 м/мин. Это значение линейной скорости соответствует рекомендуемым при обработке материалов режущим инструментом из быстрорежущей стали.

4. Рекомендуем проводить ионно-плазменную обработку образцов в течении 30–40 минут при токе 90А, температуре 200 °С, вакууме  $5 \cdot 10^{-3}$  МПа, давление азота 0,097 Па и пропана – 100Па.

ЛИТЕРАТУРА

1. Хирвонен Дж.К. Ионная имплантация. М.: Металлургия, 1985. – 285 с.
2. Х. Риссел, И. Руге. Ионная имплантация: Пер. с нем. В.В. Климова, В.Н. Пальянова. / Под ред. М.И. Гусевой. – М.: Наука. Главная редакция физико-математической литературы, 1983. – 360с.
3. Попок, Н.Н. Ионные источники: виды, конструкции, применение = Ion sources: variety, construction, application: пособие для студентов и магистрантов машиностроительной специальности/ Н.Н. Попок, С.В. Дербуш, А.Н. Попок. – Новополоцк: Полоц. гос. ун-т, 2017. – 72с.

УДК 621.924.1

## РАЗРАБОТКА БЛОЧНО-МОДУЛЬНЫХ РЕЖУЩИХ ИНСТРУМЕНТОВ ДЛЯ СТАНКОВ С ЧПУ

Д.А. БАШЛАЧЁВ

(Представлено: д-р техн. наук., проф. Н.Н. ПОПОК)

Проведён анализ модульных инструментальных систем, включающий виды инструментальных модулей, механизмы крепления модулей, инструментальные системы для многоцелевой обработки, который позволил определить современные тенденции в области конструирования и производства инструментальных систем. Рассмотрены конструктивные схемы различных типов блочно-модульных режущих инструментов и предложены конструкции фрезы и зенкера для применения их на станках с ЧПУ «Рободрил».

**Введение.** Большой резерв повышения эффективности машиностроения заложен в создании рациональной системы инструментального обеспечения отечественных предприятий по опыту ведущих зарубежных фирм, например, "Sandvik Coromant" (Швеция), "Mitsubishi" (Япония), "Isca" (Израиль), "Hertel" (Германия) и других [1]. Предложенные фирмами системы предусматривают проектирование широкой гаммы режущих инструментов с унифицированными конструктивными элементами для каждого типа инструмента, применение прогрессивных технологий их изготовления, включающих нанесение износостойких покрытий на лезвия, разработку рекомендаций по режиму работы режущих инструментов с учётом свойств обрабатываемого и инструментального материалов, диагностику и ремонт инструмента в процессе эксплуатации. Такой подход к инструментальному обеспечению предприятий позволяет уменьшить номенклатуру режущих инструментов, повысить срок их службы и, в конечном итоге, сократить материальные и финансовые затраты на проектирование, производство и эксплуатацию режущих инструментов [2, 3].

**Основная часть.** В настоящее время на кафедре ТиОМП ведётся работа по конструированию и изготовлению блочно-модульных режущих инструментов для фрезерных станков с ЧПУ фирмы Fanuc [4, 5].

Отличительными особенностями инструмента для данных станков являются использованные конусные хвостовики японского стандарта JIS B6339 и ограниченные габаритные размеры корпусных модулей.

На рис. 1 представлена конструктивная схема торцевой фрезы для фрезерного станка с ЧПУ. Резцовые блоки 1, устанавливаются в отверстиях корпусного модуля 2. Блоки крепятся в корпусном модуле с помощью зажима, состоящего из сухаря упорного 3 и сухаря резьбового 4. Корпусный модуль с закреплёнными резцовыми блоками устанавливается на хвостовик 5 и зажимается винтом 7, торцовые шпонки 6 предотвращают от проворачивания корпусного модуля и придают ему жёсткости. В хвостовик вворачивается наконечник 8 для затягивания фрезы в шпиндель станка.

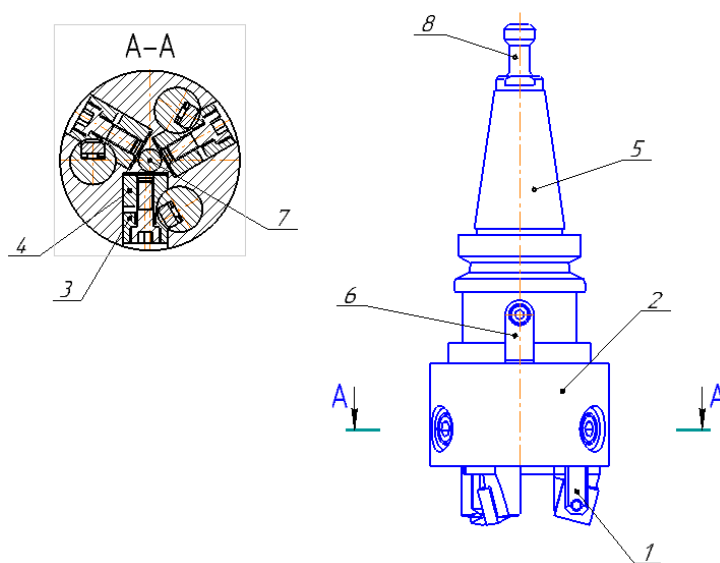


Рисунок 1. – Конструктивная схема торцевой фрезы

На рис. 2 показана 3D модель конструкции торцевой фрезы.

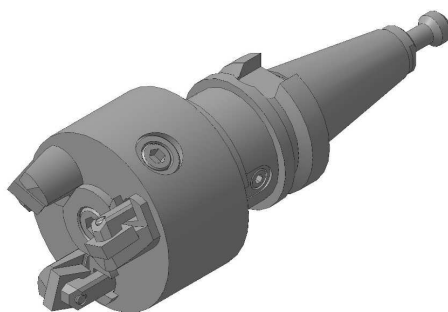


Рисунок 2. – 3D модель конструкции торцовой фрезы

Конструктивная схема зенкера для фрезерного станка с ЧПУ (рис. 3) включает резцовые блоки 1, которые устанавливаются в отверстиях корпусного модуля 2. Резцовые блоки установлены в корпусный модуль под углом 15°, что позволяет обрабатывать глухие отверстия и получать проточку с углом 90°. Блоки крепятся в корпусном модуле с помощью зажима, состоящего из сухаря упорного 3 и сухаря резьбового 4. Корпусный модуль с закреплёнными резцовыми блоками устанавливается на хвостовик 5 и зажимается винтом 7, торцовые шпонки 6 предотвращают от проворачивания корпусного модуля и придают ему жёсткости. В хвостовик вворачивается наконечник 8 для затягивания зенкера в шпиндель станка.

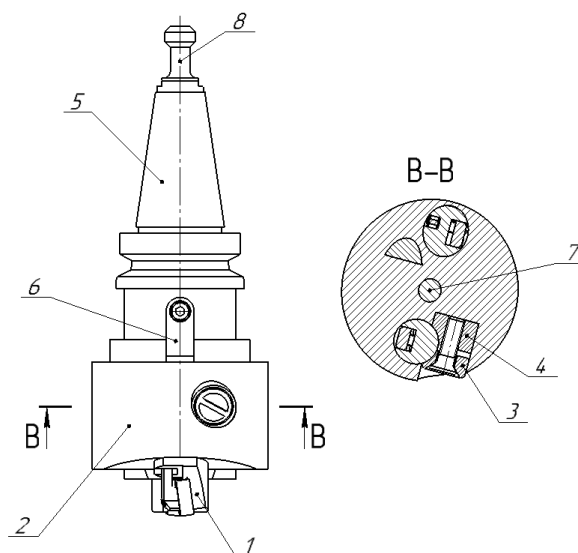


Рисунок 3. – Конструктивная схема зенкера

На рис. 4 показана 3D модель конструкции зенкера.



Рисунок 4. – 3D модель конструкции зенкера

**Заключение.** Рассмотренные конструктивные схемы различных типов блочно-модульных режущих инструментов и предложенные конструкции фрезы и зенкера для применения их на станках с ЧПУ «Рободрил», отличительными особенностями которых являются: хвостовик В30, а минимально возможный размер блоков резцовых, что позволило изготовить торцовую фрезу и зенкер диаметром  $\varnothing 63$  мм и количеством блоков резцовых равным 2–3 штук.

## ЛИТЕРАТУРА

1. Попок Н.Н. Анализ тенденций проектирования инструментальных систем. Часть 1. Термины и системы обозначений режущих пластин, державок и модулей. Вестник ПГУ. Серия В. Промышленность. Прикладные науки №3, 2012г. – 71–81 с.
2. Попок Н.Н. Анализ тенденций проектирования инструментальных систем. Часть 2. Механизмы крепления модулей. Вестник ПГУ. Серия В. Промышленность. Прикладные науки №11, 2012г. – 11–19 с.
3. Попок Н.Н. Анализ тенденций проектирования инструментальных систем. Часть 3. Инструментальные системы для многоцелевой обработки. Вестник ПГУ. Серия В. Промышленность. Прикладные науки №3, 2013г. – 19–37 с.
4. Конструирование и оснащение технологических комплексов / А.М. Русецкий [и др.]; под общ. ред. А.М.Русецкого – Минск: Беларус. наука, 2014. – 316 с.
5. Попок Н.Н., Максимчук А.С., Портянко С.А. Совершенствование системы закрепления режущих пластин и блоков резцовых в блочно-модульных режущих инструментах. Вестник ПГУ. Серия В. Промышленность. Прикладные науки №3, 2015. – 16–22 с.

УДК 621.693

**ПОВЫШЕНИЕ РЕСУРСА РАСПРЕДЕЛИТЕЛЬНЫХ ВАЛОВ  
СОВЕРШЕНСТВОВАНИЕМ ТЕХНОЛОГИЙ ИХ ВОССТАНОВЛЕНИЯ НАПЛАВКОЙ****М.В. ВОРОБЬЁВ, М.Г. УЛЬСКИЙ***(Представлено: канд. техн. наук, доц. Т.В. ВИГЕРИНА)*

*По результатам экспериментальных исследований установлена эффективность использования технологии наплавки для восстановления эксплуатационных характеристик стальных распределительных валов автомобильных и тракторных двигателей.*

Восстановление изношенных деталей автомобилей обеспечивает экономию металла, топлива, энергетических и трудовых ресурсов, а также рациональное использование природных ресурсов и охрану окружающей среды. Для восстановления работоспособности изношенных деталей автомобилей требуется в 5–8 раз меньше технологических операций по сравнению с изготовлением новых деталей. Важным резервом снижения издержек в агропромышленном комплексе при производстве продукции является обеспечение высокой надёжности эксплуатируемого оборудования и техники [1–3]. Поэтому остаются актуальными исследования, направленные на разработку современных технологических процессов восстановления и упрочнения деталей. Ресурс двигателя внутреннего сгорания зависит в основном от состояния кривошипно-шатунного и газораспределительного механизмов, а распределительный вал – это главное звено газораспределительного механизма. Шейки распределительного вала подвержены трению скольжения при больших скоростях и высоких удельных давлениях. Наиболее характерными дефектами распределительных валов являются: износ, задиры шеек и др., поэтому износостойкость является одним из свойств материала, определяющих ресурс распределительных валов [4, 5].

Повышение эффективности и снижение трудоёмкости ремонта сельскохозяйственной техники непосредственно связано с использованием современных способов восстановления деталей и новых материалов. К числу таких способов относится наплавка, которая позволяет использовать порошковые износостойкие наплавочные сплавы, обеспечивающие повышение срока службы деталей в 2–8 раз [6]. Перед традиционными дугowymi способами наплавка имеет следующие преимущества:

- высокую производительность и низкую энергоёмкость;
- позволяет получать наплавленные слои малой толщины с относительно низким тепловым воздействием на деталь;
- обеспечивает низкую степень перемешивания основного и наплавочного материала, что способствует сохранению в наплавленном металле исходных физико-механических свойств наплавочного материала;

В настоящее время использование наплавки на ремонтных предприятиях сдерживается отсутствием специализированного оборудования и технологий восстановления конкретных деталей.

Целью настоящей работы являлось экспериментальное исследование влияния наплавленного материала на износостойкость наплавленных покрытий, а также влияние состава наплавляемого материала на структуру полученных покрытий.

Исследования влияния наплавленного слоя на износостойкость и свойства поверхности трения проводились по схеме трения «палец-диск». Индентор изготовлен из Стали 45 в форме стержня.

Всего изготовлено 9 образцов и разделены на 3 группы:

1. Образцы первой группы были подвергнуты термообработке: закалка с последующим высоким отпусканием до твердости не ниже HRC 54. Общее время нагрева (время нагрева и выдержки) деталей при закалке брали из расчёта 1 минута на 1 мм наименьшего размера наибольшего сечения.

2. Вторая группа образцов была наплавлена порошком ПГ-19М-01. Порошок предназначен для нанесения слоя с высокими антифрикционными свойствами, имеет низкий коэффициент трения. За счет наличия пористости работает как спеченный антифрикционный материал. Сопротивление ударам удовлетворительное. Порошок применяется для восстановления трущихся поверхностей деталей машин и агрегатов.

Метод восстановления и (или) упрочнения: напыление без оплавления. Может служить компонентом смесей для прессования изделий в порошковой металлургии. Рекомендуемая толщина напыляемых покрытий до 3,0 мм. Возможно получение более толстых слоев. Химический состав порошка, масс. %: 8,5–10,5 Al; 4 Fe. Основа – Cu. Твердость наплавленного слоя: 65–72 HRB.

3. Третья группа была наплавлена износостойким хромистым чугуном (ИЧХ). Химический состав порошка, масс. %: 0,5 Mo; 0,4 V; 0,8 Ni; 17–19 Cr.

В качестве контртела выбран диск Ø70 x 6 мм из свинцовистой бронзы БрС30.

Оценка триботехнических свойств проводилась на универсальной машине трения MODEL: MMW – 1A, STRIAL NO.:1705029, JINAN KASOV TESTING EQUIPMENT CO., LTD вертикального типа с компьютерным управлением процессом тестирования при различных режимах трения.

Качество формирования наплавленного слоя, его химический состав и структура в большой мере зависят от режима наплавки. Всегда желательно максимальное постоянство режима. Ввиду этого заслуживает предпочтения наплавка постоянным током, так как в заводских сетях переменного тока часто бывают резкие колебания напряжения, что отрицательно влияет на стабильность режима и, следовательно, на форму наплавленного валика.

Режим наплавки желательно выбрать так, чтобы было обеспечено: отличное формирование каждого наплавленного валика, максимальная производительность наплавки (в кг/час), минимальное (но вполне надежное) проплавление основного металла или ранее наплавленного слоя, минимальный припуск для механической обработки. Основными факторами, определяющими режим наплавки, являются: число электродов, ток, напряжение дуги, скорость перемещения дуги, вылет электродов, шаг наплавки.

На основании практических данных и рекомендаций технической литературы [6], исходя из габаритов и массы детали, а также возможностей пространственного ведения процесса наплавки следует рекомендовать следующие режимы наплавки, представленные в таблице 1.

Таблица 1. – Режимы наплавки

Параметры	ПГ-19М-01	ИЧХ18
Наплавка вершины кулачка		
Сила тока, А	120–140	125–150
Напряжение, В	25	30
Наплавка цилиндрической поверхности кулачка		
Сила тока, А	160–170	180–190
Напряжение, В	30	30
Наплавка опорной шейки вала		
Сила тока, А	170–180	190–200
Напряжение, В	30	35

На основании результатов полученных при испытании на машине трения MODEL: MMW – 1A был построен график силы трения каждого из видов образцов (рис. 1).

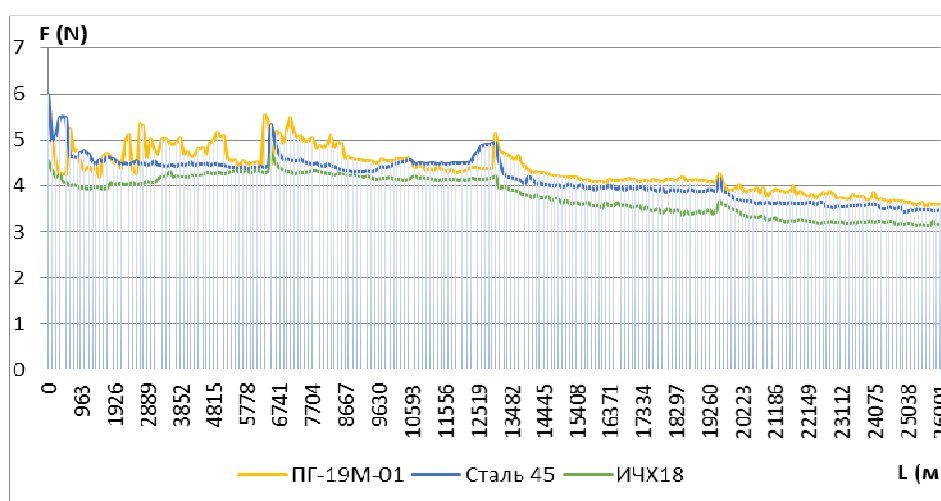


Рисунок 1. – График силы трения  $F(N)$  от пути трения

В первую очередь можно выделить участки с общим характером движения графиков. Все образцы в начале пути проходят этап притирки, это видно по хаотичному движению графика. Незначительные скачки графика на пути трения 6500, 13000 и 19500 метров связаны с остановкой машины трения для проведения взвешиваний, а также замены масла. Этап притирки можно проследить как по показаниям графика, так и по цвету смазочного материала (рис. 2).



Рисунок 2. – Цвет масла: а – после первого этапа трения ИЧХ18 (0-6500м)  
б – после второго этапа (6500–13000 м)

Интенсивность изнашивания наплавленного материала исследовали массовым методом путем взвешивая образцов через определенный промежуток времени (3600 с) в момент испытаний на машине трения. Результаты измерений представлены в виде графика (рис. 3).

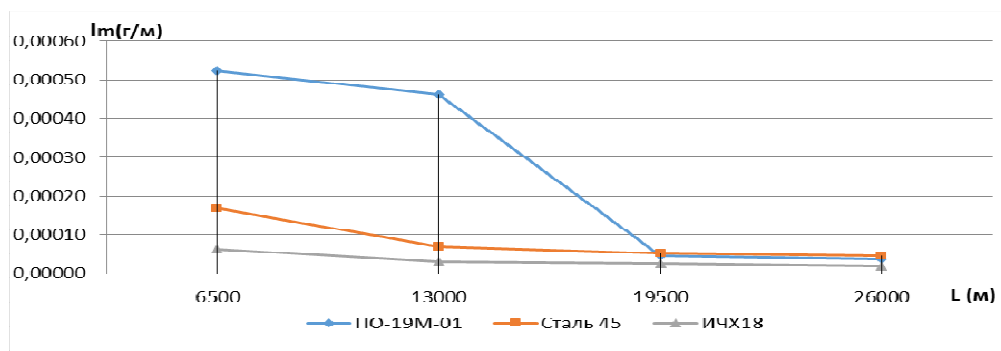


Рисунок 3. – Зависимость массового износа  $\Delta m$  от пути трения

Наблюдается более сильная и продолжительная потеря веса в момент притирки образцов, предположительно причиной стало недостаточное время шлифовки образцов после наплавки, однако как видно на графике после притирки показания износостойкости наплавленного слоя сравнимы со Сталью 45 и даже незначительно превосходят. Образцы, наплавленные износостойким хромистым чугуном, имеют минимальные массовые изменения на всех этапах трения, относительно других образцов.

Одной из важных характеристик металла является его структура. Микроструктура – строение металла, видимое при больших увеличениях с помощью микроскопа. Микроскопический анализ даёт возможность определить величину, форму и расположение зерен, мелкие неметаллические включения, мелкие трещины и другие дефекты, которые могут понизить механические свойства металлических изделий. Поэтому контроль структуры производится на всех этапах изготовления изделий, начиная от выплавки металла и кончая термической обработкой готовых деталей и их отделочными операциями.

Микроструктура покрытия, полученного наплавкой порошком ПГ-19М-01 представлена на рисунке 4.

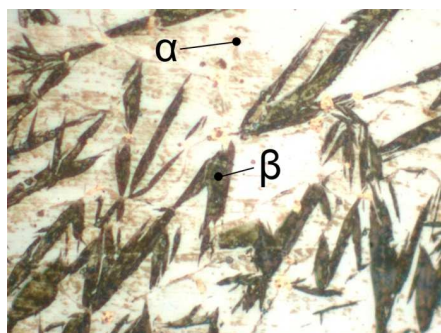


Рисунок 4. – Микроструктура покрытия, полученного наплавкой порошком ПГ-19М-01



Для более детального анализа были проведены измерения твердости по глубине наплавленного слоя, по результатам которых можно выделить 3 зоны: основа (наплавленный металл), переходный слой и покрытие.

В переходной зоне наблюдается значительное увеличение микротвердости (1200HV) в два раза превышающее среднее показатели на поверхности наплавленного слоя.

Порошок ПГ19М-01 по своим прочностным свойствам практически не уступает стали: по антифрикционным выше, по технологичности наплавки превышает, т.к. температура плавления приблизительно в 1,7–2 раза ниже. Отрицательные эффекты от применения данного порошка: разные теплофизические свойства со сталью, бронза с железом образуют гальванопару и при нарушении сплошности покрытия сталь интенсивно разрушается

Аналогичным способом был проведен анализ микроструктуры покрытия, полученного наплавкой высокохромистого износостойкого чугуна (рис. 5).

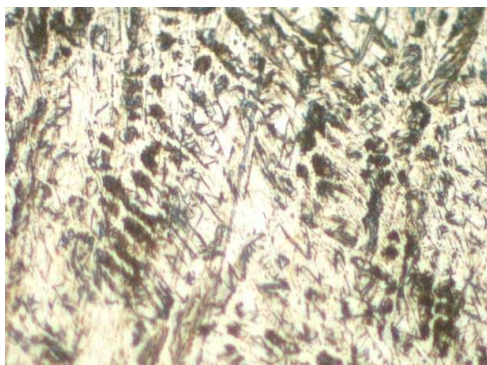


Рисунок 5. – Микроструктура наплавленного порошком ИЧХ18 покрытия

Структура имеет ярко выраженное дендритное строение, скорее всего заэвтектического типа микротвердость 560HV.

Низкий коэффициент трения обеспечивается высокой твердостью (низкой шероховатостью, отсутствием пор и задигов). В процессе трения сохранялась повышенная задиристость из-за отсутствия средства поверхностей. Микротвердость покрытия из износостойкого чугуна на поверхности на 10% выше, чем микротвердость основы.

Анализ полученных зависимостей интенсивности изнашивания от пути трения показал, что интенсивность изнашивания материала покрытия, образцов наплавленных порошком ПГ19М-01 сопоставима с интенсивностью изнашивания образцов из Стали 45, а для образцов с покрытием ИЧХ18 ниже в 1,1 раза.

На основании проведенных экспериментальных исследований для восстановления стальных распределительных валов рекомендована наплавка износостойким хромистым чугуном, позволяющим получить покрытие с высокой износостойкостью и низким коэффициентом трения.

#### ЛИТЕРАТУРА

1. Власов, В.М. Техническое обслуживание и ремонт автомобилей / В.М. Власов, С.В. Жанказиев, С.М. Круглов [и др.]. – 2-е изд., стер. – М.: Издательский центр «Академия», 2004. – 480 с.
2. Шадричев, В.А. Ремонт автомобилей / В.А. Шадричев. – М.: Высшая школа, 1970. – 479 с.
3. Новиков, А.Н. Восстановление и упрочнение деталей автомобилей: учебное пособие / А.Н. Новиков, М.П. Стратулат, А.Л. Севостьянов.– Орловский государственный технический университет.– Орел, 2006. – 336 с.
4. Бурумкулов, Ф.Х. Электроискровые технологии восстановления и упрочнения деталей машин и инструментов (теория и практика) / Ф.Х. Бурумкулов, П.П. Лезин, П.В. Сенин [и др.]; – МГУ им. Н.П. Огарева. – Саранск: Тип. «Красный Октябрь», 2003. – 504 с.
5. Бурумкулов, Ф.Х. Восстановление и упрочнение деталей электроискровым методом / Ф.Х. Бурумкулов, А.В. Беляков, Л.М. Лельчук, [и др.] // Сварочное производство. – 1998.– № 2. – 192 с.
6. Глазков, Ю.Е. Восстановление деталей сельскохозяйственных машин наплавкой: метод. указания / Ю.Е. Глазков, А.И. Попов. – Тамбов: Тамбовский гос. тех. ун-т, 2004. – 13 с.

