

Документ подписан простой электронной подписью
 Информация о владельце:
 ФИО: Стрельникова Светлана Евгеньевна
 Должность: ректор
 Дата подписания: 28.09.2026 16:46:35
 Уникальный программный ключ:
 ed7e7ef871ef28d870011250a92f17921078cc18

**Автономная некоммерческая образовательная организация высшего образования
 «Контур Университет»**



УТВЕРЖДАЮ

Ректор

КОНТУР
УНИВЕРСИТЕТ

С.Е. Стрельникова

28.09.2026 г.

**МОДУЛЬ 2. ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА
 ИНФОРМАЦИОННЫХ СИСТЕМ
 Web-технологии**

рабочая программа дисциплины (модуля)

Учебный план 09.03.03_ИИС_2026.plx
 09.03.03 Прикладная информатика

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **3 ЗЕТ**

Часов по учебному плану 108

Виды контроля в семестрах:

в том числе:

зачеты 6

аудиторные занятия 48

самостоятельная работа 58

часов на контроль 2

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	6 (3.2)		Итого	
Неделя	17 2/6			
Вид занятий	уп	рп	уп	рп
Лекции	16	16	16	16
Практические	32	32	32	32
Итого ауд.	48	48	48	48
Контактная работа	48	48	48	48
Сам. работа	58	58	58	58
Часы на контроль	2	2	2	2
Итого	108	108	108	108

Рабочая программа дисциплины

Web-технологии

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки
09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

09.03.03 Прикладная информатика

Одобрена Ученым советом вуза

Протокол от 15.09.2026 № 05.

Одобрена учебно-методической комиссией

Протокол от 10.09.2026 г. № 01

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	Цель освоения дисциплины:
1.2	Формирование у обучающихся системы знаний о современных Web-технологиях, протоколах, архитектуре и принципах разработки клиент-серверных приложений, а также развитие практических навыков создания пользовательских интерфейсов, разработки REST API и интеграции Web-приложений с внешними сервисами.
1.3	Задачи изучения дисциплины:
1.4	Ознакомить с архитектурой и протоколами Web (HTTP/HTTPS, REST, SOAP, WebSocket), принципами построения клиент-серверных приложений (SPA, PWA);
1.5	Сформировать знания о технологиях клиентской разработки: HTML5, CSS3, JavaScript, TypeScript и современных фронтенд-фреймворках (React, Angular, Vue.js);
1.6	Изучить принципы разработки и интеграции Web-сервисов и REST API для взаимодействия с внешними системами;
1.7	Развить навыки создания адаптивных и интерактивных пользовательских интерфейсов с учетом принципов UI/UX дизайна;
1.8	Сформировать умения разрабатывать одностраничные приложения (SPA) с использованием современных фреймворков и интегрировать их с бэкенд-сервисами.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В.12
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Программирование
2.1.2	Проектирование информационных систем
2.1.3	Информационные технологии и сервисы
2.1.4	Введение в профессиональную деятельность
2.2	Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Интеллектуальные информационные системы

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ИНДИКАТОРЫ ИХ ДОСТИЖЕНИЯ
ПК-4: Способен применять методы и программные средства сбора, обработки и анализа информации для информационно-аналитической поддержки принятия управленческих решений
ПК-4.1: Интерпретирует возможности и ограничения различных ИТ-решений для решения управленческих задач (автоматизация отчётности, прогнозирование, мониторинг KPI) на основе знаний о современных ИТ-архитектурах (клиент-сервер, облачные вычисления, технологии хранения и передачи данных)
Знать: современные ИТ-архитектуры (клиент-серверная, облачные вычисления, микросервисная); принципы и протоколы передачи данных в Web (HTTP/HTTPS, REST, SOAP, WebSocket); технологии хранения данных и их применение в Web-приложениях Уметь: интерпретировать возможности и ограничения различных ИТ-решений для управленческих задач; выбирать архитектурный подход в зависимости от требований к Web-приложению; оценивать применимость различных технологий для автоматизации отчётности и мониторинга Владеть: навыками анализа ИТ-решений с точки зрения их архитектурных особенностей; методами оценки применимости Web-технологий для решения управленческих задач; инструментами сравнительного анализа архитектурных подходов

ПК-3: Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации
ПК-3.4: Проектирует техническую архитектуру ИТ-сервиса (модули, компоненты, взаимодействия) на основе стандартов прототипирования, эргономики ПО и принципов построения API с использованием UML
Знать: стандарты прототипирования и эргономики программного обеспечения; принципы построения API (REST, SOAP); методологию проектирования с использованием UML (диаграммы классов, компонентов, последовательности) Уметь: проектировать техническую архитектуру ИТ-сервиса с выделением модулей и компонентов; разрабатывать API для взаимодействия компонентов системы; использовать UML для визуализации архитектурных решений Владеть:

навыками проектирования архитектуры Web-приложений;
методами документирования API и архитектурных решений;
инструментами моделирования архитектуры ИТ-сервисов

