

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Агафонов Александр Викторович

Должность: директор филиала

Дата подписания: 18.08.2026 21:01:02

Уникальный программный ключ:

2539477a8ecf706dc9cff164bc411eb6d324a86d

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ЧЕБОКСАРСКИЙ ИНСТИТУТ (ФИЛИАЛ) МОСКОВСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

Отдел дополнительного профессионального образования



УТВЕРЖДАЮ

Директор филиала

/А.В. Агафонов/

2026г.

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

«Проектирование, разработка и оптимизация веб-приложений»
(наименование программы)

Чебоксары, 2026

1. ОБЩАЯ ХАРАКТЕРИСТИКА ДОПОЛНИТЕЛЬНОЙ ПРОФЕССИОНАЛЬНОЙ ПРОГРАММЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

1.1. Нормативная база

Программа разработана в соответствии с требованиями:

- 1) Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- 2) Приказ Министерства науки и высшего образования Российской Федерации от 24 марта 2025 г. № 266 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»
- 3) Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.01 – Информатика и вычислительная техника, утверждённого приказом Минобрнауки РФ № 929 от 19 сентября 2017 г.;
- 4) Профессионального стандарта 06.001 «Программист», утверждённого приказом Минтруда РФ от 20 июля 2022 г. № 424н;
- 5) Профессионального стандарта 06.015 «Специалист по информационным системам», утверждённого приказом Минтруда РФ от 13 июля 2023 г. № 586н.

1.2. Цель дополнительной профессиональной программы повышения квалификации (далее – Программа)

Программа повышения квалификации направлена на совершенствование и (или) получение новой компетенции, необходимой для профессиональной деятельности, и (или) повышение профессионального уровня в рамках имеющейся квалификации

1.3. Требования к поступающему для обучения по программе

К освоению программы допускаются лица, имеющие среднее профессиональное и (или) высшее образование; лица, получающие среднее профессиональное и (или) высшее образование.

1.4. Объем освоения программы

Объем освоения программы – 36 академических часов.
Сроки обучения - 2 недели.

1.5. Форма обучения

Реализация программы предусмотрена в заочной форме с использованием дистанционных образовательных технологий.

1.6. Документ, выдаваемый по результатам освоения программы

По результатам освоения программы выдается удостоверение о повышении квалификации, образец которого устанавливается Чебоксарским институтом (филиалом) федерального государственного автономного образовательного учреждения высшего образования «Московский политехнический университет» (далее – филиал).

Лица, не прошедшие итоговую аттестацию или получившие на итоговой аттестации неудовлетворительные результаты, а также лица, освоившие часть программы и (или) отчисленные из Чебоксарского института (филиала) Московского политехнического университета, получают справку об обучении или о периоде обучения.

При освоении программы параллельно с получением среднего профессионального образования и (или) высшего образования удостоверение о повышении квалификации выдается одновременно с получением соответствующего документа об образовании и о квалификации.

2. ХАРАКТЕРИСТИКА НОВОЙ И (ИЛИ) СОВЕРШЕНСТВУЕМОЙ КОМПЕТЕНЦИИ, НЕОБХОДИМОЙ ДЛЯ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Наименование должности (ей) (ЕКС):

Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения:

№ п/п	Обобщенная трудовая функция (ПС) или Квалификационные характеристики (ЕКС) или Вид деятельности	Компетенции/Трудовые функции (ПС)
1	Создание (модификация) и сопровождение ИС. Проектирование ИС; разработка баз данных; применение технологий модульного тестирования; создание пользовательской документации (ПС 06.015)	ПК-1 Способен выполнять работы и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы
2	Разработка, отладка, проверка работоспособности, модификация программного обеспечения. Разработка, отладка и тестирование программного обеспечения; создание баз данных ИС (ПС 06.001, 06.015)	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение
3	Разработка, отладка, модификация и поддержка системного программного обеспечения. Применение современных технологий программирования (ПС 06.001, 06.015)	ПК-3 Способен разрабатывать компоненты системных программных продуктов

п

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

В результате освоения программы слушатель должен усовершенствовать следующие знания, умения и получить практический опыт, необходимый для профессиональной деятельности:

Код компетенции (только код)	Знания (знать)	Умения (уметь)	Практический опыт (владеть)
ПК-1	31. Принципы проектирования информационных систем и веб-приложений 32. Технологии модульного тестирования ИС 33. Стандарты документирования программного обеспечения	У1. Проектировать архитектуру веб-приложения (frontend/backend) У2. Применять технологии тестирования компонентов У3. Создавать пользовательскую документацию к веб-приложению	В1. Навыками создания и развёртывания полноценного веб-приложения В2. Навыками организации репозитория (Git) и версионирования кода. В3. Навыками создания технической документации к ИС
ПК-2	34. Основные языки программирования для backend и frontend (PHP, JavaScript, Python). 35. Структуры реляционных баз данных, SQL-запросы	У4. Составлять алгоритмы, писать и отлаживать код на JavaScript, PHP. У5. Работать с базами данных: MySQL, SQLite (создание схем, написание запросов,	В4. Навыками разработки frontend-интерфейсов (HTML, CSS, JS). В5. Навыками backend-разработки и работы с базами данных из PHP.

	(CRUD, JOIN, агрегация). 36. Принципы построения архитектурного стиля взаимодействия между программами REST API.	интеграция с PHP). У6. Разрабатывать и документировать управляемые ресурсы и идентифицировать их по идентификации URL в рамках построения архитектурного стиля взаимодействия между программами REST API.	В6. Навыками интеграции frontend и backend информационных систем через архитектурный стиль REST API.
ПК-3	37. Современные информационные технологии и программные средства при решении задач разработки веб-приложений. 38. Принципы работы веб-серверов (Apache, Nginx), протоколов HTTP/HTTPS. 39. Основы контейнеризации (Docker): образы, контейнеры, Docker Compose.	У7. Выбирать современные программные средства для разработки веб-приложений. У8. Устанавливать и настраивать веб-серверы. У9. Работать с открытыми платформами контейнеризации типа Docker: создавать образы, запускать контейнеры, настраивать сети.	В7. Навыками системной установки и конфигурирования веб-серверов и Docker-контейнеров в среде ОС Linux/Windows. В8. Настройкой системного взаимодействия программных интерфейсов, выполнения системных команд устройств и десктопов, технологиями ADB, GRPC, компиляцией AOT Native. В9. Навыками развёртывания веб-приложений с использованием Docker Compose.

