

УДК 004.021

**ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА  
ДЛЯ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ  
ТУРИСТИЧЕСКОГО ЦЕНТРА УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ  
«ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**А.И. ПОПОВ**

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

*Рассмотрены принципы построения графического интерфейса пользователя в приложении на языке программирования Java. Проведен сравнительный анализ двух наиболее популярных фреймворков по созданию пользовательского интерфейса, с их кратким описанием, анализом преимуществ и недостатков каждого. Приведено краткое описание спроектированного интерфейса для разрабатываемого приложения.*

В каждом современном языке программирования существует собственный набор методов и средств для разработки графического интерфейса. В языке программирования Java, для настольных приложений, чаще всего используются такие библиотеки как JavaFX с проектированием интерфейса на основе fxml-файлов и Swing с непосредственным созданием элементов в коде программы.

В представленной статье приведем сравнение некоторых технологий проектирования интерфейса настольных приложений, написанных на языке Java.

Как правило, современные приложения строятся по шаблону MVC (Model-View-Controller или Модель-Вид-Контроллер). Идея данного шаблона проста – разделение обязанностей: задача контроллера – обработка действий пользователя (нажатие по кнопкам, обработка запросов к серверу и т.д.); модель – предоставляет контроллеру представление данных, запрашиваемых пользователем; вид – обеспечивает представление данных, полученных из модели.

В Swing элементы пользовательского интерфейса размещаются непосредственно при помощи кода приложения. При этом указываются параметры, присваиваемые элементу управления, от его координат относительно родительского элемента, до различных свойств элемента. Возможно выбирать из предоставляемого набора средств, реализованных в стандартных классах библиотеки Swing, либо реализовать необходимые свойства самостоятельно [1].

Перечислим основные преимущества библиотеки Swing:

- содержит достаточно богатый и удобный набор элементов пользовательского интерфейса;
- практически не зависит от той платформы, на которой должна выполняться программа, следовательно, меньше подвержена ошибкам, характерным для конкретной платформы;
- обеспечивает одинаковое восприятие конечными пользователями приложений с графический пользовательский интерфейс на разных платформах;
- встроенный редактор форм почти во всех средах разработки;
- на базе свинга есть много расширений типа SwingX;
- поддержка различных стилей (Look And Feel).

Однако можно указать на следующие недостатки:

- окно с множеством компонентов начинает подтормаживать;
- работа с менеджерами компоновки становится затруднительной в сложных интерфейсах.

Листинг 1 демонстрирует пример создания контекстного меню с одним элементом с использованием библиотеки Swing.

```
JMenuBar mb = new JMenuBar();
JMenu m1 = new JMenu("FILE");
mb.add(m1);
JMenuItem m22 = new JMenuItem("Save as");
m1.add(m22);
```

**Листинг 1. – Пример создания меню приложения на Swing**

JavaFX является более современным решением для разработки пользовательского интерфейса настольных приложений на Java. Для отрисовки используется графический конвейер, что значительно ускоряет работу приложения. Набор встроенных компонентов обширен, есть отдельные компоненты для отрисовки графиков. Реализована поддержка мультимедийного контента, множество эффектов отображения, анимации. Внешний вид всех компонентов можно легко изменить с помощью CSS-стилей.

Для создания компонентов используются специальные xml/fxml-файлы, в которых при помощи тегов, параметров и свойств создаются элементы пользовательского интерфейса. Также существует большое количество графических редакторов для проектирования интерфейса на JavaFX с последующей интерпретацией в xml формат. Листинг 2 демонстрирует пример создания контекстного меню с одним элементом с использованием библиотеки JavaFX. Данный код размещается в отдельном fxml файле, доступ ко всем элементам происходит по присвоенному им id.

```
<MenuBar layoutY="2.0" prefHeight="25.0" prefWidth="600.0">
  <Menu mnemonicParsing="false" text="File" id="menuBar">
    <items>
      <MenuItem mnemonicParsing="false", text="Close" id="menuItem"/>
    </items>
  </Menu>
</MenuBar>
```

### Листинг 2. – Создание меню с помощью JavaFX

Достоинства фреймворка JavaFX:

- быстрая работа за счет графического конвейера при компиляции и запуске проекта это ускоряет процесс обработки кода и работу приложения;
- множество различных компонентов. Фреймворк позволяет использовать для создания интерфейса, как набор стандартных элементов, так и сторонние пользовательские элементы;
- поддержка стилей. JavaFX поддерживает CSS стилизацию всех компонентов интерфейса;
- утилиты для создания программы. Существует большое количество графических утилит, позволяющих моделировать интерфейс при помощи графического отображения элементов с последующим преобразованием в fxml;
- приложение можно запускать как десктопное и в браузере как часть страницы. Построение интерфейса при помощи fxml файлов и CSS стилей позволит при необходимости перенести интерфейс приложения на веб-форму.

Недостатки JavaFX:

- фреймворк активно развивается, поэтому случаются и падения, и некоторые «глюки», избежать непредвиденных проблем можно не используя элементы, добавленные в последнем обновлении и не злоупотребляя различными пользовательскими расширениями для фреймворка;
- JavaFX полностью привязан к платформе языка Java, однако так как данный проект будет разрабатываться как настольное приложение на языке Java. Однако данный минус не является существенным. Проблемы установки программы на компьютеры пользователей решаются установкой на них JVM [2].