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Примечание
	Раздел 1.			
1.1	Введение в Web-технологии: клиент-серверная архитектура, доменные имена, хостинг /Лек/	6	2	
1.2	Введение в Web-технологии: клиент-серверная архитектура, доменные имена, хостинг /Пр/	6	2	
1.3	Введение в Web-технологии: клиент-серверная архитектура, доменные имена, хостинг /Ср/	6	2	
1.4	HTML5: структура веб-страницы, семантическая вёрстка, формы и таблицы /Лек/	6	1	
1.5	HTML5: структура веб-страницы, семантическая вёрстка, формы и таблицы /Пр/	6	2	
1.6	HTML5: структура веб-страницы, семантическая вёрстка, формы и таблицы /Ср/	6	4	
1.7	CSS3: основы стилизации, селекторы, блочная модель, позиционирование /Лек/	6	1	
1.8	CSS3: основы стилизации, селекторы, блочная модель, позиционирование /Пр/	6	2	
1.9	CSS3: основы стилизации, селекторы, блочная модель, позиционирование /Ср/	6	4	
1.10	CSS Flexbox и Grid: современные подходы к вёрстке и созданию макетов /Лек/	6	1	
1.11	CSS Flexbox и Grid: современные подходы к вёрстке и созданию макетов /Пр/	6	2	
1.12	CSS Flexbox и Grid: современные подходы к вёрстке и созданию макетов /Ср/	6	4	
1.13	Адаптивная и резиновая вёрстка: медиазапросы, работа с разными устройствами /Лек/	6	1	
1.14	Адаптивная и резиновая вёрстка: медиазапросы, работа с разными устройствами /Пр/	6	2	
1.15	Адаптивная и резиновая вёрстка: медиазапросы, работа с разными устройствами /Ср/	6	4	
1.16	Введение в JavaScript: переменные, типы данных, функции, условные операторы и циклы /Лек/	6	1	
1.17	Введение в JavaScript: переменные, типы данных, функции, условные операторы и циклы /Пр/	6	2	
1.18	Введение в JavaScript: переменные, типы данных, функции, условные операторы и циклы /Ср/	6	4	
1.19	Работа с DOM: поиск элементов, изменение содержимого и стилей, создание элементов /Лек/	6	1	
1.20	Работа с DOM: поиск элементов, изменение содержимого и стилей, создание элементов /Пр/	6	2	
1.21	Работа с DOM: поиск элементов, изменение содержимого и стилей, создание элементов /Ср/	6	4	
1.22	Обработка событий: клики, ввод текста, отправка форм, валидация данных /Лек/	6	1	
1.23	Обработка событий: клики, ввод текста, отправка форм, валидация данных /Пр/	6	2	
1.24	Обработка событий: клики, ввод текста, отправка форм, валидация данных /Ср/	6	4	
1.25	Асинхронный JavaScript: работа с сервером, получение данных, обновление страницы без перезагрузки /Лек/	6	1	
1.26	Асинхронный JavaScript: работа с сервером, получение данных, обновление страницы без перезагрузки /Пр/	6	2	

1.27	Асинхронный JavaScript: работа с сервером, получение данных, обновление страницы без перезагрузки /Ср/	6	4	
1.28	Основы серверной разработки: обработка запросов, маршрутизация, взаимодействие с базой данных /Лек/	6	1	
1.29	Основы серверной разработки: обработка запросов, маршрутизация, взаимодействие с базой данных /Пр/	6	2	
1.30	Основы серверной разработки: обработка запросов, маршрутизация, взаимодействие с базой данных /Ср/	6	4	
1.31	REST API: принципы, методы, работа с форматами JSON и XML /Лек/	6	1	
1.32	REST API: принципы, методы, работа с форматами JSON и XML /Пр/	6	2	
1.33	REST API: принципы, методы, работа с форматами JSON и XML /Ср/	6	4	
1.34	Аутентификация и авторизация: сессии, JWT-токены, защита маршрутов /Лек/	6	1	
1.35	Аутентификация и авторизация: сессии, JWT-токены, защита маршрутов /Пр/	6	2	
1.36	Аутентификация и авторизация: сессии, JWT-токены, защита маршрутов /Ср/	6	4	
1.37	Введение в современные фронтенд-фреймворки: компонентный подход, управление состоянием /Лек/	6	1	
1.38	Введение в современные фронтенд-фреймворки: компонентный подход, управление состоянием /Пр/	6	2	
1.39	Введение в современные фронтенд-фреймворки: компонентный подход, управление состоянием /Ср/	6	4	
1.40	Проектирование Web-приложений: архитектурные паттерны (MVC), UML-диаграммы /Лек/	6	1	
1.41	Проектирование Web-приложений: архитектурные паттерны (MVC), UML-диаграммы /Пр/	6	3	
1.42	Проектирование Web-приложений: архитектурные паттерны (MVC), UML-диаграммы /Ср/	6	4	
1.43	Интеграция Web-приложений с внешними сервисами, работа с API и визуализация данных /Лек/	6	1	
1.44	Интеграция Web-приложений с внешними сервисами, работа с API и визуализация данных /Пр/	6	3	
1.45	Интеграция Web-приложений с внешними сервисами, работа с API и визуализация данных /Ср/	6	4	
1.46	/Зачёт/	6	2	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации

Темы для обсуждения на практических занятиях:

Практическая работа № 1 «Анализ клиент-серверной архитектуры Web-приложений» (к теме 1.1)

Тема: Введение в Web-технологии: клиент-серверная архитектура, доменные имена, хостинг.

Цель: сформировать понимание клиент-серверной архитектуры, принципов работы доменных имён и хостинга.

Задания:

Нарисуйте схему клиент-серверного взаимодействия при открытии веб-страницы: от ввода адреса в браузере до отображения страницы.

Изучите структуру URL-адреса. Разберите любой сайт на составляющие: протокол, домен, путь, параметры.

Проведите анализ двух хостинг-провайдеров по критериям: стоимость, объём диска, поддержка баз данных, наличие SSL-сертификата. Составьте таблицу сравнения.

Проверьте, какие HTTP-заголовки передаются при открытии сайта (используйте инструменты разработчика в браузере).

Запишите основные заголовки запроса и ответа.

Форма отчётности: Схема взаимодействия, анализ структуры URL, сравнительная таблица хостингов.

Практическая работа № 2 «Создание веб-страницы с использованием HTML5» (к теме 1.2)

Тема: HTML5: структура веб-страницы, семантическая вёрстка, формы и таблицы.

Цель: сформировать навыки создания структурированных веб-страниц с использованием семантических тегов HTML5.

Задания:

Создайте HTML-страницу с использованием семантических тегов (header, nav, main, section, article, aside, footer). Тема — «Моя будущая профессия — специалист по интеллектуальным системам».

Сверстайте таблицу с данными о курсах валют (минимум 5 строк, 3 столбца: валюта, покупка, продажа). Используйте теги table, thead, tbody, tfoot.

Создайте форму обратной связи с полями: имя, email, телефон, тема обращения, текст сообщения, выбор приоритета (высокий, средний, низкий), кнопка отправки. Все поля должны иметь корректные атрибуты.

Добавьте на страницу ссылки на внешние ресурсы, изображения (локальные и внешние), маркированный и нумерованный списки.

Форма отчётности: HTML-файл, скриншот страницы.

Практическая работа № 3 «Стилизация веб-страницы с использованием CSS3» (к теме 1.3)

Тема: CSS3: основы стилизации, селекторы, блочная модель, позиционирование.

Цель: сформировать навыки стилизации веб-страниц с использованием CSS3.

Задания:

Возьмите HTML-страницу из предыдущей работы. Создайте внешний CSS-файл и подключите его.

Примените стили к различным элементам: задайте цвета (фоновые, текстовые), шрифты (Google Fonts), размеры, отступы, рамки.

Используйте различные типы селекторов: универсальный, по тегу, по классу, по идентификатору, вложенные, дочерние, соседние.