4. СОДЕРЖАНИЕ ПРОГРАММЫ

4.1. Учебный план

№ п/п	Наименование дисциплин / разделов / модулей/тем	Всего, час.	Аудиторные занятия, час			Учебные занятия с использованием ДОТ, час			Самостоятельная работа, час.	Формы аттестации	
			лекции	лабораторные работы	практические занятия	лекции	лабораторные работы	практические занятия		Зачет	Экзамен
1	2	3	4	5	6	7	8	9	10	11	12
Модуль 1. Архитектура и инфраструктура веб-приложений											
1.1	Установка и настройка Web-сервера.	6				2		2	2		
1.2	Контейнеризация приложений.	6				2		2	2		
Модуль 2. Базы данных веб-приложений											
2.1	Основы SQL и реляционных баз данных. Подключение MySQL, SQLite и PHPMyAdmin.	6				2		2	2		

Модуль 3. Разработка веб-приложений										
3.1	Frontend-разработка (HTML, CSS, JavaScript).	8				2		2	4	
3.2	Backend-разработка и REST API	8				2		2	4	
4.	Итоговая аттестация	2								2
Итого		36				10		10	14	2

4.2. Календарный учебный график

Период обучения (недели, месяцы) ¹⁾	Наименование дисциплины/модуля
1-2 день	Архитектура и инфраструктура веб-приложений
3 день	Базы данных веб-приложений
4-5 день	Разработка веб-приложений
6 день	Итоговая аттестация

¹⁾ Точный порядок реализации модулей/разделов (дисциплин) обучения определяется в расписании занятий при наборе группы

4.3. Содержание программы

Цель: подготовка квалифицированных кадров, обладающих современными знаниями и умениями в проектировании, разработке и оптимизации веб-приложений.

Программа предназначена для студентов, инженеров, технических специалистов, руководителей предприятий, сотрудников отделов автоматизации и ИТ, желающих повысить свою квалификацию в сфере проектирования, разработки и оптимизации веб-приложений.

Преимущества прохождения программы:

- Получение актуальных знаний и навыков в области современных информационных технологий.
- Возможность карьерного роста и повышение конкурентоспособности на рынке труда.
- Доступ к современным технологиям программирования и инструментальным средствам разработки, проектирования и оптимизации веб-приложений.

Таким образом, программа «Проектирование, разработка и оптимизация веб-приложений» является важным инструментом для профессионального развития специалистов, работающих в сфере информационных технологий.

Планируемые результаты обучения:

- ПК-1. Способен выполнять работы и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы:

Знания:

1. Знание современных технологий проектирования веб-приложений: понимание принципов архитектуры клиент-сервер, моделей взаимодействия приложений, методов анализа требований и проектирования интерфейсов.
2. Понимание процессов разработки веб-приложений: подходов непрерывной их интеграции, тестирования, доставки и развертывания.

3. Знание HTML/CSS, JavaScript, PHP/Laravel, Node.js и других инструментов, применяемых в разработке веб-приложений.

4. Основы информационной безопасности веб-приложений: понимание угроз кибератак, уязвимости.

5. Принципы кроссбраузерной совместимости: знание особенностей отображения интерфейсов в разных браузерах и операционных системах, техники адаптации под мобильные устройства.

Умения:

1. Анализировать существующие процессы управления в организациях разных типов и масштабов.

2. Анализ и проектирование информационной системы: способность анализировать требования пользователей, проектировать базы данных, разрабатывать техническую документацию и спецификации.

3. Оптимизация производительности веб-приложений: умение выбирать эффективные инструменты и методы оптимизации скорости загрузки страниц, улучшения масштабируемости и надежности веб-сервисов.

4. Интеграция сервисов API и сторонних решений: интеграция REST API, аутентификация, работа с сервисами Google Maps, Yandex.Cloud.

5. Мониторинг работоспособности и отказоустойчивость: внедрение мониторинга состояния сервера и приложения, применение кластеризации и балансировки нагрузки для предотвращения сбоев.

Навыки:

1. Владения современными языками программирования и фреймворками.

2. Методы тестирования и оценки качества веб-приложений: знакомство с инструментами автоматизированного тестирования.

3. Администрирование баз данных: установка, настройка и обслуживание СУБД MySQL, PostgreSQL.

4. Методики повышения доступности веб-интерфейсов.

5. Модели развертывания и эксплуатации веб-приложений: облачные технологии, контейнеризация (Docker), виртуализация серверов, командные интерфейсы.

Эти знания, умения и навыки позволят специалисту успешно реализовывать проекты по автоматизации бизнес-процессов и управлению информационными системами организаций различного масштаба.

- ПК-2. Способен разрабатывать требования и проектировать программное обеспечение:

Знания:

1. Методы моделирования предметной области.

2. Архитектурные паттерны и шаблоны проектирования.

3. Основные принципы юзабилити-дизайна: Понимание эргономичных подходов к проектированию интерфейсов, стандарты и рекомендации по улучшению удобства пользования.

4. Основы безопасности веб-разработки.

5. Современные подходы к разработке требований, формализации требований, составления технического задания и функциональной спецификации.

Умения:

1. Сбор и документирование требований: Навык интервьюирования заказчиков, анкетирования конечных пользователей, составление подробных функциональных спецификаций.

2. Создание прототипов интерфейсов.

3. Проектирование архитектуры веб-приложений: Способность выбора оптимальной структуры проекта, распределение компонентов по слоям, выбор подходящих паттернов проектирования.

4. Формулирование и оценка технических рисков: Оценка возможных трудностей реализации функционала, определение критически важных аспектов и принятие мер минимизации риска.

5. Умения выделять ядро продуктовых функций, формулировать минимальные жизнеспособные продукты (MVP), строить приоритеты развития продуктов.

Навыки:

1. Работа с командами разработчиков: Эффективное взаимодействие с разработчиками, тестировщиками, дизайнерами и заказчиками, ведение совместной проектной документации.

2. Постановка целей и контроль исполнения: Использование инструментов управления проектами, организация встреч по статусу работ и промежуточному контролю этапов проекта.

3. Использование специализированных инструментов: Владение средствами визуального моделирования.

