МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Казанский национальный исследовательский технологический университет»
(ФГБОУ ВО «КНИТУ»)
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
(ИДПО ФГБОУ ВО «КНИТУ»)

УТВЕРЖДАЮ
Директор ИДПО ФГБОУ ВО «КНИТУ»
_____ Ю.Н.Зиятдинова
3 сентября 2024 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

«Программирование и IT-технологии»
(745 акад. часов)

Лицензия ФГБОУ ВО «КНИТУ» серия 90Л01, № 0009203, рег. №2165 от 27.05.2016

Программа утверждена на заседании учебно-методической комиссии ИДПО ФГБОУ ВО «КНИТУ» (протокол от 02 09 2024 г. № 8)

Председатель учебно-методической комиссии _____ В.В. Кондратьев

Программа утверждена на заседании Ученого совета ИДПО ФГБОУ ВО «КНИТУ» (протокол от 03 09 .2024 г. № 7)

Председатель Ученого совета ИДПО ФГБОУ ВО «КНИТУ»
_____ Ю.Н.Зиятдинова

Казань, 2024 г.

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель реализации программы

Формирование у слушателей компетенций в области разработки программного обеспечения, информационных технологий и цифровой трансформации бизнеса. Программа направлена на подготовку квалифицированных кадров, обладающих необходимыми знаниями и практическими навыками для эффективного решения профессиональных задач в сфере программирования и IT-индустрии. Она обеспечивает возможность освоения современных методов проектирования, анализа, тестирования и сопровождения ПО, формирования умения применять современные инструменты и технологии.

1.2 Планируемые результаты обучения

Слушатель должен знать:

– Основные принципы и подходы к разработке программного обеспечения.
– Современные языки программирования и платформы разработки.
– Инструменты управления проектами и командной работой.

Слушатель должен уметь:

– Проектировать и разрабатывать программное обеспечение различного уровня сложности.
– Проводить тестирование и отладку программных продуктов.
– Использовать системы контроля версий и инструменты коллективной разработки.
– Применять методы оптимизации производительности и масштабирования приложений.
– Организовывать и управлять проектной деятельностью в команде разработчиков.

Слушатель должен владеть:

– Навыками разработки программного обеспечения на различных языках программирования.
– Методиками проектирования баз данных и организации хранения информации.
– Технологиями интеграции приложений и API-интерфейсов.
– Практическим опытом работы с системами виртуализации и облачными технологиями.
– Умениями документирования и представления результатов своей работы.
– Профессиональной терминологией и коммуникационными навыками взаимодействия с заказчиками и коллегами.

1.3. Формируемые компетенции

Выпускник, освоивший программу, должен обладать профессиональными компетенциями, соответствующими видам профессиональной деятельности, на которые ориентирована программа:

– Способность анализировать требования заказчика и формулировать технические задания на разработку программного продукта.
– Возможность эффективно организовывать процессы разработки и тестирования программного обеспечения.
– Владение методами распределенных систем обработки данных.
– Понимание принципов сетевого взаимодействия и протоколов передачи данных.
– Навык администрирования серверных платформ и операционных систем.
– Способность понимать бизнес-процессы компаний и адаптировать информационные технологии под конкретные нужды предприятия.
– Осведомленность о перспективах развития цифровых технологий и умение своевременно внедрять инновационные решения.
– Коммуникативные способности для успешного взаимодействия с представителями разных подразделений организаций.
– Гибкость мышления и способность быстро осваивать новые технологии и подходы в условиях динамично меняющегося рынка информационных услуг.

1.4. Программа разработана на основе:
профессионального стандарта Специалист по информационным системам (приказ Минтруда России от 13.07.2023 № 586н)

требований ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии (бакалавриат), утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 926

Категория слушателей: *студенты, аспиранты, преподаватели и другие категории граждан.*

Общий объем занятий: *745 ак. часов*

Срок обучения: 3 уч. года

Форма обучения: очно-заочная

Календарный учебный график

Образовательный процесс по программе может осуществляться в течение всего учебного года. Занятия проводятся по мере комплектования групп.

График обучения **Форма обучения**	**Ауд. часов** **в день**	**Дней** **в неделю**
Очно-заочная	3	2

Учебный план

№	Наименование дисциплины	ОТ час.	Аудиторные/ занятия, час.		СРС	Форма контроля
			Лк	ПЗ, СЗ, ЛЗ	час	
1	2	3	4	5	6	7
1.	Язык программирования Python	57	21	21	15	зачет
2.	Основы верстки и веб-разработки на языке JavaScript	57	21	21	15	зачет
3.	Интеграция QA-практик в разработку офисных решений	60	21	18	21	экзамен
4.	Фреймворк Django	57	21	15	21	зачет
5.	Алгоритмы и структуры данных	96	30	21	45	зачет экзамен
6.	Создание интерактивных web приложений на Фреймворк Django	63	15	27	21	экзамен
7.	Электронные презентации	39	9	9	21	зачет
8.	Разработка web-приложений с использованием фреймворков языка Python (FastAPI, Flask, Django)	57	15	21	21	зачет
9.	Операционные системы	39	9	12	18	зачет
10.	SQL и база данных	54	15	21	18	зачет
11.	Мультимедиатехнологии и тестирование пользовательского опыта (UX)	49	18	21	10	зачет
12.	Тестирование офисных приложений и автоматизация процессов	49	12	12	25	экзамен
13.	Современные JavaScript-фреймворки для разработки web-приложений	56	15	21	20	зачет
14.	Курсовая работа	10			10	диф. зачет
15.	Итоговая аттестация	2			2	итоговый междисциплинарный экзамен
Итого		**745**	**222**	**240**	**283**	

