

Учреждение образования
«Полоцкий государственный университет имени Евфросинии Полоцкой»

УТВЕРЖДАЮ

Ректор учреждения образования
«Полоцкий государственный университет
имени Евфросинии Полоцкой»

Ю.Я. Романовский

«28» _____ 2024 г.

Регистрационный № УД — 432 24 /уч.

МОДУЛЬ «ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Учебная программа учреждения образования
по учебной дисциплине для специальности
1-40 03 01 «Искусственный интеллект»

2024 г.

Учебная программа составлена на основе образовательного стандарта по специальности высшего образования ОСВО 1-40 03 01-2021 и учебного плана специальности 1-40 03 01 «Искусственный интеллект» (регистрационный № 72-22/уч. ФИТ от 22.07.2022г. для дневной формы получения высшего образования).

СОСТАВИТЕЛИ:

Рогулев Владимир Сергеевич, старший преподаватель кафедры технологий программирования учреждения образования «Полоцкий государственный университет имени Евфросинии Полоцкой»

Чертков Валерий Михайлович, канд. техн. наук, доцент, доцент кафедры технологий программирования учреждения образования «Полоцкий государственный университет имени Евфросинии Полоцкой»

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования учреждения образования «Полоцкий государственный университет имени Евфросинии Полоцкой»
(протокол № 5 от «28» 05 2024)

Методической комиссией факультета информационных технологий учреждения образования «Полоцкий государственный университет имени Евфросинии Полоцкой»
(протокол № 6 от «26» 06 2024)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Учебная дисциплина «Технологии программирования» является ключевой в подготовке специалистов по информационным технологиям. Она закладывает основу для понимания принципов разработки программного обеспечения, применимых в различных сферах, таких как веб-разработка, системное программирование, разработка микросервисов и распределенных систем.

Воспитательное значение учебной дисциплины «Технологии программирования» заключается в формировании у обучающихся правильной культуры и стиля программирования и научного мировоззрения; развитии исследовательских умений, аналитических способностей, креативности, необходимых для решения научных и практических задач; развитии познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

ЦЕЛИ, ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины:

Дисциплина «Технологии программирования» направлена на формирование у студентов фундаментальных знаний и навыков в области разработки программного обеспечения. Она охватывает как теоретические основы программирования, так и современные технологии и инструменты, используемые в процессе создания, тестирования и сопровождения программных продуктов. Основная цель — подготовка студентов к самостоятельной разработке сложных программных систем, использующих эффективные алгоритмы, современные фреймворки и инструменты автоматизации.

Задачи учебной дисциплины:

- изучить основные парадигмы программирования (процедурное, объектно-ориентированное, функциональное) и их применение;
- ознакомиться с алгоритмами и структурами данных, необходимыми для эффективного решения задач;
- разобраться с принципами многопоточности и асинхронного программирования для повышения производительности приложений;
- изучить паттерны проектирования и их роль в построении гибких и поддерживаемых программных систем;
- овладеть навыками работы с системами контроля версий (Git) и методологиями управления проектами (Agile, Scrum);

- изучить современные инструменты разработки, такие как фреймворки для веб-разработки, контейнеризация (Docker) и системы оркестрации (Kubernetes);

- овладеть навыками тестирования программного обеспечения и автоматизации процессов с использованием CI/CD пайплайнов.

Базовой учебной дисциплиной для учебной дисциплины «Технологии программирования» является «Языки программирования». В свою очередь учебная дисциплина «Технологии программирования» является базой для учебной дисциплины компонента учреждения образования «Методы машинного обучения».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Технологии программирования» формируются следующие **компетенции**:

специализированная:

СК-16: строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного и иных парадигм программирования.

В результате изучения учебной дисциплины обучающийся должен:

знать:

- основные парадигмы программирования (процедурное, объектно-ориентированное, функциональное) и их отличия;
- алгоритмы и структуры данных, включая сортировки (пузырьковая сортировка, быстрая сортировка) и структуры данных (списки, стеки, очереди, деревья, графы);
- принципы объектно-ориентированного программирования (ООП): инкапсуляция, наследование, полиморфизм, абстракция;
- основные паттерны проектирования: Singleton, Factory, Observer, Decorator и другие;
- принципы многопоточности и асинхронного программирования;
- основные методы тестирования программного обеспечения: модульное тестирование (юнит-тестирование), интеграционное и функциональное тестирование;
- принципы работы с API (REST, SOAP), основные методы взаимодействия с внешними сервисами;
- принципы работы с системами контроля версий (Git): создание и управление ветками, разрешение конфликтов, слияние изменений;
- основы CI/CD (непрерывная интеграция и доставка) и роль автоматизации в процессах разработки;
- основные методологии управления проектами (Agile, Scrum) и принципы организации командной работы;