4. Составление наглядных докладов, проведение демонстраций проектов перед руководством и клиентами, подготовка технической документации.

5. Оценка готовности и соответствия стандартам: Проведение аудитов соответствия корпоративным и отраслевым стандартам качества и требованиям безопасности.

Эти знания, умения и навыки обеспечивают готовность выпускников грамотно определять цели, задачи и структуру будущего программного продукта, обеспечивать качество процесса разработки и добиваться успешного завершения проектов по созданию веб-приложений.

- **ПК-3. Способен разрабатывать компоненты системных программных продуктов:**

Знания:

1. Структура и архитектура веб-приложений: Понимание модели клиент-сервер, трехуровневой архитектуры, REST API и основных архитектурных стилей.

2. Программирование на стороне клиента и сервера: Глубокое знание основ фронтенд-технологий (HTML, CSS, JavaScript) и бэкенд-языков (PHP, Node.js).

3. Фреймворки и библиотеки для веб-разработки: Освоение популярных инструментов и платформ.

4. Безопасность веб-приложений: Общие угрозы и меры противодействия (защита от инъекций, межсайтового скриптинга, подделки запросов).

5. Принципы модульного проектирования и повторного использования кода: Применение SOLID-принципов, паттернов проектирования и методов организации модульной структуры.

Умения:

Разработка интерфейсов веб-приложений: Создание интерактивных элементов, анимаций, адаптивных макетов, доступность для мобильных устройств и слабовидящих пользователей.

Реализация серверной части веб-приложений: Работа с базой данных, реализация REST API, логика обработки HTTP-запросов, кэширование данных.

Автоматизированное тестирование и отладка: Настройка тестов (E2E, интеграционные тесты), фиксация багов и выполнение рефакторинга кода.

Оптимизация производительности веб-приложений: Профилирование и ускорение загрузки страницы, уменьшение объема передаваемых данных, оптимизация базы данных.

Поддержка и сопровождение веб-приложений: Организация регулярного обновления компонентов, мониторинг ошибок и исправления дефектов.

Навыки:

1. Командная работа над проектом: Совместная разработка с распределением ролей, постановка задач и контроль сроков выполнения.

2. Настройка инфраструктуры разработки: Установка окружения (Git, Docker), запуск локальных серверов и баз данных.
3. Планирование релизов и развертывание: Автоматическое развёртывание изменений, организация релиза новых версий и обновление существующих приложений.
4. Работа с документацией: Поддержание актуальной технической документации, написание инструкций по установке и настройке веб-приложений.
5. Непрерывное саморазвитие: Регулярное повышение квалификации, освоение новых технологий и участие в сообществе разработчиков.

Таким образом, знания, умения и навыки позволяют выпускникам уверенно заниматься разработкой качественных и эффективных компонентов системных программных продуктов в сфере веб-технологий.

Содержание дисциплины

Модуль 1. Архитектура и инфраструктура веб-приложений.

Тема 1.1. Установка и настройка Web-сервера.

Выбор и установка веб-сервера (Apache, Nginx). Выбор дистрибутива и программного обеспечения. Анализ требований к серверу Сравнение Apache, Nginx и альтернатив. Определение совместимости с ОС. Подготовка операционной системы. Настройка системы (обновления, безопасность). Установка необходимых зависимостей и библиотек. Конфигурация сети для сервера. Основные принципы работы: HTTP/HTTPS, конфигурационные файлы, виртуальные хосты. Конфигурация сервера. Настройка виртуальных хостов. Управление модулями и расширениями. Работа с конфигурационными файлами (перенаправления, Rewrite правила). Оптимизация производительности. Кэширование статического и динамического контента. Балансировка нагрузки. Настройка параметров соединения. Мониторинг и диагностика. Использование файла .htaccess. Настройка SSL/TLS-сертификатов. Генерация и установка сертификатов. Настройка шифров и протоколов безопасности. Автоматизация обновления сертификатов. Управление доступом и аутентификация. Ограничение доступа по IP. Реализация базовой и продвинутой аутентификации. Использование firewall и других средств контроля доступа.

Тема 1.2. Контейнеризация приложений (Docker).

Что такое контейнеризация? Какие задачи она решает? Контейнеризация против виртуализации (VM): сравнение подходов, плюсы и минусы. История Docker. Архитектура Docker: компоненты и их назначение (Движок, Демон, Клиент, Репозиторий). Установка Docker Desktop (Windows/Mac) / Docker Engine (Linux). Пошаговая инструкция.

Основные команды командной строки Docker: запуск, просмотр, остановка, удаление контейнеров, просмотр образов. Работа с Docker Hub: поиск и загрузка готовых образов из репозитория Docker. Запуск простого контейнера из готового. Простейший пример.

Основы контейнеризации: отличие от виртуализации. Архитектура Docker (Daemon, Client, Registry). Основные команды: run, ps, stop, rm. Создание Dockerfile. Сборка и публикация образов (Docker Hub). Тома Docker: обзор типов томов (именованные тома, привязки). Разница и применение. Преимущества использования для управления несколькими контейнерами. Основные команды. Введение в оркестрацию контейнеров: Kubernetes, Docker Swarm. Обзор основных понятий и возможностей (по выбору). Сети Docker: обзор типов сетей (мост, хост, нет). Выбор подходящего типа для разных задач. Docker Compose для многоконтейнерных приложений. Сетевые настройки и тома (volumes).

Модуль 2. Базы данных веб-приложений.

Тема 2.1. Основы SQL и реляционных баз данных. Подключение MySQL, SQLite и PHPMyAdmin.

Реляционная модель данных. Типы данных. Введение в базы данных. Что такое база данных? Типы баз данных. Реляционные базы данных: основные понятия (таблица, поле, запись, ключ). Проектирование реляционных баз данных. Сущность-связь (ER) модели. Нормализация баз данных. Зачем нужна нормализация? Язык SQL: Обзор. Основные операторы SQL. SQL-операторы: SELECT, INSERT, UPDATE, DELETE. Фильтрация (WHERE), сортировка (ORDER BY), группировка (GROUP BY). JOIN: INNER, LEFT, RIGHT, CROSS. Подзапросы. Нормализация БД. Индексы и транзакции. Защита от SQL-инъекций (prepared statements).