** ОТ – общая трудоемкость, Лк – лекции, ПЗ – практические занятия, СЗ – семинарские занятия, ЛЗ – лабораторные занятия, ДОТ – дистанционные образовательные технологии, СРС – самостоятельная работа слушателя*

Содержание учебных дисциплин (модулей)

Дисциплина 1. Язык программирования Python

№ п/п	Наименование тем	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в Python	История возникновения и развития языка Python. Установка интерпретатора и настройка рабочей среды. Особенности синтаксиса и структуры языка. Переменные, типы данных и операции. Базовые конструкции управления потоком выполнения (условия, циклы).
2.	Функциональное программирование	Понятие функций, аргументов и возвращаемых значений. Лямбда-функции и анонимные функции. Рекурсия и её применение. Модули и пакеты в Python. Работа с исключительными ситуациями и обработка ошибок.
3.	Объектно-ориентированное программирование	Классы и объекты, наследование и полиморфизм. Конструкторы и деструкторы объектов. Свойства классов и метаклассы. Декораторы и их использование. Принципы SOLID и паттерны проектирования.
4.	Библиотеки и фреймворки	Стандартная библиотека Python. Использование сторонних библиотек (NumPy, Pandas, Matplotlib). Веб-разработка с использованием Django и Flask. Автоматизация задач и работа с файлами. Тестирование и профилирование кода.
5.	Продвинутые темы	Многопоточность и асинхронное программирование. Оптимизация производительности и производительность памяти. Сборка и упаковка проектов. Интеграция Python с другими языками и средами. Машинное обучение и Data Science с использованием Python.
Практические и/или семинарские занятия		Не предусмотрено
Лабораторные занятия		Практическая часть курса предполагает выполнение лабораторных работ и проектов, направленных на закрепление изученного материала. Студенты выполняют задания по созданию небольших программ, анализу данных, работе с внешними источниками данных и разработкой веб-приложений.
Самостоятельная работа слушателя		Выполнение домашних заданий и индивидуальных проектов. Изучение дополнительной литературы и документации по языку Python. Участие в онлайн-курсах и специализированных платформах (например, Coursera, Stepik, Habr Academy). Проведение самостоятельных исследований и экспериментов с кодом. Подготовка презентаций и докладов по отдельным тематикам курса.
Используемые образовательные технологии		Интерактивные лекции: активное обсуждение теории, демонстрация примеров и разбор сложных вопросов. Лабораторные работы: приобретение опыта работы с различными библиотеками и инструментами Python в ходе выполнения учебных задач. Проектная деятельность: создание собственного программного продукта или участие в реальных проектах.

Дисциплина 2. Основы верстки и веб-разработки на языке JavaScript

№ п/п	Наименование тем	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в JavaScript и основы веб-разработки	Роль JavaScript в современной веб-разработке Подключение JavaScript к HTML (inline, <script>, внешние файлы) Основы синтаксиса: переменные, типы данных, операторы Управляющие конструкции (условия, циклы) Функции (объявление, вызов, параметры, замыкания)
2.	Работа с DOM (Document Object Model)	Что такое DOM и его структура Методы поиска элементов (getElementById, querySelector и др.) Изменение содержимого, атрибутов и стилей элементов События (click, submit, keypress и др.) и обработчики событий Динамическое создание и удаление элементов
3.	Асинхронность в JavaScript	Понятие асинхронности Callback-функции и их ограничения Промисы (Promise) и их методы (then, catch, finally) Async/Await для работы с асинхронным кодом Fetch API для работы с HTTP-запросами
4.	Работа с формами и валидация	Получение данных из форм Валидация вводимых данных (регулярные выражения, кастомные проверки) Отправка данных на сервер (AJAX, Fetch) Обработка ошибок при отправке данных
5.	Хранение данных на клиенте	Cookies: чтение и запись LocalStorage и SessionStorage Индексированные базы данных (IndexedDB)
6.	Основы работы с библиотеками и фреймворками	Подключение сторонних библиотек (jQuery, Lodash) Обзор современных фреймворков (React, Vue, Angular) Знакомство с Node.js и npm/yarn
7.	Тестирование и отладка кода	Инструменты разработчика в браузере Консольные методы (console.log, console.error и др.) Тестирование с помощью Jest или Mocha Линтеры (ESLint) и форматирование (Prettier)
Практические и/или семинарские занятия		Написание простых скриптов (калькулятор, конвертер валют) Разработка "To-Do List" с добавлением/удалением задач Создание модального окна по клику на кнопку Загрузка данных с JSONPlaceholder API Создание погодного приложения с OpenWeatherMap API Форма регистрации с валидацией email/пароля Интеграция с mock-сервером (например, MockAPI)
Лабораторные занятия		Не предусмотрены
Самостоятельная работа слушателя		Изучение MDN Web Docs (раздел "JavaScript Basics") Создание мини-проекта: "Таймер обратного отсчета" Реализация игры "Кликер" (счетчик кликов + улучшения) Изучение Event Delegation (разбор примеров) Разработка "Чтение RSS-ленты" (парсинг JSON/XML) Конспект: "Разница между Promise.all и Promise.race" Реализация формы обратной связи с отправкой в Telegram (через бота)
Используемые образовательные технологии		Интерактивные тренажеры (Codecademy, freeCodeCamp) Видеолекции + конспектирование в Notion Работа в парах (peer programming) Разбор кода через CodePen/JSFiddle Демонстрация через Postman (REST API) Геймификация: соревнование "Кто быстрее получит данные" Интерактивные формы на Google Forms (примеры валидации)