уметь:

- разрабатывать программы на одном или нескольких языках программирования (Python, Java, C++ и др.), используя базовые и продвинутые конструкции;

- применять алгоритмы и структуры данных для решения задач с различными требованиями по производительности и ресурсоемкости;

- использовать принципы ООП для проектирования и разработки модульных и масштабируемых систем;

- реализовывать асинхронное программирование и многопоточность для повышения производительности приложений;

- применять паттерны проектирования для построения гибких архитектур программных систем;

- разрабатывать, тестировать и отлаживать программные продукты, используя автоматизированные инструменты для модульного и интеграционного тестирования;

- работать с внешними API, делать запросы к удалённым сервисам и обрабатывать данные;

- использовать систему контроля версий Git для управления проектами, работы в команде и координации изменений в коде;

- настраивать и применять системы CI/CD для автоматического тестирования и развертывания программного обеспечения;

- работать в команде, используя современные методологии управления проектами (Scrum, Kanban), участвовать в планировании, ретроспективах и оценке задач;

владеть:

- навыками написания чистого и поддерживаемого кода, следуя принципам SOLID и другим методологиям разработки;

- техниками работы с инструментами разработки и отладки кода, включая использование интегрированных сред разработки (IDE);

- методами проектирования архитектуры программных систем с использованием паттернов проектирования и современных фреймворков;

- навыками работы с системами контроля версий (создание репозитория, работа с ветками, разрешение конфликтов);

- инструментами для управления процессом разработки в командах, включая работу с GitHub/GitLab для ведения проектов и автоматизации;

- технологиями контейнеризации приложений (Docker) и оркестрации (Kubernetes) для развёртывания программных продуктов;

- средствами автоматизированного тестирования и инструментами CI/CD для внедрения непрерывной интеграции и доставки;

- организацией командной работы, включая планирование задач, распределение ролей и взаимодействие в рамках проекта с использованием Scrum.

Согласно учебному плану, учебная программа изучения дисциплины «Технологии программирования» рассчитана следующим образом:

Курс	3
Семестр	5
Всего часов по учебной дисциплине	108
Аудиторных часов по учебной дисциплине	48
В том числе:	
Лекции, часов	32
Лабораторные занятия, часов	16
Самостоятельная работа студентов	60
Трудоемкость дисциплины, зачетные единицы	3
Форма промежуточной аттестации	экзамен

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. ОСНОВЫ ПРОГРАММИРОВАНИЯ

Тема 1. ВВЕДЕНИЕ В ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Обзор технологий программирования и парадигм: процедурное, объектно-ориентированное, функциональное.

Тема 2. АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

Основы алгоритмов и структур данных, таких как списки, деревья, хеш-таблицы.

Тема 3. ТИПЫ ДАННЫХ И ТИПИЗАЦИЯ

Различие типов данных, строгая и динамическая типизация, системы типов в языках программирования.

Тема 4. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП)

Основные принципы ООП: инкапсуляция, наследование, полиморфизм и абстракция.

Тема 5. ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Введение в парадигму функционального программирования, использование функций высшего порядка и замыканий.

Тема 6. АСИНХРОННОЕ ПРОГРАММИРОВАНИЕ И МНОГОПОТОЧНОСТЬ

Основы асинхронного программирования, работа с потоками, проблемы многопоточности и подходы их решения.

Раздел 2. СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ ПО

Тема 7. ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ

Классические паттерны проектирования, их использование для построения гибких и масштабируемых систем.

Тема 8. МЕТАПРОГРАММИРОВАНИЕ И РЕФЛЕКСИЯ

Принципы метапрограммирования, использование динамической генерации и анализа кода.

Тема 9. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Основы юнит-тестирования, тесты на производительность, автоматизация тестирования.

Тема 10. ФРЕЙМВОРКИ ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЙ

Обзор современных фреймворков (Django, React, Spring), их использование для создания приложений.