УДК 621.693

## ВОССТАНОВЛЕНИЕ ДЕТАЛЕЙ АВТОМОБИЛЯ ТИПА «ВАЛ»

**М.В. ВОРОБЬЁВ, М.Г. УЛЬСКИЙ***(Представлено: канд. техн. наук, доц. Т.В. ВИГЕРИНА)*

*В работе была изучена номенклатура восстанавливаемых деталей автомобиля, установлены наиболее часто встречающиеся дефекты и предложен наиболее производительный и менее энергоемкий способ восстановления деталей типа «вал» наплавкой порошковыми материалами.*

Автомобиль представляет собой сложную техническую систему, элементы которой имеют различные характеристики устойчивости к потере работоспособного состояния. На них влияют как внутренние конструктивные факторы, зависящие от назначения и свойств элемента, так и совокупность внешних факторов, определяемых как условия эксплуатации автомобиля.

Современный автомобиль состоит из 15–20 тыс. деталей, из которых 7–9 тысяч теряют свои первоначальные свойства при работе, причем около 3–4 тысяч деталей имеют срок службы меньший, чем у автомобиля в целом. Все это вызывает наибольшие простои автомобилей, ресурсные затраты в эксплуатации [1].

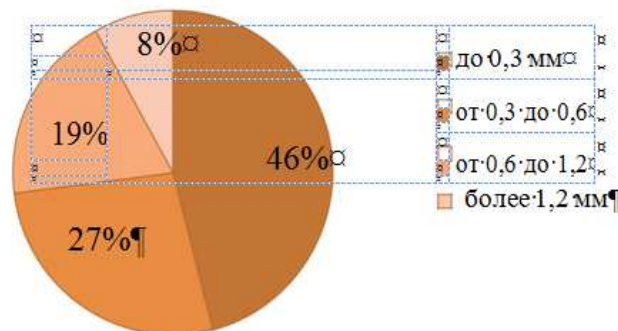
Целью работы являлся анализ номенклатуры восстанавливаемых деталей автомобиля, выявление наиболее часто встречающихся дефектов у деталей типа «вал» и способов их восстановления.

Анализ литературных источников показал, что для более 70% изношенных деталей автомобильной техники рационально было бы повторное использование после восстановления. Это значительно снижает ресурсные затраты автотранспортных предприятий, а, кроме того, это экономически оправдано для ремонтного производства. Затраты на восстановление деталей в большинстве случаев не превышают 25–30% их стоимости, а при квалифицированном назначении технологии восстановления достигается 100%-й ресурс.

Автомобильные детали типа «вал» составляют большую часть номенклатуры восстанавливаемых деталей. В большинстве случаев именно эти детали лимитируют ресурс узлов и агрегатов машин. Коэффициент их восстановления при капитальном ремонте машин составляет 0,25–0,95. Длина восстанавливаемых валов составляет 100–4000 мм, однако более 90% этих деталей имеют длину не более 1000 мм. Диаметры валов равны 12–210 мм, но у 98% валов диаметр не превышает 60 мм. Среднее значение массы составляет около 3 кг.

У деталей типа «вал» наиболее часто дефекты появляются на посадочных поверхностях под подшипники и резьбовых поверхностях.

Поверхности под подшипники восстанавливают при износе более 0,017–0,060 мм; поверхности неподвижных соединений (места под ступицы со шпоночными пазами и др.) за счет дополнительных деталей – при износе более 0,04–0,13 мм; поверхности подвижных соединений – при износе более 0,4–1,3 мм; под уплотнения – более 0,15–0,20 мм. Шпоночные пазы восстанавливают при износе по ширине более 0,065–0,095 мм; шлицевые поверхности – при износе более 0,2–0,5 мм [2].



**Рисунок 1. – Анализ дефектов деталей типа «вал» по степени износа**

Из всей совокупности восстанавливаемых поверхностей валов 46% изнашиваются до 0,3 мм; 27% – от 0,3 до 0,6 мм; 19% – от 0,6 до 1,2 мм и 8% – более 1,2 мм.

Основным требованием, которое необходимо выполнить при восстановлении валов, является обеспечение размеров и шероховатости восстанавливаемых поверхностей, их твердости, сплошности покрытия, прочности сцепления нанесенных слоев с основным металлом, а также симметрично-

сти, соосности, радиального и торцового биений обработанных поверхностей, параллельности боковых поверхностей зубьев шлицевых и шпоночных пазов оси вала.

Валы автомобильной техники изготавливают преимущественно из среднеуглеродистой и низколегированной сталей. Их подвергают поверхностной закалке токами высокой частоты, цементации с последующей закалкой, нормализации.

Анализ литературных источников показал, что дефекты деталей автомобилей принято делить на три группы: механические повреждения, химико-тепловые повреждения и износ.

Повреждения деталей механическим способом происходит в результате повреждения его поверхности трещинами, рисками и задирами, а также возможен изгиб вала, его поломка или скручивание.

Дефекты химико-теплого характера возникают в результате сложных взаимодействий при тяжелых условиях работы узлов и агрегатов, в которых они установлены. К таким повреждениям относятся: коробление, коррозия, раковины и т.д. Данные повреждения встречаются реже, по сравнению с другими.

Разнообразие условий эксплуатации обуславливает различные виды изнашивания рабочих поверхностей деталей автомобилей. Для валов характерными видами изнашивания являются абразивное, коррозионно-механическое, гидроабразивное, гидроэрозионное и кавитационное.

Анализ причин неисправности деталей типа «вал» показал, что большей частью (более 70%) основных дефектов является изнашивание деталей, которые работают в сопряжениях типа вал – подшипник. Особенно износы валов характерны в типе трения – подшипник скольжения.

В автомобильной технике сопряжения типа вал – подшипник скольжения применяют в цилиндро-поршневой группе (коленчатый вал – вкладыши – блок двигателя, поршневой палец – шатун), в головке блока цилиндров (ГБЦ) (распределительный вал – ГБЦ), в подвеске автомобиля (шкворень – цапфа поворотного кулака), в турбокомпрессоре (вал ротора – втулка подшипника). При эксплуатации автомобилей в этих сопряжениях возникает естественный износ, приводящий к поломке и выходу из строя данного узла или всего агрегата. В ходе ремонта в более 50% случаев помимо износа корпусной втулки требуется замена либо восстановление вала, также из-за износа, причем стоимость втулок составляет 5–10% от стоимости самих валов. В связи с этим восстановление валов в сопряжениях с подшипником скольжения и выбор методов восстановления является актуальной задачей.

Основная масса дефектов распределительного вала сводится к значительному износу и задирам на поверхностях кулачков и шеек, трещинам, царапинам на поверхностях под сальники, разрушению крепежных отверстий или резьбы в них, а также шпоночных пазов и посадочных мест под штифты, шкивы и шестерни (рис. 2).

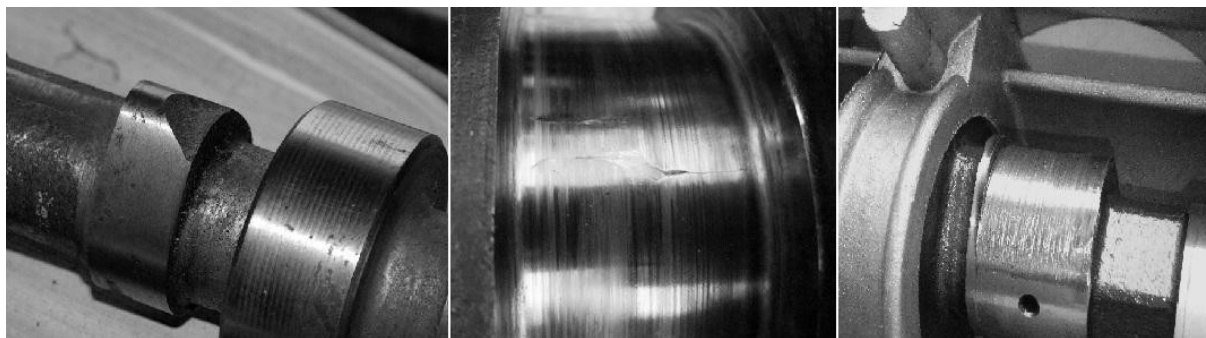


Рисунок 2. – Виды дефектов распределительного вала

Причины, вызывающие многие из вышеперечисленных дефектов, аналогичны с причинами износа коленчатого вала двигателя: недостаточное давление в системе смазки, недостаточный уровень масла, некачественное масло, сильный перегрев, приводящий к разжижению масла, попадание в масло бензина (дизельного топлива), приводящее к разжижению масла, засоренного масляного фильтра, и работа двигателя на грязном масле.

Преимущественное применение при восстановлении валов получили следующие виды наплавки: в среде углекислого газа, вибродуговая в различных защитных средах, в природном газе и под флюсом. Эти процессы применяют преимущественно при износах более 0,5 мм. Для восстановления поверхностей, работающих в условиях неподвижных сопряжений, широко распространена электроконтактная

приварка металлического слоя (ленты, проволоки). Преимущества электроконтактной приварки: незначительный нагрев деталей, возможность приварки металлического слоя различной твердости и износостойкости, уменьшение расхода наплавочных материалов, возможность регулирования толщины наносимого слоя в зависимости от износа, значительное повышение производительности и улучшение условий труда специалистов.

По сравнению с другими способами восстановления деталей наплавка позволяет получать покрытие на поверхности деталей необходимой толщины и заданного химического состава, высокой твердости и износостойкости.

В общем объеме работ по восстановлению деталей на ремонтных предприятиях различные способы восстановления составляют: наплавка под слоем флюса 32%, вибродуговая наплавка 12%, наплавка в среде углекислого газа 20%, наплавка порошковой проволокой без флюсовой или газовой защиты 10%, плазменная наплавка 1,5%, электроконтактное напекание 6%, гальванические способы 5%, электромеханическая обработка 1%, электрошлаковая наплавка 1,5%, заливка деталей жидким металлом 2% и восстановление деталей полимерами 5%.

Перед традиционными дуговыми способами наплавка порошковыми материалами имеет следующие преимущества [3]:

- высокую производительность и низкую энергоёмкость;
- позволяет получать наплавленные слои малой толщины с относительно низким тепловым воздействием на деталь;
- обеспечивает низкую степень перемешивания основного и наплавочного материала, что способствует сохранению в наплавленном металле исходных физико-механических свойств наплавочного материала;

Хорошие результаты при наплавке дает использование порошковой проволоки, в состав которой входят феррохром, ферротитан, ферромарганец, графитовый и железный порошки. Наплавку выполняют под слоем флюса или в среде защитного газа, но при введении в проволоку соответствующих компонентов возможна наплавка и без флюсовой или газовой защиты. Порошковую проволоку изготавливают на специальных станках методом волочения. Исходным материалом служит лента из низкоуглеродистой стали порошок, содержащий необходимые элементы.

При износе неподвижных поверхностей до 0,2 мм эффективно электромеханическое высаживание и выглаживание. Восстановление деталей этим способом не требует дополнительного материала, а при выглаживании поверхности происходит упрочнение поверхностного слоя, повышается износостойкость и усталостная прочность [4].

Гальванические процессы применяются только при массовом восстановлении однотипных деталей [5].

При небольших износах для восстановления шлицевых валов рекомендуется холодное пластическое деформирование.

При износе зубьев по толщине до 0,5 мм на их нерабочей наружной поверхности холодным пластическим деформированием на гидравлическом прессе с помощью шлиценакатной головки формируют технологическую канавку. Металл, вытесненный из канавки, заполняет боковую изношенную поверхность зуба и увеличивает наружный диаметр вала, обеспечивая минимально необходимый припуск для механической обработки шлицевой поверхности.

Упрочнение методами электроискровой обработки применяют для повышения: износостойкости и твердости поверхности деталей машин, работающих в условиях повышенных температур в инертных газах, жаростойкости и коррозионной стойкости поверхности, долговечности металлорежущего, деревообрабатывающего, слесарного и другого инструмента, а так же для создания шероховатости под последующее гальваническое покрытие, облегчения пайки обычным припоем труднопаяемых материалов (нанесение промежуточного слоя, например меди), увеличения размеров изношенных деталей при ремонте, изменения свойств поверхностей изделий из цветных металлов и инструментальных сталей.

Анализ дефектов деталей типа «вал» показал, что наиболее часто дефекты появляются на посадочных поверхностях под подшипники и резьбовых поверхностях. Было установлено, что из всей совокупности восстанавливаемых поверхностей валов 46% изнашиваются до 0,3 мм; 27% – от 0,3 до 0,6 мм; 19% – от 0,6 до 1,2 мм и 8% – более 1,2 мм. Анализ литературных источников показал, что наплавка с использованием порошковых износостойких наплавочных сплавов обеспечивает повышение срока службы деталей в 2–8 раз и нашла широкое применение при восстановлении деталей типа «вал».

## ЛИТЕРАТУРА

1. Власов, В.М. Техническое обслуживание и ремонт автомобилей / В.М. Власов, С.В. Жанказиев, С.М. Круглов [и др.]. – 2-е изд., стер. – М.: Издательский центр «Академия», 2004. – 480 с.
2. Молодык, Н.В., Восстановление деталей машин / Н.В. Молодык, А.С. Зенкин – М.: Машиностроение, 1989. – 480 с.
3. Шадричев, В.А. Ремонт автомобилей / В.А. Шадричев. – М.: Высшая школа, 1970. – 479 с.
4. Бурумкулов, Ф.Х. Восстановление и упрочнение деталей электроискровым методом / Ф.Х. Бурумкулов, А.В. Беляков, Л.М. Лельчук, [и др.] // Сварочное производство. – 1998.– № 2.– 192 с.
5. Новиков, А.Н. Технологические основы восстановления и упрочнения деталей сельскохозяйственной техники из алюминиевых сплавов электрохимическими способами: Дис. д-ра. техн. наук: / А.Н. Новиков – М., 1999. – 346 с.

УДК 620(071)

**ЦИНКОВАНИЕ ЭЛЕМЕНТОВ КУЗОВА АВТОМОБИЛЯ**

**А.В. ГУЩА, А.Т. БЕЙТУЛЛАЕВ**  
(Представлено: В.В. КОСТРИЦКИЙ)

*Сталь является самым популярным материалом для изготовления кузовов на сегодняшний день. Сталь легко подвергается коррозионному воздействию и производители применяют прием оцинкование деталей. И нам стало интересно, возможно ли провести цинкование в домашних условиях.*

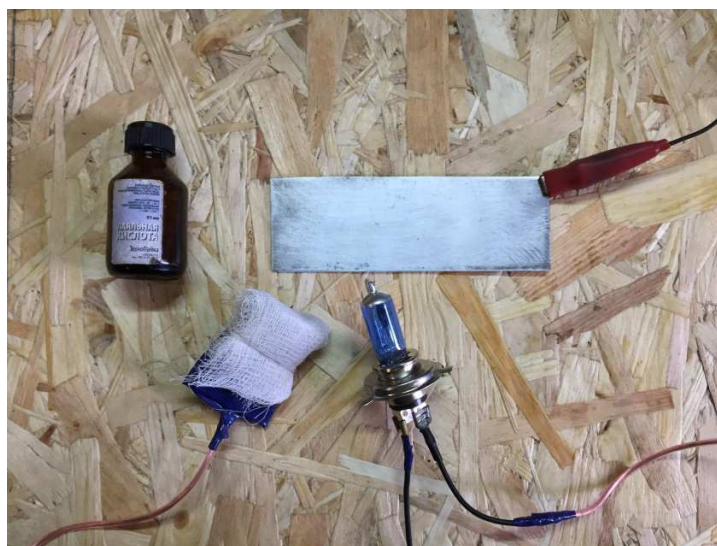
Для начало разберемся что такое цинкование. Цинкование – покрытие металла слоем цинка для защиты от коррозии. Метод защиты основан на следующем принципе. Большинство металлов (например, цинк, олово, алюминий) окисляются на воздухе, при этом на поверхности металла образуется плотная защитная плёнка из соединений окисленного металла. Эта плёнка препятствует проникновению кислорода вглубь металла и таким образом останавливает дальнейшее окисление металла. Толщина цинкового слоя зависит от температуры и продолжительности процесса цинкования и колеблется от 6 мкм для гальванического цинкования до 1,5 мм.

Существует несколько способов цинкование: горячее, холодное, гальваническое, термодиффузное и газодинамическое цинкование.

Чаще всего применяют горячее цинкование, принцип заключается в следующем. Необходимое изделие из металла опускают в емкость, где находится расплавленный цинк температурой около 460 градусов Цельсия. На момент процесса оцинковки происходит высокая степень соединения между непосредственно цинком и металлическим предметом.

Как это сделали мы. Наш способ цинкования относится к гальваническому. Гальваническое цинкование представляет собой электролитический метод нанесения тонкого защитного слоя цинка на изделие, поверхность которого должна быть предварительно подготовлена. Этот способ позволяет получать покрытия толщиной 5–40 мкм.

Нам понадобилось: батарейка, лампочка, провода, бинт, паяльная кислота, нить и две железные пластины 0,7 мм. Все работы проводим в защитных очках, респираторе и перчатках. Разбираем батарейку так, чтобы остался один корпус, т.к. он цинковый. На лампочке находим контакты массы и дальнего или ближнего света и подключаем к корпусу батарейки. Затем хорошо изолируем и наматываем бинт. Конечная схема подключения на лампочку подаем +, массу на деталь и масса от лампочки на батарейку (рис. 1).



**Рисунок 1. – Конечная схема подключения**

Бинт хорошенько пропитываем паяльной кислотой, подключаем схему, даем немного прогреться, и цинкуем пластину с двух сторон. Затем покрываем 2/3 части пластин грунтом и 1/3 краской и лаком (рис. 2).

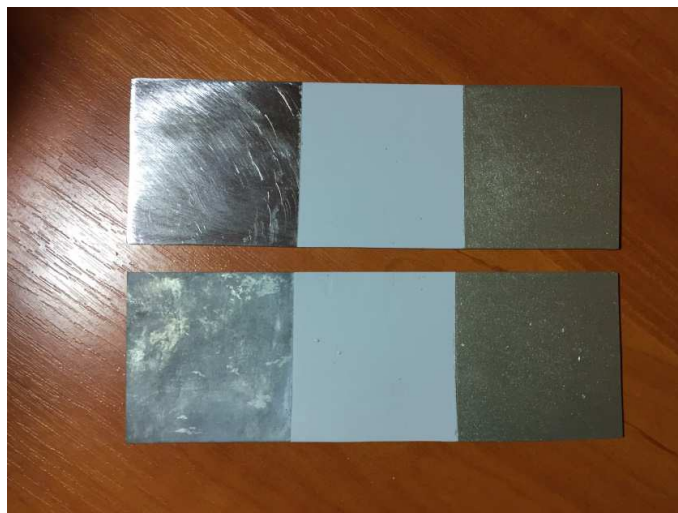


Рисунок 2. – Готовые пластины

Пластины мы положили на улице и каждый день поливали водой, первую ржавчину мы заметили через 4 дня. Далее так же поливали водой и наблюдали. Для ускорения процесса поместили пластины в агрессивную среду, соль с водой, и через 3 дня ржавчины стало намного больше. На данный момент пластины выглядят вот так (рис. 3).



Рисунок 3. – Распространение коррозии

Соблюдая правила безопасной работы с химическими растворами и электричеством, вы без проблем сможете выполнить оцинковку кузова своего автомобиля без помощи мастеров из СТО. Это значит, что защитить автомобиль от коррозии и ржавчины в гаражных условиях реально и несложно. Это существенно повысит качество ремонта и продлит жизнь кузова автомобиля. Такой метод цинкования машины некоторые хоть и называют «кустарным», но это действительно проверенный, самый простой и главное – эффективный способ борьбы с коррозией.

#### ЛИТЕРАТУРА

1. История транспорта. М: Махаон, 2007 г. – 127 с.
2. Гальванические покрытия в машиностроении. Справочник. Под ред. М.А. Шлугера, Л.Д.Тока. – М.: Машиностроение, 1985: Том 1, – 240 с.
3. Лайнер В.И., Гальванические покрытия лёгких сплавов. – М.: Metallurgizdat, 1959. – 138 с.
4. Хитрюк, В.А. Автомобильные эксплуатационные материалы: учеб. пособие / В.А. Хитрюк, А.К. Трубилов. – Минск : РИПО, 2013. – 323 с.

УДК 620(071)

## СОВРЕМЕННЫЕ КУЗОВНЫЕ МАТЕРИАЛЫ

**А.В. ГУЩА, А.Т. БЕЙТУЛЛАЕВ**  
(Представлено: В.В. КОСТРИЦКИЙ)

Кузов является одной из самых наиважнейших деталей автомобиля. В его основные качества в первую очередь должны входить безопасность, прочность, относительная при этом дешевизна, но в тоже время он должен быть оптимально удобным для всех пассажиров салона авто и отличаться стилем и дизайном. Между производителями нет единого мнения, какой из кузовных материалов наиболее лучше подходит для производства.

Мы расскажем вам о современных кузовных материалах и рассмотрим их плюсы и минусы.

**Стальной кузов.** Стальной кузов может быть различной вариантности сплава, что дает совершенно непохожие свойства его разновидностям. Например, низкоуглеродистая листовая сталь толщиной 0,6–2,5 мм. Листовая сталь обладает отличной пластичностью она же и позволяет производить из себя наружные панели деталей кузова, которые порой могут иметь довольно необычную и сложную форму. Логично, что высокопрочные сорта обладают изрядной энергоемкостью и отличной прочностью, поэтому этот вид стали применяют в производстве силовых деталей кузова (рис. 1).



Рисунок 1. – Стальной кузов

При всех этих плюсах недостатки у стали все же имеются и существенные. Так, например, неудобно то, что стальные детали имеют не малый вес, а также подвержены коррозионным процессам, что вынуждает производителей использовать приемы оцинковки стальных деталей и параллельно искать альтернативные варианты кузовных материалов.

**Алюминиевый кузов.** Сегодня все чаще можно услышать об использовании в производстве кузовов для авто такого материала как алюминий, точнее его сплавы с медью, магнием, марганцем, кремнием и цинком. Этот металл, не подвержен образованию ржавчины на деталях корпуса, а сам алюминиевый кузов при такой же прочности и жесткости весит в 2 раза меньше, чем его стальной собрат.

При всех своих качествах у алюминия имеется весомый недостаток – это хорошая проводимость шума и вибрации. Поэтому автопроизводителям приходится усиливать кузов противозвуковой изоляцией, что, в конечном счете, приводит к удорожанию машины, да и сам металл стоит дороже стали.

**Пластиковый кузов.** Большой популярностью пользуются полиуретан, поливинилхлорид, полипропилен, АБС-пластик и стеклопластик. Пластик легче даже вышеупомянутого алюминия, ему можно придать любую замысловатую форму, да и покраска его обходится намного дешевле, ведь провести ее можно уже на стадии производства, используя различные химические добавки. Этот материал уж точно не знает, что такое коррозия. Но недостатков у пластика гораздо больше и они довольно значимые.



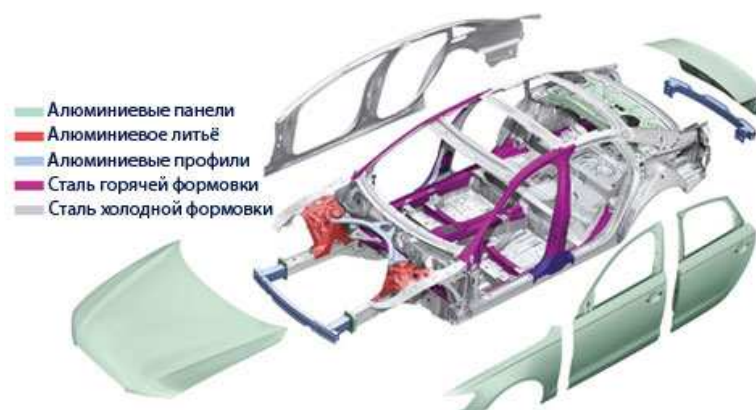


Рисунок 2. – Алюминиевый кузов

Свойства пластика меняются под влиянием различных температур – мороз делает пластик более хрупким, а жара размягчает этот материал. По этим причинам и ряду других из пластика нельзя изготавливать те детали, на которые оказываются довольно высокие силовые нагрузки, ремонту некоторые пластиковые детали и вовсе не поддаются, и требуют полной своей замены. Именно это привело к тому, что на сегодняшний день из пластика изготавливают лишь навесы, бампера и крылья.

**Композитный кузов.** Еще одним видом материала для изготовления кузова являются композитные материалы. Это «гибридный» материал, получаемый из нескольких соединенных вместе. Такое производство делает композитный кузов оптимальным по качествам, так как в нем соединяется все лучшее от каждого компонента. Кроме того, композитные материалы более долговечны, из них можно изготавливать самые крупные и сплошные детали, что, несомненно, упрощает само производство. К композитным материалам относится, например, углеволокно, которое, кстати, используется в производстве чаще всего. Из углеволокна изготавливают остовы к кузовам для суперкаров. К минусам данного материала можно отнести трудоемкость при его использовании в автомобилестроении. Иногда даже необходим ручной труд, что, конечно, в итоге сказывается на цене. Еще один недостаток – это практически невозможность восстановления деталей из углепластика после деформации при авариях. Все это способствует тому, что массово автомобили в углепластиковом кузове практически не выпускаются.