Дисциплина 3. Интеграция QA-практик в разработку офисных решений

№ п/п	Наименование тем	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в QA и тестирование офисных решений	Введение в понятие Quality Assurance (QA). Отличие QA от QC (Quality Control). Зачем нужен QA в разработке офисных решений. Основные этапы цикла разработки и их связь с QA.
2.	Подходы и стратегии тестирования	Типы тестирования: модульное, интеграционное, системное, нагрузочное, регрессионное. Автоматизированное vs ручное тестирование. Лучшие практики разработки тестовых наборов. Инструменты для автоматизации тестирования.
3.	Методологии и процессы QA	Waterfall, Agile, Scrum и Kanban — совместимость с QA-практиками. Документирование и управление багами (bug tracking systems). Внедрение CI/CD и DevOps в процессы QA. Метрики и индикаторы качества ПО.
4.	Автоматизация тестирования офисных решений	Инструменты автоматизации тестирования: Selenium, Appium, JUnit, NUnit, PyTest и др. Настройка автотестов и интеграций с системами сборки и деплоя. Плюсы и минусы автоматизации тестирования офисных приложений.
5.	Тестирование офисного ПО на платформе Microsoft Office	Специфика тестирования Excel, Word, PowerPoint и Outlook. API и макросы VBA — инструменты для упрощения тестирования. Взаимодействие с облачными сервисами Microsoft (Sharepoint, OneDrive).
6.	Тестирование офисных решений на платформе Google Workspace	Особенности тестирования Docs, Sheets, Slides и Forms. Интеграции с облачным хранилищем Google Drive. Практические советы по ускорению тестирования облачных офисных решений.
Практические и/или семинарские занятия		Практическое занятие по установке и настройке инструмента Selenium WebDriver. Семинар по сравнению ручного и автоматизированного тестирования в проекте. Практическое занятие по созданию и запуску автотестов для Sharepoint. Проектная работа по внедрению CI/CD для проекта офисного приложения.
Лабораторные занятия		Лабораторная работа по созданию набора модульных тестов для Word Add-In.
Самостоятельная работа слушателя		Чтение рекомендованной литературы и изучение документации по QA и тестированию. Подготовка и выполнение индивидуальных проектов по автоматизации тестирования офисного решения. Самостоятельное изучение популярных инструментов для тестирования офисных приложений. Подготовка отчета по самостоятельному исследованию QA-процессов в существующих проектах.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Практические занятия с непосредственным созданием и испытанием автотестов. Использование инструментальных средств для демонстрации практических навыков (Selenium IDE, Visual Studio Code, Postman и др.) Использование дистанционных форматов обучения (видеоконференции, онлайн-чаты, форумы).

	Обратная связь и индивидуальные консультации преподавателя.

Дисциплина 4. Фреймворк Django

№ п/п	Наименование тем	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в Django	Установка и настройка Django. Основные компоненты Django: Models, Views, Templates, URLs. Проектная структура Django и начальная конфигурация.
2	Основы работы с базой данных	Модельные классы и настройки ORM. CRUD-операции с объектами через ORM. Создание и модификация таблиц, индексов и ограничений.
3	Шаблоны и представление данных	Система шаблонов Django и шаблонные фильтры. Передача данных из View в Template. Отображение списков и детализированных страниц.
4.	Маршрутизация и обработчики запросов	URLconf и настройка маршрутов. Обработчики GET и POST-запросов. Встроенные классы CBV (Class-Based Views).
5.	Пользователи и аутентификация	Модели пользователей и регистрация аккаунта. Аутентификация и авторизация. Разграничение прав доступа.
6.	Формы и обработка ввода	Создание HTML-форм с помощью Django Form API. Валидация данных, обработка ошибок и AJAX-валидация. CSRF-защита и безопасность.
7.	API и RESTful сервисы	Создание REST API с помощью Django Rest Framework. Авторизация и токены JWT. Рабочее взаимодействие фронтенда и бекенда.
8.	Развёртывание и оптимизация	Настройки сервера и production-конфигурация. Работа с GitHub/GitLab и continuous integration. Deployment на Heroku, Docker и AWS
Практические и/или семинарские занятия		Практическое занятие по созданию простейшего Django-проекта. Практическое занятие по созданию форм и обработке HTTP-запросов. Проектная работа по развёртыванию Django-приложения на облачном хостинге.
Лабораторные занятия		Лабораторная работа по настройке ORM и баз данных. Лабораторная работа по созданию REST API с Django Rest Framework.
Самостоятельная работа слушателя		Чтение рекомендованной литературы и изучение документации Django. Самостоятельное выполнение лабораторных работ и заданий по созданию Django-приложений. Подготовка и защита проекта по разработке полнофункционального веб-приложения. Изучение дополнительных библиотек и инструментов для Django-разработки.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Практические занятия с пошаговым созданием и тестированием Django-приложений. Использование инструментальных средств (PyCharm, VSCode, PostgreSQL, MySQL, GitHub Actions, Docker и др.). Онлайн-курсы и вебинары по продвинутым темам Django-разработки. Обратная связь и индивидуальные консультации преподавателя.