Поэкспериментируйте с блочной моделью: задайте padding, margin, border для разных элементов. Сравните, как ведут себя блочные и строчные элементы.

Используйте позиционирование: relative, absolute, fixed для размещения элементов на странице (например, фиксированная шапка или плавающая панель).

Форма отчётности: HTML-файл, CSS-файл, скриншот страницы.

Практическая работа № 4 «Создание макета с использованием Flexbox и Grid» (к теме 1.4)

Тема: CSS Flexbox и Grid: современные подходы к вёрстке и созданию макетов.

Цель: сформировать навыки создания макетов с использованием Flexbox и Grid.

Задания:

Создайте макет карточки товара (изображение, название, описание, цена, кнопка) с использованием Flexbox. Товары должны выстраиваться в строку и переноситься на следующую строку при недостатке места.

Создайте макет страницы с «шахматной» сеткой (2 или 3 колонки) с использованием CSS Grid. Элементы должны занимать разное количество колонок и строк.

Реализуйте макет сайта с шапкой, основным контентом (2 колонки: основная и боковая) и подвалом. Используйте Grid для всей страницы и Flexbox для внутренних элементов.

Создайте галерею изображений с использованием Grid. Изображения должны адаптироваться по размеру и сохранять пропорции.

Форма отчётности: HTML-файл, CSS-файл, скриншот страницы.

Практическая работа № 5 «Адаптивная вёрстка с использованием медиазапросов» (к теме 1.5)

Тема: Адаптивная и резиновая вёрстка: медиазапросы, работа с разными устройствами.

Цель: сформировать навыки создания адаптивных интерфейсов для разных устройств.

Задания:

Возьмите макет из предыдущей работы. Добавьте медиазапросы для трёх точек перелома: десктоп (>1024px), планшет (768-1024px), мобильное устройство (<768px).

Настройте отображение элементов для разных устройств: на мобильных устройствах колонки должны располагаться вертикально, меню должно сворачиваться в гамбургер, шрифты уменьшаться.

Используйте относительные единицы измерения (% , rem , vh , vw) вместо фиксированных пикселей.

Добавьте мета-тег viewport для корректного отображения на мобильных устройствах.

Протестируйте адаптивность с помощью инструментов разработчика в браузере (эмуляция разных устройств).

Форма отчётности: HTML-файл, CSS-файл с медиазапросами, скриншоты для трёх устройств.

Практическая работа № 6 «Программирование на JavaScript: основы» (к теме 1.6)

Тема: Введение в JavaScript: переменные, типы данных, функции, условные операторы и циклы.

Цель: сформировать базовые навыки программирования на JavaScript.

Задания:

Напишите программу для расчёта стоимости заказа с учётом скидки: если сумма больше 5000 рублей — скидка 10%, если больше 10000 — скидка 15%, иначе скидки нет.

Напишите функцию, которая принимает массив чисел и возвращает: сумму всех чисел, среднее арифметическое, максимальное и минимальное значение.

Используя циклы, выведите в консоль все числа от 1 до 100, которые делятся на 3 и на 5 одновременно.

Напишите программу для проверки, является ли введённая строка палиндромом (читается одинаково слева направо и справа налево).

Создайте объект «Товар» с полями: название, цена, количество на складе. Напишите функцию, которая рассчитывает общую стоимость всех товаров на складе.

Форма отчётности: HTML-файл с подключённым JavaScript (консоль), код в виде отдельных функций.

Практическая работа № 7 «Работа с DOM в JavaScript» (к теме 1.7)

Тема: Работа с DOM: поиск элементов, изменение содержимого и стилей, создание элементов.

Цель: сформировать навыки управления содержимым и структурой веб-страницы через JavaScript.

Задания:

Создайте HTML-страницу с пустым списком. Напишите JavaScript-код, который добавляет элементы в список по нажатию на кнопку (значение берётся из поля ввода).

Напишите функцию, которая меняет цвет фона страницы случайным образом при нажатии на кнопку.

Создайте страницу с таблицей. Добавьте JavaScript-код для подсчёта суммы значений в столбце (например, «цена» или «количество») и отображения итоговой суммы.

Реализуйте фильтрацию списка: при вводе текста в поле поиска отображаются только те элементы, которые содержат введённый текст.

Создайте галерею изображений: миниатюры внизу, большое изображение в центре. При клике на миниатюру меняется большое изображение.

Форма отчётности: HTML-файл, CSS-файл, JavaScript-файл.

Практическая работа № 8 «Обработка событий и валидация форм» (к теме 1.8)

Тема: Обработка событий: клики, ввод текста, отправка форм, валидация данных.

Цель: сформировать навыки обработки событий и валидации форм на стороне клиента.

Задания:

Создайте форму регистрации с полями: имя (обязательное), email (обязательный, корректный формат), пароль (минимум 6 символов), подтверждение пароля. Реализуйте валидацию на JavaScript: при отправке формы проверяются все поля, ошибки отображаются под соответствующими полями.

Реализуйте динамическую проверку пароля: при вводе пароля отображается индикатор надёжности (слабый/средний/сильный) в зависимости от длины и наличия специальных символов.

Добавьте маску ввода для поля телефона: автоматическое форматирование при вводе (например, +7 (XXX) XXX-XX-XX).

Реализуйте «живой» поиск по списку товаров: при вводе текста в поле поиска список фильтруется в реальном времени.

Добавьте анимацию при наведении на кнопку (изменение цвета, размера, тени).

Форма отчётности: HTML-файл, CSS-файл, JavaScript-файл.

Практическая работа № 9 «Асинхронный JavaScript: работа с сервером» (к теме 1.9)

Тема: Асинхронный JavaScript: работа с сервером, получение данных, обновление страницы без перезагрузки.

Цель: сформировать навыки выполнения асинхронных запросов к серверу и обновления страницы.

Задания:

Используя публичное API (например, для получения курсов валют, прогноза погоды), получите данные и отобразите их на странице. Используйте fetch с async/await.

Добавьте на страницу кнопку «Обновить данные», которая повторно запрашивает данные с сервера без перезагрузки страницы.

Реализуйте обработку ошибок: при недоступности сервера или ошибке в данных показывайте сообщение пользователю.

Создайте страницу для поиска: пользователь вводит запрос, данные подгружаются с сервера в реальном времени (debounce для уменьшения числа запросов).

Отобразите индикатор загрузки (спиннер) во время выполнения запроса к серверу.