Архитектура MySQL-сервера. Преимущества и недостатки MySQL. Установка и настройка MySQL. Установка MySQL-сервера. Настройка параметров подключения. Создание учетных записей пользователей. Работа с MySQL из командной строки. Подключение к MySQL-серверу. Введение в PHPMyAdmin. Что такое PHPMyAdmin? Установка и настройка PHPMyAdmin. Использование PHPMyAdmin для управления базами данных. Создание и удаление баз данных. Использование PHPMyAdmin для управления таблицами. Создание и удаление таблиц. Изменение структуры таблиц. Использование PHPMyAdmin для управления данными. Просмотр данных. Добавление, изменение и удаление данных. Выполнение SQL-запросов. Работа с PHPMyAdmin: управление БД, таблицами, данными. Особенности SQLite: однофайловая СУБД, применение в небольших проектах. Подключение к MySQL из PHP (mysqli, PDO). Подключение к SQLite из PHP (PDO, SQLite3). Выполнение запросов, обработка результатов, обработка ошибок. Выполнение SQL-запросов из PHP. Подготовка SQL-запроса. Выполнение SQL-запроса. Обработка результатов запроса. Предотвращение SQL-инъекций. Что такое SQL-инъекции? Использование подготовленных выражений (prepared statements). Экранирование данных. Обработка ошибок. Обработка ошибок при подключении к базе данных. Обработка ошибок при выполнении SQL-запросов.

Модуль 3. Разработка веб-приложений.

Тема 3.1. Frontend-разработка (HTML, CSS, JavaScript).

Семантический HTML: структура документа, основные теги. CSS: селекторы, каскадность, блочная модель, Flexbox, Grid, медиазапросы, адаптивная верстка. JavaScript: переменные, типы данных, функции, массивы, объекты, работа с DOM. События и обработчики. Асинхронность: Promise, async/await. Fetch API, работа с JSON. Основы валидации форм на клиентской стороне. Инструменты фронтенд-разработки: npm/yarn, Webpack/Parcel, ESLint, Prettier.

Тема 3.2. Backend-разработка и REST API.

Основы PHP: синтаксис, функции, ООП. Обработка HTTP-запросов (GET/POST). Работа с формами и файлами. Синтаксис языка, структура фреймворка, маршрутизация, шаблонизаторы. Базы данных: Реляционные базы данных (MySQL, PostgreSQL) или NoSQL. Принципы REST API: HTTP-методы, статус-коды, форматы данных (JSON). Проектирование API-маршрутов. моделирование данных, ORM (Object-Relational Mapping). API (Application Programming Interface): Принципы RESTful API, HTTP-методы (GET, POST, PUT, DELETE), форматы данных (JSON, XML), сериализация/десериализация. Аутентификация и авторизация: Сессии, токены (JWT), OAuth 2.0. Безопасность бэкенда: Валидация данных, защита от SQL-инъекций, XSS, CSRF, CORS. Интеграция фронтенда и бэкенда через fetch/AJAX. Аутентификация и сессии. Основы безопасности: XSS, CSRF, CORS. Деплой приложения (Docker + веб-сервер). Основы Git: коммиты, ветки, удалённый репозиторий. Архитектура веб-приложения: Одностраничные приложения (SPA), многостраничные приложения (MPA). Обмен данными: Fetch API, AJAX, CORS. Управление состоянием: Redux, Vuex, Context API (для фронтенда). Развертывание приложения: Подготовка к развертыванию, хостинг, Docker. Тестирование: Модульное тестирование, интеграционное тестирование, end-to-end тестирование.

5. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

5.1. Кадровое обеспечение образовательного процесса

Отличительными особенностями дополнительной профессиональной программой является ориентация на компетентностный подход, позволяющий развивать и наращивать необходимые компетенции для решения профессиональных задач практической деятельности слушателей.

Реализация программы обеспечивается научно-педагогическими кадрами, имеющими высшее профессиональное образование или среднее профессиональное образование в области, соответствующей преподаваемому предмету, без предъявления требований к стажу работы, либо высшее профессиональное образование или среднее профессиональное образование и дополнительное профессиональное образование по направлению деятельности в образовательном учреждении без предъявления требований к стажу работы.

5.2. Материально-техническое обеспечение

Указывается перечень учебных аудиторий для проведения лекций, семинарских/практических и лабораторных занятий, самостоятельной работы обучающихся с указанием лабораторного оборудования, демонстрационных приборов, оборудования для практических занятий, мультимедийных средств, тренажеров, симуляторов, наглядных пособий, используемых для освоения дисциплины.

Учебный процесс по программе проводится в лекционных аудиториях, оборудованных видеопроекторным оборудованием для презентаций, средствами звуковоспроизведения, экраном и имеющих выход в сеть Интернет.

Практические занятия проводятся в помещениях, оборудованных учебной мебелью, оснащенных мультимедийными средствами обучения и компьютерных классах с выходом в Интернет.

Для организации самостоятельной работы слушателей предусмотрены компьютерные классы и библиотека, имеющая рабочие места для слушателей, оснащенные компьютерами с доступом к базам данных и сети Интернет.

5.3. Учебно-методическое и информационное обеспечение

5.3.1. Нормативные документы и ГОСТы

1. ГОСТ Р 21.1101–2013 «Единая система конструкторской документации (ЕСКД). Правила выполнения проектной и рабочей документации».
2. ГОСТ 34.601-90 «Автоматизированные системы. Стадии создания».
3. ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»
4. ГОСТ Р МЭК 61131-3-2016 «Программируемые контроллеры. Часть 3. Языки программирования.
5. ГОСТ 34.603-92 «Виды испытаний автоматизированных систем»

5.3.2. Основная литература

1. Гниденко И. Г. Технологии и методы программирования: учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2026. — 241 с. — (Высшее образование). — ISBN 978-5-534-18130-2.

— Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/583784> (дата обращения: 09.03.2026).

2. Веб-разработка : учебник для вузов / под общей редакцией О. В. Ратановой, Н. А. Ребус, А. Ю. Анисимова. — Москва : Издательство Юрайт, 2026. — 211 с. — (Высшее образование). — ISBN 978-5-534-21194-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590626> (дата обращения: 09.03.2026).

3. Сысолетин, Е. Г. Разработка интернет-приложений : учебник для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под научной редакцией Л. Г. Доросинского. — Москва : Издательство Юрайт, 2025. — 80 с. — (Высшее образование). — ISBN 978-5-534-17124-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562916> (дата обращения: 09.03.2026).