Дисциплина 5. Алгоритмы и структуры данных

№ п/п	Наименование тем	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в язык программирования C#	История и предназначение языка C#. Основные компоненты среды разработки .NET. Первая программа на C#.
2	Основы синтаксиса и операторов	Типы данных и операторы. Ветвления (if, switch-case). Циклы (for, while, foreach).
3	Объектно-ориентированное программирование	Классы и объекты. Наследование, инкапсуляция и полиморфизм. Интерфейсы и абстрактные классы.
4.	Работа с коллекциями и LINQ	Стандартные коллекции (.NET Collections). Generics и IEnumerable. LINQ-запросы и lambda-выражения.
5	Работа с файлами и сериализация	Потоки и файлы (FileStream, StreamReader/Writer). XML и JSON сериализация. Работа с файловой системой Windows.
6	Графический интерфейс и WPF	Основы графического интерфейса (WPF). Binding и MVVM-паттерн. Разработка GUI-приложений на C#.
Практические и/или семинарские занятия		Практическое занятие по созданию простой консольной программы на C#. Практическое занятие по работе с коллекциями и LINQ. Практическое занятие по многопоточному программированию.
Лабораторные занятия		Лабораторная работа по созданию классов и объектов. Лабораторная работа по сериализации и десериализации данных.
Самостоятельная работа слушателя		Чтение рекомендованной литературы и изучение документации по C#. Самостоятельное выполнение лабораторных работ и заданий по созданию программ на C#. Подготовка и защита проекта по разработке приложения на C#. Изучение дополнительных библиотек и инструментов для C#.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Практические занятия с пошаговым созданием и тестированием программ на C#. Использование инструментальных средств (Visual Studio, ReSharper, JetBrains Rider и др.). Онлайн-курсы и вебинары по продвинутым темам C# и .NET. Обратная связь и индивидуальные консультации преподавателя.

Дисциплина 6. Создание интерактивных web-приложений на Фреймворк Django

№ п/п	Наименование разделов (тем, элементов и т.д.)	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в интерактивные веб-приложения	Понятие интерактивности и необходимость интерактивных элементов. Отличия статических и динамических веб-приложений. Современные подходы к разработке интерактивных приложений.
2.	Основы JavaScript и DOM	Введение в JavaScript: переменные, функции, объекты. Работа с Document Object Model (DOM). Основные события браузера и их обработка.

3.	Взаимодействие с сервером через AJAX	Основы AJAX и его реализация на стороне клиента. Отправка асинхронных запросов к серверу Django. Получение и отображение данных на странице без перезагрузки страницы.
4.	Front-end библиотека jQuery	Использование jQuery для упрощения работы с DOM. Анимации и эффектные переходы. Интеграция jQuery с Django приложениями.
5.	Интерактивные формы и Django Forms	Создание интерактивных форм с поддержкой JavaScript. Валидация полей формы на стороне клиента и сервера. Использование Custom Validation в Django Forms.
6.	Создание RESTful API на Django	Основные принципы RESTful API. Создание API Endpoints с помощью Django Rest Framework. Доступ к данным через API и их отображение на фронте.
7.	Ajax и взаимодействие с API	Асинхронные запросы к REST API. Обработка ответов от API и обновление UI без перезагрузки страницы. Best practices при работе с AJAX и RESTful API.
8.	Безопасность и производительность	Cross-Site Request Forgery (CSRF) защита. Безопасность JavaScript и XSS атаки. Оптимизация загрузки страниц и производительности приложения.
9.	Реализация Fullstack приложения	Проектирование структуры Fullstack-приложения. Полноценная интеграция backend (Django) и frontend (JavaScript/jQuery). Практическое воплощение интерактивных элементов и клиент-серверного взаимодействия.
Практические и/или семинарские занятия		Практическое занятие по созданию интерактивных элементов с использованием JavaScript. Практическое занятие по созданию RESTful API с Django Rest Framework. Проектная работа по разработке и реализации интерактивного веб-приложения.
Лабораторные работы		Лабораторная работа по интеграции AJAX-запросов в Django-приложение. Лабораторная работа по созданию полноценного fullstack-приложения с использованием Django и JavaScript.
Самостоятельная работа слушателя		1. Чтение рекомендованной литературы и изучение документации по JavaScript и Django. 2. Самостоятельное выполнение лабораторных работ и заданий по созданию интерактивных веб-приложений. 3. Подготовка и защита проекта по разработке интерактивного веб-приложения. 4. Изучение дополнительных библиотек и инструментов для JavaScript и Django.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Практические занятия с пошаговым созданием и тестированием интерактивных веб-приложений. Использование инструментальных средств (PyCharm, VSCode, Chrome Developer Tools, PostgreSQL, MySQL, Django Rest Framework и др.). Онлайн-курсы и вебинары по продвинутым темам Django и JavaScript. Обратная связь и индивидуальные консультации преподавателя.