У каждого типа кузовов есть свои достоинства и недостатки. Тут уж все зависит от вкусов потребителей, то есть нас с вами.

#### ЛИТЕРАТУРА

1. Карл Дамшен. «Ремонт автомобильных кузовов». Серия «Автомеханик». Сокр. пер. с нем. В.С. Тулова под ред. А.Ф. Синельникова. - М.: ООО «Книжное издательство «За рулём», 2007. – 240 с.: ил.
2. Гордиенко В.Н. Ремонт кузовов отечественных легковых автомобилей. М.: АТЛАС-ПРЕСС, 2006 – 256 с.: ил.
3. Вишневецкий Ю.Т. Кузовные работы: Учебное пособие. – М.: Дашков и Ко, 2006. – 304 с.
4. Костенко, В.И. Эксплуатационные материалы ( для автомобильного транспорта) : учеб. Пособие / В.И. Костенко [и др.]. Санкт-Петербург : Изд-во СЗТУ, 2005. – 165 с.

УДК 69:658.26

**ВЛИЯНИЕ ВЕЛИЧИНЫ РАЗНОСТЕННОСТИ ТРУБ  
НА ТЕХНОЛОГИЧЕСКИЕ СВОЙСТВА ТРУБОПРОВОДОВ****Д.О. РОДУЛЕВИЧ***(Представлено: канд. техн. наук, доц. С.В. ПИЛИПЕНКО)*

*Выполнен анализ степени влияния величины разностенности и величина овальности трубы на величину допустимого давления при прокачке жидкости через трубопровод. Выполнены расчеты, позволившие оценить влияние толщины стенки и овальности трубы на коэффициент запаса прочности трубопровода и массу его погонного метра. Сделан вывод о возможности эксплуатации трубопровода с трубами, имеющими колебания величины толщины стенки в нижней части допуска (в пределах ГОСТа).*

**Введение.** При проектировании соединительных трубопроводов в машинах и агрегатах принимают, что трубы имеют идеально круглое поперечное сечение и постоянную толщину стенки [1]. Однако, на практике это не так, величина толщины стенки каждой трубы не постоянная, она колеблется в определенных пределах (ограниченных величиной допуска, указанного в стандарте на данный вид прокатной продукции) [2, 3]. Эти колебания наблюдаются как в поперечном сечении, так и вдоль трубы. Чем меньше допуски, тем, как правило, стоимость трубы выше. Изменение геометрических параметров трубы оказывают определенное влияние на ее эксплуатационные характеристики.

**Целью статьи.** Является анализ влияния величины разностенности трубопровода на и его способность сопротивляться внутреннему давлению.

**Анализ исследований.** Виды и функции металлических трубопроводов в автомобиле.

В конструкции автомобилей с двигателем внутреннего сгорания применяются следующие трубопроводы:

- впускной трубопровод;
- выпускной трубопровод;
- соединительный трубопровод ГБО (Газобаллонного оборудования);
- трубопроводы тормозных систем и т.д.

Впускной трубопровод. Главной функцией впускного трубопровода является подвод горючей смеси к цилиндрам, а также подогрев горючей смеси посредством тепла охлаждающей жидкости, которая циркулирует через рубашку трубы. Не маловажным фактором для впускного трубопровода является равномерное распределение горючей смеси к отдельным цилиндрам. Если речь идет о металлическом впускном трубопроводе, то его изготавливают из алюминия [4], давление в этом трубопроводе колеблется в пределах 20–35 кПа.

Выпускной трубопровод. Данный трубопровод служит для отвода отработавших газов из цилиндров. Выпускной трубопровод должен обеспечивать лучшую очистку цилиндров от отработавших газов, оказывая минимальное сопротивление перемещению газов и обеспечивая большее наполнение. В основном выпускной трубопровод изготавливают из чугуна.

Соединительный трубопровод ГБО. Функционал данного трубопровода представляет собой безопасную подачу КППГ (компримированного природного газа) под высоким давлением из баллонов к двигателю внутреннего сгорания. Соединительный трубопровод изготавливают из бесшовной холоднокатаной трубы из стали марки Сталь 20. Трубопроводы для систем питания ГСН изготавливают из медных трубок, давление газа в них колеблется в пределах 1,0–1,2 МПа [5].

Трубопроводы тормозных систем в автомобилях предназначены для передачи усилия от педали к основному цилиндру за счет тормозной жидкости. Трубки подведены к суппортам и соединяются с ними конусными муфтами. При эксплуатации автомобиля тормозные трубки переносят высокие нагрузки – давление внутри них при обычном торможении составляет около 60 бар (6 МПа). Если машина нагружена и движется со скоростью 100 км/ч, рабочее давление повышается до 100 бар (10 МПа), а при экстренном торможении оно достигает 200–300 бар (20–30 МПа.) [6–7].

На рисунке 1 представлено распределение толщины стенки по периметру трубы на примере волочения толстостенной трубы с размерами 25x2,5 мм через 21,4x1,7 мм до тонкостенной трубы 20x1 мм [8]. На данных графиках, на примере изменения геометрических параметров поперечного сечения при роликовом волочении одной трубы, представлены два возможных вида поперечной разностенности – эксцентричная и симметричная. Большинство труб имеют одну из этих видов разностенностей [9].

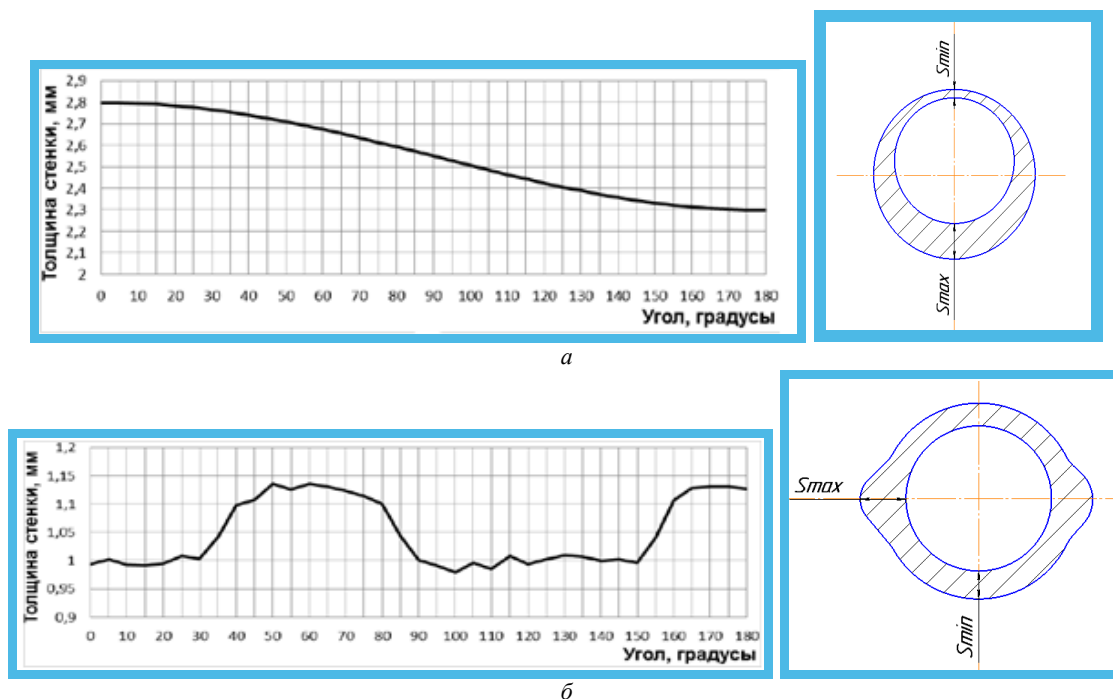


Рисунок 1 – Распределение толщины стенки по периметру поперечных сечений [8]:  
 а – труба с эксцентричной разностенностью; б – труба с симметричной разностенностью

Допустимое рабочее давление на участок трубопровода без учета дефектов и разностенности трубы можно рассчитать по соответствующей формуле [10]:

$$P_{доп} = \frac{2 \cdot S_{ном} \cdot \sigma_{\sigma} \cdot m}{n \cdot D_{вн} \cdot K_1 \cdot K_{назн.}} \text{ (МПа)}, \tag{1}$$

- где  $S_{ном}$  – номинальная толщина стенки трубы, мм;  
 $\sigma_{\sigma}$  – временное сопротивление разрыву металла трубы, МПа;  
 $m$  – коэффициент категоричности трубопровода;  
 $n$  – коэффициент надежности по нагрузке;  
 $D_{вн}$  – внутренний диаметр трубопровода, мм;  
 $K_1$  – коэффициент надежности по материалу;  
 $K_{назн.}$  – коэффициент надежности по назначению трубопровода.

**Основной материал.** Наиболее нагруженным является трубопровод тормозной системы. Для анализа влияния величины разностенности трубопровода на и его способность сопротивляться внутреннему давлению выбираем трубу бесшовную холоднокатаную согласно ГОСТ 617-2006 «Трубы медные и латунные круглого сечения общего назначения. Технические условия». Материал медной трубы для тормозной системы (М1) имеет следующие механические характеристики:  $\sigma_{\sigma} = 270$  МПа (трубы холоднокатаные, полутвердое состояние материала);  $\sigma_T = 180$  МПа [11, 12]. Для анализа выбран труба диаметром 4,76 (с допуском: +0; -0,15 мм) и толщиной стенки 0,8 (с допуском: ±0,08 мм, или ±0,07 для повышенной точности). Во время расчетов, значение всех коэффициентов в формуле (1) были приравнены к единице. Расчеты проводились в одном из математических редакторов. Результаты расчетов показаны в таблице 1

Таблица 1. – Влияние точности геометрических характеристик поперечного сечения трубы на допустимое давление в медной трубе 4,76x0,8 мм

$S_{max}$		$S_{min}$	
$D_{внутр.max}$	$D_{внутр.min}$	$D_{внутр.max}$	$D_{внутр.min}$
3	2,9	3,32	3,22
$P_{доп}$ (МПа)			
158,4	163,8621	117,1084	120,7453
Запас прочности (при $P_{раб.} = 50$ МПа)			
3,2	3,3	2,3	2,4

Как видно из таблицы, коэффициент запаса прочности при минимальной толщине стенки и максимальном значении диаметра трубы минимальный (2,3), но это вполне достаточно для работы трубопровода. При минимальном значении диаметра трубы и максимальной толщине стенки коэффициент запаса прочности принимает наибольшее значение (3,3). В целом, трубы с рядовыми допусками на величину разностенности вполне удовлетворяют требованиям для трубопроводов тормозных систем и заменять более дорогие, высокоточные трубы не имеет смысла.

Так как геометрические характеристики поперечного сечения напрямую влияют на массу труб, целесообразно рассчитать экономию материала при изготовлении 1000 м трубы.

Массу трубы можно рассчитать по следующей формуле

$$m = \rho * V, \text{ кг} \quad (2)$$

где  $\rho$  – плотность материала трубы, кг/м<sup>3</sup>;  
V – объём материала трубы, м<sup>3</sup>.  
Объём материала трубы

$$V = S * L, \text{ м}^3 \quad (3)$$

где S – площадь поперечного сечения трубы, мм<sup>2</sup>;  
L – длина трубы.  
Площадь поперечного сечения трубы

$$S = \frac{\pi}{4} (D^2 - D_{\text{вн}}^2), \text{ мм}^2 \quad (4)$$

где D – наружный диаметр трубы, мм;  
D<sub>вн</sub> – внутренний диаметр трубы, мм.

Геометрические характеристики поперечного сечения исследуемой трубы приведены в таблице 1. Во время расчётов значение длины трубы принимается 1 м. Расчёты проводились в одном из математических редакторов. Результаты расчётов показаны в таблице 2.

Таблица 2. – Влияние точности геометрических характеристик поперечного сечения трубы на массу в медном трубопроводе 4,76x0,8 мм

$S_{\text{max}}$		$S_{\text{min}}$	
$D_{\text{внутр. max}}$	$D_{\text{внутр. min}}$	$D_{\text{внутр. max}}$	$D_{\text{внутр. min}}$
3	2,9	3,32	3,22
$m$ (кг)			
0,095896	0,093425	0,081696	0,079674

**Выводы.** Трубы, имеющие колебания толщины стенки в нижней части допуска (Согласно ГОСТ, от  $S_{\text{среднее}}$  до  $S_{\text{min}}$ ) имеют достаточную степень запаса прочности для безопасной эксплуатации автомобильных трубопроводов. Из-за уменьшения массы труб возможна экономия дорогостоящего материала труб. Как видно из таблицы 2, масса метра трубы при минимальной толщине стенки и минимальном диаметре принимает минимальное значение (0,079647 кг). А разность между максимальным и минимальным значениями массы трубы составляет 0,016222 кг, что при изготовлении 1000 м труб даёт экономию материала в 16,222 кг. Согласно таблице 1 коэффициент запаса прочности трубы с минимальной толщиной стенки и минимальным диаметром (а, следовательно, и с минимальной массой) равен 2,4, что является оптимальным вариантом для производства трубопроводов тормозных систем автомобилей.

#### ЛИТЕРАТУРА

1. Куликов Ю. А. Динамика трубопроводов летательных аппаратов: Дис. д-ра техн. наук: 05.07.03: Йошкар-Ола, 1995. – 282 с.
2. Пилипенко С.В. Исследование изменения разностенности труб в ходе прокатки на стане ХПТ / С.В. Пилипенко // Сталь, №3 – 2016 – 32–37 с.
3. Григоренко В.У. Исследование изменения разностенности холоднокатаных труб / В.У. Григоренко, С.В. Пилипенко // Сталь, №9 – 2008 – 62–63 с.

4. <https://ustroistvo-avtomobilya.ru/sistemy-vpryska/vpuskny-e-truboprovody/>
5. <https://auto.kombat.com.ua/reduktoryi-vyisokogo-davleniya-gazoballonного-oborudovaniya-gbo-avtomobilye/>
6. <http://www.braz.by/reverse/brake/brake-pipes/?of=s10136>
7. <http://systemsauto.ru/brake/brake.html>
8. Вагапов Е. Н. Разработка математической модели роликового волочения труб на длинной оправке с целью прогнозирования точности: Дис. канд техн. наук: 05.02.09: Екатеринбург, 2012. – 162 с.
9. Столетний М.Ф. Точность труб / М.Ф. Столетний, Е.Д. Клемперт – М.: Metallurgy, 1975 – 239 с.
10. Абрамов И.П. Расчет максимально допустимого рабочего давления при длительной эксплуатации магистральных нефтепроводов / И.П. Абрамов, И.Ю. Подалов // Записки Горного института. Т.167. Часть 2 – 2006. – 184–185 с.
11. ГОСТ 617-2006. Трубы медные и латунные круглого сечения общего назначения. Технические условия. М., 2008. – 34 с.
12. <http://docs.cntd.ru/document/gost-617-2006>

УДК 621.01/004.896

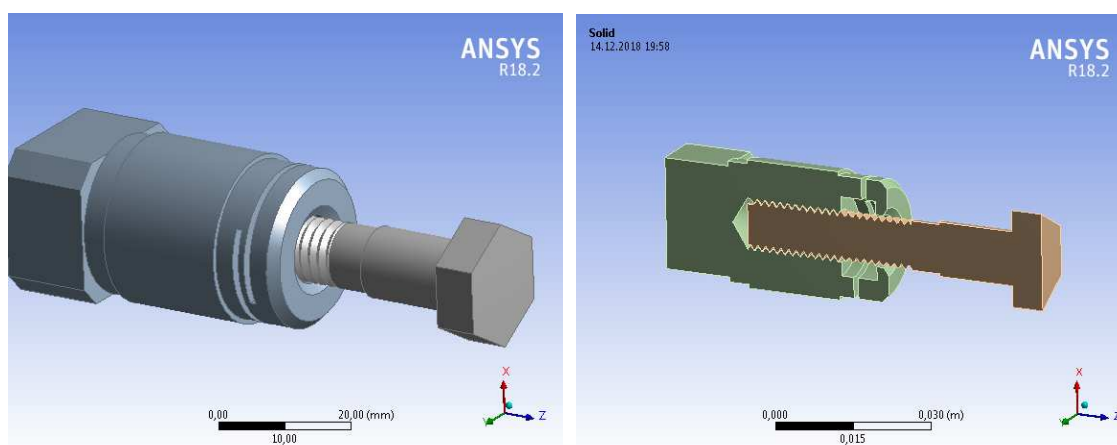
**ОПТИМИЗАЦИЯ ГЕОМЕТРИЧЕСКИХ ПАРАМЕТРОВ  
РАСТОЧНЫХ ИНСТРУМЕНТОВ НА ОСНОВЕ МОДЕЛИРОВАНИЯ  
РЕЗЬБОВЫХ СОЕДИНЕНИЙ****Е.М. ТИХОН***(Представлено: д-р техн. наук, проф. Н.Н. ПОПОК)*

*Рассмотрена система блочно-модульных расточных режущих инструментов с микрометрической настройкой лезвий, произведено моделирование резьбового соединения винт-втулка проанализирована модель напряжений и перемещений в резьбовом соединении, разработана математическая модель расчета напряжений и перемещений в блочно-модульных расточных режущих инструментах, даны рекомендации по проектированию высокоточных расточных головок.*

Точность сборных режущих инструментов с регулируемыми резцовыми вставками (блоками), например, расточных режущих инструментов зависит от точности исполнения резьбового соединения «винт-гайка». Для повышения точности этого резьбового соединения чаще всего используется разрезная гайка (втулка), которая позволяет регулировать зазоры в резьбе и тем самым, повышать точность перемещения резцовых вставок [1].

Как правило, соединение винта с втулкой происходит в сжатом состоянии втулки. Приведение резьбового соединения в рабочее положение производится путем снятия напряжения сжатия, в результате которого происходит компенсация зазоров в резьбовом соединении [2]. При этом ставится задача по выбору размеров нарезаемых пазов во втулке – глубины, ширины, шага пазов и других параметров, обеспечивающих требуемую точность перемещений.

Для решения этой задачи произведено моделирование резьбового соединения винт-втулка в программной среде «ANSYS» (рис. 1).

**Рисунок 1. – Трехмерная модель втулки и винта**

Моделируется резьба М12 с шагом 0,5 и 1 мм, выполненная с точностью 6g. Глубина прорезаемого паза – 18 и 21 мм, ширина паза 1 и 1,5 мм, шаг паза – 2 и 2,5 мм (ширина выступа 1 и 1,5 мм). Составляется план эксперимента (моделирования), включающий три фактора – глубину, ширину, шаг паза во втулке, варьируемые на двух уровнях – максимальном и минимальном их значениях. При изменении осевой нагрузки на резьбу от 50 до 200Н рассматривается напряжение в резьбовом соединении и перемещения винта.

Цель эксперимента (моделирования) – определение предельной осевой нагрузки, при которой гарантируется минимальный зазор в резьбовом соединении и требуемая точность перемещений резцовой вставки.

Анализ результатов моделирования показывает (рис. 2), что с увеличением нагрузки величины перемещения и напряжения растут, причем наиболее приемлемыми для последующих исследований являются образцы под номера 1, 4, 5 и 8, так как не возникает слишком больших и слишком малых нагрузок. Из этих вариантов выбирается наиболее рациональный образец под номером 1, так как он имеет наибольшие перемещения и обеспечивает минимальные зазоры в резьбовом соединении при любом значении нагрузки.

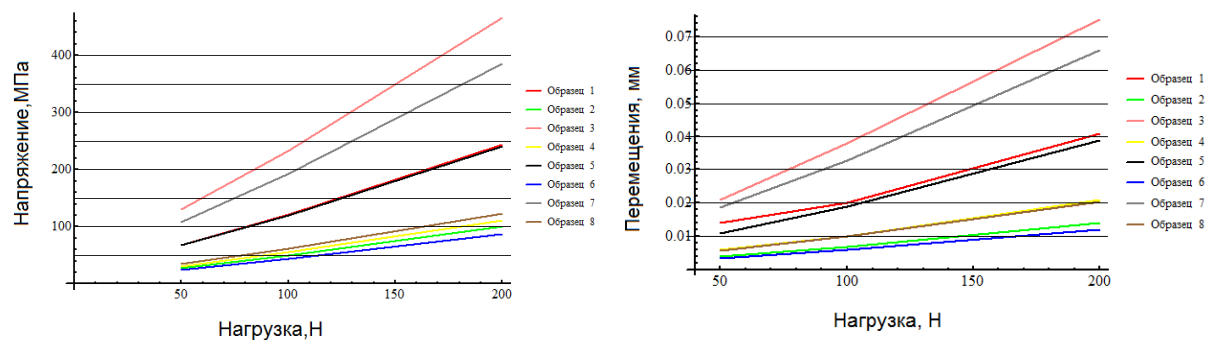


Рисунок 2. – Графики зависимости напряжений и перемещений от нагрузки

На основе результатов моделирования была разработана *математическая модель* расчета напряжений и перемещений в блочно-модульных расточных режущих инструментах, позволяющая оптимизировать геометрические параметры винта и разрезной втулки в механизмах настройки режущих лезвий (рис. 3).



Рисунок 3. – Схема физической модели перемещений

Формулы, описывающие данную модель:

$$y_1 = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3;$$

$$y_2 = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3;$$

где  $y_1 = \Delta$  – величина перемещения, мм;

$y_2 = \sigma$  – величина напряжения, МПа;

$x_1, x_2, x_3$  – факторы:

$x_1 = b$  – ширина паза, мм;

$x_2 = t$  – шаг паза, мм;

$x_3 = B$  – глубина паза, мм;

$a_1, a_2, a_3, b_1, b_2, b_3$  – коэффициенты регрессии.

Результаты расчетов величин перемещений и напряжений представлены в таблицах 1 и 2.

Статистическая обработка результатов расчетов позволила получить формулы в следующем виде:

$$\Delta = -0,0283 - 0,00124 \cdot b - 0,0175 \cdot t + 0,00325 \cdot B;$$

$$\sigma = 39,285 - 11,575 \cdot b - 75,95 \cdot t + 6,23 \cdot B.$$

Таблица 1. – Результаты расчетов напряжений

№	Параметры пазов			Напряжение $\sigma$ , МПа		
	Ширина b, мм	Шаг t, мм	Глубина B, мм	Нагрузка 50Н	Нагрузка 100Н	Нагрузка 200Н
1	0,5	1	18	55,98	101,74	213,51
2	0,5	2	18	25,57	50,2	100,43
3	0,5	1	21	89,33	192,36	421,6
4	0,5	2	21	46,45	65,37	110,77
5	1	1	18	53,01	120,48	240,96
6	1	2	18	24,5	43,76	87,51
7	1	1	21	83,08	162,42	344,84
8	1	2	21	34,27	61,42	122,77

Таблица 2. – Результаты расчетов перемещений

№	Параметры пазов			Перемещение $\Delta$ , мм		
	Ширина b, мм	Шаг t, мм	Глубина B, мм	Нагрузка 50Н	Нагрузка 100Н	Нагрузка 200Н
1	0,5	1	18	0,014	0,02	0,041
2	0,5	2	18	0,0039	0,007	0,014
3	0,5	1	21	0,021	0,038	0,075
4	0,5	2	21	0,0059	0,01	0,021
5	1	1	18	0,011	0,019	0,039
6	1	2	18	0,0034	0,006	0,012
7	1	1	21	0,0185	0,033	0,066
8	1	2	21	0,0057	0,01014	0,0203

В результате исследования установлены зависимости параметров пазов в разрезной втулке резьбового микрометрического соединения необходимых для достаточного предварительного натяга в резьбовом соединении, необходимого для выборки зазоров в соединении при его эксплуатации.

Получена математическая модель расчета напряжений и перемещений в блочно-модульных точных режущих инструментах, позволяющая оптимизировать геометрические параметры винта и разрезной втулки в механизмах настройки режущих лезвий.

#### ЛИТЕРАТУРА

1. Н.Н. Попок, Анализ тенденций проектирования инструментальных систем. Часть 1. Термины и системы обозначений режущих пластин, державок и модулей // Вестник ПГУ, сер. 1 – №3 – 2012. – 71–81 с.
2. Н.Н. Попок, Анализ тенденций проектирования инструментальных систем. Часть 2. Механизмы крепления модулей // Вестник ПГУ, сер. 1, №11, 2012. – 11–19 с.



