

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шутов Олег Леонтьевич

Должность: Директор

Дата подписания: 06.06.2026 11:49:43

Уникальный программный ключ:

2ee6ded937fc2877009a3b03e0f0a7f33d8083d5

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ПРОФЕССИОНАЛЬНАЯ
ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ**

**«КУБАНСКИЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ»
(АНПО «КУБАНСКИЙ ИПО»)**

ОТДЕЛЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

РАБОЧАЯ ПРОГРАММА

учебной дисциплины

ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

по специальности

**09.02.11 РАЗРАБОТКА И УПРАВЛЕНИЕ ПРОГРАММНЫМ
ОБЕСПЕЧЕНИЕМ**

направленность программы: Веб-разработка

Краснодар, 2026

СОГЛАСОВАНО

Зам. директора по КОД и МР

_____/ Т.В. Першакова
28.05.2026 г.**УТВЕРЖДАЮ**

Директор АНПОО «Кубанский ИПО»

_____/ О.Л. Шутов
Приказ №38-О от 28.05.2026 г.**ОДОБРЕНО**

Педагогическим советом

Протокол №6 от 28.05.2026 г

РАССМОТРЕНО

на заседании УМО

«Информационные системы и
программирование»

Протокол № 5 от 15.05.2026г.

Председатель _____ / С.А. Пясецкий

Рабочая программа учебной дисциплины ОП.06 Основы алгоритмизации и программирования предназначена для реализации образовательной программы подготовки специалистов среднего звена.

Разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением (Приказ Министерства Просвещения Российской Федерации от 24 февраля 2025 г. № 138, зарегистрированного Министерством Юстиции России 31 марта 2025 г. № 81696) с учетом примерной образовательной программы, разработанной Федеральным учебно-методическим объединением в системе среднего профессионального образования по укрупненным группам профессий, специальностей 09.00.00 Информатика и вычислительная техника, с учетом профессиональных стандартов: «Программист» (Приказ Министерства труда и социальной защиты РФ от 20 июля 2022 г. № 424н, зарегистрирован Министерством юстиции Российской Федерации от 22 августа 2022г. №69720); «Разработчик Web и мультимедийных приложений» (Приказ Министерства труда и социальной защиты РФ от 18 января 2017 г. № 44н, зарегистрирован Министерством юстиции Российской Федерации от 31 января 2017 г. № 45481) и компетенции «Веб- технологии».

Организация-разработчик: АНПОО «Кубанский ИПО»

Разработчик:

Пясецкий С.А., преподаватель АНПОО «Кубанский ИПО»

Рецензенты:

1. Варкентин В.Ф. – преподаватель, АНПОО «Кубанский ИПО»
Квалификация по диплому: преподаватель информатики
2. Маслиев Р.О, генеральный директор ООО «Старт Эксперт»

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	13
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ ..	15

1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

1.1 Место дисциплины в структуре основной образовательной программы:

Учебная дисциплина ОП.06 Основы алгоритмизации и программирования является обязательной частью общепрофессионального цикла образовательной программы в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

Особое значение дисциплина имеет при формировании и развитии общих и профессиональных компетенций:

ОК.01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК.02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК.04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК.09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1. Проектировать модули программного обеспечения;

ПК 2.2. Разрабатывать модули программного обеспечения;

ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения;

1.2 Цель и планируемые результаты освоения дисциплины:

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания.

Код ПК, ОК	Умения	Знания
ПК.2.1., ПК.2.2., ПК.2.3., ОК.01, ОК.02, ОК.04, ОК.09	<ul style="list-style-type: none">– распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части;– определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации;– взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности;– понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы;– разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;– анализировать требования и определять функциональность модуля;– создавать интерфейсы для взаимодействия с другими модулями и системами;– обеспечивать безопасность, производительность и масштабируемость при разработке модулей;– оптимизировать проектируемые модули для повышения их эффективности и качества;– определять структуру и интерфейсы модулей;	<ul style="list-style-type: none">– актуальный профессиональный и социальный контекст, в котором приходится работать и жить;– номенклатура информационных источников, применяемых в профессиональной деятельности;– язык программирования, основные конструкции, синтаксис;– структуры данных;– принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP;– работа с инструментальным программным обеспечением;– многопоточность в программных модулях;– основные принципы проектирования модулей программного обеспечения;– языки программирования и технологии для реализации модулей;– принципы создания интерфейсов для взаимодействия с другими модулями и системами;– принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей;– методы и подходы к интеграции модулей и компонентов;