Форма отчётности: HTML-файл, CSS-файл, JavaScript-файл.

Практическая работа № 10 «Создание простого сервера на Node.js» (к теме 1.10)

Тема: Основы серверной разработки: обработка запросов, маршрутизация, взаимодействие с базой данных.

Цель: сформировать навыки создания простого серверного приложения.

Задания:

Установите Node.js. Создайте простой HTTP-сервер, который обрабатывает GET-запросы и возвращает HTML-страницу «Hello World!».

Реализуйте маршруты: / (главная), /about (страница о сайте), /api/data (возвращает JSON-ответ). Используйте модуль http или express.

Организируйте хранение данных в JSON-файле: реализуйте CRUD-операции (создание, чтение, обновление, удаление) для списка задач (To-Do).

Внедрите работу с базой данных (например, SQLite): создайте таблицу users, реализуйте добавление и получение пользователей.

Реализуйте обработку POST-запросов: принимайте данные из формы, сохраняйте их на сервере, возвращайте ответ об успешном сохранении.

Форма отчётности: Код сервера, примеры запросов с помощью Postman или curl.

Практическая работа № 11 «Разработка и тестирование REST API» (к теме 1.11)

Тема: REST API: принципы, методы, работа с форматами JSON и XML.

Цель: сформировать навыки проектирования и тестирования REST API.

Задания:

Разработайте REST API для управления списком товаров (название, цена, количество, категория). Реализуйте следующие эндпоинты:

GET /api/products — получить список всех товаров.

GET /api/products/:id — получить товар по id.

POST /api/products — добавить новый товар.

PUT /api/products/:id — обновить товар.

DELETE /api/products/:id — удалить товар.

Протестируйте API с помощью Postman. Для каждого метода подготовьте примеры запросов и ответов.

Реализуйте пагинацию (постраничный вывод) для списка товаров (GET /api/products?page=1&limit=10).

Добавьте сортировку и фильтрацию (по цене, категории, наличию).

Документируйте API: опишите каждый эндпоинт, параметры запроса, структуру ответа, коды ошибок.

Форма отчётности: Код API, коллекция Postman, документация API.

Практическая работа № 12 «Аутентификация и авторизация» (к теме 1.12)

Тема: Аутентификация и авторизация: сессии, JWT-токены, защита маршрутов.

Цель: сформировать навыки реализации аутентификации и авторизации в Web-приложениях.

Задания:

Реализуйте регистрацию пользователя: сохраняйте хеш пароля (bcrypt) в базе данных.

Реализуйте вход пользователя: проверяйте пароль, генерируйте JWT-токен и возвращайте его клиенту.

Настройте защищённые маршруты: доступ к ним возможен только при наличии валидного JWT-токена.

Реализуйте обновление токена (refresh token) или установите время жизни токена.

Создайте страницу профиля пользователя, доступную только авторизованным пользователям.

Реализуйте выход из системы (логат): удаление токена на клиенте.

Форма отчётности: Код сервера, примеры запросов в Postman, клиентская часть с авторизацией.

Практическая работа № 13 «Создание простого SPA с использованием фреймворка» (к теме 1.13)

Тема: Введение в современные фронтенд-фреймворки: компонентный подход, управление состоянием.

Цель: сформировать навыки создания одностраничного приложения с использованием фреймворка.

Задания (на примере React или Vue.js):

Установите и настройте проект с использованием фреймворка (Create React App или Vue CLI).

Создайте компоненты: App, Header, Footer, TodoList, TodoItem, AddTodo.

Реализуйте состояние приложения: список задач (массив объектов с полями id, text, completed).

Реализуйте функциональность: добавление задачи, удаление задачи, отметка задачи как выполненной, фильтрация (все/активные/выполненные).

Интегрируйте SPA с вашим REST API: загружайте список задач с сервера, отправляйте изменения (добавление, удаление, обновление) на сервер.

Добавьте стилизацию: сделайте интерфейс удобным и адаптивным.

Форма отчётности: Код проекта (ссылка на репозиторий), скриншоты работающего приложения.

Практическая работа № 14 «Проектирование архитектуры Web-приложения» (к теме 1.14)

Тема: Проектирование Web-приложений: архитектурные паттерны (MVC), UML-диаграммы.

Цель: сформировать навыки проектирования архитектуры Web-приложений и документирования решений.

Задания:

Разработайте архитектуру Web-приложения для управления задачами (To-Do лист) с использованием паттерна MVC: опишите модель (данные), представление (интерфейс), контроллер (логика обработки запросов).

Постройте диаграмму классов для вашего приложения (UML): укажите классы, их свойства, методы и связи.

Постройте диаграмму компонентов: покажите, как связаны клиентская и серверная части, база данных.

Постройте диаграмму последовательности для одного из сценариев использования (например, добавление задачи).

Разработайте схему базы данных для приложения (ER-диаграмма): укажите таблицы, поля, связи (минимум 3 таблицы).

Форма отчётности: UML-диаграммы, ER-диаграмма, описание архитектуры.

Практическая работа № 15 «Интеграция Web-приложения с внешними сервисами» (к теме 1.15)

Тема: Интеграция Web-приложений с внешними сервисами, работа с API и визуализация данных.

Цель: сформировать навыки интеграции Web-приложений с внешними API и визуализации данных.

Задания:

Выберите внешний API (например, для получения курсов валют, прогноза погоды, данных о фильмах, новостей). Получите данные и отобразите их на странице.

Интегрируйте полученные данные с вашим SPA: создайте отдельный компонент для отображения данных (например, виджет погоды или таблица курсов валют).

Реализуйте визуализацию данных с использованием библиотеки для построения графиков (Chart.js, D3.js или аналоги).

Отобразите данные в виде диаграммы или графика (например, изменение курса валют за неделю, прогноз температуры).

Добавьте функциональность обновления данных по расписанию (каждые 5-10 секунд) или по нажатию кнопки.

Обработайте ошибки интеграции: если API недоступен, покажите понятное сообщение пользователю.

Оптимизируйте запросы: используйте кэширование для уменьшения количества запросов к внешнему API.

Форма отчётности: Код проекта, скриншоты с визуализацией данных.