Тема 11. МИКРОСЕРВИСНАЯ АРХИТЕКТУРА

Введение в микросервисную архитектуру, её преимущества и проблемы, сравнение с монолитами.

Тема 12. КОНТЕЙНЕРИЗАЦИЯ И ОРКЕСТРАЦИЯ

Основы работы с Docker и Kubernetes для развертывания и управления контейнерами.

Раздел 3. УПРАВЛЕНИЕ ПРОЕКТАМИ И КОМАНДНАЯ РАБОТА

Тема 13. СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ (GIT)

Принципы работы с Git, создание веток, слияние, разрешение конфликтов, коллективная работа над кодом.

Тема 14. МЕТОДОЛОГИИ УПРАВЛЕНИЯ ПРОЕКТАМИ: AGILE И SCRUM

Основы гибких методологий управления проектами, роли в команде и жизненный цикл проекта.

Тема 15. НЕПРЕРЫВНАЯ ИНТЕГРАЦИЯ И РАЗВЕРТЫВАНИЕ (CI/CD)

Введение в автоматизацию сборки, тестирования и развертывания программных систем.

Тема 16. DEVOPS ПРАКТИКИ

Совмещение разработки и эксплуатации, автоматизация процессов, мониторинг и поддержка инфраструктуры.

Учебно-методическая карта учебной дисциплины «Технологии программирования»
Дневная форма получения высшего образования

Дневная форма получения высшего образования									
Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Литература	Формы контроля знаний	
		лекции	практические занятия	лабораторные занятия	управляемая самостоятельная работа студента				
					3	4			5
1	2						7	8	
Пятый семестр									
Раздел 1. Основы программирования									
ВВЕДЕНИЕ В ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ									
Тема 1	Обзор технологий программирования и парадигм: процедурное, объектно-ориентированное, функциональное.	2					[5,6,7,8]	УО	
АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ									
Тема 2	Основы алгоритмов и структур данных, таких как списки, деревья, хеш-таблицы.	2					[5,6,7,8]	УО	
Лабораторная работа №1									
	Основы программирования и базовые конструкции.			2			МУ лр	ЛР*	
ТИПЫ ДАННЫХ И ТИПИЗАЦИЯ									
Тема 3	Различие типов данных, строгая и динамическая типизация, системы типов в языках программирования.	2					[5,6,7,8]	УО	
Лабораторная работа №2									
	Алгоритмы сортировки и работы со структурами данных.			2			МУ лр	ЛР*	

Тема 4	ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП) Основные принципы ООП: инкапсуляция, наследование, полиморфизм и абстракция.	2					[5,6,7,8]	УО
	Лабораторная работа №3 Объектно-ориентированное программирование		2				МУ _{ЛР}	ЛР*
Тема 5	ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ Введение в парадигму функционального программирования, использование функций высшего порядка и замыканий.	2					[5,6,7,8]	УО
Тема 6	АСИНХРОННОЕ ПРОГРАММИРОВАНИЕ И МНОГОПОТОЧНОСТЬ Основы асинхронного программирования, работа с потоками, проблемы многопоточности и подходы их решения.	2					[5,6,7,8]	УО
	Лабораторная работа №4 Асинхронное программирование и многопоточность.		2				МУ _{ЛР}	ЛР*
Раздел 2. Современные технологии разработки ПО								
Тема 7	ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ Классические паттерны проектирования, их использование для построения гибких и масштабируемых систем.	2					[5,6,7,8]	УО
	Лабораторная работа №5 Паттерны проектирования.		2				МУ _{ЛР}	ЛР*
Тема 8	МЕТАПРОГРАММИРОВАНИЕ И РЕФЛЕКСИЯ Принципы метапрограммирования, использование динамической генерации и анализа кода.	2					[5,6,7,8]	УО
	Лабораторная работа №6 Работа с API и интеграция систем.		2				МУ _{ЛР}	ЛР*
Тема 9	ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Основы юнит-тестирования, тесты на производительность, автоматизация тестирования.	2					[5,6,7,8]	УО
	Лабораторная работа №7 Тестирование программного обеспечения.		2				МУ _{ЛР}	ЛР*
Тема 10	ФРЕЙМВОРКИ ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЙ Обзор современных фреймворков (Django, React, Spring), их использование для создания приложений.	2					[5,6,7,8]	УО