<ul style="list-style-type: none"> – анализировать требования к модулю и определять его функциональность; – создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля; – проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами; – учитывать требования к масштабируемости, производительности и безопасности при проектировании модуля; – интегрировать модули и компоненты, обеспечивая их взаимодействие; – работать с различными форматами данных и протоколами передачи данных. – <i>разрабатывать простые веб-интерфейсы (HTML и базовый JS) для демонстрации программ;</i> – <i>отправлять HTTP-запросы из программы и обрабатывать ответы в формате JSON;</i> – <i>использовать Git для контроля версий учебных проектов;</i> – <i>отлаживать код с помощью встроенных инструментов среды разработки;</i> – <i>писать асинхронный код для неблокирующих операций (например, загрузка данных);</i> – <i>обрабатывать события мыши, клавиатуры и формы в веб-сценариях;</i> – <i>создавать простые модули с экспортом/импортом (ES6-модули или аналоги).</i> 	<ul style="list-style-type: none"> – принципы безопасности при интеграции модулей и компонентов. – <i>принципы работы HTTP/HTTPS, методы запросов (GET, POST, PUT, DELETE);</i> – <i>форматы передачи данных: JSON, XML, их парсинг и генерация;</i> – <i>базовые принципы REST API;</i> – <i>основы Git: инициализация репозитория, commit, push, pull, ветки;</i> – <i>отладка (debugging) с использованием breakpoints и логирования;</i> – <i>основы асинхронного программирования (колбэки, промисы, async/await);</i> – <i>принципы событийно-ориентированного программирования (браузерные события, обработчики).</i>
--	--

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
ОБЪЕМ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ДИСЦИПЛИНЫ	144
<i>в том числе вариативная часть</i>	<i>72</i>
- теоретическое обучение	54
- практические занятия	72
в т.ч. в форме практической подготовки	72
- промежуточная аттестация	18
в том числе:	
консультации	12
экзамен	6

Тематический план учебной дисциплины

Наименование разделов и тем	Количество аудиторных часов				
	Всего	в т.ч. в форме практической подготовки	самост. работа	теоретич. обучение	практич. занятия
Раздел 1. Введение в программирование	46	26	-	20	26
Тема 1.1 Основы алгоритмизации, языки и системы программирования. Основные элементы языка. Типы данных. Основы структурного программирования	46	26	-	20	26
Раздел 2. Технологии программирования	44	26	-	18	26
Тема 2.1 Модульное программирование	20	6	-	14	6
Итого за семестр 3	66	32	-	34	32
Тема 2.1 Модульное программирование	24	20	-	4	20
Раздел 3. Разработка приложений	36	20	-	16	20
Тема 3.1 Этапы разработки приложений	36	20	-	16	20
Консультации	12	-	-	-	-
Экзамен	6	-	-	-	-
Итого за семестр 4	78	40	-	20	40
ВСЕГО	144	72	-	54	72