4. Никсон Р. Создаём динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. — М., 2025.

5. Дакетт Д. PHP и MySQL. Серверная веб-разработка. — М., 2023.

5.3.3. Дополнительная литература

1. Полуэктова Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 204 с. — (Высшее образование). — ISBN 978-5-534-18645-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588458> (дата обращения: 09.03.2026).

2. Тузовский А. Ф. Проектирование и разработка web-приложений : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Высшее образование). — ISBN 978-5-534-16300-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561176> (дата обращения: 09.03.2026).

3. Web-разработки в asp. Net web forms : учебник для вузов / С. Т. Гуляева, В. В. Миронов, Н. О. Котелина, И. И. Лавреш. — Москва : Издательство Юрайт, 2025. — 134 с. — (Высшее образование). — ISBN 978-5-534-19885-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/569218> (дата обращения: 09.03.2026).

4. Маркин А. В. Программирование на SQL : учебник и практикум для вузов / А. В. Маркин. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 805 с. — (Высшее образование). — ISBN 978-5-534-18371-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589589> (дата обращения: 09.03.2026).

5.3.4. Электронные образовательные ресурсы

1. Российский портал научно-технических публикаций eLibrary.ru
2. ЭБС «ЮРАЙТ» (<https://urait.ru/>);
3. ЭБС «IPR BOOKS» (<http://www.iprbookshop.ru/>);
4. Документация MDN Web Docs (<https://developer.mozilla.org/>);
5. Официальная документация Docker (<https://docs.docker.com/>).

5.3.5. Лицензионное и свободно распространяемое программное обеспечение

1. Windows 7 OLPNLAcDmc Windows Server 2012
2. Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License
3. Microsoft Office Standard 2007 (Microsoft DreamSpark Premium Electronic Software Delivery Academic (Microsoft Open License)
4. Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License

5. Microsoft Office Standard 2019(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License
6. Eclipse IDE for Java EE Developers, NET Framework, JDK 8, Microsoft SQL Server Express Edition,
7. Microsoft Visual Studio 2019
8. Гарант- справочно-правовая система

Учебный процесс проводится в компьютерных классах, оснащённых рабочими местами с подключением к интернету. На каждом ПК установлено необходимое программное обеспечение:

1. Веб-сервер Apache / Nginx;
2. Docker Desktop (Windows / macOS) или Docker Engine (Linux);
3. СУБД MySQL, SQLite; PHPMyAdmin;
4. Браузер (Chrome / Firefox) с инструментами разработчика;
5. Visual Studio Code с расширениями (PHP, JavaScript, HTML/CSS);
6. Git, Node.js, npm;
7. Kaspersky Endpoint Security, Windows 10/11 или Ubuntu 22.04.

5.3.6. Современные профессиональные базы данных и информационные справочные системы

Профессиональная база данных и информационно-справочные системы	Информация о праве собственности (реквизиты договора)
Университетская информационная система РОССИЯ https://uisrussia.msu.ru/	Тематическая электронная библиотека и база для прикладных исследований в области экономики, управления, социологии, лингвистики, философии, филологии, международных отношений, права. свободный доступ
научная электронная библиотека Elibrary http://elibrary.ru/	Научная электронная библиотека eLIBRARY.RU - это крупнейший российский информационно-аналитический портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты более 26 млн научных статей и публикаций, в том числе электронные версии более 5600 российских научно-технических журналов, из которых более 4800 журналов в открытом доступе свободный доступ
сайт Института научной информации по общественным наукам РАН. http://www.inion.ru	Библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам ведутся с начала 1980-х годов. Общий объём массивов составляет более 3 млн. 500 тыс. записей (данные на 1 января 2012 г.). Ежегодный прирост — около 100 тыс. записей. В базы данных включаются аннотированные описания книг и статей из журналов и сборников на 140 языках, поступивших в Фундаментальную библиотеку ИНИОН РАН. Описания статей и книг в базах данных снабжены шифром хранения и ссылками на полные тексты источников из Научной электронной библиотеки.
Федеральный портал «Российское образование» [Электронный ресурс] – http://www.edu.ru	Федеральный портал «Российское образование» – уникальный интернет-ресурс в сфере образования и науки. Ежедневно публикует самые актуальные новости, анонсы событий, информационные материалы для широкого круга читателей. Еженедельно на портале размещаются эксклюзивные материалы, интервью с ведущими специалистами – педагогами, психологами, учеными, репортажи и аналитические статьи.

	Читатели получают доступ к нормативно-правовой базе сферы образования, они могут пользоваться самыми различными полезными сервисами – такими, как онлайн-тестирование, опросы по актуальным темам и т.д.
Информационные технологии – периодическое научно-техническое издание в области информационных технологий, автоматизированных систем и использования информатики в различных приложениях novtex.ru	Издательство выпускает теоретические и прикладные научно-технические журналы, обеспечивающие научной, производственной, обзорно-аналитической и образовательной информацией руководящих работников и специалистов промышленных предприятий, научных академических и отраслевых организаций, а также учебных заведений в области приоритетных направлений развития науки и технологий.
Ассоциация инженерного образования России http://www.ac-raee.ru/	Совершенствование образования и инженерной деятельности во всех их проявлениях, относящихся к учебному, научному и технологическому направлениям, включая процессы преподавания, консультирования, исследования, разработки инженерных решений, оказания широкого спектра образовательных услуг, обеспечения связей с общественностью, производством, наукой и интеграции в международное научно-образовательное пространство. свободный доступ

5.4. Образовательные технологии, используемые в процессе реализации программы

В процессе реализации программы используются образовательные технологии *проблемно-ориентированного обучения, проектного обучения, развития критического мышления, ситуационных задач, дискуссий и т.д.*

Программа может реализовываться с использованием дистанционных образовательных технологий.

6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценка качества освоения программы осуществляется в форме текущего контроля успеваемости и/или промежуточной аттестации по дисциплине/модулю/разделу и итоговой аттестации слушателей.

6.1. Текущий контроль и промежуточная аттестация

Для проведения промежуточной аттестации знаний слушателей предусмотрен вид контроля: *тестирование*.

Задания для проведения промежуточной аттестации слушателей представлены в рабочей программе по каждому модулю.