## ХИМИЧЕСКИЕ ТЕХНОЛОГИИ

УДК 665.662.5

СВОЙСТВА ДЕПАРАФИНИЗИРОВАННЫХ ФРАКЦИЙ,  
ПОЛУЧЕННЫХ ИЗ ОСТАТОЧНОГО ПРОДУКТА ПРОЦЕССА «ЮНИКРЕКИНГ»

П.Ф. ГРИШИН

(Представлено: канд. техн. наук, доц. А.А. ЕРМАК)

Приведены результаты изучения свойств депарафинизированных фракций, полученных в результате фракционирования и последующей сольвентной депарафинизации остаточного продукта процесса «Юникрекинг». Показана возможность использования вышеуказанных фракций в качестве сырья для получения белых масел.

**Введение.** Остаточный продукт процесса гидрокрекинга вакуумных газойлей по технологии «Юникрекинг» фирмы UOP представляет собой сложную смесь высококипящих соединений, состоящую, преимущественно, из парафиновых и нафтеновых углеводородов, с низким содержанием серы и фактически полным отсутствием тяжелых металлов. В связи с этим он может быть использован не только при получении топлив, но и как сырьё для нефтехимической промышленности, производства высококачественных базовых масел, а также в качестве компонента пластичных смазок [1–4].

Одним из перспективных направлений комплексной переработки остаточного продукта процесса гидрокрекинга является получение белых масел [3; 5]. Белые масла представляют собой глубоко очищенные парафино-нафтеновые углеводороды с низким содержанием или полным отсутствием ароматических, серо- и азотсодержащих соединений. По степени очистки белые масла подразделяются на две группы: медицинские и технические. Первые находят применение в фармацевтической, косметической и пищевой промышленности. Технические белые масла используются в химической промышленности в качестве растворителей инициаторов и каталитических комплексов, пластификаторов полимеров, а также находят применение в текстильной, полиграфической, бумажной и других отраслях промышленности [5].

Целью настоящей работы является изучение свойств депарафинизированных фракций, выделенных из остаточного продукта процесса гидрокрекинга, а также оценка возможности их использования в качестве сырья для получения белых масел.

**Исследовательская часть.** В качестве объекта исследований были выбраны два образца, полученные путём фракционирования и сольвентной депарафинизации остаточного продукта процесса гидрокрекинга вакуумных газойлей. Кривые кипения исследуемых образцов представлены на рисунке 1.

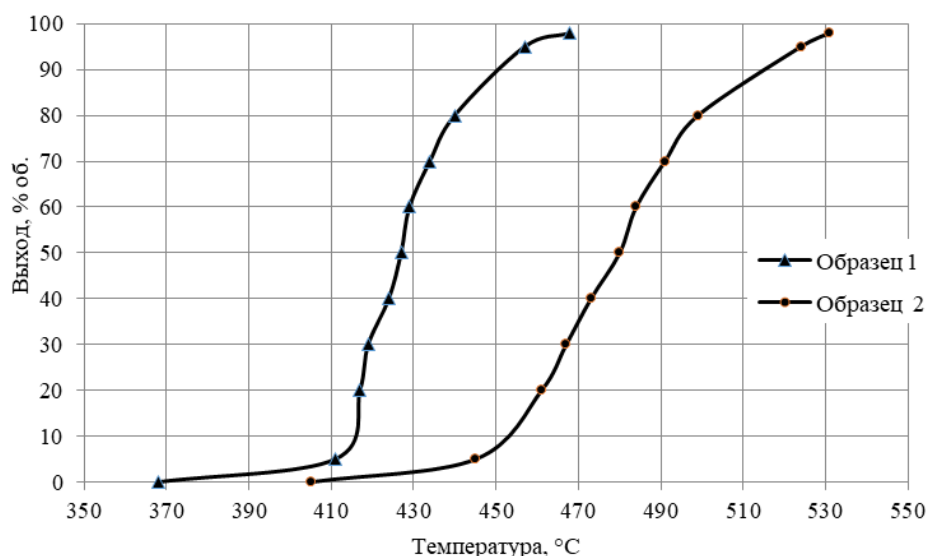


Рисунок 1. – Кривые кипения образцов депарафинизированных фракций

Физико-химические свойства исследуемых образцов представлены в таблице.

Таблица. – Физико-химические свойства образцов депарафинизированных фракций

Наименование показателя	Значение	
	Образец 1	Образец 2
Вязкость кинематическая при 40 °С, мм <sup>2</sup> /с	22,12	41,16
Вязкость кинематическая при 100 °С, мм <sup>2</sup> /с	4,541	6,915
Индекс вязкости	120	127
Показатель преломления при 20 °С	1,4652	1,4678
Плотность при 20 °С, кг/м <sup>3</sup>	839,3	843,8
Кислотное число, мг КОН / гр.	0,025	0,011
Температура текучести, °С	минус 18	минус 12
Температура вспышки, °С	224	248
Цвет, ед. ЦНТ	1	1,5
Групповой состав, % масс.:		
– парафино-нафтяные углеводороды	98,34	98,08
– 1 группа ароматических углеводородов	0,8	1,19
– 2 группа ароматических углеводородов	0,11	0,15
– 3 группа ароматических углеводородов	0,11	0,17
– 4 группа ароматических углеводородов	0,39	0,15
– смолы	0,25	0,26
Массовая доля серы, %	0,0054	0,0006

Исследуемые образцы представляют собой фракции с высоким содержанием парафино-нафтяных углеводородов, незначительным содержанием серы, ароматических углеводородов и смол. С повышением температуры кипения увеличивается кинематическая вязкость, показатель преломления, плотность, температура вспышки и застывания. При этом ухудшается цвет.

Для получения из исследуемых образцов белых масел необходимо удалить смолы и ароматические углеводороды. В связи с этим их необходимо подвергнуть доочистке, например, адсорбционной очистке с использованием активированной глины [6]. Регулировать свойства получаемых белых масел, в частности, кинематическую вязкость, можно путём компаундирования фракций. Кинематическая вязкость смесей исследуемых образцов представлена на рисунке 2.

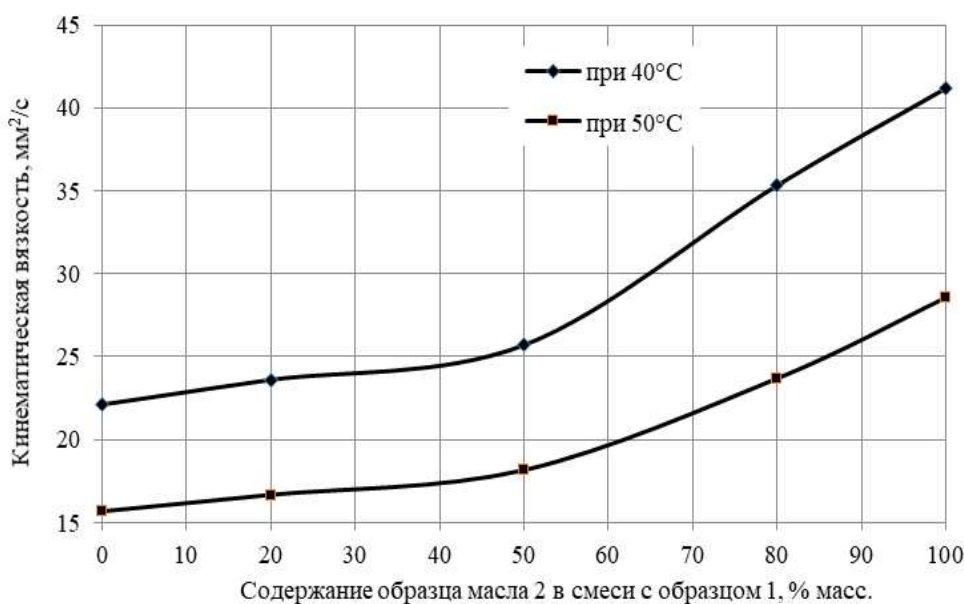


Рисунок 2. – Вязкость смесей исследуемых образцов

**Вывод.** Таким образом, путём фракционирования остаточного продукта процесса гидрокрекинга на несколько фракций с последующей их депарафинизацией и доочисткой, могут быть получены различные марки белых масел.

## ЛИТЕРАТУРА

1. Ермак, А.А. Свойства и перспективные направления переработки остаточного продукта процесса «Юникрекинг»./ А.А. Ермак, С.В. Покровская, И.В. Бурая и др.// Вестник Полоцкого государственного университета. Серия В, Промышленность. Прикладные науки. – 2015. – № 11. – 115–120 с. (<http://elib.psu.by:8080/handle/123456789/14387>)
2. Фамутдинов, Р.Н. Определение качества сырья для высокоиндексных масел из остатка гидрокрекинга/ Р.Н. Фамутдинов, С.В. Дезорцев // Башкирский химический журнал. – 2013. – Т. 20, № 4. – 37–39 с.
3. Sung, H.Y. VHVI Base Oils and White Oils from Fuels Hydrocracker Bottoms/ H.Y. Sung, S.H. Kwon, J.P. Andre// Asia Fuels and Lubricants Conference, Singapore, January 25–28, 2000. – p. 21
4. Булавка, Ю.А. Смазочная композиция на основе отходов нефтехимии для консервации техники [Электронный ресурс]. – Режим доступа:<http://elib.psu.by:8080/handle/123456789/21985> – Дата доступа: 24.09.2019.
5. Пиголева, И.В. Разработка технологии получения белых масел для эмульсионных вакцин из остаточных продуктов гидрокрекинга: диссертация ... кандидата технических наук: 05.17.07; [Место защиты: ФГАОУ ВО «Российский государственный университет нефти и газа имени И.М. Губкина»], 2018. – 156 с.
6. Ермак А.А., Гришин П.Ф., Артеменок Н.А. Перспективы применения активированной глины для доочистки остаточного продукта процесса «Юникрекинг» // Вестник Полоцкого государственного университета. Серия В, Промышленность. Прикладные науки. - 2019. - № 3. - 111–117 с.

УДК 665.761.6

## СПОСОБЫ ПОЛУЧЕНИЯ БЕЛЫХ МАСЕЛ

**П.Ф. ГРИШИН***(Представлено: канд. техн. наук, доц. А.А. ЕРМАК)*

*Рассмотрены способы получения белых масел из нефтяного сырья. Изучены свойства остаточного продукта процесса гидрокрекинга, очищенного активированной глиной. Показана возможность использования адсорбционного метода очистки при получении белых масел. Предложена блок-схема процесса очистки.*

**Введение.** Белые масла представляют собой высокоочищенные фракции нефтяных и парафиновых углеводородов без цвета, вкуса и запаха. Несмотря на малотоннажное производство, белые масла находят широкое применение в областях, где чистота продукта имеет первостепенное значение: производство парфюмерно-косметических изделий, медицина и ветеринария, синтез и переработка полимеров, производство резинотехнических изделий, текстильных материалов, упаковка продуктов питания и т.д. Обширный набор областей применения белых масел сопровождается различными специфическими требованиями к ним, как по физико-химическим показателям, так и степени очистки, что определяет выбор способа их получения.

Важно отметить, что на нефтеперерабатывающих мощностях ведущих зарубежных нефтяных компаний белые масла выпускают с применением гидрокаталитических процессов, в то время как отечественное производство представлено предприятиями, занимающимися производством медицинских субстанций и выпускающими вазелиновые и лампадные масла по технологии сернокислотной очистки [1].

**Основная часть.** Процессы сернокислотной очистки применяются для удаления непредельных, гетероорганических, смолисто-асфальтовых соединений и полициклических ароматических углеводородов из масляных фракций парафинового сырья для получения масел малотоннажного и специального ассортимента. При обработке серной кислотой перечисленные выше нежелательные компоненты масляных фракций подвергаются реакциям сульфирования с образованием кислых эфиров, сульфокислот, сульфонов, дисульфидов и т.д. В процессе протекают также побочные реакции, катализируемые серной кислотой: алкилирование аренов алкенами, полимеризация алкенов, поликонденсация аренов, смол и др. [2]. Сернокислотная очистка позволяет получать белые масла медицинского качества, однако процесс сопровождается выходом побочных продуктов, которые представляют опасность для экологии, а их утилизация затруднена. Поэтому в последние годы эти процессы заменяются на более совершенные и эффективные, в частности, экстракционные и гидрогенизационные процессы.

В отличие от сернокислотной очистки, в ходе которой происходит удаление нежелательных компонентов, гидрокаталитические процессы направлены на преобразование смол, гетероатомных соединений серы, кислорода, азота, ароматических и *n*-парафиновых углеводородов в более желательные соединения за счет каталитического взаимодействия с водородом. Данные технологии позволяют не только увеличить выход целевого продукта, но также обеспечить непрерывность процесса без образования побочных продуктов. Основными преимуществами гидрокаталитических процессов являются: высокий выход целевого продукта; отсутствие экологически опасных отходов; непрерывность процесса; высокое качество получаемых белых масел; гибкость процесса, обусловленная вариативностью ассортимента белых масел за счет изменения технологических параметров [1]. Для получения высокоиндексных базовых масел применяются различные комбинации гидрогенизационных процессов, включающие в себя гидрокрекинг, каталитическую гидроизодепарафинизацию, гидроочистку, гидроконверсию рафината, селективную очистку, гидроизомеризацию парафинов. Производство белого масла путем гидроочистки остаточного продукта гидрокрекинга или гидроконверсии рафината сопряжено с затратами на строительство новых установок и производство водорода. Оптимальным вариантом для выпуска малотоннажной продукции высокого качества является комбинация процесса гидрокрекинга вакуумных газойлей и последующей адсорбционной очистки остаточного продукта. На рисунке 1 представлена поточная схема, включающая в себя различные комбинации способов получения белых масел.

Одним из важнейших технических показателей у белого масла является процентное распределение углеводородов. Особые требования предъявляются к содержанию ароматических углеводородов, т.к. они токсичны. Технические белые масла должны содержать менее 7% масс., а медицинские белые масла менее 0,1% масс. ароматических углеводородов. Другим важнейшим показателем является цвет. в нормах ASTM D 1500. Как у технического, так и у медицинского белого масла этот показатель, определённый по ASTM D 1500, должен быть менее 0,5 [3].

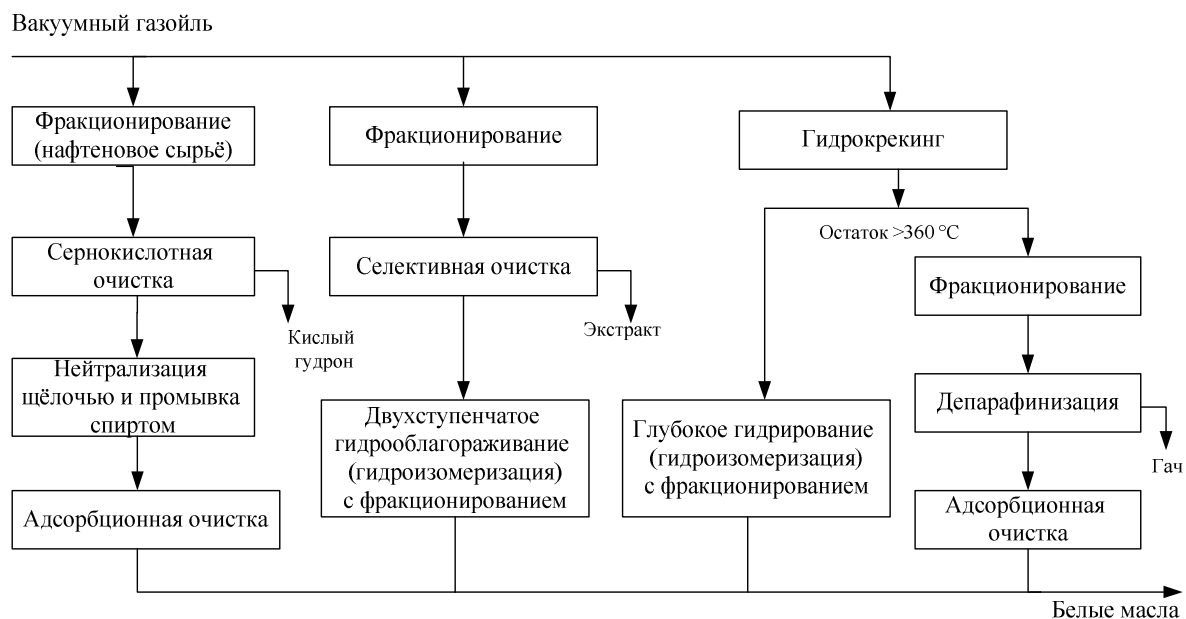


Рисунок 1. – Способы получения белых масел

Для достижения данных показателей адсорбционной очисткой остаточного продукта процесса гидрокрекинга вакуумного газойля необходим подбор эффективного адсорбента. В промышленности в качестве адсорбентов используются отбельные глины. Путем адсорбционной очистки можно существенно улучшить цвет и стабильность при хранении масел и жиров. Для очистки применяются разные классы отбельных глин.

Первая группа образует класс высокоактивных, основанных, главным образом, на монтмориллоните, отбельных глин НРВЕ (high performance bleaching earth). Эта группа включает, в частности, активированные кислотой монтмориллониты, причем кислотное активирование осуществляют дорогостоящим способом: путем dealюминирования необожженных глин концентрированными кислотами при высоких температурах. При таком способе получают отбельные глины с очень высокой удельной поверхностью и большим объемом пор. Недостатком этих высокоактивных отбельных глин является то, что из-за dealюминирования кислотами при получении скапливаются большие количества кислых, богатых солями сточных вод, которые можно обработать или утилизировать только путем дорогостоящих процессов.

Другая группа образует класс естественно активных глин. Эти отбельные глины натурального происхождения применяются для очистки жиров и масел уже сотни лет. Эти естественно активные системы (называемые также флоридин или фуллеровы земли) можно получить с очень малыми затратами. Однако они имеют очень незначительную отбеливающую силу, так что в большинстве своем они не подходят для очистки трудноотбеливаемых масел и жиров.

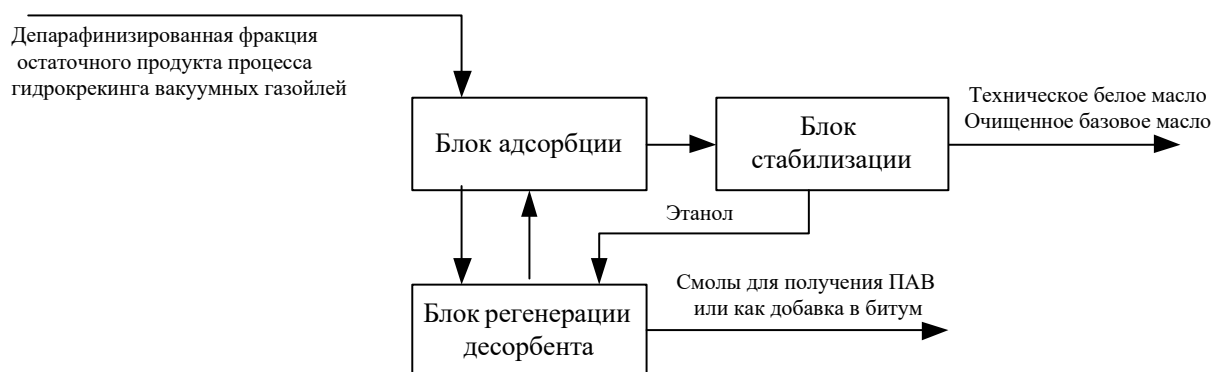
Компромисс между низкой стоимостью получения и приемлемой активностью представляет третий класс отбельных глин, так называемые поверхностно-активированные системы SMBE (surface modified bleaching earth; поверхностно-активированные отбельные глины). Здесь в естественно активную необожженную глину вводятся небольшие количества кислоты и этим достигается активирование "in situ". Для этого способа используются необожженные глины, содержащие аттапульгит и хормит. Они имеют очень высокую для природных необожженных глин удельную поверхность: от примерно 100 до 180 м<sup>2</sup>/г, и объем пор от примерно 0,2 до 0,35 мл/г. Так как соли, образованные при кислотном активировании, или непрореагировавшая часть кислот не вымываются, они остаются на продукте и по меньшей мере частично откладываются также в порах. Из-за этого активированные кислотами отбельные глины, как правило, не достигают такой же эффективности, какая достигается с высокоактивными отбельными глинами (НРВЕ), получаемыми dealюминированием кислотой. Однако простой способ получения делает возможным сравнительно экономичное производство, так как не накапливается никаких кислых сточных вод [4].

Проведенные исследования показали, что при адсорбционной очистке остаточного продукта процесса гидрокрекинга с помощью активной глины возможно получение белого масла. Свойства очищенного продукта представлены в таблице.

Таблица. – Свойства очищенного продукта

Показатель	Значение
Вязкость кинематическая при 40 °С, мм <sup>2</sup> /с	40,82
Вязкость кинематическая при 100 °С, мм <sup>2</sup> /с	6,88
Индекс вязкости	127
Показатель преломления при 50 °С	1,4558
Показатель преломления при 20 °С	1,4661
Цвет, ед. ЦНТ (ASTM D 1500)	0
Плотность при 20 °С, кг/м <sup>3</sup>	842,0
Кислотное число, мг КОН/гр.	0,0014
Групповой состав, % масс.:	
– парафино-нафтяные углеводороды	99,72
– 1 группа ароматических углеводородов	0,27
– 2 группа ароматических углеводородов	0
– 3 группа ароматических углеводородов	0
– 4 группа ароматических углеводородов	0
– смолы	0,01

Исходя из показателей качества полученного продукта, можно сделать вывод о пригодности данного масла для технических нужд. При проведении процесса адсорбции на активированной глине адсорбент осмолается и изменяет цвет с белого на угольно-черный цвет. В связи с этим существует необходимость дальнейшей промывки и регенерации адсорбента [5]. Принципиальная блок-схема установки адсорбционной очистки депарафинизированной фракции остаточного продукта процесса гидрокрекинга вакуумных газойлей представлена на рисунке 2.



**Рисунок 2. – Блок-схема установки адсорбционной очистки активированной глиной депарафинизированной фракции остаточного продукта процесса гидрокрекинга вакуумных газойлей**

Установка адсорбционной очистки активированной глиной депарафинизированной фракции остаточного продукта процесса гидрокрекинга вакуумных газойлей должна состоять из:

- блока адсорбции, включающего в себя несколько абсорберов работающих параллельно по схеме «адсорбция – десорбция – промывка»;
- блока регенерации десорбента, представляющего собой смесь этилового спирта с бензолом или толуолом, и промывочного этилового спирта. В состав блока входит две ректификационные колонны и ёмкости хранения десорбента и спирта;
- блока стабилизации очищенного масла, включающего в себя стабилизационную колонну и ёмкости хранения очищенного масла.

**Вывод.** Белые масла могут быть получены различными способами: сернокислотной очисткой нафтяного сырья, с использованием процессов селективной очистки и гидрооблагораживания масляных фракций, путём глубокого гидрирования остаточного продукта процесса гидрокрекинга. Одним из перспективных и относительно простых способов получения технических белых масел является адсорбционная очистка остаточного продукта процесса гидрокрекинга вакуумного газойля с применением ак-

тивных глин. Для оценки возможности использования очищенного продукта в медицинской и парфюмерной промышленности необходимы дальнейшие исследования.

## ЛИТЕРАТУРА

1. Пиголева, И.В. Разработка технологии получения белых масел для эмульсионных вакцин из остаточных продуктов гидрокрекинга: дис. ... канд. техн. наук : 05.17.07 / ФГАОУ ВО «Российский государственный университет нефти и газа имени И.М. Губкина». – 2018. – 156 с.
2. Виск В., Клюр А. Производство белого масла.// IV международный нефтяной конгресс. Том IV. Технология переработки нефти и сланцев. – М.: Государственно научно-техническое издательство нефтяной и горно-топливной литературы, 1956. – 265–286 с.
3. Использование белых масел и некоторые требования к ним.// [Электронный ресурс]. – Режим доступа: <http://autokomplekt.com/info/articles/the-use-of-white-oils-and-some-requirements/> – Дата доступа: 19.09.2019.
4. Патент RU 2379104. МПК В01J 20/12. Глины с высокой удельной поверхностью для получения отбельных глин, а также способ активирования этих глин. / Шурц Клаус (DE). Заявка №2007145413/15, Заявлено 08.06.2005; Опубл. 20.01.2010. Бюл. №2
5. Ермак А.А., Гришин П.Ф., Артеменок Н.А. Перспективы применения активированной глины для доочистки остаточного продукта процесса «Юникрекинг» // Вестник Полоцкого государственного университета. Серия В, Промышленность. Прикладные науки. – 2019. – № 3 . – 111–117 с.