Дисциплина 7. Электронные презентации

№ п/п	Наименование разделов (тем, элементов и т.д.)	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в электронные презентации	Понятие и назначение электронных презентаций. История и развитие технологий презентаций. Основные инструменты для создания презентаций (Microsoft PowerPoint, Apple Keynote, Google Slides).
2.	Основы дизайна и композиции	Цветовые палитры и шрифты. Иллюстрации, фотографии и иконки. Layouts и Grid Systems.

3.	Создание эффективных слайдов	Принципы хорошего слайда: минимализм, ясность, логика. Вставка медиа-элементов (текст, изображения, видео, анимация). Оптимизация контента для удобства восприятия.
4.	Дополнительные инструменты и технологии	Использование дополнений и шаблонов для PowerPoint. Cloud-презентации и совместная работа (Google Slides). Live-анимация и интерактивные элементы.
5.	Подготовка презентаций для различных аудиторий	Презентации для образовательных целей. Презентации для бизнеса и продажи. Презентации для публичных выступлений и конференций.
Практические и/или семинарские занятия		Практическое занятие по созданию простой презентации в PowerPoint. Практическое занятие по вставке и оптимизации медиа-контента. Подготовка и защита проекта по созданию презентации для определенной аудитории. Практическое занятие по проведению презентаций и работе с аудиторией.
Лабораторные работы		Не предусмотрены
Самостоятельная работа слушателя		Чтение рекомендованной литературы и учебников по курсу. Подготовка реферата по одной из тем дисциплины. Самостоятельное создание и оформление презентации. Выполнение расчетных заданий по визуализации данных.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Использование мультимедийных средств и интернет-ресурсов.

Дисциплина 8. Разработка web-приложений с использованием фреймворков языка Python (FastAPI, Flask, Django)

№ п/п	Наименование разделов (тем, элементов и т.д.)	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в экосистему Python для веб-разработки	Роль Python в бэкенд-разработке. Синхронная и асинхронная модели работы. Обзор фреймворков Django, Flask, FastAPI
2.	Основы веб-технологий и протокол HTTP	Архитектура клиент-сервер и жизненный цикл запроса. Методы HTTP, заголовки, коды состояния.
3.	Микрофреймворк Flask	Архитектура Flask-приложения. Маршрутизация и обработка запросов. Шаблоны Jinja2 для разделения логики и представления. Расширения (Flask-SQLAlchemy, Flask-Login).
4.	Высокопроизводительный FastAPI	Типизация данных, асинхронность, автоматическая документация (Swagger/ReDoc). Модели Pydantic для валидации. Маршруты и параметры запросов. Real-time взаимодействие через WebSocket.
5.	Полнофункциональный Django	Архитектурный паттерн MVT (Model-View-Template). Структура проекта и приложений. Django ORM для работы с базой данных. Система администрирования Django Admin. Шаблоны и статические файлы.
Практические и/или семинарские занятия		Создание сайта-визитки на Flask. Реализация асинхронного REST API на FastAPI. Разработка приложения на Django.
Лабораторные работы		Не предусмотрены
Самостоятельная работа слушателя		Изучение официальной документации фреймворков. Выполнение лабораторных работ и курсового проекта. Сравнительный анализ применимости фреймворков.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Практические занятия с пошаговым созданием и тестированием веб-приложений. Обратная связь и индивидуальные консультации преподавателя.

Дисциплина 9. Операционные системы

№ п/п	Наименование тем	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в операционные системы	Понятие и назначение операционных систем. История развития операционных систем. Классификация и типы операционных систем.
2.	Архитектура операционных систем	Ядро и пользовательское пространство. Процессы и потоки. Многозадачность и многопоточность.
3	Управление процессами	Планирование и диспетчеризация процессов. Синхронизация и взаимоисключение. Проблемы гонок и тупиков.
4.	Управление памятью	Виртуальная память и страничная организация. Управление физической памятью. Подкачка и свопинг.
5	Файловые системы	Организация файлов и каталогов. Методы доступа к файлам. Безопасность и права доступа.
6	Ввод-вывод и драйверы устройств	Устройство ввода-вывода и каналы передачи данных. Драйверы устройств и работа с ними. Буферизация и кэширование.
7	Безопасность операционных систем	Уязвимости и угрозы. Защита и шифрование данных. Обеспечение целостности и доступности.
Практические и/или семинарские занятия		Практическое занятие по управлению процессами в Linux. Практическое занятие по организации ввода-вывода. Практическое занятие по работе с драйверами устройств.
Лабораторные занятия		Лабораторная работа по работе с файловой системой. Лабораторная работа по настройке безопасности в Windows.
Самостоятельная работа слушателя		Чтение рекомендованной литературы и учебников по курсу. Подготовка реферата по одной из тем дисциплины. Самостоятельное выполнение лабораторных работ и заданий по управлению операционной системой. Выполнение расчетных заданий по моделированию работы ОС.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Использование мультимедийных средств и интернет-ресурсов. Обратная связь и индивидуальные консультации преподавателя. Решение кейсов и задач, близких к реальным жизненным ситуациям.