Тема 11	МИКРОСЕРВИСНАЯ АРХИТЕКТУРА Введение в микросервисную архитектуру, её преимущества и проблемы, сравнение с монолитами.	2					[5,6,7,8]	УО
Тема 12	КОНТЕЙНИРИЗАЦИЯ И ОРКЕСТРАЦИЯ Основные работы с Docker и Kubernetes для развертывания и управления контейнерами.	2					[5,6,7,8]	УО
Раздел 3. Управление проектами и командная работа								
Тема 13	СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ (GIT) Принципы работы с Git, создание веток, слияние, разрешение конфликтов, коллективная работа над кодом. Лабораторная работа №8 Использование Git и систем контроля версий.	2					[5,6,7,8]	УО
Тема 14	МЕТОДОЛОГИИ УПРАВЛЕНИЯ ПРОЕКТАМИ: AGILE И SCRUM Основные гибких методологий управления проектами, роли в команде и жизненный цикл проекта.	2				2	МУ _{ЛР}	ЛР*
Тема 15	НЕПРЕРЫВНАЯ ИНТЕГРАЦИЯ И РАЗВЕРТЫВАНИЕ (CI/CD) Введение в автоматизацию сборки, тестирования и развертывания программных систем.	2					[5,6,7,8]	УО
Тема 16	DEVOPS ПРАКТИКИ Совмещение разработки и эксплуатации, автоматизация процессов, мониторинг и поддержка инфраструктуры.	2					[5,6,7,8]	УО
Итого за пятый семестр		32				16		

*мероприятия текущего контроля

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ЛИТЕРАТУРА

Основная:

1. Городняя, Л. В. Парадигма программирования : учебное пособие для вузов / Л. В. Городняя. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 232 с. — ISBN 978-5-8114-6680-1. - Текст: электронный // Лань: электронно-библиотечная система. URL: <https://e.lanbook.com/book/151660>. - Режим доступа: для авториз. пользователей.

2. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. - 2 издание, исправленное и дополненное. - Москва : Юрайт, 2023. - 432 с.

3. Васильева, И. Н. Криптографические методы защиты информации : учебник и практикум для вузов / И. Н. Васильева. - Москва : Юрайт, 2023. - 349 с.

4. Деза, Е. И. Введение в криптографию. Теоретико-числовые основы защиты информации : учебное пособие / Е. И. Деза, Л. В. Котова. - издание стереотипное. - Москва : ЛЕНАНД, 2022. - 368 с.

5. Павлов, Л. А. Структуры и алгоритмы обработки данных [Электронный ресурс]: учебник для вузов / Л. А. Павлов, Н. В. Первова. — Санкт-Петербург : Лань, 2021. — 256 с. // Лань: электронно-библиотечная система. — Режим доступа: по подписке: URL: <https://e.lanbook.com/book/156929> (дата обращения: 24.09.2024).

6. Скворцова, Л. А. Структуры и алгоритмы обработки данных [Электронный ресурс]: учебно-методическое пособие / Л. А. Скворцова, К. В. Гусев, С. М. Трушин. — Москва : РТУ МИРЭА, 2021. — 235 с. // Лань : электронно-библиотечная система. — Режим доступа: по подписке: URL: <https://e.lanbook.com/book/218699> (дата обращения: 24.09.2024).

7. Кораблин, Ю. П. Структуры и алгоритмы обработки данных [Электронный ресурс]: учебно-методическое пособие / Ю. П. Кораблин, В. П. Сыромятников, Л. А. Скворцова. — Москва : РТУ МИРЭА, 2020. — 219 с. // Лань: электронно-библиотечная система. — Режим доступа: по подписке: URL: <https://e.lanbook.com/book/163860> (дата обращения: 24.09.2024).

8. Апанасевич, С. А. Структуры и алгоритмы обработки данных. Линейные структуры [Электронный ресурс]: учебное пособие / С. А. Апанасевич. — Санкт-Петербург : Лань, 2022. — 136 с. // Лань: электронно-библиотечная система. — Режим доступа: по подписке: URL: <https://e.lanbook.com/book/206261> (дата обращения: 24.09.2024).

Дополнительная:

9. Кнут, Д. Э. Искусство программирования: учеб. пособие: пер. с англ. Т.1 : Основные алгоритмы / Д. Э. Кнут. - 3-е изд. - М. : Изд. дом Вильямс, 2000. - 712с.