Примерные вопросы к зачету:

1. Что такое клиент-серверная архитектура? Опишите компоненты и их взаимодействие.
2. Какие существуют типы Web-приложений (статический сайт, динамический сайт, SPA, PWA, микросервисы)? В чём их отличия?
3. Как устроена работа доменных имён? Опишите роль DNS.
4. Что такое протокол HTTP? Опишите структуру HTTP-запроса и HTTP-ответа.
5. Какие методы HTTP-запросов существуют? Для чего используется каждый метод?
6. Что такое HTTPS? Как он обеспечивает безопасность передачи данных?
7. Что такое REST API? Какие принципы лежат в основе REST?
8. Какова структура HTML-документа? Что такое семантическая вёрстка и зачем она нужна?
9. Какие семантические теги HTML5 вы знаете? Для чего используется каждый из них?

10. Как создать таблицу и форму в HTML? Какие теги используются?
11. Какие основные типы селекторов в CSS вы знаете? В чём их различия?
12. Что такое блочная модель CSS? Из каких компонентов она состоит?
13. Какие виды позиционирования есть в CSS? В чём их различие?
14. Что такое Flexbox? Для решения каких задач он используется?
15. Что такое CSS Grid? В чём отличие Grid от Flexbox?
16. Что такое адаптивная вёрстка? Что такое медиазапросы и как они работают?
17. Какие основные типы данных в JavaScript вы знаете?
18. В чём разница между var, let и const в JavaScript?
19. Что такое функция в JavaScript? Как объявить функцию и передать параметры?
20. Что такое DOM? Как получить доступ к элементам страницы и изменить их через JavaScript?
21. Что такое события в JavaScript? Как обрабатывать события?
22. Что такое асинхронное программирование в JavaScript? Что такое промисы и async/await?
23. Что такое fetch? Как с его помощью получить данные с сервера?
24. Как создать простой HTTP-сервер на Node.js? Что такое маршрутизация?
25. Что такое REST API? Как проектируются эндпоинты в REST API?
26. Что такое JWT-токен? Как он используется для аутентификации?
27. Какие основные угрозы безопасности существуют в Web-приложениях (XSS, CSRF, SQL-инъекции) и как с ними бороться?
28. Что такое SPA (Single Page Application)? В чём её преимущества перед традиционными сайтами?
29. Что такое компонентный подход в разработке интерфейсов? Какие современные фреймворки вы знаете?
30. Какие архитектурные паттерны используются при разработке Web-приложений (MVC, MVVM)? Как организовать интеграцию Web-приложения с внешними сервисами?

5.2. Темы письменных работ

не предусмотрены

5.3. Фонд оценочных средств

Тестовые задания:

ВОПРОСЫ НА СООТВЕТСТВИЕ (№ 1–5)

Инструкция: к каждому элементу из левого столбца подберите ОДИН соответствующий элемент из правого столбца. Ответ запишите в виде сочетания цифр и букв (например: 1-А, 2-Б, 3-В, 4-Г). Каждая буква используется только один раз.

ПК-3.4. Вопрос № 1. Соответствие между HTTP-методом и его описанием

Установите соответствие между HTTP-методом и его описанием:

HTTP-метод Описание

1. GET А. Создание нового ресурса на сервере
2. POST Б. Удаление ресурса на сервере
3. PUT В. Получение данных с сервера
4. DELETE Г. Полное обновление существующего ресурса
5. PATCH Д. Частичное обновление ресурса

Ответ: 1 __, 2 __, 3 __, 4 __, 5 __

ПК-3.4. Вопрос № 2. Соответствие между протоколом и его описанием

Установите соответствие между протоколом и его описанием:

Протокол Описание

1. HTTP А. Безопасная передача данных в Web (с шифрованием)
2. HTTPS Б. Двусторонняя связь в реальном времени
3. WebSocket В. Протокол передачи гипертекста
4. REST Г. Архитектурный стиль для построения API

Ответ: 1 __, 2 __, 3 __, 4 __

ПК-3.4. Вопрос № 3. Соответствие между методом массива в JavaScript и его описанием

Установите соответствие между методом и его описанием:

Метод Описание

1. map() А. Создаёт новый массив с элементами, прошедшими проверку
2. filter() Б. Преобразует массив в строку с разделителем
3. join() В. Создаёт новый массив, применяя функцию к каждому элементу
4. reduce() Г. Сворачивает массив к одному значению

Ответ: 1 __, 2 __, 3 __, 4 __

ПК-4.1. Вопрос № 4. Соответствие между свойством CSS и его описанием

Установите соответствие между свойством CSS и его описанием:

Свойство Описание

1. display: flex А. Активация CSS Grid для элемента
2. display: grid Б. Активация Flexbox для элемента
3. flex-direction В. Направление оси в Flexbox (row/column)
4. grid-template-columns Г. Определение количества и ширины колонок в Grid

Ответ: 1 __, 2 __, 3 __, 4 __

ПК-3.4. Вопрос № 5. Соответствие между термином REST API и его описанием

Установите соответствие между термином и его описанием:

Термин Описание

1. Эндпоинт А. Формат данных, используемый в REST API
2. JSON Б. Уникальный URL для доступа к ресурсу
3. Статус-код В. Числовой код результата выполнения запроса
4. Заголовки Г. Дополнительные параметры запроса/ответа

Ответ: 1 __, 2 __, 3 __, 4 __

ВОПРОСЫ С ОДНИМ ВЕРНЫМ ОТВЕТОМ (№ 6–15)

Инструкция: прочитайте текст, выберите правильный вариант ответа и запишите только букву выбранного варианта ответа (А, Б, В или Г).

ПК-3.4. Вопрос № 6. Какой тег HTML используется для создания ссылки?

- А) <link>
- Б) <a>
- В) <href>
- Г) <url>

Ответ:

ПК-3.4. Вопрос № 7. Какой HTTP-метод используется для получения данных с сервера?

- А) POST
- Б) PUT
- В) GET
- Г) DELETE

Ответ:

ПК-3.4. Вопрос № 8. Какое свойство CSS используется для задания цвета текста?

- А) background-color
- Б) color
- В) font-size
- Г) text-style

Ответ:

ПК-3.4. Вопрос № 9. Какой метод JavaScript используется для добавления элемента в конец массива?

- А) push()
- Б) pop()
- В) shift()
- Г) unshift()

Ответ:

ПК-3.4. Вопрос № 10. Какой протокол обеспечивает защищённую передачу данных в Web?

- А) HTTP
- Б) HTTPS
- В) FTP
- Г) WebSocket

Ответ:

ПК-3.4. Вопрос № 11. Какой статус-код HTTP означает успешное выполнение запроса?