Дисциплина 10. SQL и база данных

№ п/п	Наименование разделов (тем, элементов и т.д.)	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1	Введение в базы данных	Понятие базы данных и СУБД. История и развитие баз данных. Классификация и типы баз данных.
2.	Реляционные базы данных	Таблицы, строки и столбцы. Нормализация данных. Связи между таблицами (первичные ключи, внешние ключи).
3.	Основы языка SQL	SELECT-запросы и фильтрация данных. Агрегатные функции и GROUP BY. JOIN-оператора и объединение таблиц.
4.	Управление данными	INSERT, UPDATE и DELETE-запросы. Транзакции и их свойства (ACID).

		Индексация и оптимизация запросов.
5.	Хранение и индексирование данных	Типы индексов и их использование. Оптимизация производительности запросов. Хранение больших объемов данных.
6.	Проектирование и создание баз данных	Проектирование ER-диаграмм. Импорт и экспорт данных. Безопасность и резервное копирование.
Практические и/или семинарские занятия		Практическое занятие по созданию базы данных и нормализации данных. Практическое занятие по использованию JOIN-операторов. Практическое занятие по оптимизации запросов и индексации данных
Лабораторные работы		Не предусмотрены
Самостоятельная работа слушателя		Чтение рекомендованной литературы и учебников по курсу. Подготовка реферата по одной из тем дисциплины. Самостоятельное выполнение лабораторных работ и заданий по SQL-запросам. Выполнение расчетных заданий по проектированию и нормализации базы данных.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Использование мультимедийных средств и интернет-ресурсов. Обратная связь и индивидуальные консультации преподавателя. Решение кейсов и задач, близких к реальным жизненным ситуациям.

Дисциплина 11. Мультимедиатехнологии и тестирование пользовательского опыта (UX)

№ п/п	Наименование тем	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в мультимедиа-технологии	Понятие мультимедиа и их применение. История и развитие мультимедийных технологий. Классификация и виды мультимедийных приложений.
2.	Основы UX и Usability	Понятие пользовательского опыта (User Experience). Основные принципы и подходы к UX-дизайну. Критерии удобства использования (Usability).
3.	Проектирование мультимедийных интерфейсов	Прототипирование и wireframing. Визуальный дизайн и user-friendly интерфейсы. Оптимизация навигации и меню.
4.	Технологии создания мультимедийных приложений	Adobe Animate, After Effects, Premiere Pro и другие инструменты. Создание анимации и интерактивных элементов. Интеграция аудио и видео в мультимедийный контент.
5.	Тестирование пользовательского опыта (UX-тестирование)	Виды и методы UX-тестирования. User Testing и Eye Tracking. Heuristic Evaluation и Card Sorting.
6.	Трендовые направления в мультимедиа и UX	AR/VR-технологии и их использование. Голосовые помощники и чат-боты. Accessibility и инклюзивный дизайн.
Практические и/или семинарские занятия		Практическое занятие по созданию прототипа интерфейса. Практическое занятие по проектированию мультимедийного контента. Практическое занятие по проведению UX-тестирования. Практическое занятие по анализу и улучшению пользовательского опыта. Практическое занятие по интеграции мультимедийных элементов в веб-приложения.
Лабораторные занятия		Не предусмотрены
Самостоятельная работа слушателя		Чтение рекомендованной литературы и учебников по курсу. Подготовка реферата по одной из тем дисциплины. Самостоятельное создание мультимедийного проекта.

	Выполнение расчетных заданий по анализу пользовательского опыта.
Используемые образовательные технологии	Лекции и семинары с применением активных форм обучения. Использование мультимедийных средств и интернет-ресурсов. Обратная связь и индивидуальные консультации преподавателя. Решение кейсов и задач, близких к реальным жизненным ситуациям.

Дисциплина 12. Тестирование офисных приложений и автоматизация процессов

№ п/п	Наименование разделов (тем, элементов и т.д.)	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий
1.	Введение в тестирование офисных приложений.	Понятие и цели тестирования. Этапы тестирования: юнит-тестирование, интеграционное тестирование, системное тестирование. Особенности тестирования офисных приложений (MS Office, LibreOffice и т.д.).
2.	Типы и методики тестирования	White-box и black-box тестирование. Functional и non-functional тестирование. Тестирование совместимости и безопасности.
3.	Инструменты для тестирования офисных приложений	AutoIt, Sikuli, Eggplant и другие инструменты. Использование инструментов автоматизации (Selenium, AutoHotkey). Введение в скрипты и макросы для автоматизации тестирования.
4.	Тестирование и отладка макросов и скриптов	Создание и отладка макросов на VBA. Automation Anywhere и UiPath. Авто-тестирование макросов и скриптов.
5.	Автоматизация тестирования офисных приложений	Инструментальные средства для автоматизированного тестирования MS Office и LibreOffice. Подготовка тестовых данных и проведение автоматизированных тестов. Оптимизация и отладка автоматизированных тестов.
6.	Практическое внедрение и мониторинг	Внедрение автоматизированных процессов в рабочий цикл. Инструменты мониторинга производительности и надежности. Улучшение и оптимизация процессов автоматизации.
Практические и/или семинарские занятия		Практическое занятие по тестированию функционала MS Word и Excel. Практическое занятие по настройке CI/CD для офиса. Практическое занятие по автоматизации тестирования и мониторингу производительности.
Лабораторные работы		Лабораторная работа по созданию макросов и скриптов для автоматизации. Лабораторная работа по тестированию безопасности офисных приложений.
Самостоятельная работа слушателя		Чтение рекомендованной литературы и учебников по курсу. Подготовка реферата по одной из тем дисциплины. Самостоятельное выполнение лабораторных работ и заданий по автоматизации процессов. Выполнение расчетных заданий по тестированию офисных приложений.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Использование мультимедийных средств и интернет-ресурсов. Обратная связь и индивидуальные консультации преподавателя. Решение кейсов и задач, близких к реальным жизненным ситуациям.