Приложение будет иметь несколько форм, одна из которых считается главной и при запуске программы отображается первой. При закрытии главной формы приложения, прекращается работа всего приложения, при этом также закрываются все открытые окна приложения.

Главная форма приложения будет представлять из себя окно, с меню выбора таблиц из базы данных. После выбора таблицы будут выводиться все хранимые данные в таблицу, расположенную в форме. При помощи боковой панели пользователь сможет управлять данными: заносить новые, редактировать или удалять старые, осуществлять поиск по какому-либо параметру. Кроме того, с помощью кнопок можно будет вызывать дополнительные формы для просмотра графика работы сотрудников и проведения определённым сотрудником мероприятий.

Макеты разработанных форм продемонстрированы на рисунках 1, 2.

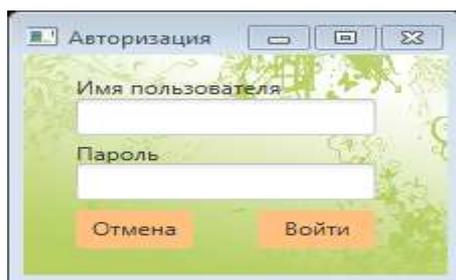
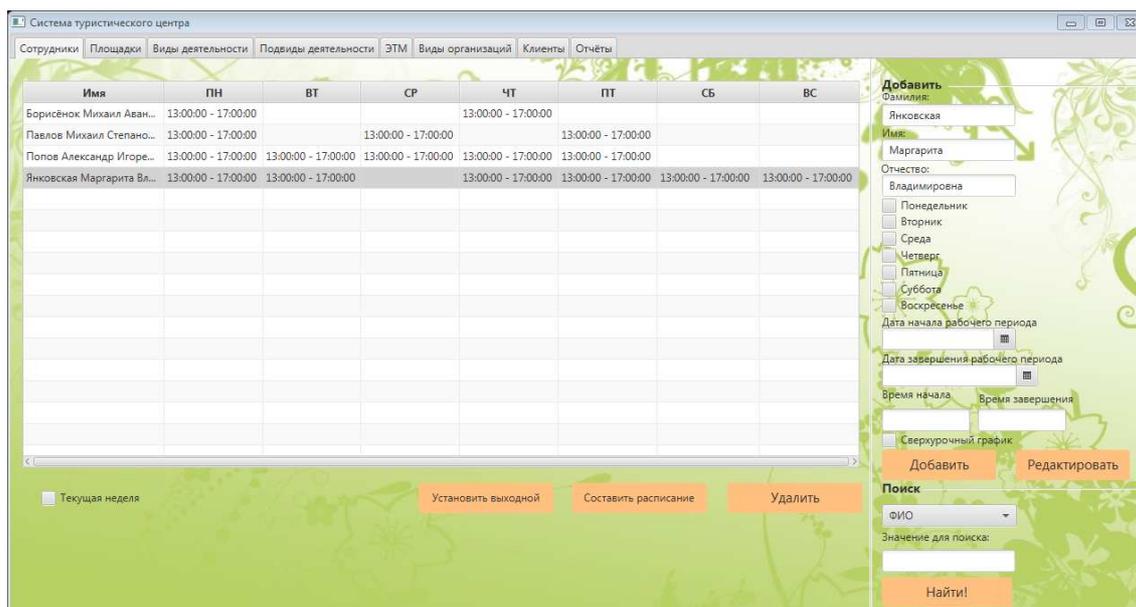


Рисунок 1. – Окно авторизации пользователя



**Рисунок 2. – Рабочее окно приложения**

Кроме того, приложение должно генерировать отчёты по хранимым данным. Пример формы отчётов продемонстрирован на рисунке 3.

	A	B	C	D	E
1	<b>Отчёт о проводимых мероприятиях</b>				
2	<b>Название мероприятия</b>	<b>Сотрудник</b>	<b>Дата проведения</b>	<b>Начало экскурсии</b>	<b>Конец экскурсии</b>
3	Экскурсия в кадетский корпус	Янковская Маргарита Владимировна	2018-04-30	08:45:00	09:15:00
4	Экскурсия в кадетский корпус	Янковская Маргарита Владимировна	2018-05-01	08:30:00	09:00:00
5	Обзорная экскурсия	Попов Александр Игоревич	2018-05-22	15:00:00	16:00:00
6					
7					
8					
9					
10	Подпись руководителя ТЦ:	М.В. Янковская			
11					
12					
13					
14					

**Рисунок 3. – Пример формы сгенерированного отчёта**

Таким образом, после анализа наиболее популярных фреймворков для реализации пользовательского интерфейса настольных приложений было принято решение использовать библиотеку JavaFX. Для облегчения проектирования интерфейса будет использоваться графическая утилита JavaFX SceneBuilder. После выбора средств разработки интерфейса составлены макеты основных экранных форм для приложения и примеры формы генерируемых приложением отчётов.

#### ЛИТЕРАТУРА

1. Краткий обзор GUI-фреймворков для Java [Электронный ресурс] / EAX. – Режим доступа: <https://eax.me/java-swing/>. – Дата доступа: 09.09.18.
2. GUI на Java [Электронный ресурс] / Хакер. – Режим доступа: <https://xaker.ru/2014/09/10/java-gui/>. – Дата доступа: 09.09.18.