- А) 200
- Б) 404
- В) 500
- Г) 403

Ответ:

ПК-3.4. Вопрос № 12. Какое свойство CSS используется для создания адаптивной вёрстки?

- А) display
- Б) position
- В) @media
- Г) flex

Ответ:

ПК-3.4. Вопрос № 13. Какой метод в JavaScript используется для выполнения запроса к серверу?

- А) fetch()
- Б) request()
- В) send()
- Г) get()

Ответ:

ПК-3.4. Вопрос № 14. Какой HTML-тег используется для создания неупорядоченного списка?

- А)
- Б)
- В)
- Г) <list>

Ответ:

ПК-3.4. Вопрос № 15. Какое свойство CSS используется для создания отступа внутри элемента?

- А) margin
- Б) padding
- В) border
- Г) spacing

Ответ:

ВОПРОСЫ С НЕСКОЛЬКИМИ ВЕРНЫМИ ОТВЕТАМИ (№ 16–20)

Инструкция: прочитайте текст, выберите не менее двух правильных вариантов ответов и запишите только буквы выбранных вариантов ответов (А, Б, В или Г).

ПК-3.4. Вопрос № 16. Какие методы HTTP-запросов относятся к безопасным (не изменяющим состояние сервера)?

- А) GET
- Б) POST
- В) PUT
- Г) DELETE

Ответ:

ПК-3.4. Вопрос № 17. Какие теги HTML5 являются семантическими?

- А) <header>
- Б) <div>
- В) <section>
- Г) <article>

Ответ:

ПК-3.4. Вопрос № 18. Какие типы позиционирования существуют в CSS?

- А) static
- Б) relative
- В) absolute
- Г) fixed

Ответ:

ПК-4.1. Вопрос № 19. Какие форматы данных используются в REST API?

- А) JSON
- Б) XML
- В) HTML
- Г) CSV

Ответ:

ПК-3.4. Вопрос № 20. Какие из перечисленных инструментов относятся к фронтенд-фреймворкам/библиотекам?

- А) React
- Б) Node.js
- В) Angular
- Г) Vue.js

Ответ:

ВОПРОСЫ С ОДНИМ АРГУМЕНТИРОВАННЫМ ОТВЕТОМ (№ 21–25)

Инструкция: Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.

ПК-3.4. Вопрос № 21. Какой протокол следует использовать для передачи данных между клиентом и сервером в реальном времени (например, для чата)?

- А) HTTP
- Б) WebSocket
- В) FTP

Ответ:

ПК-3.4. Вопрос № 22. Какое расширение файла используется для каскадных таблиц стилей?

- А) .html
- Б) .css
- В) .js

Ответ:

ПК-3.4. Вопрос № 23. Какой метод в JavaScript используется для преобразования объекта в строку JSON?

А) JSON.stringify()

Б) JSON.parse()

В) JSON.encode()

Ответ:

ПК-3.4. Вопрос № 24. Какой атрибут тега <input> используется для указания обязательного заполнения поля в форме?

А) required

Б) mandatory

В) validate

Ответ:

ПК-4.1. Вопрос № 25. Какой подход к разработке позволяет создавать приложения, которые работают без перезагрузки страницы и загружают только необходимые данные?

А) Многостраничное приложение (MPA)

Б) Одностраничное приложение (SPA)

В) Статический сайт

Ответ:

ВОПРОСЫ НА УСТАНОВЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ (№ 26–30)

Инструкция: Прочитайте текст, установите правильную последовательность действий и запишите аргументы, обосновывающие выбранный порядок.

ПК-3.4. Вопрос № 26. Установите правильную последовательность этапов выполнения GET-запроса.

А) Отправка запроса серверу

Б) Получение HTML-страницы

В) Ввод URL в браузере

Г) Обработка ответа и отображение страницы

Ответ:

ПК-3.4. Вопрос № 27. Установите правильную последовательность подготовки и отправки формы на сервер.

А) Обработка данных и отправка запроса

Б) Заполнение формы пользователем

В) Валидация данных на стороне клиента

Г) Получение ответа и обновление страницы

Ответ:

ПК-3.4. Вопрос № 28. Установите правильную последовательность этапов обработки запроса в REST API.

А) Маршрутизация к соответствующему обработчику

Б) Получение HTTP-запроса на сервере

В) Формирование и отправка JSON-ответа клиенту

Г) Выполнение бизнес-логики и работа с БД

Ответ:

ПК-4.1. Вопрос № 29. Установите правильную последовательность этапов создания SPA.

А) Интеграция с REST API

Б) Разработка компонентов интерфейса

В) Настройка маршрутизации

Г) Управление состоянием приложения

Ответ:

ПК-3.4. Вопрос № 30. Установите правильную последовательность действий при разработке Web-приложения.

А) Проектирование архитектуры (UML-диаграммы)

Б) Разработка и тестирование кода

В) Постановка задачи и сбор требований

Г) Внедрение и развёртывание приложения

Ответ:

5.4. Перечень видов оценочных средств и критерии оценивания

Контроль результатов освоения дисциплины является формой управления качеством образования.

Контроль результатов освоения дисциплины осуществляется в виде текущего контроля и промежуточной аттестации (зачета).

Система текущего контроля включает:

1. контроль работы обучающихся на лекционных и практических занятиях;
2. контроль участия в совместной работе группы;
3. контроль выполнения индивидуальных заданий;
4. контроль выполнения обучающимися заданий для самостоятельной работы.

Промежуточная аттестация осуществляется в виде зачета, который проводится в устной либо письменной форме.

Задачи промежуточной аттестации:

1. определить общий уровень освоения дисциплины в целом;

2. определить уровень основных знаний по каждой теме курса;
3. объективизировать результат контроля, минимизировать возможную субъективность преподавателя.

Критерии и рубрики оценивания Элемент контроля/Критерий/Индикаторы/Уровни (0–5)/Комментарий к уровню «5»

Практическая работа № 1 «Анализ клиент-серверной архитектуры Web-приложений» (к теме 1.1)

Элемент контроля: Понимание клиент-серверной архитектуры, принципов работы доменных имён и хостинга / ПК-4.1 / 0 / 3 / 5 / Схема клиент-серверного взаимодействия содержит все ключевые компоненты и этапы; анализ структуры URL выполнен корректно; сравнительная таблица хостинг-провайдеров заполнена по 4 критериям с обоснованными выводами; HTTP-заголовки проанализированы и зафиксированы; сделаны аргументированные выводы о работе Web-приложений.