Дисциплина 12. Современные JavaScript-фреймворки для разработки web-приложений

№ п/п	Наименование разделов (тем, элементов и т.д.)	Содержание обучения по темам, наименование и тематика лабораторных (практических и/или семинарских) занятий, самостоятельной работы слушателя и используемых образовательных технологий

1.	Введение в экосистему JavaScript-фреймворков	Назначение и роль фреймворков в современной веб-разработке. Обзор эволюции от jQuery до современных SPA-фреймворков. Сравнительный анализ ключевых игроков: React, Vue.js, Angular и Svelte. Критерии выбора фреймворка для проекта.
2.	Системы управления состоянием приложения	Причины усложнения управления состоянием в масштабируемых приложениях. Обзор архитектурных паттернов: Flux, Redux, Vuex/Pinia. Понятие централизованного хранилища, мутаций и строгой однонаправленной передачи данных для предсказуемости поведения интерфейса.
3.	Взаимодействие с сервером и управление данными	Интеграция фронтенд-фреймворков с REST API и GraphQL. Организация асинхронных запросов с помощью Fetch API и специализированных клиентов. Стратегии обновления клиентского хранилища.
4.	Современные архитектурные решения	Обзор мета-фреймворков (Next.js, Remix) и их роли в решении проблем рендеринга, роутинга и оптимизации производительности. Ключевые концепции: серверный рендеринг (SSR), статическая генерация (SSG) и инкрементальная регенерация (ISR).
5.	Тестирование и отладка приложений	Уровни тестирования фронтенда: модульное, компонентное и сквозное (E2E) тестирование. Обзор популярных инструментов (Jest, Vitest, Testing Library, Playwright) и их применение для обеспечения качества и стабильности кодовой базы.
6.	Сборка, оптимизация и развертывание	Инструменты сборки: обзор Vite, Webpack и их роли в оптимизации производственных сборок. Методы повышения производительности: разделение кода (Code Splitting) и ленивая загрузка. Процесс развёртывания приложения на современных хостинг-платформах (Vercel, Netlify)
Практические и/или семинарские занятия		Разработка простого интерактивного SPA-приложения на одном из фреймворков (React или Vue). Практикум по созданию и композиции переиспользуемых компонентов. Решение кейса по внедрению централизованного хранилища состояний в приложение. Практическое занятие по оценке и улучшению производительности приложения с помощью инструментов профилирования.
Лабораторные работы		Не предусмотрены
Самостоятельная работа слушателя		Изучение официальной документации и рекомендованных руководств по выбранному фреймворку. Самостоятельное выполнение заданий по доработке функциональности лабораторных проектов.
Используемые образовательные технологии		Лекции и семинары с применением активных форм обучения. Использование мультимедийных средств и интернет-ресурсов. Обратная связь и индивидуальные консультации преподавателя. Решение кейсов и задач, близких к реальным жизненным ситуациям.

Требования к промежуточной и итоговой аттестации

Промежуточная аттестация проводится в виде тестирований и/или решения практических задач по каждой дисциплине на тему пройденного материала. Оценка уровня освоения дисциплины осуществляется по двухбалльной системе («зачет», «незачет»).

По итогам экзамена оценивание слушателя осуществляется по пятибалльной шкале.

Итоговая аттестация производится в форме междисциплинарного экзамена.

Лицам, успешно освоившим программу профессиональной переподготовки и прошедшим итоговую аттестацию, выдается удостоверение о профессиональной переподготовке.

Организационно-педагогические условия реализации ДПП

№ п/п	Наименование дисциплин (модулей), разделов (тем, элементов и т.д.)	Фамилия, имя, отчество, год рождения	Ученая степень, ученое звание	Стаж	Основное место работы, должность	Место работы и должность по совместительству (если есть)
1.	Язык программирования Python	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
2.	Основы верстки и веб-разработки на языке JavaScript	Алексеев Владислав Алексеевич		5 лет	преподаватель	
3.	Интеграция QA-практик в разработку офисных решений	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
4.	Фреймворк Django	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
5.	Создание интерактивных web приложений на Фреймворк Django	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
6.	Алгоритмы и структуры данных	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
7.	Разработка web-приложений с использованием фреймворков языка Python (FastAPI, Flask, Django)	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
8.	Электронные презентации	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
9.	Операционные системы	Хакимова Елена Геннадьевна		18 лет	КНИТУ, ст. преп ОПП	
10.	SQL и база данных	Агафонова Дарья Олеговна		3 лет		
11.	Мультимедиа-технологии и тестирование пользовательского опыта (UX)	Агафонова Дарья Олеговна		3 лет		
12.	Тестирование офисных приложений и автоматизация процессов	Агафонова Дарья Олеговна		3 лет		
13.	Современные JavaScript-фреймворки для разработки web-приложений	Алексеев Владислав Алексеевич		5 лет	преподаватель	

Материально-техническое обеспечение

Занятия проводятся в аудиториях института дополнительного профессионального образования (ауд. №№ 218), оборудованной компьютерной и проекционной техникой.