10. Кнут, Д. Э. Искусство программирования: учеб. пособие: пер. с англ. Т.3 : Сортировка и поиск / Д. Э. Кнут. - 2-е изд. - М. : Изд. дом Вильямс, 2000. - 822с.

Е. В. Суркова

11. Котов, Ю.А. Криптографические методы защиты информации: стандартные шифры. Шифры с открытым ключом [Электронный ресурс]/ Ю.А. Котов; Новосибирский государственный технический университет. - Новосибирск : Новосибирский государственный технический университет, 2017. - 67 с. : ил., табл. - Режим доступа: по подлинке. URL: <https://biblioclub.ru/index.php?page=book&id=574782>

12. Кручинин, В.В. Технологии программирования [Электронный ресурс]: учебное пособие / В.В. Кручинин ; Федеральное агентство по образованию, Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). - Томск : ТУСУР, 2013. - 272 с. : ил. – Режим доступа: по подписке: <https://biblioclub.ru/index.php?page=book&id=480536>

13. Аверченков, В.И. Криптографические методы защиты информации [Электронный ресурс]: учебное пособие. / В.И. Аверченков, М.Ю. Рытов, С.А. Шпичак. - 2-е изд., стер. - Москва : ФЛИНТА, 2017. – 201 с. - Режим доступа: по подписке: <https://biblioclub.ru/index.php?page=book&id=574862>.

14. Дроздов, С.Н. Структуры и алгоритмы обработки данных [Электронный ресурс]: учебное пособие / С.Н. Дроздов; Министерство образования и науки РФ, Южный федеральный университет, Инженерно-технологическая академия. Таганрог: Издательство Южного федерального университета, 2016. - 228 с. - Режим доступа: по подписке: <http://biblioclub.ru/index.php?page=book&id=493032>

15. Ролич, О.Ч. Технологии программирования: курс лекций / О.Ч. Ролич. - Минск: БГУ, 2008. - 144 с.

16. Костюкова, Н.И. Комбинаторные алгоритмы для программистов [Электронный ресурс]: / Н.И. Костюкова. - 2-е изд, исправ/. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. 217 с.: ил. - Режим доступа: по подписке: <https://biblioclub.ru/index.php?page=book&id=429067>

ПЕРЕЧЕНЬ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Основы программирования и базовые конструкции.
2. Алгоритмы сортировки и работы со структурами данных
3. Объектно-ориентированное программирование
4. Асинхронное программирование и многопоточность
5. Паттерны проектирования.
6. Работа с API и интеграция систем.
7. Тестирование программного обеспечения.
8. Использование Git и систем контроля версий.

ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ, НАГЛЯДНЫХ И ДРУГИХ ПОСОБИЙ, МЕТОДИЧЕСКИХ УКАЗАНИЙ И МАТЕРИАЛОВ, ТЕХНИЧЕСКИХ СРЕДСТВ ОБУЧЕНИЯ, ОБОРУДОВАНИЯ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

1. ПЭВМ с основной конфигурацией не хуже процессор – Intel Celeron, объем ОЗУ – 2Гб.
2. Операционная система – Windows 7 (8 или 10).
3. Языки программирования: Python, Java, C++, JavaScript и другие.
4. Средства для разработки и тестирования: IDE (Visual Studio Code, PyCharm), инструменты для автоматизации тестирования (JUnit, pytest).
5. Инструменты контейнеризации: Docker, Kubernetes.
6. Системы контроля версий: Git, GitHub/GitLab.
7. Платформы для CI/CD: Jenkins, GitHub Actions, GitLab CI/CD.

ПЕРЕЧЕНЬ ВОПРОСОВ ДЛЯ ПРОВЕДЕНИЯ ЭКЗАМЕНА

1. Что такое программирование и какие существуют его основные парадигмы?
2. Чем отличаются процедурное, объектно-ориентированное и функциональное программирование?
3. Объясните понятие переменной и какие типы данных могут использоваться в программировании.
4. В чем отличие статической и динамической типизации?
5. Что такое структуры данных и каковы их основные виды?
6. Опишите основные структуры данных: списки, стеки, очереди и их применение.
7. В чем заключается принцип работы алгоритма сортировки пузырьком?
8. Как работает алгоритм быстрой сортировки (QuickSort)?
9. Что такое инкапсуляция, наследование, полиморфизм и абстракция в ООП?
10. Объясните, что такое класс и объект в объектно-ориентированном программировании.
11. Как реализовать наследование в ООП и в чем его преимущество?
12. Как работает механизм полиморфизма в объектно-ориентированных языках программирования?