Критерии оценивания сформированности компетенций слушателей в ходе текущего контроля и промежуточной аттестации:

№ п/п	Наименование модулей, разделов, тем	Предмет оценивания (проф. компетенции)	Основные показатели оценки результатов	Форма и вид аттестации	Критерии оценивания
1.	Модуль 1. Архитектура и инфраструктура веб-приложений.	ПК 1	31-39, У1-У9, В1-В9	Промежуточная аттестация. Зачёт (тестирование)	Тест выполнен: более 70% правильных ответов - «зачтено», 69 % и менее – «не зачтено»
2.	Модуль 2. Базы данных веб-приложений	ПК 2	31-39, У1-У9, В1-В9	Промежуточная аттестация. Зачёт (тестирование)	Тест выполнен: более 70% правильных ответов - «зачтено», 69 % и менее – «не зачтено»
3.	Модуль 3. Разработка веб-приложений	ПК 3	31-39, У1-У9,	Промежуточная аттестация.	Тест выполнен: более 70% правильных

			B1-B9	Зачёт (тестирование)	ответов - «зачтено», 69 % и менее – «не зачтено»
--	--	--	-------	-------------------------	--

Модуль 1. Архитектура и инфраструктура веб-приложений.

Вопрос 1

Какой из следующих веб-серверов является наиболее популярным?

- A) Apache
- B) Paint
- C) Notepad
- D) Filezilla

Вопрос 2

Какой протокол используется для передачи данных в интернете?

- A) FTP
- B) HTTP
- C) SMTP
- D) DNS

Вопрос 3

Какой файл конфигурации используется для настройки веб-сервера Apache?

- A) httpd.conf
- B) config.xml
- C) server.cfg
- D) apache.config

Вопрос 4

Какой командой запускается веб-сервер Apache на Linux?

- A) apache start
- B) systemctl start apache2
- C) run apache
- D) start httpd

Вопрос 5

Какой порт, по умолчанию используется для HTTP?

- A) 80
- B) 443
- C) 21
- D) 25

Вопрос 6

Какой формат файла используется для установки пакета на Ubuntu?

- A) .exe
- B) .deb
- C) .rpm
- D) .zip

Вопрос 7

Что означает SSL в контексте веб-серверов?

- A) Secure Socket Layer
- B) Simple Socket Layer
- C) Secure Server Login

D) Simple Secure Link

Вопрос 8

Что позволяет файл .htaccess?

- A) Писать PHP код
- B) Переопределять настройки сервера в каталоге
- C) Изменять системные файлы
- D) Управлять базой данных

Вопрос 9

Что такое кэширование на веб-сервере?

- A) Хранение данных для ускорения доступа
- B) Уменьшение размера файлов
- C) Удаление ненужных файлов
- D) Защита данных

Вопрос 10

Что такое Docker?

- A) Инструмент для управления виртуальными машинами
- B) Система для развертывания приложений в контейнерах
- C) Язык программирования
- D) Операционная система

Вопрос 11

Какой файл используется для определения конфигурации контейнера в Docker?

- A) Dockerfile
- B) docker-compose.yml
- C) config.json
- D) containerfile

Вопрос 12

Как просмотреть все запущенные контейнеры Docker?

- A) `docker ps -a`
- B) `docker containers`
- C) `docker ps`
- D) `docker show active`

Вопрос 13

Как остановить работающий контейнер Docker?

- A) `docker stop <id>`
- B) `docker kill <id>`
- C) `docker halt <id>`
- D) `docker end <id>`

Вопрос 14

Что делает команда `docker-compose up`?

- A) Останавливает контейнеры
- B) Запускает и создаёт контейнеры
- C) Удаляет контейнеры
- D) Загружает образы

Вопрос 15

Что такое Docker Hub?

- A) Локальное хранилище образов
- B) Служба хранения и обмена образами
- C) Утилита настройки Docker
- D) Инструмент мониторинга

Вопрос 16

Как проверить версию Docker?

- A) `docker --version`
- B) `docker version --show`
- C) `docker info version`
- D) `docker check version`

Вопрос 17

Какой командой посмотреть логи контейнера?

- A) `docker logs <id>`
- B) `docker view <id>`
- C) `docker output <id>`
- D) `docker show logs <id>`

Вопрос 18

Что такое виртуальный хост в Apache?

- A) Физический сервер
- B) Конфигурация для нескольких сайтов на одном сервере
- C) Прокси-сервер
- D) База данных

Вопрос 19

Какой параметр в Nginx задаёт корневую директорию сайта?

- A) `root`
- B) `home`
- C) `path`
- D) `dir`

Вопрос 20

Как называется файл для Docker Compose?

- A) `compose.json`
- B) `docker-compose.yml`
- C) `containers.yml`
- D) `services.yml`

Вопрос 21

Что обозначает HTTPS?

- A) Hypertext Transfer Protocol Secure
- B) High Transfer Protocol Service
- C) Hyper Transfer Proxy Secure
- D) Host Transfer Protocol Secure

Вопрос 22

Что такое балансировка нагрузки на веб-сервере?

- A) Распределение входящих запросов между несколькими серверами
- B) Сжатие данных

- C) Шифрование данных
- D) Кэширование статики

Вопрос 23

Какой инструмент используется для тестирования доступности веб-сервера?

- A) ping
- B) curl
- C) netstat
- D) ifconfig

Вопрос 24

Что такое Docker-образ?

- A) Запущенный контейнер
- B) Шаблон для создания контейнеров
- C) Лог-файл
- D) Конфигурационный файл

Вопрос 25

Какой тип сети Docker используется по умолчанию?

- A) host
- B) none
- C) bridge
- D) overlay

Вопрос 26

Что означает SSL-сертификат?

- A) Документ, удостоверяющий подлинность сайта
- B) Лицензия программного обеспечения
- C) Ключ доступа к серверу
- D) Файл конфигурации

Вопрос 27

Какой модуль Apache используется для перенаправлений URL?

- A) mod_rewrite
- B) mod_redirect
- C) mod_url
- D) mod_proxy

Вопрос 28

Что такое Docker Volume?

- A) Внешнее хранилище данных для контейнеров
- B) Тип сети
- C) Образ контейнера
- D) Параметр CPU

Вопрос 29

В каком формате указываются зависимости в docker-compose.yml?