УДК 661.715.6

## НАПРАВЛЕНИЯ ИСПОЛЬЗОВАНИЯ ТЯЖЕЛОЙ СМОЛЫ ПИРОЛИЗА

Е.А. ШУЛЬГА, Н.С. ВАШКОВА

(Представлено: канд. хим. наук, доц. С.Ф. ЯКУБОВСКИЙ)

*Увеличение объёмов производства этилена и других низших непредельных углеводородов термическим пиролизом углеводородного сырья вызывает проблему разработки эффективных схем переработки образующихся жидких продуктов пиролиза (смол), в особенности фракции выкипающей выше 180 °С – тяжёлой смолы пиролиза (ТСП).*

Установлено, что тяжёлые смолы образующиеся при пиролизе разного углеводородного сырья, имеют близкие характеристики – примерно одинаковый элементарный и групповой составы [1]. Основным компонентом этих углеводородов являются бициклические и трициклические ароматические углеводороды и углеводороды, содержащие большое число циклов.

Считается, что рациональное использование ТСП может осуществляться по двум вариантам – топливному и химическому. По первому варианту ТСП является компонентом котельного топлива, по второму ТСП служит ценным нефтехимическим сырьём для производства игольчатого кокса, технического углерода, сажи, волокнообразующих и электродных пеков, стеклоуглерода, компонентов лаков, пропиточных, строительных и других материалов [2].

Одним из направлений применения ТСП без разделения её на отдельные фракции является получение пластификатора бетонных смесей реакцией сульфирования концентрированной серной кислотой с последующей нейтрализацией полученной сульфомассы щелочными реагентами [3]. Пластификаторы позволяют разжижать бетонную смесь для удобства дальнейшей переработки. Применение пластификаторов довольно универсально: с их помощью можно повысить плотность (водонепроницаемость) и прочность бетона, уменьшив количество воды в смеси при сохранении её подвижности и получить бетон с низкой усадкой.

Из ТСП возможно также получение поверхностно-активных веществ. Реакция происходит при взаимодействии алкилпроизводных полициклических углеводородов, содержащихся в ТСП, с малеиновым ангидридом. Для этого смолу пиролиза вначале нагревают с малеиновым ангидридом, а потом полученный продукт сульфатируют концентрированной серной кислотой с последующей нейтрализацией сульфомассы щелочными агентами [4, 5].

Тяжёлая смола пиролиза используется в качестве одного из компонентов антисептика для пропитки древесины. Антисептические пропиточные составы используются для защиты древесины от био-разрушений [6].

Авторы работы [7] предложили использовать ТСП для получения товарного нефтяного нафталина. Для этой цели должна использоваться широкая нафталиновая фракция, выкипающая в пределах 205–230 °С.

Анализ работы установок пиролиза нефтяных фракций позволил авторам работы [8] выявить, что в бензиновой колонне может быть отобрана промежуточная фракция ТСП, которая содержит в зависимости от точки отбора от 30 до 55% масс. нафталина.

Найден способ получения нафталина без использования больших энерго- и ресурсозатратных процессов [9–11]. Данный способ позволяет получать нафталин более высокого качества, что положительно сказывается на использовании ТСП в качестве сырья для получения последнего.

В настоящее время актуальной проблемой является дефицит нефтяного игольчатого кокса, производство которого оказывает влияние на развитие электродной промышленности.

Игольчатый кокс используется для получения высококачественных графитовых электродов. Электроды должны иметь высокую механическую прочность, электропроводность, низкое содержание серы и низкий КТР.

Для получения игольчатого кокса из ТСП используется фракция выкипающая выше 230 °С. Полученный пек разбавляется растворителем, при температуре 120–140 °С достигается разделение фаз экстракта и рафината, удаление растворителя из экстракта и коксование экстракта при температуре 502 °С [12].

Нами предлагается три направления использования ТСП, приведенные на рисунке.

Проведённый краткий обзор направлений использования тяжёлой смолы пиролиза показывает, что данный продукт нефтепереработки имеет большой потенциал применения в нефтехимии, электродной промышленности и других областях. Использование данного ресурса является целесообразным, так как дальнейшая переработка тяжёлой смолы пиролиза становится рентабельной и эффективной.



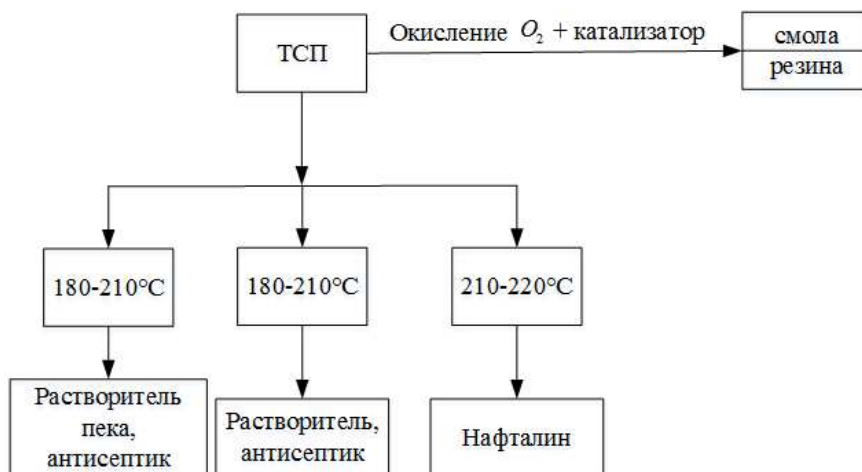


Рисунок. – Разгонка тяжёлой смолы пиролиза по Энглеру

## ЛИТЕРАТУРА

1. В.М. Капустин, М.Г. Рудин. Химия и технология переработки нефти. – М.: Химия, 2013 – 496 с.
2. Переработка жидких продуктов пиролиза / А.Д. Беренц [ и др.]. – М.: Химия, 1985 – 212 с.
3. Расширение сырьевой базы для производства суперпластификаторов / А.П. Шведов, С.Ф. Якубовский, А.В. Зубова // Вестник Полоцкого государственного университета – Новополоцк: ПГУ, 2007 – 68–72 с.
4. В.В. Сасковец, Влияние фракционного состава ТСП на свойства полученных ПАВ / В.В. Сасковец, Н.В. Кожарская, Е.Я. Гамбург // Нефтепереработка и нефтехимия. – 1992. – № 11.
5. Использование малеинового ангидрида при синтезе ПАВ на основе ТСП / В.В. Сасковец [ и др.] // Нефтепереработка и нефтехимия – 1993, № 4.
6. Патент № 2303522 – Антисептик нефтяной для пропитки древесины – жидкость товарная консервационная / Л.В. Долматов, А.Ф. Ахметов, С.Н. Караван, 2007 г.
7. Получение стандартного нафталина из тяжёлых жидких продуктов пиролиза бензиновых фракций без применения концентрированной  $H_2SO_4$  / В.М. Беднов [и др.] // Химия и химическая технология. – 1975. – № 9.
8. Изучение возможностей получения нафталина из вторичного нефтехимического сырья. Ресурсосберегающие и экологически чистые технологии / С.В. Покровская, С.М. Ткачёв, С.Ф. Якубовский. – Гродно, 1997 – 207 с.
9. Технология переработки тяжёлой смолы пиролиза углеводородного сырья в контексте импортозависимости страны / Ю.А. Булавка, С.Ф. Якубовский // Актуальные задачи нефтеперерабатывающего и нефтехимического комплекса. Материалы конференции РГУ нефти и газа им. И.М. Губкина. – 21–23 ноября 2018 г. – Москва, 25–27 с.
10. Инновационный подход к переработке тяжёлой смолы пиролиза углеводородного сырья / Ю.А. Булавка, С.Ф. Якубовский, С.С. Хохотов, В.А. Ляхович // Актуальные проблемы развития нефтегазового комплекса России: Материалы XII Всероссийской научно-технической конференции – Москва, 2018. – 209 с.
11. Получение товарных продуктов из тяжёлой смолы пиролиза / Ю.А. Булавка, С.Ф. Якубовский, В.А. Ляхович // Актуальные вопросы современного химического и биохимического материаловедения: материалы V Международной молодёжной научно-практической школы-конференции (г. Уфа, 4–5 июня 2018 г.) / отв. Ред. О.С. Куковинец. – Уфа: РИЦ БашГУ, 2018 – 54–57 с.
12. Перспектива расширения сырьевой базы для получения игольчатого кокса / И.Р. Хайрудинов, А.А. Тихонов, М.М. Ахметов // Башкирский химический журнал. – 2011. – №3.

УДК 004.622: 53.082.52

**УСТАНОВКА ДЛЯ ИССЛЕДОВАНИЯ ПАРАМЕТРОВ ИСТОЧНИКОВ  
ИСКУССТВЕННОГО СВЕТА****С.Ю. ЗМИТРОВИЧ***(Представлено: канд. физ.-мат. наук, доц. С.А. ВАБИЩЕВИЧ)*

*Представлена принципиальная схема работы автоматизированной установки, позволяющей анализировать физико-технические параметры источников искусственного света с помощью оптических датчиков. Разработанное устройство может быть использовано для организации автоматизированного тестового неразрушающего контроля на любом этапе производства и эксплуатации светотехнических устройств и приборов.*

*Ключевые слова: источники искусственного освещения, спектр света, фотометрические параметры, автоматизация эксперимента.*

**Введение.** Актуальной проблемой при подборе для использования в промышленных и бытовых условиях осветительных приборов является достоверный анализ технических параметров источников искусственного света (ИИС) с целью определения оптимальных режимов их эксплуатации. Известно, что отклонение физико-технических характеристик ИИС от регламентируемых норм или несоответствие заявленным характеристикам изделия может негативно сказаться на самочувствии и здоровье человека. С другой стороны, при некорректном подборе осветительных приборов создаются условия для возникновения короткого замыкания в электрической цепи и нарушаются условия пожарной безопасности.

Цель настоящей работы заключалась в создании установки на основе микроконтроллерного устройства для автоматического исследования и последующего анализа физико-технических характеристик и параметров источников искусственного света.

Актуальность работы заключается в автоматизации проведения исследований, что позволяет повысить достоверность экспериментальных данных за счет снижения величины случайной погрешности, сократить время измерений, расширить диапазон и повысить эффективность использования датчиковой аппаратуры [1,2].

**Принципиальная схема и принцип работы устройства.** Основными физико-техническими параметрами, исследуемыми в настоящей работе, являются:

- спектральная характеристика – параметр источника света, характеризующий его волновой состав;
- коэффициент пульсаций – характеристика, оценивающая относительные колебания освещённости;
- световой поток – мощность светового излучения;
- сила света – характеризует величину световой энергии, переносимой в некотором направлении в единицу времени;
- номинальная мощность – потребляемая мощность, требуемая для нормальной работы источника света;
- световая отдача лампы – оценочная характеристика, показывающая отношение светового потока к потребляемой мощности.

Основу автоматизированной установки для исследования параметров составляет микроконтроллерное устройство, которое может быть построено на базе микроконтроллеров, отладочных плат AVR или STM. Указанное устройство проводит измерения, используя собственный модуль аналого-цифрового преобразователя (АЦП), который позволяет конвертировать аналоговую величину в цифровой код. Программное обеспечение является инициатором обмена данными между персональным компьютером и микроконтроллером посредством виртуального COM порта на базе USB [3,4].

Структурная схема и графическое представление концепта установки представлены на рисунках 1 и 2 соответственно.

Принципиальный алгоритм работы системы может быть описан следующим образом:

- 1) ИИС 5 включается в патрон 2, питающие провода которого идут от ваттметра 7;
- 2) через программное обеспечение настраивается графическое изображение измеряемых величин и микроконтроллерное устройство 3 с подключёнными к нему оптоэлектронными датчиками 1, 4 и 6;
- 3) подключается электропитание ваттметра 7;
- 4) ваттметр 7 начинает считывать потребляемую электрическую энергию от ИИС 5;

- 5) свет, излучаемый от ИИС 5, падает на оптоэлектронные датчики 1, 4 и 6, которые, в свою очередь, начинают фиксировать измеряемые параметры светового потока, силы света, мерцаний ИИС, спектральную характеристику;
- 6) после проведенных испытаний происходит расчёт световой отдачи ИИС 5;
- 7) в автоматическом режиме проводится анализ полученных параметров, сравнение их с техническими характеристиками на ИИС 5 и регламентируемыми нормами на прибор.

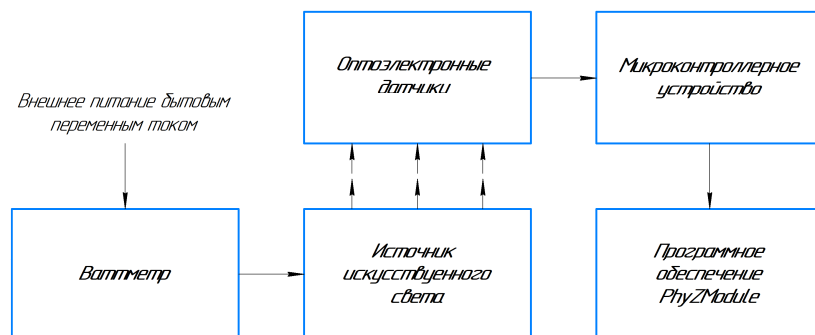


Рисунок 1. – Структурная схема установки для измерения параметров источников искусственного света

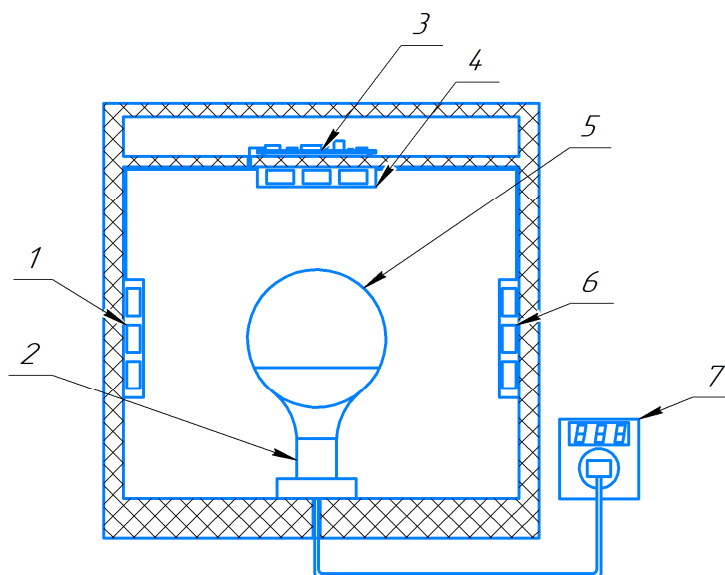


Рисунок 2. – Макет установки для измерения параметров источников искусственного света

**Заключение.** Представленная в работе работы автоматизированная установка позволяет с высокой точностью определять физико-технические параметры источников искусственного света с помощью оптических датчиков. Анализ осуществляемый после получения экспериментальных данных позволяет исследователю дать заключение о соответствии указанных параметров техническому паспорту прибора или устройства и регламентирующим эксплуатационным нормам, а также о пригодности использования того или иного источника искусственного света в определенных условиях.

Разработанное устройство может быть использовано для организации автоматизированного тестового неразрушающего контроля на любом этапе производства и эксплуатации светотехнических устройств и приборов.

#### ЛИТЕРАТУРА

1. Змитрович, С.Ю. Автоматизированная система физических измерений/ С.Ю.Змитрович, С.А.Вабищевич, Д.Н.Шабанов// Вестник Полоцкого государственного университета. – 2019. – №4. – С.45-49.
2. Змитрович, С.Ю., Вабищевич, С.А. Обработка аналоговых сигналов датчиковой аппаратуры. Информационно-коммуникационные технологии: достижения, проблемы, инновации (ИКТ-2018) [Элек-

тронный ресурс]: электронный сборник статей I международной научно-практической конференции, посвященной 50-летию Полоцкого государственного университета, Новополоцк, 14–15 июня 2018 г. / Полоцкий государственный университет. – Новополоцк, 2018. – 1 электрон. опт. диск (CD-ROM). С.272-275.

3. STMicroelectronics [Электронный ресурс] / STM32CubeMX for STM32 configuration and initialization C code generation – Режим доступа: <https://clck.ru/ERtM3>

4. STMicroelectronics [Электронный ресурс] / About ST – Режим доступа: [https://www.st.com/content/st\\_com/en/about/st\\_company\\_information/who-we-are.html](https://www.st.com/content/st_com/en/about/st_company_information/who-we-are.html)

УДК621.91.01

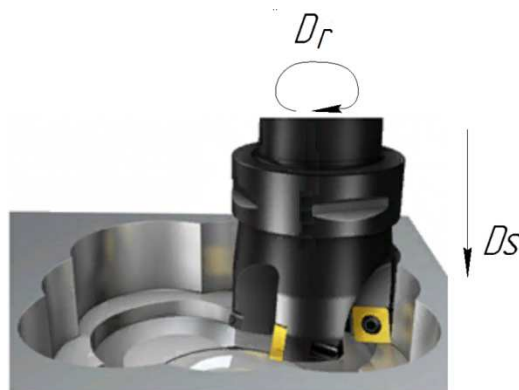
**СРАВНЕНИЕ ПЛУНЖЕРНОГО И КОНТУРНОГО ФРЕЗЕРОВАНИЯ**

**Е.О. ЖИХОРЕВ**

*(Представлено: д-р техн. наук, проф. Н.Н. ПОПОК)*

Являясь одним из наиболее широко применяемых способов механической обработки, фрезерование может классифицироваться по различным признакам: в зависимости от расположения шпинделя станка - горизонтальное и вертикальное, от типа фрезы - концевое, торцевое, фасонное и т.д., движения подачи фрезы - попутное и встречное. Кроме этих видов обработки существует еще плунжерное фрезерование. В настоящее время этот способ все чаще применяется для обработки поверхностей деталей.

Плунжерное фрезерование (рисунок 1) представляет собой вид обработки, при котором производится последовательное врезание концевой фрезы в обрабатываемый материал вдоль оси, при этом обработка ведется торцевой частью инструмента. В соответствии с ГОСТ 25761 фрезерование - лезвийная обработка с вращательным главным движением резания при постоянном радиусе его траектории, сообщаемым инструменту, и хотя бы одним движением подачи, направленным перпендикулярно оси главного движения резания [1]. В случае же плунжерного фрезерования рабочая подача осуществляется вдоль оси инструмента, что подпадает под определение осевой обработки согласно ГОСТ 25761. Из имеющихся видов осевой обработки плунжерное фрезерование больше соответствует цекованию.



**Рисунок 1. – Плунжерное фрезерование**

Плунжерное фрезерование широко применяется при обработке карманов в корпусных деталях. С помощью SolidCAM проведем сравнение производительности обработки при плунжерном и контурном фрезеровании [2,3]. Результаты сравнения этих видов обработки приведены в таблице.

Таблица. – Результаты расчетов

	Плунжерное фрезерование		Контурное фрезерование	
Режимы резания	рабочая подача (по оси Z) – 30 мм/мин величина перекрытия – 10...70% кол-во оборотов – 1600 об/мин подача для чистовой обработки стенок - 100 мм/мин		рабочая подача (по оси XY) – 100 мм/мин по оси Z – 30 мм/мин кол-во оборотов – 1600 об/мин	
Размер кармана (ДхШхГ)	35x15x40	50x25x40	35x15x40	50x25x40
Время обработки, мин	8,3 ÷ 25,2	33 ÷ 69,5	9,5	16,9

В зависимости от величины перекрытия (величины врезания инструмента в материал) и размеров кармана время обработки может существенно изменяться. Кроме того, у плунжерного фрезерования можно выделить и другие недостатки:

- необходимость получение предварительного отверстия для первого прохода фрезы, либо врезания по спирали или маятниковой подаче.
- большое количество остаточного металла, в связи с этим требуется чистовая обработка поверхности контурным фрезерованием. Также при плунжерном фрезеровании возникает проблема выбора диа-

метра фрезы: при небольшом диаметре высота гребешка  $H$  (рисунок 2) будет небольшой и надобность в полустиковой обработке исключается, однако вероятность поломки инструмента будет высока, а время обработки больше. С другой стороны, использование фрезы большого диаметра позволит уменьшить время обработки и исключить возможность выхода инструмента из строя, однако придется вводить в управляющую программу полустиковой проход из-за большого размера остаточного материала.

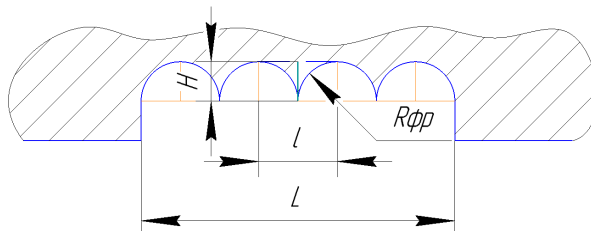


Рисунок 2. – Схема расположения остаточного материала

На рисунке 2 видна схема расположения остаточных гребешков после фрезерования.  $l$  - величина перемещения фрезы,  $L$  - длина обрабатываемой стороны,  $R_{фр}$  - радиус инструмента. Можно вывести простую зависимость количества гребешков, остающихся после обработки:

$$N_{гр} = (L/l) - 1 = (L/2R_{фр}) - 1$$

Величину же врезания в материал многие компании, занимающиеся разработкой и продажей режущего инструмента, рекомендуют брать приблизительно равной ширине используемой пластины.

**Вывод.** Плунжерное фрезерование применяется при:

- при вылете инструмента более  $4xD_{фр}$ ;
- при низкой стабильности процесса резания, так как при осевых нагрузках обеспечивается большая жесткость инструмента, чем при обработке по контуру, из-за чего точность обработки выше;
- при полустиковой обработке углов, карманов и других конструктивных элементов;
- при обработке труднообрабатываемых материалов, таких как титановые сплавы;
- при ограниченном крутящем моменте или мощности станка.

Кроме того, при плунжерном фрезеровании устанавливаются повышенные требования к режущему инструменту: большой объем стружечной канавки, внутренние каналы для подвода СОЖ, тангенциальное крепление пластин на торцевой части фрезы для увеличения жесткости пластины и уменьшения изгибающих напряжений.

#### ЛИТЕРАТУРА

1. База стандартов [Электронный ресурс] // Главная/ Общероссийский классификатор стандартов / Общие положения. Терминология. Стандартизация. Документация / Виды обработки резанием. Термины и определения общих понятий: <http://engeneer.ru/gost-25761-83>- Дата доступа: 10.09.2019
2. Руководство SolidCAM 2006 R10.1 , 106 с.
3. SandvikCoromant [Электронный ресурс] // Старт / Техническая информация / Фрезерование / Фрезерование отверстий и карманов / Плунжерное фрезерование: <https://www.sandvik.coromant.com/ru-ru/knowledge/milling/milling-holes-cavities-pockets/pages/plunge-milling.aspx>- Дата доступа: 10.09.2019

УДК 621.91.01

**ИССЛЕДОВАНИЕ ЦИКЛОВ ВРЕЗАНИЯ  
ПРИ ФРЕЗЕРОВАНИИ УСТУПОВ КОРПУСНЫХ ДЕТАЛЕЙ НА СТАНКАХ С ЧПУ**

**Е.О. ЖИХОРЕВ**

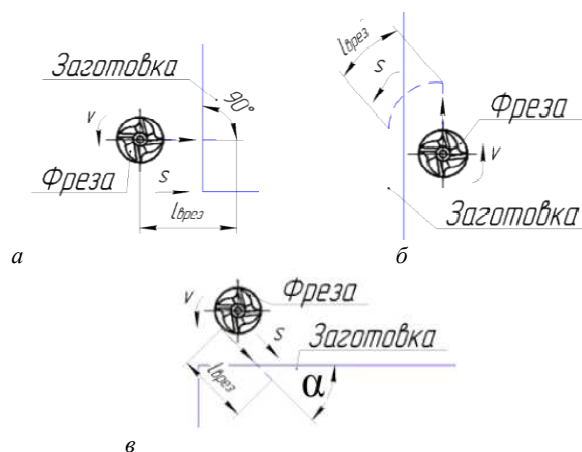
(Представлено: д-р техн. наук, проф. Н.Н. ПОПОК)

Корпусные детали составляют значительный процент изделий, изготавливаемых на машиностроительных производствах Республики Беларусь. Конфигурация корпусных деталей включает большое количество конструктивных элементов (КЭ): пазов, карманов, уступов, бобышек, отверстий. Ранее проведенный анализ частоты встречаемости КЭ корпусных деталей, обрабатываемых на ОАО "Измеритель", показал, что уступы встречаются во всех 100% корпусов [1].

В связи с этим исследование циклов обработки уступов на фрезерных станках с ЧПУ является актуальным, так как от способа врезания зависит период стойкости инструмента. Поэтому одной из задач технолога является выбор наиболее рационального способа врезания.

Рассмотрим применяемые методы врезания инструмента при обработке уступов:

- 1) по нормали (Рис. 1.а);
- 2) по дуге (Рис. 1.б);
- 3) под углом (Рис. 1.в).