13. Что такое интерфейсы и абстрактные классы? В чем их различие?
14. Чем функциональное программирование отличается от процедурного?
15. Объясните принцип работы функций высшего порядка и приведите примеры их использования.
16. Что такое замыкания и как они используются в функциональном программировании?
17. Какие основные механизмы используются для реализации многопоточности в современных языках программирования?
18. Каковы основные преимущества и недостатки многопоточности?
19. Что такое асинхронное программирование и как оно помогает улучшить производительность программ?
20. Что такое паттерны проектирования и зачем они нужны?
21. Объясните паттерн Singleton и его применение.
22. Что такое паттерн "Наблюдатель" (Observer) и где он используется?
23. Чем отличаются структурные и поведенческие паттерны проектирования?
24. Какова роль метапрограммирования в современных языках программирования?
25. Что такое рефлексия и как её можно использовать в приложениях?
26. Объясните понятие модульного тестирования (юнит-тестирования) и его назначение.
27. Чем интеграционное тестирование отличается от модульного?
28. Какие существуют методы автоматизированного тестирования программного обеспечения?
29. Объясните, что такое тестирование на производительность и как его проводить.
30. В чем заключается основное преимущество использования фреймворков при разработке программного обеспечения?
31. Какую роль играют ORM-фреймворки при работе с базами данных?
32. Опишите основные принципы работы с микросервисной архитектурой.
33. Чем микросервисная архитектура отличается от монолитной архитектуры?
34. Как микросервисы взаимодействуют друг с другом в распределенной системе?
35. Какие инструменты используются для контейнеризации приложений?
36. Какова роль Docker в контейнеризации и развертывании приложений?
37. Что такое Kubernetes и как он используется для оркестрации контейнеров?
38. Каковы основные преимущества использования контейнеров в разработке ПО?
39. В чем заключается процесс интеграции сторонних API в программное обеспечение?
40. Какие существуют методы для обработки ошибок при взаимодействии с внешними API?
41. Что такое REST API и каковы его основные принципы?
42. Чем отличается REST от SOAP и какие преимущества у REST API?

43. Какие существуют способы аутентификации и авторизации при работе с API?
44. Что такое система контроля версий (VCS) и зачем она нужна?
45. Какие задачи решает Git в разработке программного обеспечения?
46. Объясните процесс создания ветки в Git и зачем это нужно.
47. Как происходит процесс слияния веток в Git и что такое конфликт слияния?
48. Как происходит управление версиями в команде с помощью Git?
49. Что такое "pull request" и как его используют в командной разработке?
50. Объясните основные принципы методологии управления проектами Agile.
51. Что такое Scrum и как он применяется в рамках Agile-подхода?
52. В чем разница между Agile и Waterfall методологиями?
53. Какие роли выделяются в Scrum-команде и каковы их обязанности?
54. Что такое спринт и какие этапы он включает?
55. Что такое CI/CD и как оно улучшает процесс разработки ПО?
56. Какие инструменты используются для автоматизации CI/CD процессов?
57. Объясните, как работает пайплайн в системах CI/CD.
58. Каковы основные этапы процесса деплоя при использовании CI/CD?
59. Что такое DevOps и каковы его ключевые принципы?
60. В чем заключается взаимосвязь между DevOps и автоматизацией процессов разработки?

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

При изучении дисциплины используются следующие формы самостоятельной работы:

- подготовка к устным опросам на лекции;
- самостоятельное изучение теоретического материала;
- подготовка к защите отчетов по лабораторным работам;
- систематизация полученных знаний при подготовке к экзамену.

Содержание самостоятельной работы студентов дневной формы получения образования

Вид самостоятельной работы	Тематическое содержание и используемые источники	Количество часов
1	2	3
Подготовка к устным опросам на лекции	Тема 1 Литература: [1,2,3,4]	2
	Тема 2 Литература: [1,2,3,4]	2
	Тема 3 Литература: [1,2,3,4]	2
	Тема 4 Литература: [1,2,3,4]	2
	Тема 5 Литература: [1,2,3,4]	2
	Тема 6 Литература: [1,2,3,4]	2
	Тема 7 Литература: [1,2,3,4]	2
	Тема 8 Литература: [1,2,3,4]	2
	Тема 9 Литература: [1,2,3,4]	2
	Тема 10 Литература: [1,2,3,4]	2