- A) JSON
- B) YAML
- C) XML
- D) CSV

Вопрос 30

Что такое reverse проху (обратный прокси)?

- A) Сервер, передающий запросы клиента к бэкенд-серверам
- B) Клиент, скрывающий свой IP
- C) Тип базы данных
- D) Протокол шифрования

Модуль 2. Базы данных веб-приложений.

Вопрос 1:

Что такое реляционная база данных?

- A) БД, хранящая данные в таблицах со связями
- B) БД без структуры
- C) Файловое хранилище
- D) Кэш-сервер

Вопрос 2:

Что такое PHPMyAdmin?

- A) Графический интерфейс для работы с MySQL
- B) Сервер баз данных
- C) Язык программирования
- D) Фреймворк для PHP

Вопрос 3:

Что такое SQLite?

- A) Полнофункциональная СУБД с сервером
- B) Встраиваемая СУБД без сервера
- C) Дистрибуция MySQL
- D) Язык запросов

Вопрос 4:

Какой запрос создает новую базу данных в MySQL?

- A) `INSERT DATABASE db;`
- B) `CREATE DATABASE db;`
- C) `ADD DATABASE db;`
- D) `NEW DATABASE db;`

Вопрос 5:

Как подключиться к SQLite в PHP?

- A) `new SQLiteDatabase('file.db');`
- B) `new SQLite('file.db');`
- C) `new PDO('sqlite:file.db');`
- D) `new MySQL('file.db');`

Вопрос 6:

Какой SQL-оператор выбирает данные?

- A) `GET FROM table;`
- B) `SELECT * FROM table;`
- C) `EXTRACT FROM table;`
- D) `FIND IN table;`

Вопрос 7:

Какой оператор добавляет записи в таблицу?

- A) ADD
- B) INSERT
- C) UPDATE
- D) CREATE

Вопрос 8:

Что означает JOIN в SQL?

- A) Удаление записей
- B) Объединение данных из нескольких таблиц
- C) Создание индекса
- D) Обновление данных

Вопрос 9:

Что такое первичный ключ (PRIMARY KEY)?

- A) Уникальный идентификатор записи
- B) Внешний ключ
- C) Индекс таблицы
- D) Тип данных

Вопрос 10:

Что такое SQL-инъекция?

- A) Атака через вредоносный SQL-код в запросе
- B) Добавление данных в БД
- C) Ошибка подключения
- D) Резервное копирование

Вопрос 11:

Как защититься от SQL-инъекций?

- A) Использовать prepared statements
- B) Хранить пароли в открытом виде
- C) Отключить логирование
- D) Увеличить размер таблиц

Вопрос 12:

Что делает оператор WHERE?

- A) Сортирует данные
- B) Фильтрует строки по условию
- C) Группирует данные
- D) Объединяет таблицы

Вопрос 13:

Что такое нормализация БД?

- A) Процесс устранения избыточности данных
- B) Шифрование данных
- C) Создание резервных копий
- D) Индексирование

Вопрос 14:

Какой тип данных хранит целые числа в MySQL?

- A) TEXT
- B) INT

- C) VARCHAR
- D) FLOAT

Вопрос 15:

Какой разряд VARCHAR в MySQL?

- A) 255 байт
- B) 65535 символов
- C) 4096 байт
- D) Неограниченный

Вопрос 16:

Что делает оператор ORDER BY?

- A) Фильтрует данные
- B) Сортирует результат запроса
- C) Удаляет записи
- D) Создает индекс

Вопрос 17:

Что такое внешний ключ (FOREIGN KEY)?

- A) Ключ для шифрования
- B) Поле, ссылающееся на первичный ключ другой таблицы
- C) Уникальный индекс
- D) Автоинкрементное поле

Вопрос 18:

Что делает команда DROP TABLE?

- A) Очищает таблицу от данных
- B) Удаляет таблицу из базы данных
- C) Создает таблицу
- D) Переименовывает таблицу

Вопрос 19:

Что такое транзакция в базе данных?

- A) Набор операций, выполняемых как единое целое
- B) Запрос на выборку
- C) Резервная копия
- D) Индекс

Вопрос 20:

Для чего служит директива mysqli_connect в PHP?

- A) Для остановки соединения
- B) Для создания соединения с MySQL
- C) Для выполнения запросов
- D) Для вывода данных

Вопрос 21:

Что такое агрегатная функция SQL?

- A) Функция, выполняющая вычисления над набором строк (SUM, AVG, COUNT)
- B) Функция создания таблиц
- C) Функция шифрования
- D) Функция форматирования строк

Вопрос 22:

Что делает оператор GROUP BY?

- A) Сортирует данные
- B) Группирует строки с одинаковыми значениями
- C) Фильтрует строки
- D) Обновляет данные

Вопрос 23:

Как называется процесс объединения данных из двух таблиц без условия?

- A) CROSS JOIN
- B) INNER JOIN
- C) LEFT JOIN
- D) FULL JOIN

Вопрос 24:

Что такое индекс в базе данных?

- A) Структура для ускорения поиска данных
- B) Уникальный ключ
- C) Тип запроса
- D) Резервная копия таблицы

Вопрос 25:

Какой оператор изменяет существующую запись?

- A) CHANGE
- B) UPDATE
- C) MODIFY
- D) ALTER

Вопрос 26:

Что такое ORM?

- A) Object-Relational Mapping — технология связи объектов и таблиц БД
- B) Тип сервера
- C) Протокол доступа
- D) Язык разметки

Вопрос 27:

Что хранит тип данных TEXT в MySQL?

- A) Большие текстовые данные
- B) Числа с плавающей точкой
- C) Дату и время
- D) Бинарные данные

Вопрос 28:

Что такое подзапрос (subquery)?

- A) Запрос внутри другого SQL-запроса
- B) Резервный запрос
- C) Запрос к другому серверу
- D) Шаблонный запрос

Вопрос 29:

Чем SQLite отличается от MySQL?

- A) SQLite не требует отдельного сервера, хранит данные в файле

- B) MySQL работает медленнее
- C) SQLite поддерживает больше пользователей
- D) MySQL не поддерживает транзакции

Вопрос 30:

Какая команда PHP получает данные из массива результатов MySQL?

- A) `mysqli_fetch_assoc()`
- B) `mysql_get_row()`
- C) `fetch_row()`
- D) `db_fetch()`

Модуль 3. Разработка веб-приложений.