**Рисунок 1. – Траектории врезания**  
а – по нормали; б – по дуге; в – под углом 45°

На Рисунке 1 представлены схемы врезания, где обозначены: направление подачи фрезы (s), направление вращения инструмента (v), длина врезания (l<sub>врез</sub>).

Для определения сил резания, действующих на инструмент при различных способах врезания, проведен эксперимент, в ходе которого фиксировались напряжение и сила тока, потребляемые двигателем главного движения вертикально-фрезерного станка модели FSS-500. Параметры обработки представлены в таблице.

Таблица. – Параметры обработки

	Врезание по дуге	Врезание под углом 45°	Врезание под углом 90°
Используемый инструмент	Фреза концевая Р6М5 Ø10		
Материал заготовки	Алюминиевый сплав Д16		
Длина врезания l <sub>врез</sub> , мм	14,7	12,72	12
Глубина обработки t, мм	10	10	10
Режимы резания			
s, мм/мин	100	100	100
n, об/мин	1600	1600	1600
V, м/мин	50,24	50,24	50,24

В ходе проведения эксперимента при обработке заготовки (Рис. 2) значения данных параметров фиксировались при помощи видеосъемки машинных параметров системы ЧПУ.

На основе анализа видеосъемки получены данные по силе тока, напряжению и скорости вращения шпинделя и построены соответствующие графики (Рис. 3,4,5)



Рисунок 2. – Заготовка после проведения эксперимента

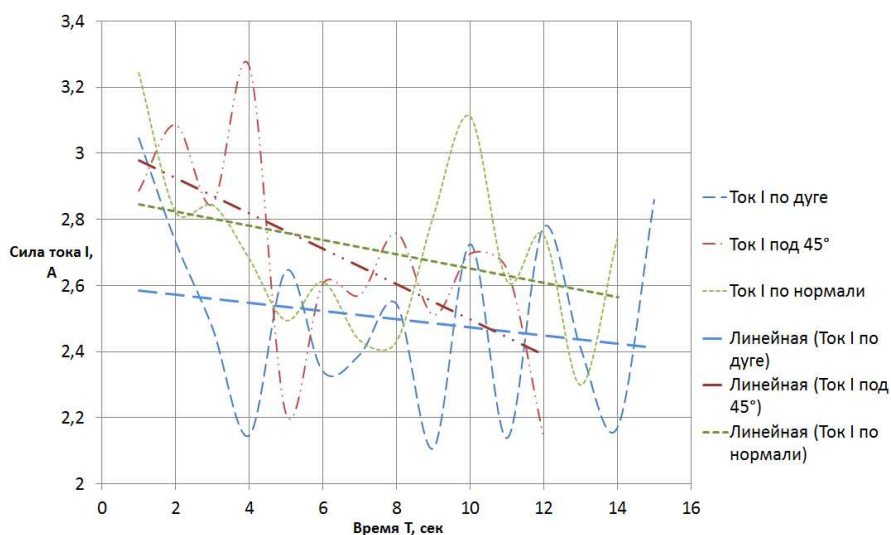


Рисунок 3. – График силы тока

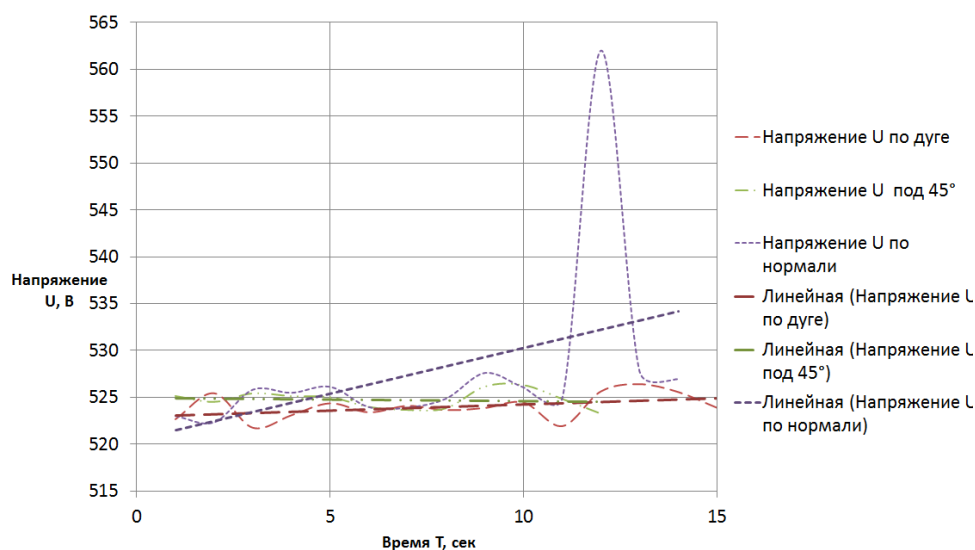


Рисунок 4. – График напряжения



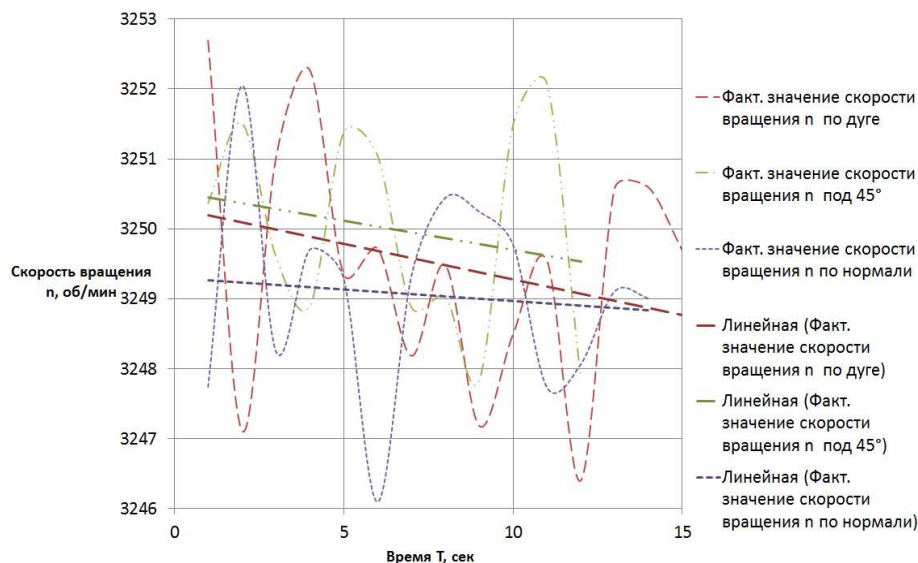


Рисунок 5. – График скорости вращения шпинделя

Анализ графиков показывает, что:

- наиболее резкое падение тока наблюдается при врезании под углом 45°, с 3 до 2,4 А. В остальных двух случаях разница между максимальным и минимальным значениями тока составляет около 0,2 А;
- наибольшее увеличение напряжения – с 522 до 534 В–приходится на врезание по нормали. При врезании под углом и по дуге напряжение колеблется в пределах 523-525 В;
- во всех трех случаях наблюдается падение скорости вращения шпинделя, из-за того, что по мере врезания увеличивается глубина резания, и как следствие возрастает нагрузка на шпиндель станка.
- Несмотря на то, что врезание по дуге является наиболее выгодным способом, на практике при разработке УП используются все три способа ввиду того, что не всегда при обработке конструктивных элементов есть достаточно свободного пространства для движения инструмента.

#### ЛИТЕРАТУРА

1. Y. Zhykharau, N. Popok Analysis of constrictive elements of housing details processed on CNC milling machine / European and National Dimension in Research. Machine-building X Junior Researcher's Conference, Novopolotsk, 10-11 May 2018, p. 158-159.

УДК 62091

## СИСТЕМА ОРИЕНТАЦИИ СОЛНЕЧНЫХ КОЛЛЕКТОРОВ В ПРОСТРАНСТВЕ

**В.К. КОВАЛЕВСКИЙ***(Представлено: канд. техн. наук, доц. В.Ф. ЯНУШКЕВИЧ)*

В статье исследовано влияние ориентации и угла наклона солнечного коллектора на его производительность. Установлено, что для поглощения максимального количества солнечной энергии плоскость солнечного коллектора должна быть всегда перпендикулярна солнечным лучам. Проанализировано освещение солнцем Земной поверхности в зависимости от времени суток и года. Установлено, что для монтажа солнечных коллекторов необходимо знать оптимальную ориентацию в пространстве абсорбера солнечного коллектора. Для оценки оптимального ориентирования коллекторов учитывается вращение Земли вокруг Солнца и вокруг своей оси, а также изменение расстояния от Солнца. В статье представлен расчет угла наклона солнечного коллектора для зимнего и летнего периода в Новополоцке, а также формулы расчета суммарной дневной солнечной радиации на горизонтальную поверхность и количество удельной энергии, вырабатываемой солнечным коллектором.

Солнечный коллектор – гелиоустановка (для сбора тепловой энергии Солнца), способная нагревать материал-теплоноситель.

В настоящее время использование солнечных коллекторов для воспроизводства тепловой энергии не проекты будущего, а действующие и реализуемые программы во многих странах мира. Солнечные коллекторы очень широко используются в Америке, Австралии, Европе для отопления промышленных и бытовых помещений, для горячего водоснабжения производственных процессов и бытовых нужд.

Для увеличения производительности коллекторов предлагается использовать специальные селективно-поглощающие покрытия абсорбера, концентраторы солнечного излучения, антиотражательные покрытия на остеклении и др.

Обычно рекомендуется направлять солнечный коллектор в южную сторону под углом 35-40° к горизонту (рис.1). При этом фиксированном положении коллектор сможет работать на полную мощность, т.е. располагаться под углом 90° к солнечным лучам, небольшой промежуток времени[1].

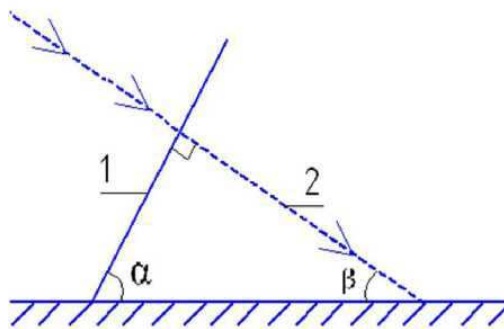


Рисунок 1. – Схема расположения солнечного коллектора:

1 – солнечный коллектор; 2 – солнечные лучи;

α – угол расположения солнечного коллектора; β – угол падения солнечных лучей

Для повышения эффективности работы коллекторов, необходимо рассчитать углы наклона солнечного коллектора по отношению к солнцу.

Рассмотрим необходимый диапазон регулирования солнечными коллекторами, максимальный угол падения солнечных лучей - β в Новополоцке в зимнее время составляет 13°, в летнее время угол падения солнечных лучей - 52°.

Рассчитаем угол наклона солнечного коллектора для зимнего и летнего периода, используя рисунок 1. Наибольшее количество солнечной энергии поглощается коллектором при его расположении под прямым углом к направлению солнечных лучей. Сумма углов треугольника составляет 180°, следовательно:

$$\alpha = 180^\circ - 90^\circ - \beta$$

Для зимнего периода времени рекомендуемый угол наклона составит  $\alpha = 77^\circ$ . Для летнего периода времени рекомендуемый угол наклона составит  $\alpha = 38^\circ$ .

Получаем, что в Новополоцке угол наклона коллектора должен лежать в диапазоне от  $20^\circ$  - минимальный угол расположения вакуумных коллекторов и до  $77^\circ$  - максимальный угол для зимнего периода. Суммарная дневная солнечная радиация на горизонтальную поверхность определяется [2]:

$$H = H_0 \left( a + b \frac{S}{S_0} \right)$$

где  $H_0$  - коэффициент, равный атмосферному значению солнечной радиации, а и в - постоянные коэффициенты для заданного района;  $S$  - действительная продолжительность солнечного сияния для заданного района, ч;  $S_0$  - возможная продолжительность солнечного сияния, ч.

Количество удельной энергии, вырабатываемой солнечным коллектором [2]:

$$Q_{y0} = K_f \left( H \cdot \gamma - \frac{T_n - T_{o.c}}{R_n} 3,6 \cdot 10^{-3} \right)$$

где  $K_f$  - коэффициент переноса солнечной энергии к жидкости,  $\gamma$  - коэффициент пропускания солнечного излучения прозрачным покрытием,  $T_n$  - температура приемной поверхности коллектора, К,  $T_{o.c}$  - температура окружающей среды, К,  $R_n$  - термическое сопротивление приемной поверхности типичного коллектора.

Установлено, что при стандартной установке коллектора в направлении южной стороны, солнце окажется вне зоны доступа коллектора на 28,8% от общего времени солнца над горизонтом.

#### ЛИТЕРАТУРА

1. Ширман, Я. Д. Теоретические основы радиолокации / Я. Д. Ширман, В. Н. Голиков, И. Н. Бусыгин, Г. А. Костин. – М.: Советское радио, 1970. – С. 559.
2. Саплин, Л.А. Энергоснабжение сельскохозяйственных потребителей с использованием возобновляемых источников: Учебное пособие / Л.А. Саплин, С.К. Шерязов, О.С. Пташкина-Гирина, Ю.П. Ильин. – Челябинск: ЧГАУ, 2000. – С.194.
3. Метрология и радиоизмерения. Учебно – методический комплекс для студ. спец. 1 – 390101 «Радиотехника» / В. Ф. Янушкевич. – Новополоцк, ПГУ, 2010. – С. 304.

УДК 62091

## СТРУКТУРНАЯ СХЕМА СИСТЕМЫ УПРАВЛЕНИЯ СОЛНЕЧНЫМ КОЛЛЕКТОРОМ

**В.К. КОВАЛЕВСКИЙ***(Представлено: канд. техн. наук, доц. В.Ф. ЯНУШКЕВИЧ)*

В статье разработана структурная схема системы управления солнечным коллектором. Определены основные функциональные части изделия, их назначение и взаимосвязи. Схема разработана с учетом угла наклона солнечного коллектора для зимнего и летнего периода в Новополоцке, а также суммарной дневной солнечной радиации на горизонтальную поверхность и количества удельной энергии, вырабатываемой солнечным коллектором.

Структурная системы управления солнечным коллектором приведена на рисунке 1[1]. Разработка проведена на основе методики, изложенной в работах [2 – 5].

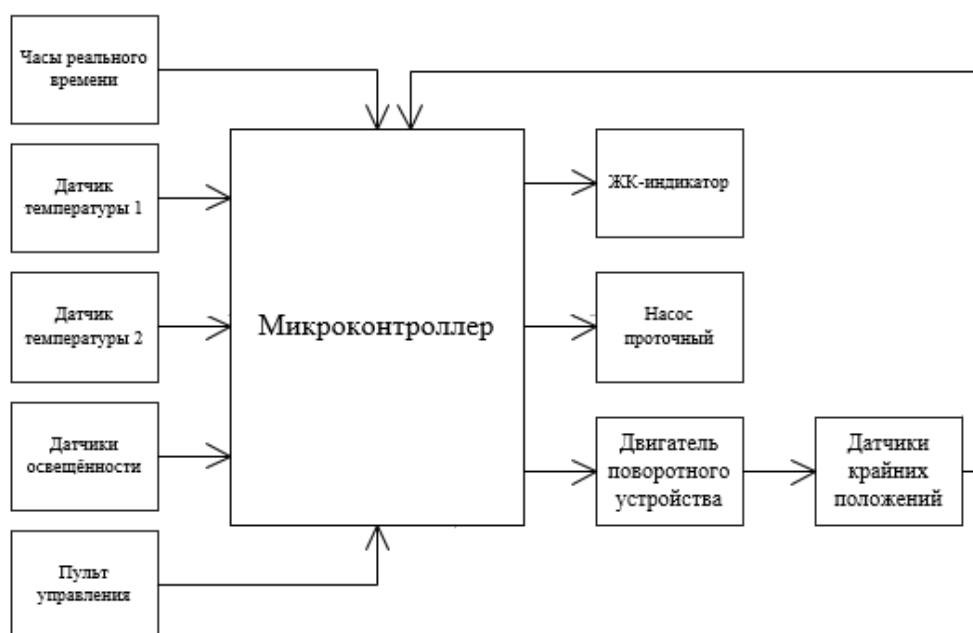


Рисунок 1. - Структурная схема системы управления солнечным коллектором

**Часы реального времени.**

Часы работают или в 24-часовом или 12-часовом формате с индикатором AM/PM (до полудня, после полудня). Подключение часов реального времени к микропроцессу упрощено за счет синхронной последовательной связи.

**Датчик температуры.**

Диапазон измерения температуры жидкости составляет от минус 55 до плюс 125 °С.

**Датчик освещённости.**

В качестве датчика освещённости будут использованы фоторезисторы. Фоторезистор представляет из себя полупроводниковый радиоэлемент, который меняет свое сопротивление в зависимости от освещения.

**Микроконтроллер** — микросхема, предназначенная для управления электрическими устройствами. Контроллеры для солнечных коллекторов предназначены для повышения эффективности и автоматизации процесса нагрева воды.

**ЖК-индикатор.** Это устройство размерностью 1602 (т.е. по 16 символов в двух строках), на базе контроллера HD44780, являются одними из самых простых, доступных и востребованных дисплеев для разработки различных электронных устройств. Его можно встретить, как и в простых, так и в промышленных устройствах.

**Пульт управления** — устройство для контроля и управления работой устройств и процессов.

**Насос проточный.** Устройством для движения теплоносителя в циркуляционном контуре является насос. С помощью него создается необходимый перепад давлений, компенсирующий гидравлические сопротивления в системе, и обеспечивается проток жидкости с требуемым расходом. Работа насоса характеризуется следующими рабочими характеристиками: подача, мощность и КПД.

**Двигатель поворотного устройства.**

Этот механизм, необходим в схеме для движения солнечного коллектора по траектории солнца.

**Датчики крайних положений.**

Датчик перемещения — это прибор, предназначенный для определения величины линейного или углового механического перемещения какого-либо объекта. Они необходимы нам для ориентации солнечного коллектора в пространстве. Данные датчики позволяют микроконтроллеру определять границы действия самого коллектора. Границы действия солнечного коллектора — это восход и закат солнца.

Таким образом, в статье разработана структурная схема системы управления солнечным коллектором с учетом влияния ориентации и угла наклона солнечного коллектора на его производительность. Установлено, что для поглощения максимального количества солнечной энергии плоскость солнечного коллектора должна быть всегда перпендикулярна солнечным лучам. В разработанном устройстве учитывается освещение солнцем земной поверхности в зависимости от времени суток и года.

#### ЛИТЕРАТУРА

1. Функциональные схемы и их виды [Электронный ресурс]. – Режим доступа: <http://www.moysatapparat.ru/ustanovka/ustanovka.html>
2. Ширман, Я. Д. Теоретические основы радиолокации / Я. Д. Ширман, В. Н. Голиков, И. Н. Бусыгин, Г. А. Костин. – М.: Советское радио. – 1970. – С. 559.
3. Метрология и радиоизмерения. Учебно – методический комплекс для студ. спец. 1 – 390101 «Радиотехника» / В. Ф. Янушкевич. – Новополоцк, ПГУ. – 2010. – С. 304.
4. Петровский, А. А. Микропроцессорная техника. Учебное пособие по курсовому проектированию / А. А. Петровский, Г.В. Таранов. – Минск, БГУИР. – 2005. – 51 с.
5. Микроконтроллеры AVR [Электронный ресурс]. – Режим доступа: <http://www.electromonter.info/handbook/13/ElectromagneticRelay>

УДК 621.371.39

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ GSM АНТЕНН

А.О. СТАНОВОЙ

(Представлено: канд. техн. наук, доц. В.Ф. ЯНУШКЕВИЧ)

В статье представлены результаты сравнительной оценки спроектированной GSM антенны и существующих антенн подобного принципа действия. Показана актуальность проектирования данных антенн, вызванная развитием радиоэлектронных устройств и систем связи и необходимостью конструирования GSM антенн. Использование антенн в радиотехнических системах позволяет повысить возможности систем радиосвязи, радиолокации, так как количество информации, которая передается в единицу времени, прямо пропорционально полосе частот сигнала. Показано, что данная антенна позволяет также обеспечить высокую помехозащищенность каналов связи и повысить точность при оценке взаимной ориентации движущихся объектов.

Антенна — устройство для излучения и приёма радиоволн [1].

Форма, размеры и конструкция антенн и разнообразны зависят от длины излучаемых или принимаемых волн и назначения антенны. Применяются антенны в виде отрезка провода, комбинаций из таких отрезков, отражающих металлических зеркал различной конфигурации, полостей с металлическими стенками, в которых вырезаны щели (щелевая антенна), спиралей из металлических проводов и другие.

GSM антенна – наиболее распространенная антенна, имеющая достаточно узкую диаграмму направленности в горизонтальной и вертикальной плоскостях. Антенна состоит из набора элементов: одного вибратора, к которому подводится и снимается сигнал (активный вибратор) и несколько пассивных вибраторов, один из которых является рефлектором, а остальные работают в режиме директоров. Задача рефлектора ослабить излучение в заднем, не рабочем, направлении, задача директора усилить сигнал в нужном направлении. Все вибраторы параллельны и лежат в одной плоскости, эта плоскость и определяет поляризацию антенны. Часто такую антенну называют антенной типа «волновой канал» или антенной «Уда - Яги» [2].

В связи с развитием радиоэлектронных устройств и систем связи, появилась необходимость в GSM антеннах. Использование антенн в радиотехнических системах позволяет повысить возможности систем радиосвязи, радиолокации, так как количество информации, которая передается в единицу времени, прямо пропорционально полосе частот сигнала; позволяет также обеспечить высокую помехозащищенность каналов связи; повысить точность при оценке взаимной ориентации движущихся объектов [3].

Universal Mobile Telecommunications System (далее UMTS) – Универсальная Мобильная Телекоммуникационная Система – технология сотовой связи, разработана Европейским Институтом Стандартов Телекоммуникаций (ETSI) для внедрения 3G в Европе.

С целью отличия от конкурирующих решений UMTS также часто называют 3GSM с целью подчеркнуть принадлежность технологии к сетям 3G и его преимущество в разработках с сетями стандарта GSM.

Анализ проектируемой GSM антенны сводится к следующим параметрам:

- коэффициент стоячей волны по напряжению: КСВН<2;
- минимальная частота:  $f_{min}=1920$  МГц;
- максимальная частота:  $f_{max}=2170$  МГц;
- усиление:  $K_u = 17$  Дб.

На темы, схожие GSM антенн найдены следующие патенты:

- Штыревая антенна;
- Штыревой четвертьволновой вибратор;
- Директорная антенна.

Для проведения сравнительного анализа, в таблице представлены основные технические характеристики антенн рассмотренных выше аналогов.

Таблица. – Основные технические характеристики рассмотренных аналогов и разрабатываемого устройства

Параметр/антенна	GSM	Штыревая	Четвертьволновой вибратор	Директорная
Диапазон частот, МГц	1920..2170	900..1800	1800..2600	872..960
Усиление, Дб	17	5	5,5	14
Волновое сопротивление, Ом	50	50	50	50
КСВ	1,6	1,7	1,8	1,9
Максимальная потребляемая мощность, Вт	50	60	20	60

Исходя из представленных выше данных можно сделать вывод, что разрабатываемое устройство имеет большее усиление и имеет более узкую полосу пропускания, а также коэффициент стоячей волны (КСВ) наименьший в сравнении с рассмотренными аналогами.

## ЛИТЕРАТУРА

1. Гололобов, Д. В. Фазовые методы идентификации углеводородных залежей / Д.В. Гололобов, А.О.Рак, В.Ф.Янушкевич / Инженерный вестник. – Мн. –2001. – №1(20). – С. 16-18.
2. Янушкевич, В.Ф. Электромагнитные методы поиска и идентификации углеводородных залежей / В.Ф.Янушкевич. – Новополоцк, ПГУ, 2017. – 232с.
3. Гололобов, Д.В. Взаимодействие ЧМ-сигналов с анизотропной средой / Д. В. Гололобов, Н. В. Цывис, В. Ф. Янушкевич, С. В. Калинин, Д. Л. Хадыко / Проблемы проектирования и производства радиоэлектронных средств. Сборник мат-ов 2 межд. научно-техн. конференции 15-17мая 2002г. – Новополоцк, 2002. – С.300-303.