Практическая работа № 2 «Создание веб-страницы с использованием HTML5» (к теме 1.2)

Элемент контроля: Навыки создания структурированных веб-страниц с использованием семантических тегов HTML5 / ПК-3.5 / 0 / 3 / 5 / Страница содержит все необходимые семантические теги (header, nav, main, section, article, aside, footer); таблица сверстана корректно с использованием thead, tbody, tfoot; форма содержит все требуемые поля с корректными атрибутами; присутствуют ссылки, изображения, списки; код валиден и соответствует стандартам HTML5.

Практическая работа № 3 «Стилизация веб-страницы с использованием CSS3» (к теме 1.3)

Элемент контроля: Навыки стилизации веб-страниц с использованием CSS3, понимание блочной модели и позиционирования / ПК-3.4 / 0 / 3 / 5 / Внешний CSS-файл подключен корректно; использованы различные типы селекторов; применены стили к элементам (цвета, шрифты, размеры, отступы, рамки); продемонстрировано понимание блочной модели; использованы разные виды позиционирования (relative, absolute, fixed) с корректными результатами; код чистый и структурированный.

Практическая работа № 4 «Создание макета с использованием Flexbox и Grid» (к теме 1.4)

Элемент контроля: Навыки создания макетов с использованием Flexbox и Grid / ПК-3.4 / 0 / 3 / 5 / Реализованы все три макета (карточки товаров, шахматная сетка, макет сайта); Flexbox используется корректно для выравнивания и переноса элементов; Grid используется для создания сложных сеток с разным размером ячеек; макет адаптивен и корректно отображается; код чистый, стили организованы.

Практическая работа № 5 «Адаптивная вёрстка с использованием медиазапросов» (к теме 1.5)

Элемент контроля: Навыки создания адаптивных интерфейсов для разных устройств / ПК-3.4 / 0 / 3 / 5 / Реализованы медиазапросы для 3 точек перелома; на мобильных устройствах элементы располагаются вертикально, меню свернуто, шрифты уменьшены; использованы относительные единицы измерения; добавлен мета-тег viewport; адаптивность протестирована на разных устройствах; скриншоты демонстрируют корректное отображение.

Практическая работа № 6 «Программирование на JavaScript: основы» (к теме 1.6)

Элемент контроля: Базовые навыки программирования на JavaScript / ПК-3.4 / 0 / 3 / 5 / Все 5 задач решены корректно; расчёт скидки выполнен с учётом всех условий; функции для работы с массивами реализованы правильно; циклы и условные операторы использованы корректно; строка на палиндром проверена; объект «Товар» создан, функция расчёта стоимости работает; код читаемый, переменные названы осмысленно.

Практическая работа № 7 «Работа с DOM в JavaScript» (к теме 1.7)

Элемент контроля: Навыки управления содержимым и структурой веб-страницы через JavaScript / ПК-3.4 / 0 / 3 / 5 / Реализованы все 5 заданий: добавление элементов в список, изменение цвета фона, подсчёт суммы в таблице, фильтрация списка, галерея изображений; манипуляции с DOM выполняются корректно; элементы создаются и добавляются динамически; поиск элементов выполнен правильно; интерактивность работает без ошибок.

Практическая работа № 8 «Обработка событий и валидация форм» (к теме 1.8)

Элемент контроля: Навыки обработки событий и валидации форм на стороне клиента / ПК-3.4 / 0 / 3 / 5 / Реализована полная валидация формы регистрации с отображением ошибок; индикатор надёжности пароля работает корректно; маска ввода телефона реализована; «живой» поиск по списку работает в реальном времени; анимация при наведении на кнопку реализована; обработчики событий корректны.

Практическая работа № 9 «Асинхронный JavaScript: работа с сервером» (к теме 1.9)

Элемент контроля: Навыки выполнения асинхронных запросов к серверу и обновления страницы / ПК-3.4, ПК-4.1 / 0 / 3 / 5 / Данные успешно получены с публичного API и отображены на странице; реализована кнопка обновления данных; обработка ошибок выполнена корректно; реализован поиск с debounce; индикатор загрузки отображается; код с использованием async/await соответствует современным стандартам.

Практическая работа № 10 «Создание простого сервера на Node.js» (к теме 1.10)

Элемент контроля: Навыки создания простого серверного приложения / ПК-3.4 / 0 / 3 / 5 / HTTP-сервер создан и работает; реализованы маршруты (/ , /about , /api/data); реализованы CRUD-операции для списка задач; работа с базой данных реализована; обработка POST-запросов выполнена корректно; код структурирован и документирован.

Практическая работа № 11 «Разработка и тестирование REST API» (к теме 1.11)

Элемент контроля: Навыки проектирования и тестирования REST API / ПК-3.4, ПК-4.1 / 0 / 3 / 5 / Реализованы все эндпоинты (GET, POST, PUT, DELETE); реализованы пагинация, сортировка и фильтрация; API протестирован в Postman с

примерами запросов и ответов; документация API содержит описание всех эндпоинтов, параметров и кодов ошибок; структура кода соответствует принципам REST.

Практическая работа № 12 «Аутентификация и авторизация» (к теме 1.12)

Элемент контроля: Навыки реализации аутентификации и авторизации в Web-приложениях / ПК-3.4 / 0 / 3 / 5 / Реализована регистрация с хешированием пароля; реализован вход с генерацией JWT-токена; защищённые маршруты работают корректно; реализован логгаут; страница профиля доступна только авторизованным пользователям; код соответствует принципам безопасности.

Практическая работа № 13 «Создание простого SPA с использованием фреймворка» (к теме 1.13)

Элемент контроля: Навыки создания одностраничного приложения с использованием фреймворка / ПК-3.4 / 0 / 3 / 5 / Проект настроен корректно; созданы все необходимые компоненты; реализовано состояние приложения; функциональность (добавление, удаление, отметка, фильтрация) работает; интеграция с REST API выполнена; интерфейс стилизован и адаптивен; код соответствует принципам компонентного подхода.

Практическая работа № 14 «Проектирование архитектуры Web-приложения» (к теме 1.14)

Элемент контроля: Навыки проектирования архитектуры Web-приложений и документирования решений / ПК-3.4 / 0 / 3 / 5 / Архитектура описана в соответствии с паттерном MVC; диаграмма классов содержит все классы, свойства, методы и связи; диаграмма компонентов показывает связи между частями системы; диаграмма последовательности построена для ключевого сценария; ER-диаграмма содержит минимум 3 таблицы с полями и связями; все диаграммы оформлены в соответствии со стандартами UML.