Подготовка к защите отчетов по лабораторным работам	Тема 11 Литература: [1,2,3,4]	2
	Тема 12 Литература: [1,2,3,4]	2
	Тема 13 Литература: [1,2,3,4]	2
	Тема 14 Литература: [1,2,3,4]	2
	Тема 15 Литература: [1,2,3,4]	2
	Тема 16 Литература: [1,2,3,4]	2
	Лабораторная работа № 1 [МУ _{ЛР}]	1
	Лабораторная работа № 2 [МУ _{ЛР}]	1
	Лабораторная работа № 3 [МУ _{ЛР}]	1
	Лабораторная работа № 4 [МУ _{ЛР}]	1
	Лабораторная работа № 5 [МУ _{ЛР}]	1
	Лабораторная работа № 6 [МУ _{ЛР}]	1
	Лабораторная работа № 7 [МУ _{ЛР}]	1
	Лабораторная работа № 8 [МУ _{ЛР}]	1
	Систематизация полученных знаний при подготовке к экзамену	20
	Итого за пятый семестр:	60

Перечень дополнительного информационного и учебно-методического обеспечения самостоятельной работы студентов, размещенного в Google Classroom университета:

1. Конспект лекций.
2. Методические указания к выполнению лабораторных работ.
3. Список вопросов к экзамену.

КОНТРОЛЬ КАЧЕСТВА УСВОЕНИЯ ЗНАНИЙ

Средства диагностики результатов учебной деятельности:

- устный опрос на лекции;
- отчет по лабораторным и с их устной защитой;
- экзамен;

Контроль качества усвоения знаний проводится в соответствии с Положением о рейтинговой системе оценки знаний и компетенций студентов (приказ ректора университета от 06.06.2014 № 294 (в редакции, утвержденной приказом ректора университета от 17.11.2014 № 605) в форме текущего контроля и промежуточной аттестации.

Результат текущего контроля за семестр (дневная форма получения образования, 5 семестр) оценивается отметкой в баллах по десятибалльной шкале и выводится, исходя из отметок, выставленных в ходе проведения мероприятий текущего контроля в течение семестра по следующей формуле:

$$T = \frac{LP_1 + \dots + LP_m}{k}$$

$LP_1 + \dots + LP_m$ – отметки, выставленные по результатам устных защит отчетов по лабораторным работам.

Результат текущего контроля рассчитывается как округленное среднее значение.

Промежуточная аттестация проводится в форме экзамена.

Итоговая экзаменационная отметка по дисциплине рассчитывается по формуле:

$$ИЭ = k \cdot T + (1 - k) \cdot O$$

где k – весовой коэффициент текущего контроля;

T – результат текущего контроля за семестр;

O – отметка, полученная студентом на экзамене за ответ по билету.

Весовой коэффициент k принимается равным 0,5. Информация о весовом коэффициенте доводится до студентов на первом занятии в семестре. Положительной является экзаменационная отметка не ниже 4 баллов.

ХАРАКТЕРИСТИКА ИННОВАЦИОННЫХ ПОДХОДОВ К ПРЕПОДАВАНИЮ УЧЕБНОЙ ДИСЦИПЛИНЫ

Изучение дисциплины осуществляется на лекционных и лабораторных занятиях. На лекционных занятиях студенты получают теоретические сведения о технологиях программирования направленные на развитие активного и осознанного обучения, адаптацию к изменениям в технологиях и потребностям рынка. В ходе изложения лекционного материала по дисциплине активно используются: проблемно-модульное изложение материала; традиционные словесные приемы и методы, которые активизируются постановкой проблемных вопросов и заданий, организацией учебных дискуссий в опоре на имеющуюся начальную подготовку студентов и их политехнический кругозор; интерактивные методы обучения.

На лабораторных занятиях развиваются и формируются необходимые практические умения и навыки применения языков программирования для решения различных задач. Во время проведения занятий особое внимание уделяется формированию у студентов умения планировать работу, определять эффективную последовательность ее выполнения.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ С ДРУГИМИ УЧЕБНЫМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Методы машинного обучения	Кафедра технологий программирования	<i>Предложено нет</i>	

Заведующий кафедрой
технологий программирования
канд. техн. наук, доцент



В.М. Чертков