УДК 621.371.39

## GSM АНТЕННА

А.О. СТАНОВОЙ

(Представлено: канд. техн. наук, доц. В.Ф. ЯНУШКЕВИЧ)

*В статье представлены результаты проектирования GSM антенны. Сформулированы требования к антеннам данного типа. Проведено конструирование антенны подбором всех ее размеров экспериментальным путем, придерживаясь правил теоретического анализа. Представлены результаты расчетов диаграммы направленности, коэффициента стоячей волны и коэффициента отражения антенны.*

К антеннам современных радиосистем предъявляется много требований, среди которых решающее значение имеют два. Первое требование — направленность действия, т. е. распределение электромагнитной мощности в пространстве (или реакция на приходящее электромагнитное поле при радиоприеме) по определенному закону. В одних случаях желательно обеспечить равномерность действия антенны по всем направлениям, в других требуется концентрировать излучение или осуществлять радиоприем в пределах достаточно узкого углового сектора — так называемого луча. Для формирования узкого луча размеры антенны должны во много раз превышать рабочую длину волны радиосистемы. Второе требование — излучение или радиоприем должны сопровождаться минимальными потерями электромагнитной мощности на нагрев проводников и диэлектриков антенны, т. е. антенна должна иметь высокий КПД. Проблема достижения высокого КПД особенно остро проявляется при создании антенн, размеры которых малы в сравнении с длиной волны [1].

GSM антенна — наиболее распространенная антенна, имеющая достаточно узкую диаграмму направленности в горизонтальной и вертикальной плоскостях. Часто такую антенну называют антенной типа «волновой канал» или антенной «Уда - Яги» [2]. Конструкция антенны состоит из набора элементов: одного вибратора, к которому подводится и снимается сигнал (активный вибратор) и несколько пассивных вибраторов, один из которых является рефлектором, а остальные работают в режиме директоров. Рефлектор ослабляет излучение в заднем, не рабочем, направлении, функция директора — усилить сигнал в нужном направлении. Все вибраторы параллельны и лежат в одной плоскости, эта плоскость и определяет поляризацию антенны.

Расчет антенны позволяет получить лишь приближенные результаты, т.к. необходимо учитывать много взаимосвязанных факторов (длину и толщину вибраторов, расстояния между ними), и связан с громоздкими вычислениями [3]. Поэтому чаще всего 1) по заданной рабочей волне  $\lambda$  рассчитывают и устанавливают длину активного вибратора  $2l$  с учетом укорочения  $\Delta l$

2) устанавливают длину рефлектора  $2l_p$  на 5 — 10% больше длины активного вибратора, а расстояние между вибраторами порядка  $0,2 \lambda$ ;

3) с помощью индикатора поля измеряют напряженность поля  $E_0$  (т.е. в сторону активного вибратора) и  $E_{180}$  (в сторону рефлектора), и регулировкой расстояния  $d_p$  между вибраторами и длины рефлектора добиваются минимального отношения  $E_{180}/E_0$  для системы из двух вибраторов;

4) добавляют директор, длину которого  $2l_d$  устанавливают короче на 10 — 15% длины активного, а расстояние  $a_d$  до последнего — порядка  $0,2 \lambda$ , и регулировкой указанных размеров добиваются минимального отношения  $E_{180}/E_0$ . Для системы из трех вибраторов (при этом может потребоваться незначительная подстройка рефлектора);

5) добавляют последовательно второй, третий и т.д. директора и производят аналогичные регулировки.

Форма диаграммы направленности (ДН), коэффициента стоячей волны (КСВ) и коэффициента отражения (S11) рассчитаны для частоты от 1,92 ГГц до 2,17 ГГц, выбран материал медь. Антенна имеет направленное излучение в направлении  $0^\circ$ , в направлении  $-180^\circ$  наблюдается обратное малое излучение ДН как показано на рисунке 1. У антенны наблюдаются хорошие согласующие свойства на отрезке частот от 1,99 ГГц до 2,17 ГГц (рисунок 2). Величина коэффициента отражения от входа уменьшается от 0 дБ на частоте 1,92 ГГц до -10,4 дБ на частоте 2,17 ГГц, на частоте 2,045 ГГц коэффициент отражения равен -6 дБ (рисунок 3).



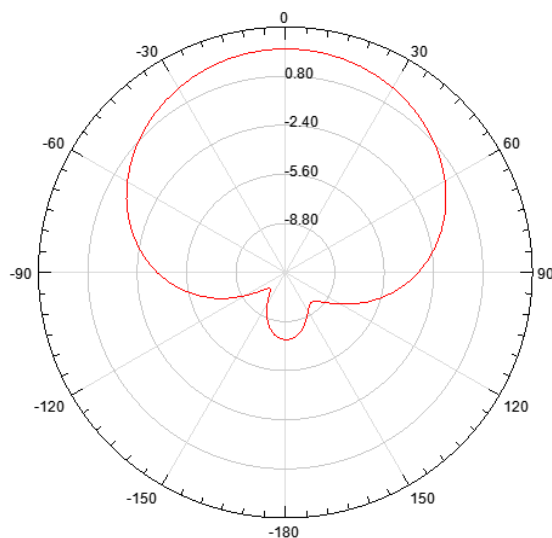


Рисунок 1. – Диаграмма направленности антенны

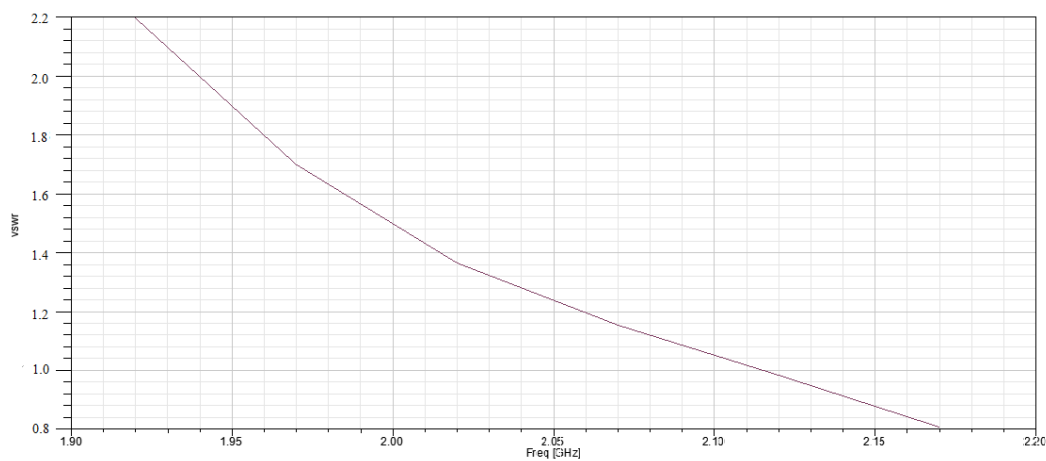


Рисунок 2. – Коэффициент стоячей волны антенны

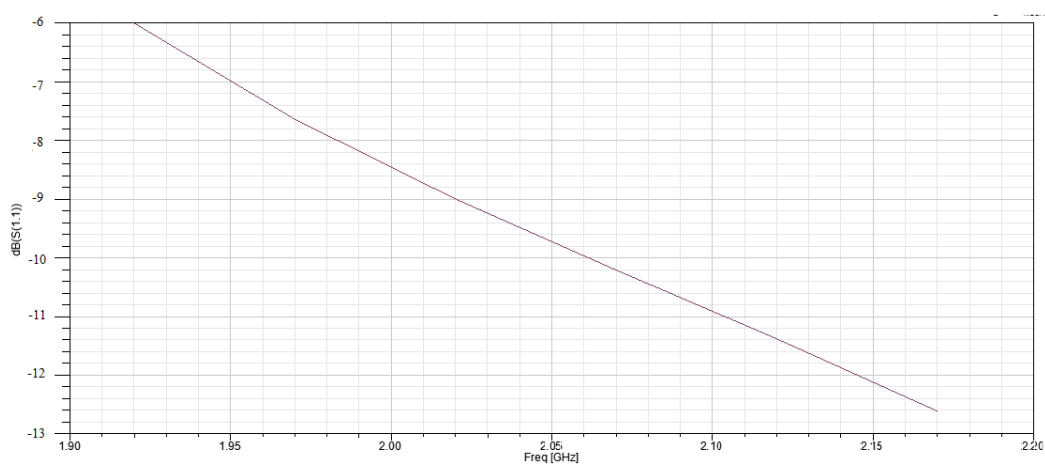


Рисунок 3. – S11 параметр антенны

Использование антенн в радиотехнических системах позволяет повысить возможности систем радиосвязи, радиолокации, так как количество информации, которая передается в единицу времени, прямо пропорционально полосе частот сигнала; позволяет также обеспечить высокую помехозащищенность каналов связи; повысить точность при оценке взаимной ориентации движущихся объектов.

## ЛИТЕРАТУРА

1. Гололобов, Д.В. Электродинамические параметры подводных источников в поле поверхностной электромагнитной волны / Д.В.Гололобов, В.Н. Москвичев, Г.П.Турук, В.Ф.Янушкевич / Тез. докл. 35 Всероссийской Межвузовской НТК. – Владивосток. – 1992. –Т1,ч.1. – С.59 – 62.
2. Янушкевич, В.Ф. Взаимодействие электромагнитных волн с анизотропными средами над углеводородными залежами в режиме радиоимпульсных сигналов. В.Ф. Янушкевич, К.И. Кременя, М.М . Иванов , Т.В. Молодечкина. Вестник ПГУ. Серия С. Фундаментальные науки. Физика. – Новопо-лоцк, 2016. – №12. – С. 45 – 50.
3. Янушкевич, В.Ф. Моделирование двухчастотного взаимодействия электромагнитных волн с углеводородной залежью / В.Ф. Янушкевич. Приборы. Справочный журнал. –1999. – №10. – С.27 – 29.

УДК 69:658.26

## ВЛИЯНИЕ ВЕЛИЧИНЫ УТОНЕНИЯ СТЕНКИ ПРИ ГИБКЕ ТРУБЫ НА ТЕХНОЛОГИЧЕСКИЕ СВОЙСТВА АВТОМОБИЛЬНЫХ ТРУБОПРОВОДОВ

**Д.О. РОДУЛЕВИЧ**

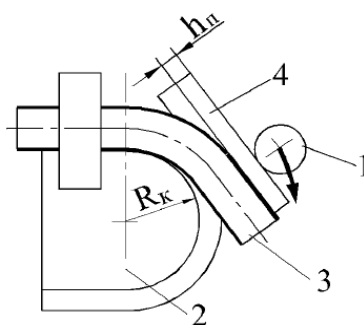
(Представлено: С.В. ПИЛИПЕНКО)

Выполнен анализ влияния величины радиуса гибки трубы на величину относительного утонения трубы и величину коэффициента запаса прочности трубопровода тормозной системы автомобиля. Установлено, что, увеличение диаметра гибки приводит к уменьшению относительного утонения стенки трубы. Из этого, при увеличении диаметра гибки увеличивается запас прочности криволинейного участка трубопровода. Уменьшение диаметра трубопровода в границах допусков приводит к увеличению запаса прочности криволинейного участка трубопровода на 2,5–3,5 % в зависимости диаметра гибки. При моделировании процесса гибки использовались две формулы, выбрана наиболее целесообразная к применению из точки зрения безопасной эксплуатации трубопроводов.

**Вступление.** Изготовление криволинейных участков трубопроводов – сложная техническая задача, требующая применения специализированного оборудования [1–3]. В ходе гибки происходит изменение проходного сечения трубопровода, труба овализируется, изменяется толщина ее стенки, могут возникнуть трещины, складки и пр. [2]. К изогнутым участкам трубопроводов ответственного назначения предъявляют жесткие технические требования и важно, чтобы наведенные в ходе гибки изменения не выводили данный участок трубопровода за предъявляемые к нему требования. Исследования характера изменения параметров трубопроводов в результате гибки являются важной научной задачей [1].

**Целью статьи** является исследование влияния величины утонения стенки трубы в результате гибки на технологические свойства автомобильных трубопроводов.

**Анализ исследований.** На рисунке 1 показана один из вариантов проведения операции изгиба трубы (рис. 1, поз. 3) обкатывающим роликом (рис. 1, поз. 1) по копиру (рис. 1, поз. 2).



**Рисунок 1.** – Один из вариантов проведения процесса гибки трубы [2, 3]

Изначально труба имеет некую величину разностенности, поэтому, нейтральная линия гибки (рис. 2,  $R_{г\text{иб.нейтр.}}$ ) находится в эксцентриситете (рис. 2,  $e$ ) относительно нейтральной линии трубы [2].

В ходе гибки, наружная часть изгибаемой трубы находится в зоне растяжения, а внутренняя – сжатия (рис. 3) [4]. В зоне растяжения стенка трубы утоняется, в зоне сжатия – утолщается (рис. 5) [5]. Кроме этого, в зоне сжатия может произойти процесс складкообразования [4–5].

Как видно (рис. 4), толщина стенки трубы в ходе гибки может изменяться в значительных пределах, особенно опасным является процесс утонения стенки трубы в зоне растяжения. Уменьшение толщины стенки может вывести ее значения за минимально допустимые (с учетом того, что бесшовная труба изначально имеет некоторую величину разностенности [6–8]). Необходимо определить в какой мере процесс утонения влияет на уменьшение коэффициента запаса прочности автомобильных трубопроводов.

Утонение стенки в зоне максимального изгиба можно определить по формуле [9–10]:

$$S_{\text{тр.деф.1}} = \frac{S_{\text{ном.}}}{1 + \frac{D_{\text{тр.нар.}} + S_{\text{ном.}}}{D_{\text{г\text{иб.вн}}} + D_{\text{тр.нар.}}}}, \quad (1)$$

или [4]:

$$S_{тр.деф.2} = \frac{D_{гиб.вн} S_{ном.} + D_{тр.нар.}}{2 \left( \frac{D_{гиб.вн}}{2} + D_{тр.нар.} \right)}. \quad (2)$$

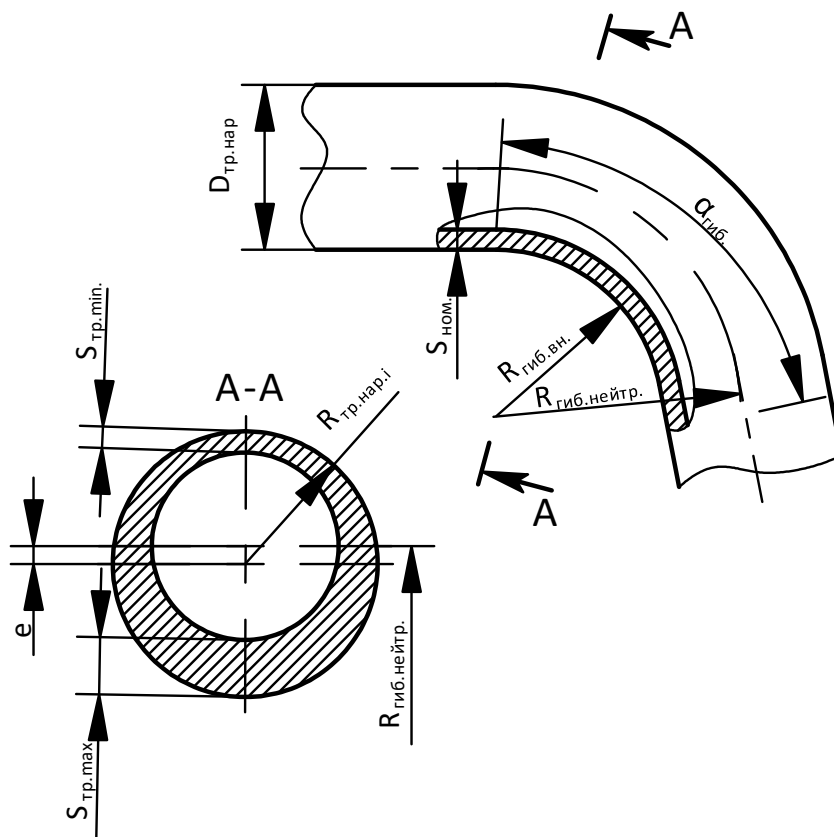


Рисунок 2. – Положение нейтральной линии деформаций в сечении трубы с учетом разнотолщинности стенки [2, 3]:  $e$  – эксцентриситет трубы

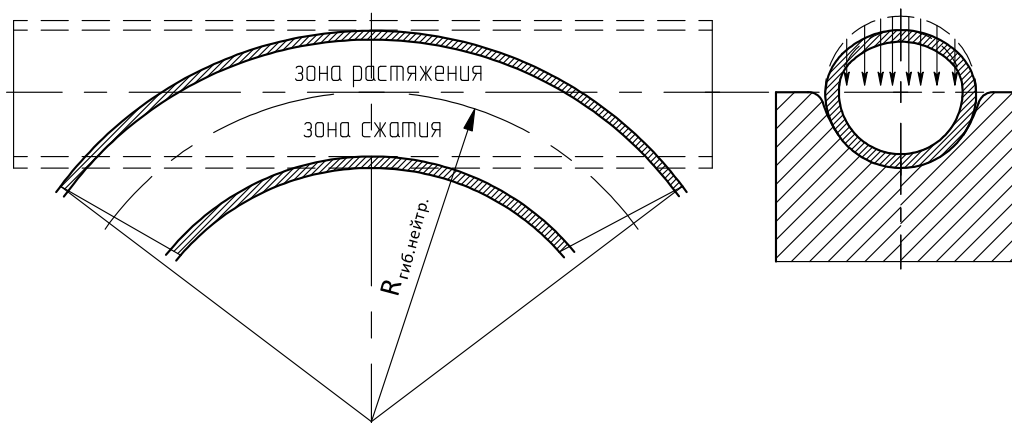


Рисунок 3. – Характерные для процесса гибки зоны деформации трубы [4]

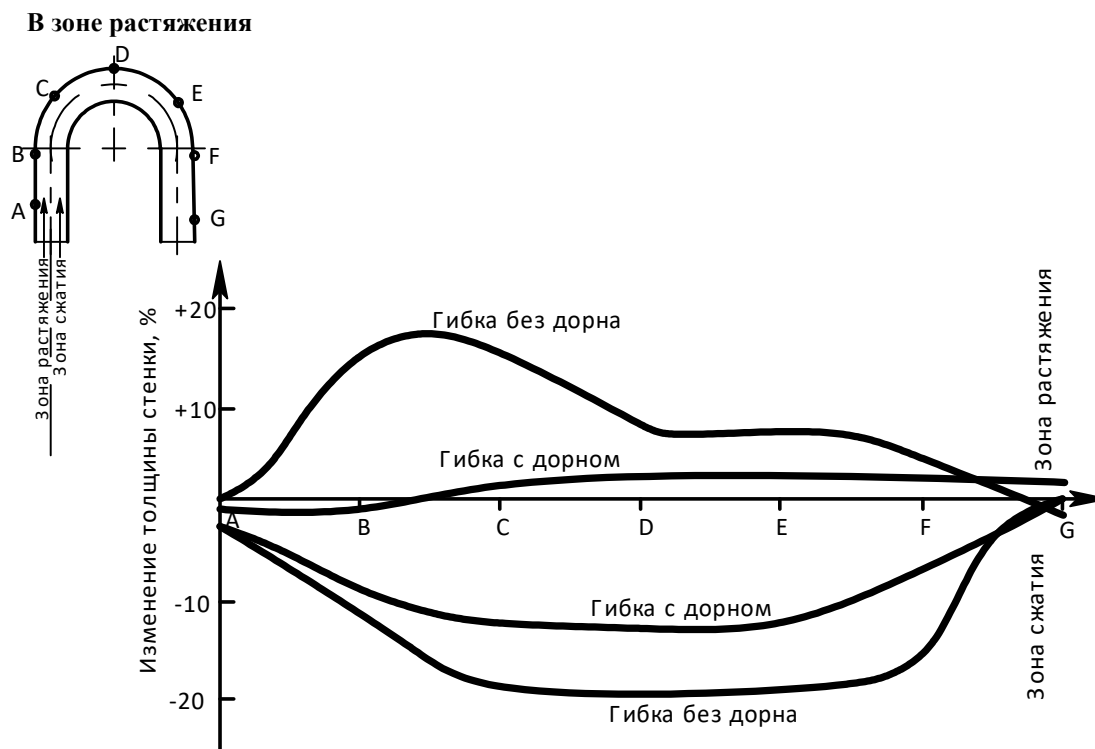


Рисунок 4. – Изменение толщины стенки в зонах гибки трубы [5]

Величина утонения стенки трубы в ходе деформации рассчитывается по формуле [3]

$$\delta_{\%} = \frac{S_{ном.} - S_{деф.}}{S_{ном.}} 100\% , \tag{3}$$

где  $S_{ном.}$  и  $S_{деф.}$  – толщина стенки до и после деформации соответственно.

Допустимое рабочее давление на участок трубопровода трубы в зависимости от диаметра и толщины стенки трубы можно рассчитать по соответствующей формуле [11]:

$$P_{доп} = \frac{2 \cdot S_{ном.} \cdot \sigma_v \cdot m}{n \cdot D_{тр.вн.} \cdot K_1 \cdot K_{назн.}} , \text{ МПа} , \tag{4}$$

где  $S_{ном.}$  – номинальная толщина стенки трубы, мм;  $\sigma_v$  – временное сопротивление разрыву металла трубы, МПа;  $m$  – коэффициент категорийности трубопровода;  $n$  – коэффициент надежности по нагрузке;  $D_{вн.}$  – внутренний диаметр трубопровода, мм;  $K_1$  – коэффициент надежности по материалу;  $K_{назн.}$  – коэффициент надежности по назначению трубопровода.

В качестве рабочего давления автомобильного трубопровода можно выбрать давление в  $P_{раб.} = 50$  Мпа.

**Основной материал.** Для исследования влияния величины утонения стенки трубы на изменение величины коэффициента запаса прочности трубопровода выбрано бесшовную трубу, которая удовлетворяет требованиям ГОСТ 617-2006 «Трубы медные и латунные круглого сечения общего назначения. Технические условия». Такая труба используется в тормозных системах автомобиля (материал – медь марки М1) и имеет следующие механические характеристики:  $\sigma_v=270$  МПа;  $\sigma_T=180$  МПа [11]. Для анализа выбран труба диаметром 4,76 (с допуском: +0; -0,1 мм) и толщиной стенки 0,8 (с допуском:  $\pm 0,08$  мм, или  $\pm 0,07$  для повышенной точности). Во время расчетов, значения всех коэффициентов в формуле (4) были приравнены к единице.

Анализируя формулы (1) и (2) можно сделать вывод о том, что на величину утонения трубы влияют:

- наружный диаметр трубы (колеблется в границах допусков);
- величина толщины стенки (колеблется в границах допусков);
- внутренний радиус гибки (выбирается эмпирически, из условий устойчивости профиля трубы).

В ходе моделирования процесса утонения использовались обе формулы (для сравнения результатов их работы). В качестве начального значения толщины стенки выбрано минимальное её значение согласно допуска ГОСТа. В качестве значений наружного диаметра выбраны экстремальные значения допуска на величину диаметра согласно ГОСТу. Внутренний диаметр гибки изменялся в пределах от 10 до 30 мм, с шагом в 5 мм. (рис. 5).

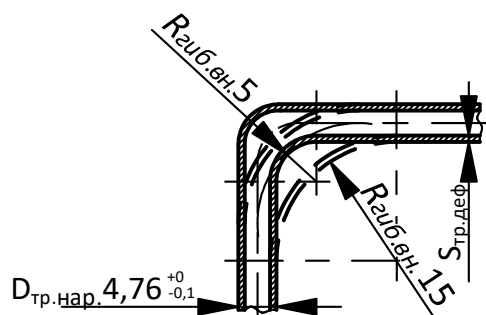


Рисунок 5. – Криволинейный участок трубопровода с разными радиусами гибки

В таблицах 1 и 2 показаны результаты расчетов изменения технологических параметров трубы в зависимости от радиуса гибки.