Практическая работа № 15 «Интеграция Web-приложения с внешними сервисами» (к теме 1.15)

Элемент контроля: Навыки интеграции Web-приложений с внешними API и визуализации данных / ПК-4.1, ПК-3.4 / 0 / 3 / 5 / Данные с внешнего API получены и отображены на странице; создан компонент для отображения данных; реализована визуализация данных с использованием библиотеки графиков; реализовано обновление данных по кнопке или по расписанию; обработка ошибок выполнена корректно; реализовано кэширование запросов.

Тест

Корректность ответов на вопросы: $\geq 90\%$ правильных ответов — 5; /75–89% — 4; /60–74% — 3; /< 60% — 2.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

Основная литература

1. 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>
2. Сысолетин, Е. Г. Разработка интернет-приложений : учебник для среднего профессионального образования / Е. Г. Сысолетин, С. Д. Ростунцев, Л. Г. Доросинский. — Москва : Издательство Юрайт, 2025. — 80 с. — (Профессиональное образование). — ISBN 978-5-534-19603-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/565692>
3. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 227 с. — (Высшее образование). — ISBN 978-5-534-17323-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://www.urait.ru/bcode/539651>
4. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588667>
5. Гаврилов, М. В. Информатика и информационные технологии : учебник для вузов / М. В. Гаврилов, В. А. Климов. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 319 с. — (Высшее образование). — ISBN 978-5-534-20354-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/558000>
6. Нечаев, А. М. Компьютерные сети : учебник и практикум для вузов / А. М. Нечаев, А. Е. Трубин, А. Ю. Анисимов [и др.] ; под научной редакцией А. М. Нечаева. — Москва : Издательство Юрайт, 2026. — 515 с. — (Высшее образование). — ISBN 978-5-534-21452-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590190>
7. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 227 с. — (Профессиональное образование). — ISBN 978-5-534-17319-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539652>
8. Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 319 с. — (Профессиональное образование). — ISBN 978-5-534-20333-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://web5.urait.ru/bcode/560669>

6.2. Электронные учебные издания и электронные образовательные ресурсы

Э1	Сетевые ресурсы Государственного автономного учреждения культуры Свердловской области «Свердловская областная универсальная научная библиотека им. В.Г. Белинского». Режим доступа: https://book.urait.ru/el_library/resursy_dostup
Э2	Образовательная платформа Юрайт. Режим доступа: https://urait.ru/

ЭЗ	Научная библиотека Cyberleninka. Режим доступа: https://cyberleninka.ru/
6.3.1 Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства	
6.3.1.1	Операционная система Astra Linux
6.3.1.2	Офисная система LibreOffice
6.3.1.3	Редакторы кода и HTML/CSS с открытой лицензией (например, Brackets, свободно распространяемое ПО, лицензия MIT); средства для создания веб-страниц (WebSite X5 Free, бесплатная версия); библиотеки JavaScript с открытыми лицензиями (jQuery, лицензия MIT); веб-браузеры с открытой лицензией (Google Chrome, Mozilla Firefox); средства для отладки веб-приложений; веб-сервер для локальной разработки (OSPanel, свободно распространяемое ПО)
6.3.1.4	Контур.Класс
6.3.1.5	Контур.Толк
6.3.1.6	ИС:Университет ПРОФ
6.3.1.7	Среда электронного обучения 3LK (русский Moodle)

7. МТО (оборудование и технические средства обучения)	
7.1	Учебные аудитории для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенные оборудованием и техническими средствами обучения:
7.2	Адрес (местоположение) учебных аудиторий Перечень оборудования и технических средств обучения
7.3	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 3 площадью 66,2 м² в помещении 501 Стол – 13 шт.
7.4	Стул – 25 шт.
7.5	ПК (с возможностью подключения к сети «Интернет») – 1 шт.
7.6	Телевизор (75”) – 1 шт.
7.7	Беспроводная система презентаций – 1 шт.
7.8	Спикерфон – 1 шт.
7.9	ПК (ноутбук) – 12 шт.
7.10	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 19 площадью 31,7 м² в помещении 501 Стол – 7 шт.
7.11	Стул – 13 шт.
7.12	ПК (с возможностью подключения к сети «Интернет») – 1 шт.
7.13	Магнитно-маркерная доска – 1 шт.
7.14	Телевизор (75”) – 1 шт.
7.15	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 21 площадью 35,6 м² в помещении 501 Стол – 7 шт.
7.16	Стул – 13 шт.
7.17	ПК (с возможностью подключения к сети «Интернет») – 1 шт.
7.18	Магнитно-маркерная доска – 1 шт.
7.19	Телевизор (75”) – 1 шт.
7.20	
7.21	Помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета:
7.22	Адрес (местоположение) помещения для самостоятельной работы обучающихся Перечень компьютерной техники и оборудования
7.23	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 5 площадью 44,3 м² в помещении 501 ПК (ноутбук) – 9 шт.
7.24	Стол – 11 шт.
7.25	Стул – 11 шт.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для успешного освоения дисциплины рекомендуется ознакомиться с рабочей программой, целями и задачами дисциплины, её связями с другими дисциплинами образовательной программы. Также рекомендуется изучить основную и дополнительную литературу, перечень заданий. Перед лекцией просматривать конспект предыдущей лекции для более глубокого восприятия материала. При подготовке к практическим занятиям изучить основную литературу, ознакомиться с дополнительной и методической литературой, учесть рекомендации преподавателя. Важно также опираться на конспекты лекций. В ходе занятий вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных понятий, явлений и процессов, научные выводы и практические рекомендации по их применению. При пропуске занятия самостоятельно изучить материал и предоставить преподавателю отчёт по пропущенной теме в часы индивидуальных консультаций. При подготовке к текущей аттестации тщательно изучать материал по блокам тем, акцентировать внимание на определениях, терминах, содержании понятий. Для этого рекомендуется использовать основную литературу, а также конспекты лекций, составленные в ходе изучения всей дисциплины. При подготовке к промежуточной аттестации в дополнение к изучению конспектов лекций, словарей, учебных пособий, целесообразно пользоваться учебной литературой, рекомендованной в программе, и электронными ресурсами. При подготовке к зачёту или экзамену организовать учёбу так, чтобы все виды работ и заданий, предусмотренные рабочей программой, были выполнены в срок. Основное в подготовке — это повторение всего материала учебной дисциплины.