Учебный процесс обеспечивается необходимым комплектом лицензионного программного обеспечения

Обучение проводится с использованием электронной обучающей среды Moodle.

Учебно-методическое обеспечение программы

Основная литература

1. Савин, Р. С. Python. Подробный справочник. — 4-е издание. — Москва: Вильямс, 2021. — ISBN 978-5-8459-2148-5.
2. Хабибуллин, И. Ш. PHP 8. Подробное руководство. — Москва: Эксмо, 2022. — ISBN 978-5-04-113988-8.
3. Роберт Мартин (Uncle Bob). Идеальный программист. Как стать профессионалом разработки ПО. — Москва: Питер, 2020. — ISBN 978-5-4461-1479-3.
4. Майкл Флетчер. HTML, CSS и JavaScript для начинающих. — Москва: Манн, Иванов и Фербер, 2022. — ISBN 978-5-00169-866-6.
5. Майк МакГрат. C++ для начинающих. — Москва: Эксмо, 2021. — ISBN 978-5-04-108549-2.
6. Стив Макконнелл. Совершенный код. Мастер-класс. — Москва: Питер, 2020. — ISBN 978-5-4461-1476-2.
7. Эрик Сэмюэл Рэймонд. Искусство UNIX-программирования. — Москва: Символ-Плюс, 2021. — ISBN 978-5-93286-241-3.
8. Никсон Роб. Изучаем PHP и MySQL. — Москва: DMK Press, 2022. — ISBN 978-5-97060-868-1.
9. Сергей Бобровский. Разработка программного обеспечения: теория и практика. — Москва: Издательство МГУ, 2021. — ISBN 978-5-9221-1685-6.
10. Андрей Александреску. Современный C++. — Москва: Диалектика, 2021. — ISBN 978-5-99094-838-3.

Дополнительная литература:

1. Уильям Сталлингс. Безопасность компьютерных сетей. — Москва: Вильямс, 2021. — ISBN 978-5-8459-2053-4.
2. Джефф Раскин. Интерфейс: новые направления в проектировании компьютерных систем. — Москва: Символ-Плюс, 2021. — ISBN 978-5-93286-199-9.
3. Мэтт Харрисон. Глубокое погружение в Python. — Москва: ДМК Пресс, 2022. — ISBN 978-5-97060-869-8.
4. Марк Летошник. Архитектура программного обеспечения. — Москва: Мир, 2021. — ISBN 978-5-03-012684-7.
5. Дэвид Бизли. Python Cookbook. — Москва: O'Reilly Media, 2022. — ISBN 978-5-907144-95-9.
6. Евгений Матвеев. Linux изнутри: ядро, системные вызовы, архитектура и низкоуровневые вещи. — Москва: DMC Press, 2021. — ISBN 978-5-97060-866-7.
7. Стивен Прэтт. Компьютерные сети и интернет. — Москва: Вильямс, 2021. — ISBN 978-5-8459-2052-7.
8. Павел Егоров. Контейнеризация и микросервисы: архитектура и разработка современных приложений. — Москва: Питер, 2022. — ISBN 978-5-4461-1477-9.
9. Эдвард Тейз. Регулярные выражения. — Москва: DMK Press, 2021. — ISBN 978-5-97060-867-4.
10. Дуглас Крокфорд. JavaScript: хорошие части. — Москва: Манн, Иванов и Фербер, 2021. — ISBN 978-5-00169-798-2.

Электронные и Internet-ресурсы:

1. Официальная документация Python — python.org/docs/
2. Официальная документация PHP — php.net/docs.php
3. Официальная документация JavaScript — developer.mozilla.org/en-US/docs/Web/JavaScript
4. Онлайн-курсы и уроки по Python — codecademy.com/learn/python
5. Бесплатные курсы по PHP и MySQL — sololearn.com/Learn/PHP/
6. Документация и курсы по JavaScript — freeCodeCamp.org/news/javascript-tutorial-for-beginners/

7. YouTube-каналы по программированию: CS Dojo, Programming with Mosh, Academind
8. Официальная документация C++ — isocpp.org/std/the-standard
9. Курс по Git и GitHub — github.githubtraining.com
10. Справочник по регулярным выражениям — regexone.com

Кадровое обеспечение программы

Образовательный процесс по дисциплинам (модулям) обеспечивается кадрами, имеющими базовое образование, соответствующее профилю дисциплины (модуля), и систематически занимающимися профессиональной деятельностью по профилю дисциплины.

Разработчики программы:

Крайсман Наталья Владимировна, канд. ист. наук, доцент кафедры «Иностранные языки в профессиональной коммуникации»

Разработчик программы:

Ст.преп КМИЦ НТ,
ст.преп.каф.ОПП _____ Хакимова Е.Г.

Руководитель программы:

Доцент кафедры ИЯПК _____ Крайсман Н.В.