Таблица 1. – Изменение технологических параметров трубы в зависимости от радиуса гибки (расчет по Ф. 1)

$D_{гиб.вн.}, мм$		10	15	20	25	30
Параметр	Внутр. диам. трубы	-				
	$P_{доп} (МПа)$					
Запас прочности (при $P_{раб.} = 50 МПа$ )	$D_{внутр. max}$	76,43	83,79	88,90	92,65	95,52
	$D_{внутр. min}$	78,86	86,49	91,77	95,64	98,60
$S_{тр.деф.1}$	$D_{внутр. max}$	1,5285	1,6758	1,7780	1,8530	1,9104
	$D_{внутр. min}$	1,5772	1,7298	1,8354	1,9128	1,9719
$\delta, \%$	$D_{внутр. max}$	0,5251	0,5637	0,5895	0,6080	0,6219
	$D_{внутр. min}$	0,5267	0,5653	0,5911	0,6095	0,6233
$\delta, \%$	$D_{внутр. max}$	27,0751	21,7116	18,1217	15,5505	13,6183
	$D_{внутр. min}$	26,8463	21,4856	17,9095	15,3539	13,4366

Таблица 2. – Изменение технологических параметров трубы в зависимости от радиуса гибки (расчет по Ф. 2)

$D_{гиб.вн.}, мм$		10	15	20	25	30
Параметр	Внутр. диам. трубы	-				
	$P_{доп} (МПа)$					
Запас прочности (при $P_{раб.} = 50 МПа$ )	$D_{внутр. max}$	93,61	98,16	101,24	103,46	105,13
	$D_{внутр. min}$	96,33	101,30	104,48	106,76	108,49
$S_{тр.деф.1}$	$D_{внутр. max}$	1,8721	1,9633	2,0248	2,0692	2,1026
	$D_{внутр. min}$	1,9266	2,0260	2,0896	2,1353	2,1697
$\delta, \%$	$D_{внутр. max}$	0,6127	0,6346	0,6491	0,6593	0,6670
	$D_{внутр. min}$	0,6127	0,6357	0,6501	0,6603	0,6679
$\delta, \%$	$D_{внутр. max}$	14,9021	11,8633	9,8540	8,4267	7,3605
	$D_{внутр. min}$	14,9021	11,7096	9,7127	8,2977	7,2426

Формула 1, по сравнению с формулой 2 дает на 80–88% большие результаты величины относительного утонения толщины стенки, однако характер изменения этого параметра в зависимости от диаметра гибки одинаков (рис. 6). Увеличение диаметра гибки приводит к уменьшению относительной величины утонения от 2 до 5% с каждым шагом (13,5 и 7,5 суммарное изменение  $\delta$ , в зависимости от применяемой для расчета формулы). Изменение диаметра трубы в границах допусков согласно ГОСТ, не приводит к значительному изменению величины  $\delta$ , при любых диаметрах гибки.

На рисунке 7 показаны графики изменения коэффициента запаса прочности трубы в зависимости от диаметрах гибки (при  $P_{раб.} = 50 МПа$ ). Из графиков видно, что увеличение диаметра гибки приводит к увеличению запаса прочности трубопровода от 9,45 до 3% с каждым шагом при расчетах по Ф. (1) (25 % общий) и от 5 до 1,6% с каждым шагом при расчетах по Ф. (2) (12,5 % общий). Колебания диаметра трубы в пределах допуска приводят к изменению коэффициента запаса прочности на 2,5–3,5 % в зависимости диаметра гибки.

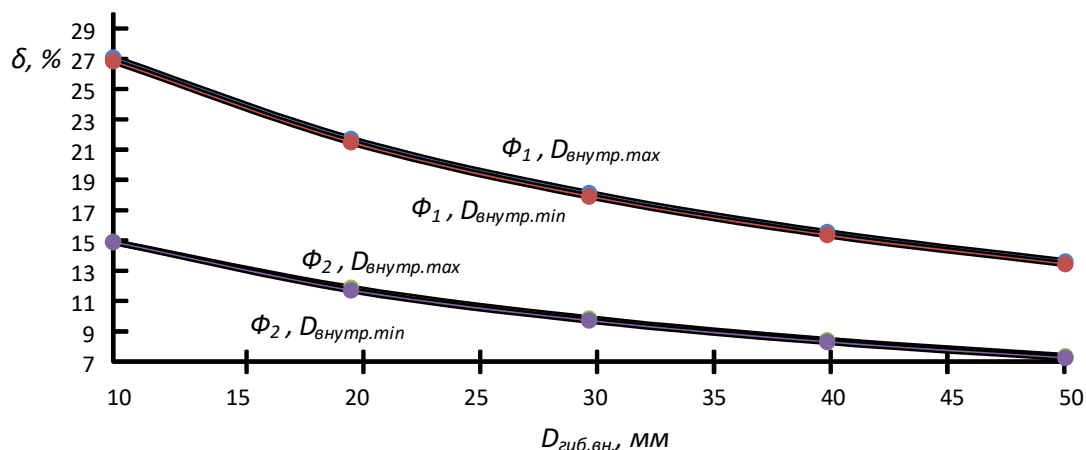


Рисунок 6. – Относительное утонение толщины стенки трубы при разных диаметрах гибки

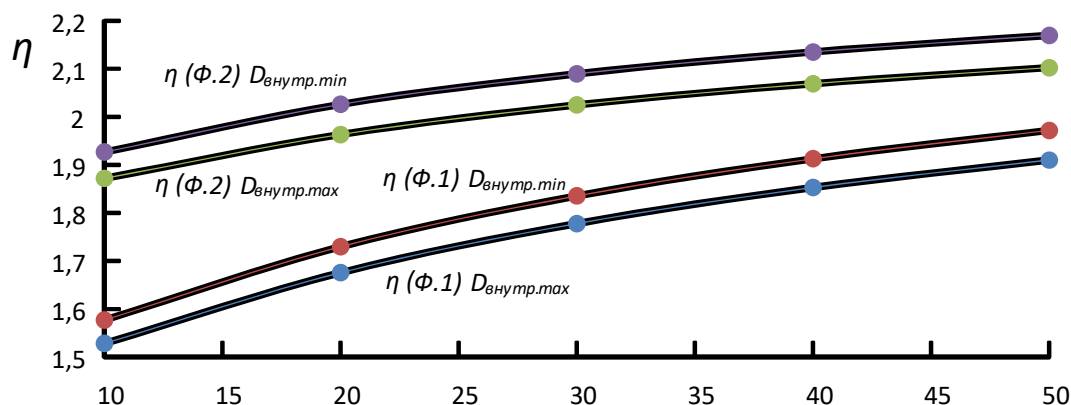


Рисунок 7. – Коэффициент запаса прочности трубы при разных диаметрах гибки (при  $P_{раб.} = 50$  МПа)

**Выводы.** Исследования влияния радиуса гибки на величину относительного утонения трубы показали, что увеличение диаметра гибки приводит к значительному уменьшению величины  $\delta$ . Так, увеличение диаметра гибки с 5 до 30 мм привело к уменьшению относительного утонения стенки трубы на 13,5 (Ф. 1) и на 7,5 (Ф. 2) %. Из этого, при увеличении диаметра гибки увеличивается запас прочности криволинейного участка трубопровода ( $\eta_{\Sigma} = 25$  % при ведении расчетов по Ф. 1 и  $\eta_{\Sigma} = 12,5$  % при ведении расчетов по Ф. 2). Уменьшение диаметра трубопровода в границах допусков приводит к увеличению запаса прочности криволинейного участка трубопровода на 2,5–3,5 % в зависимости диаметра гибки. Формула 1 дает более высокие результаты величины относительного утонения стенки. С точки зрения безопасной эксплуатации трубопроводов, желательно производить расчеты этой величины именно с ее использованием.

#### ЛИТЕРАТУРА

1. Мальцев Д.Н. Совершенствование трубогибного производства предварительным деформированием сечения заготовок: Дис. канд. техн. наук: 05.02.09: Орел, 2014. – 124 с.
2. Вдовин С.И. Инженерный метод вариационной оценки пластических деформаций / С.И. Вдовин, Т.В. Федоров. – Орел: Госуниверситет, 2013. – 93 с.
3. Никитин В.И. Проектирование станков горячей гибки труб / В.И. Никитин. – СПб.: ОАО «ЦТСС», 2011. – 236 с.
4. Мосин И.Ф. Технология изготовления деталей из труб / И.Ф. Мосин. – М.: ГНТИ машиностроительной литературы, 1962. – 171 с.
5. Трубогибы и трубогибные станки для холодной гибки труб: [ЭЛЕКТРОННЫЙ РЕСУРС] ГАЗОВИК. URL: [HTTPS://GAZOVIKPIPE.RU/TRUBOGIBY -I-TRUBOGIBOCHNYE- STANKI-DLYA-HOLODNOJ -GIBKI-TRUB](https://GAZOVIKPIPE.RU/TRUBOGIBY-I-TRUBOGIBOCHNYE-STANKI-DLYA-HOLODNOJ-GIBKI-TRUB) (ДАТА ОБРАЩЕНИЯ: 09.06.2019).

6. Пилипенко С.В. Исследование изменения разностенности труб в ходе прокатки на стане ХПТ / С.В. Пилипенко // Сталь, №3 - 2016 - с 32–37.
7. Григоренко В.У. Исследование изменения разностенности холоднокатаных труб / В.У. Григоренко, С.В. Пилипенко // Сталь, №9 - 2008 – 62–63 с.
8. Столетний М.Ф. Точность труб / М.Ф. Столетний, Е.Д. Клемперт – М.: Metallurgia, 1975 - 239 с.
9. Энциклопедия по машиностроению XXL: [Электронный ресурс] Гибка труб: <https://mash-xxl.info/pics/98106/> (дата обращения: 09.06.2019).
10. Энциклопедия по машиностроению XXL: [Электронный ресурс] Трубопроводы и присоединительная арматура [https://mash-xxl.info/page/245125212242203008235116212113157228\\_050070153189/](https://mash-xxl.info/page/245125212242203008235116212113157228_050070153189/): (дата обращения: 09.06.2019).
11. Абрамов И.П. Расчет максимально допустимого рабочего давления при длительной эксплуатации магистральных нефтепроводов / И.П. Абрамов, И.Ю. Подалов // Записки Горного института. Т.167. Часть 2 - 2006 – 184–185 с.
12. ГОСТ 617-2006. Трубы медные и латунные круглого сечения общего назначения. Технические условия. М., 2008. – 34 с.



# ПРОМЫШЛЕННОСТЬ

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

<i>Ларионов И.В.</i> Проектирование мобильного игрового приложения «Cube» под управлением операционной системы iOS .....	3
<i>Ларионов И.В.</i> Разработка мобильного игрового приложения «Cube» под управлением операционной системы iOS .....	5
<i>Лесницкий В.С.</i> Экономическая составляющая разработки web-сервиса для подачи объявлений, оценка влияния информационной среды на экономику .....	7
<i>Лесницкий В.С.</i> Выбор средств и методов решения задач, проектирование базы данных web-сервиса для подачи объявлений .....	10
<i>Солоцкий А.Ю.</i> Электронный путеводитель по городу Новополюцку для людей с ограниченными возможностями .....	13
<i>Солоцкий А.Ю.</i> Разработка пользовательского интерфейса веб-сайта, с учетом требований отзывчивости, адаптивности и повышения доступности для людей с ограниченными возможностями .....	15
<i>Цыбульский В.А.</i> Веб-сайт «Выпускники» в работе университета .....	17
<i>Цыбульский В.А.</i> Веб-сайт «Выпускники» в сфере образования .....	20
<i>Талдыкин В.Б.</i> Разработка структуры программного обеспечения для анализа компонентов тензора диэлектрической проницаемости углеводородной залежи .....	22
<i>Талдыкин В.Б.</i> Разработка интерфейса программного обеспечения для анализа компонентов тензора диэлектрической проницаемости углеводородной залежи .....	25
<i>Тращенко М.А.</i> Разработка игровых приложений на платформе Facebook .....	27
<i>Тращенко М.А.</i> Принципы разработки игравельной рекламы .....	29
<i>Лапковский К.И.</i> Модульное тестирование как альтернативный инструмент для проверки типов .....	31
<i>Лапковский К.И.</i> Универсальный null object на JavaScript .....	33
<i>Лапковский Н.И.</i> Определение точного местоположения пользователя сайта с использованием JavaScript .....	34
<i>Лапковский Н.И.</i> Преобразование речи в текст с помощью JavaScript .....	38
<i>Андреев И.С.</i> Разработка системы для скрытия информации в звуковой среде .....	41
<i>Андреев И.С.</i> Проектирование графического интерфейса системы для скрытия информации в звуковой среде .....	43
<i>Андрейчиков А.Г.</i> Анализ методов защиты приложений от несанкционированного использования .....	45
<i>Андрейчиков А.Г.</i> Нейронные сети. Сверточная нейронная сеть .....	48
<i>Балабаи М.А.</i> Как устроен Headless Chrome .....	51
<i>Балабаи М.А.</i> Sockjs для асинхронного обмена данными в реальном времени .....	53
<i>Бельский Г.Е.</i> Роль портфолио в повышении конкурентоспособности выпускника вуза .....	55
<i>Бельский Г.Е.</i> Преимущества системы составления портфолио, внедренной в учебный процесс учащихся вузов .....	58
<i>Бобрович Е.П.</i> Архитектурные решения реализации серверной части мобильной ММО-игры в космической стилистике .....	61
<i>Бобрович Е.П.</i> Мониторинг приложений с помощью Spring Boot Actuator .....	64
<i>Борисёнок К.С.</i> Проектирование интерфейсов сайта спортивного клуба .....	67
<i>Борисёнок К.С.</i> Мероприятия по внутренней SEO-оптимизации сайта спортивного клуба .....	70
<i>Вашкевич М.Ю.</i> Сравнение различных подходов обучения нейронных сетей .....	72
<i>Вашкевич М.Ю.</i> Ускорение обучения нейронной сети .....	75
<i>Виноградов А.В., Дьякова А.С.</i> BEAPP – конструктор приложений для сообществ ВКонтакте .....	78
<i>Виноградов А.В., Дьякова А.С.</i> Обзор SMM-инструментов для ВКонтакте .....	81
<i>Волков А.Н.</i> Особенности обучения детей дошкольного и школьного возраста при работе с ЭВМ .....	84
<i>Волков А.Н.</i> Развитие нейронных сверточных сетей в современном мире .....	86
<i>Гаврилов В.В.</i> Создание программной оболочки информационной сенсорной панели «Почётные граждане города Полоцка» .....	88
<i>Гаврилов В.В.</i> Разработка графического интерфейса пользователя программной оболочки информационной сенсорной панели «Почётные граждане города Полоцка» .....	92

<i>Гайдель А.П.</i> Алгоритм Кнута-Морриса-Пратта на языке Python .....	95
<i>Гайдель А.П.</i> Сервис AWS Lambda .....	98
<i>Герец А.Ю.</i> Организационно-экономические условия реализации мобильного приложения по поиску партнёров для занятия спортом .....	102
<i>Герец А.Ю.</i> Оценка рынка мобильного приложения по поиску партнёров для занятия спортом .....	106
<i>Гордейнок В.И.</i> Поворотный механизм для антенны с GPS датчиком и управлением по LAN .....	111
<i>Горовой Е.Н.</i> Анализ современной архитектуры мобильного приложения для операционной системы Андроид .....	114
<i>Горовой Е.Н.</i> Сравнительный анализ подходов реализации уровня представления данных мобильного приложения .....	116
<i>Гусевский Р.П.</i> JSX – удобная технология при разработке веб-приложений .....	118
<i>Гусевский Р.П.</i> Важность основных CSS-технологий на примере разрабатываемого игрового веб-приложения «Обучение возможностям CSS» .....	123
<i>Денисова Е.</i> Возможные варианты шифрования текстовых данных, анализ и выбор оптимального типа шифрования текстовых данных для дальнейшей реализации .....	126
<i>Денисова Е.</i> Модифицированный метод В. Сизова на случай неперiodической неограниченной функции .....	128
<i>Дысин А.В.</i> Разработка тренировочной платформы для начинающих специалистов в сфере тестирования программного обеспечения .....	130
<i>Дысин А.В.</i> Преимущества автоматизации SEO на примере популярных инструментов анализа .....	132
<i>Епанешников А.В.</i> новые возможности современного веб-программирования .....	134
<i>Епанешников А.В.</i> Анализ современных языков программирования .....	137
<i>Иванов И.И.</i> Рассмотрение прогрессивных сегментационных алгоритмов .....	139
<i>Иванов И.И.</i> Сравнительный анализ алгоритмов машинного обучения .....	142
<i>Исаков И.В.</i> Современная компьютерная стеганография в программировании .....	145
<i>Исаков И.В.</i> Атаки на стеганографическую систему цифровых водяных знаков и противодействие им .....	148
<i>Коняев Е.В.</i> Принципы и особенности проектирования охранной сигнализации в составе интегрированной системы безопасности административного здания .....	151
<i>Коняев Е.В., Савицкий К.В.</i> Безопасность как теоретический конструкт .....	153
<i>Кохановский А.В.</i> Проектирование стегосистемы, основанной на сокрытии текстовых данных в изображениях при помощи дискретных преобразований .....	157
<i>Кохановский А.В.</i> Определение степени пригодности контейнера для стеганографической модификации в стегосистеме, основанной на сокрытии текстовых данных в изображениях при помощи дискретных преобразований .....	159
<i>Красько Р.Р.</i> Архитектурные решения реализации клиентской части мобильной MMO-игры в космической стилистике .....	162
<i>Красько Р.Р.</i> Использование чистой архитектуры при разработке Android приложений .....	165
<i>Шевцов М.А.</i> Обработка данных для машинного обучения .....	167
<i>Шевцов М.А.</i> Задачи, решаемые с помощью машинного обучения .....	169
<i>Шершнев Н.О.</i> Postman как современный инструмент для тестирования API .....	171
<i>Шершнев Н.О.</i> Способы извлечения данных, использующие технологию веб-скрапинга. DiffBot как готовое решение для извлечения данных .....	173
<i>Шумянец И.А.</i> Подходы к разработке интерфейса приложения для удаленного управления Компьютером при помощи Android устройства .....	175
<i>Шумянец И.А.</i> Проектирование приложения для удаленного управления компьютером при помощи Android устройства .....	178
<i>Макарычев М.Ю., Макарычева В.А.</i> Метод обнаружения аномалий в потоках данных сетевого трафика .....	182
<i>Макарычев М.Ю., Макарычева В.А.</i> Проектирование сервиса обнаружения распределенных сетевых атак типа «Отказ в обслуживании» .....	185
<i>Малашонок С.Н.</i> Введение в компьютерное зрение .....	187
<i>Малашонок С.Н.</i> Методы компьютерного зрения для распознавания текста из изображений .....	189
<i>Маркевич К.В.</i> Подходы к разработке интерфейса для автоматизированной информационной системы инспекции по делам несовершеннолетних Полоцкого райисполкома .....	191
<i>Маркевич К.В.</i> Проектирование приложения для работы автоматизированной информационной системы комиссии по делам несовершеннолетних Полоцкого райисполкома .....	193
<i>Маханько П.В.</i> Проектирование кроссплатформенного мобильного приложения для подбора сотрудников по навыкам .....	195

<i>Маханько П.В.</i> Проектирование API и базы данных для веб-приложения, разработанного для подбора сотрудников по навыкам .....	197
<i>Медведев Н.А.</i> Цифровая дистрибуция приложения. Влияние онлайн-сервисов для пользователей и разработчиков.....	199
<i>Медведев Н.А.</i> Внешняя платежная система и ее влияние на безопасность сервиса .....	201
<i>Меницкий Е.А.</i> Основные аспекты обеспечения безопасности в медицинских информационных системах.....	203
<i>Меницкий Е.А.</i> Подходы к оптимизации сжатия изображений без видимой потери качества .....	205
<i>Мисевич И.В.</i> Анализ архитектур разработки приложений .....	208
<i>Мисевич И.В.</i> Эффективность автоматизации тестирования.....	210
<i>Морозов Д.Д.</i> Генераторы случайных чисел.....	212
<i>Морозов Д.Д.</i> Основные разновидности вредоносных программ и меры предосторожности против них .....	214
<i>Пацанков Е.Н.</i> Назначение устройства управления приборами GPIB. Актуальность разработки устройства управления приборами ПЗШИ в современном мире .....	216
<i>Пацанков Е.Н.</i> Разработка устройства управления приборами GPIB.....	219
<i>Пашкевич С.Д.</i> Реализация системы управления многопользовательской онлайн-игрой .....	222
<i>Пашкевич С.Д.</i> Применение микросервисной архитектуры при создании распределенных веб-приложений .....	225
<i>Пирович А.И.</i> Введение в WEBP формат сжатия изображений.....	228
<i>Пирович А.И.</i> Введение в интеллектуальный анализ данных .....	230
<i>Плясов В.А.</i> Алгоритм распознавания дороги на видеопоследовательности. обнаружение статических и динамических объектов на видеопоследовательности с использованием Machine Learning и сегментации изображения.....	232
<i>Плясов В.А.</i> Архитектура системы распознавания статических объектов на видеоизображениях автомобильных дорог с различным типом покрытия с использованием облачных сервисов AWS .....	235
<i>Радченко В.С.</i> Рассмотрение работы нейросетей в общем виде .....	237
<i>Радченко В.С.</i> Обучение нейросетей .....	241
<i>Самсонов К.С.</i> Проектирование веб-интерфейса для введения диалога с чат-ботом .....	244
<i>Самсонов К.С.</i> Проектирование API для чат-бота, разработанного для поиска товара внутри магазина .....	246
<i>Сергеев М.А.</i> Реализация горизонтального перемещения в приложении виртуальной реальности .....	248
<i>Сергеев М.А.</i> Реализация вертикального перемещения в приложении виртуальной реальности .....	251
<i>Сивограков А.А.</i> Цифровые водяные знаки как метод защиты изображения .....	255
<i>Сивограков А.А.</i> Разработка интерфейса для мобильного приложения .....	257
<i>Синица П.Р.</i> Сложение двух эллиптических точек и групповые операции. Сужение множества эллиптических точек на множество остатков эллиптической кривой. Графическое тестирование с интерфейсом. ....	259
<i>Синица П.Р.</i> Модифицированный метод шифрования тестовых данных эллиптическими кривыми с использованием уникальной алфавитной строки-ключа.....	262
<i>Скуковская А.А.</i> Проектирование базы данных для приложения «Умная книга рецептов» .....	267
<i>Скуковская А.А.</i> Приложение «Умная книга рецептов».....	270
<i>Смоляк А.И.</i> Разработка системы для скрытия информации с использованием шифра перестановки ....	272
<i>Смоляк А.И.</i> Проектирование графического интерфейса системы для скрытия информации с использованием шифра перестановки.....	274
<i>Станкевич К.В.</i> Метаданные и прикладное программирование .....	276
<i>Станкевич К.В.</i> Тестирование программного продукта .....	278
<i>Субботин А.В.</i> Использование языка программирования Java для разработки гибридной криптосистемы .....	281
<i>Субботин А.В.</i> Использование протокола авторизации OAuth 2.0 в пользовательском приложении.....	284

## **МАШИНОСТРОЕНИЕ**

<i>Гасс Ю.А.</i> Определение оптимальной концентрации наноалмазов в масле в зависимости от режима работы трибопары .....	286
<i>Лопатин Д.В., Гасс Ю.А.</i> Определение величины износа поверхности трения при различных режимах ее термообработки в среде смазочного материала с наноразмерными алмазосодержащими добавками .....	289

<b>Потапов А.А.</b> Анализ условий эксплуатации ударной головки гомогенизатора FB 075 и возможных путей ее изготовления с упрочнением .....	292
<b>Потапов А.А.</b> Инновационный подход к выбору методов упрочнения ударных головок гомогенизатора FBF 075 .....	297
<b>Баишлагёв Д.А., Пуйман Е.В.</b> Исследование возможностей использования покрытий из инструментальных сталей для повышения стойкости режущих инструментов .....	304
<b>Баишлагёв Д.А.</b> Разработка блочно-модульных режущих инструментов для станков с ЧПУ .....	307
<b>Воробьёв М.В., Ульский М.Г.</b> Повышение ресурса распределительных валов совершенствованием технологий их восстановления наплавкой .....	310
<b>Воробьёв М.В., Ульский М.Г.</b> Восстановление деталей автомобиля типа «вал» .....	314
<b>Гуца А.В., Бейтуллаев А.Т.</b> Цинкование элементов кузова автомобиля .....	318
<b>Гуца А.В., Бейтуллаев А.Т.</b> Современные кузовные материалы .....	320
<b>Радудевич Д.О.</b> Влияние величины разностенности труб на технологические свойства трубопроводов .....	322
<b>Тихон Е.М.</b> Оптимизация геометрических параметров расточных инструментов на основе моделирования резьбовых соединений .....	326

### **ХИМИЧЕСКИЕ ТЕХНОЛОГИИ**

<b>Гришин П.Ф.</b> Свойства депарафинизированных фракций, полученных из остаточного продукта процесса «Юникеринг» .....	329
<b>Гришин П.Ф.</b> Способы получения белых масел .....	332
<b>Шульга Е.А., Ваикова Н.С.</b> Направления использования тяжелой смолы пиролиза .....	336
<b>Змитрович С.Ю.</b> Установка для исследования параметров источников искусственного света .....	338
<b>Жихорев Е.О.</b> Сравнение плунжерного и контурного фрезерования .....	341
<b>Жихорев Е.О.</b> Исследование циклов врезания при фрезеровании уступов корпусных деталей на станках с ЧПУ .....	343
<b>Ковалевский В.К.</b> Система ориентации солнечных коллекторов в пространстве .....	346
<b>Ковалевский В.К.</b> Структурная схема системы управления солнечным коллектором .....	348
<b>Становой А.О.</b> Сравнительный анализ GSM антенн .....	350
<b>Становой А.О.</b> GSM антенна .....	352
<b>Родудевич Д.О.</b> Влияние величины утонения стенки при гибке трубы на технологические свойства автомобильных трубопроводов .....	355