Вопрос 1:

Что такое HTML?

- A) Язык разметки веб-страниц
- B) Язык программирования
- C) Протокол передачи данных
- D) База данных

Вопрос 2:

Что такое CSS?

- A) Язык стилей для оформления HTML
- B) Скриптовый язык
- C) Сервер приложений
- D) Тип базы данных

Вопрос 3:

Какой тег создаёт гиперссылку в HTML?

- A) `<link>`
- B) `<a>`
- C) `<href>`
- D) `<url>`

Вопрос 4:

Что такое DOM?

- A) Document Object Model — объектная модель документа
- B) Database Object Model
- C) Dynamic Output Mode
- D) Data Object Method

Вопрос 5:

Какое свойство CSS отвечает за цвет текста?

- A) `font-color`
- B) `text-color`
- C) `color`
- D) `background-color`

Вопрос 6:

Что такое Flexbox в CSS?

- A) Модель компоновки для одномерного выравнивания элементов
- B) Анимация элементов

- C) Шрифтовое форматирование
- D) Медиазапрос

Вопрос 7:

Что делает JavaScript-функция `addEventListener`?

- A) Добавляет обработчик события к элементу
- B) Создает новый элемент
- C) Удаляет элемент
- D) Изменяет стиль

Вопрос 8:

Что такое асинхронность в JavaScript?

- A) Выполнение операций без блокировки основного потока
- B) Синхронное выполнение
- C) Параллельные потоки
- D) Работа с сервером

Вопрос 9:

Что возвращает `fetch()` в JavaScript?

- A) Promise
- B) String
- C) Number
- D) Object

Вопрос 10:

Что такое JSON?

- A) Формат обмена данными на основе JavaScript-объектов
- B) Язык программирования
- C) База данных
- D) Протокол HTTP

Вопрос 11:

Что такое REST API?

- A) Архитектурный стиль проектирования API на основе HTTP
- B) База данных
- C) Протокол шифрования
- D) Фреймворк

Вопрос 12:

Какой HTTP-метод используется для получения данных?

- A) POST
- B) GET
- C) PUT
- D) DELETE

Вопрос 13:

Какой HTTP-метод удаляет ресурс?

- A) GET
- B) POST
- C) DELETE
- D) PATCH

Вопрос 14:

Что такое SPA (Single Page Application)?

- A) Приложение с динамической загрузкой контента без перезагрузки страницы
- B) Многостраничное приложение
- C) Мобильное приложение
- D)/Desktopное приложение

Вопрос 15:

Для чего используется тег <meta> в HTML?

- A) Для отображения заголовка страницы
- B) Для метаданных страницы (кодировка, описание)
- C) Для создания форм
- D) Для добавления скриптов

Вопрос 16:

Что такое медиазапросы CSS?

- A) Условные стили для разных устройств и экранов
- B) Стили для печати
- C) Анимации CSS
- D) Переменные CSS

Вопрос 17:

Что такое семантический HTML?

- A) HTML с тегами, несущими смысловую нагрузку (header, nav, main)
- B) HTML без стилей
- C) HTML с JavaScript
- D) Минифицированный HTML

Вопрос 18:

Что такое PHP?

- A) Серверный скриптовый язык для веб-разработки
- B) Клиентский язык
- C) Язык разметки
- D) СУБД

Вопрос 19:

Какой тип данных в JavaScript хранит набор значений?

- A) String
- B) Array
- C) Boolean
- D) Number

Вопрос 20:

Что такое async/await в JavaScript?

- A) Синтаксический сахар для работы с Promise
- B) Библиотека запросов
- C) Тип данных
- D) Событие браузера

Вопрос 21:

Что такое CORS?

- A) Политика разрешения кросс-доменных запросов

- В) Тип шифрования
- С) Протокол аутентификации
- Д) CSS-фреймворк

Вопрос 22:

Какой HTTP-статус означает успешный запрос?

- А) 200
- В) 404
- С) 500
- Д) 301

Вопрос 23:

Что такое Git?

- А) Система контроля версий
- В) IDE для разработки
- С) Сервер приложений
- Д) Пакетный менеджер

Вопрос 24:

Что делает команда `git commit`?

- А) Сохраняет изменения в локальный репозиторий
- В) Отправляет изменения на сервер
- С) Создает новую ветку
- Д) Скачивает изменения

Вопрос 25:

Что такое `npm`?

- А) Менеджер пакетов для Node.js
- В) Фреймворк JavaScript
- С) База данных
- Д) Протокол

Вопрос 26:

Что такое XSS (Cross-Site Scripting)?

- А) Атака с внедрением вредоносного скрипта в веб-страницу
- В) Ошибка CSS
- С) Тип шифрования
- Д) Метод авторизации

Вопрос 27:

Что такое аутентификация в веб-приложениях?

- А) Проверка личности пользователя
- В) Разграничение прав доступа
- С) Шифрование данных
- Д) Кэширование запросов

Вопрос 28:

Что такое JWT (JSON Web Token)?

- А) Формат токена для передачи аутентификационных данных
- В) Тип базы данных
- С) JavaScript-фреймворк
- Д) HTTP-заголовок

Вопрос 29:

Что такое CSS Grid?

- A) Двумерная система компоновки элементов
- B) Библиотека иконок
- C) Анимация CSS
- D) Тип шрифта

Вопрос 30:

Какой тег используется для создания формы в HTML?

- A) <input>
- B) <form>
- C) <button>
- D) <field>

По результатам итоговой аттестации выставляются отметки по системе: «зачтено», «не зачтено».

Критерии оценивания итоговой аттестации

Тест выполнен:

более 70% правильных ответов - «зачтено»,
69 % и менее – «не зачтено»

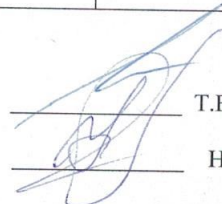
7. РАЗРАБОТЧИКИ ПРОГРАММЫ

Разработчики программы	Номера разработанных в программе дисциплин (модулей, разделов, тем) в соответствии с учебным планом
Иванов Андрей Геннадьевич, профессор кафедры «Информационные технологии и системы управления»	Модули 1,2,3

СОГЛАСОВАНО:

Начальник отдела ДПО

Заместитель директора филиала по УВР



Т.В. Лукьянова

Н.С. Малюткина