2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
СЕМЕСТР 3			
Раздел 1. Введение в программирование		46	
Тема 1.1	Содержание учебного материала	46	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
Основы алгоритмизации, языки и системы программирования.	1 Алгоритмы и их свойства Понятие алгоритма, свойства (массовость, конечность, детерминированность и др.). Способы записи: словесный, графический (блок-схемы), псевдокод, программный. Сложность алгоритмов (O-нотация). <i>Оценка сложности веб-алгоритмов (обработка запросов, поиск по массиву данных).</i>	2	
Основные элементы языка. Типы данных. Основы структурного программирования.	2 Языки программирования и среда разработки Эволюция языков программирования. Классификация: машинные, ассемблеры, высокого уровня. Компиляторы и интерпретаторы. Среда программирования (IDE). Жизненный цикл программы. Программный продукт и его характеристики. <i>Роль JavaScript и TypeScript в веб-разработке. Инструменты веб-разработчика (DevTools, VS Code).</i>	2	
	3 Типы данных и переменные Основные элементы языка. Простые типы данных (целые, вещественные, логические, символьные). Производные типы. Структурированные типы. Объявление переменных. Константы. <i>Ввод/вывод в веб-контексте: prompt, console, работа с формами HTML.</i>	2	
	4 Операторы и ввод/вывод Арифметические, логические, реляционные операторы. Ввод/вывод данных. Обработка исключений. Операторы ветвления (if, if-else, switch).	2	
	5 Циклические конструкции Циклы for, while, do-while. Вложенные циклы. Управление циклом (break, continue). Бесконечные циклы. <i>Обход JSON-массивов, полученных от API. Циклы для рендеринга списков на веб-странице.</i>	2	
	6 Массивы Одномерные массивы: объявление, инициализация, обход. Двумерные массивы: представление в памяти, обход по строкам и столбцам. Базовые алгоритмы обработки.	2	
	7 Строки и коллекции Строковые типы, основные операции. Коллекции: списки (List), словари (Map/Dictionary), множества (Set). Сравнение массивов и коллекций. <i>Парсинг строк из URL-параметров. Использование Map и Set для уникальных идентификаторов в вебе.</i>	2	

	8	Файловая система и работа с файлами Виды файлов (текстовые, бинарные). Доступ к файлам. Режимы открытия. Чтение и запись. Работа с каталогами. <i>Чтение конфигурационных файлов (JSON, YAML) для веб-приложений. Загрузка файлов через веб-интерфейс.</i>	2	
	9	Введение в HTTP и веб-архитектуру <i>Принципы клиент-серверного взаимодействия. Методы HTTP (GET, POST, PUT, DELETE). Коды ответа. Заголовки. Понятие API.</i>	2	
	10	Форматы данных в вебе: JSON и XML <i>Структура JSON (объекты, массивы). Синтаксис XML. Сравнение форматов. Парсинг и генерация. Роль в современных веб-приложениях.</i>	2	
	в том числе, практических занятий		26	
	*ПЗ №1. Создание линейные программ. <i>Реализация ввода данных через веб-форму</i>		2	
	*ПЗ №2. Составление программ разветвляющейся структуры. <i>Ветвление на основе данных из JSON-файла</i>		2	
	*ПЗ №3. Составление циклических программ. <i>Циклы для генерации HTML-списка из массива</i>		2	
	*ПЗ №4. Работа с одномерными массивами. <i>Получение массива из API и его отображение</i>		2	
	*ПЗ №5. Работа с многомерными массивами. <i>Отображение таблицы из двумерного массива на веб-странице</i>		2	
	*ПЗ №6. Обработка массивов. <i>Поиск и фильтрация по данным из JSON</i>		2	
	*ПЗ №7. Сортировка массивов. <i>Сортировка с последующим рендерингом веб-таблицы</i>		2	
	*ПЗ №8. Работа с символами и строками. <i>Обработка строк. Валидация ввода в веб-формах</i>		2	
	*ПЗ №9. Использование коллекций. <i>Использование Map для хранения состояния веб-компонентов</i>		2	
	*ПЗ №10. Работа с текстовыми файлами. <i>Загрузка и парсинг JSON-файла через веб-интерфейс</i>		2	
	*ПЗ №11. Работа с каталогами и файлами (обход, фильтрация)		2	
	*ПЗ №12. <i>Отправка HTTP-запроса (GET) и вывод ответа</i>		2	
	*ПЗ №13. <i>Парсинг JSON-ответа от тестового API</i>		2	
Раздел 2. Технологии программирования			44	
Тема 2.1 Модульное программирование	Содержание учебного материала		20	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
	11	Подпрограммы и функции Понятие подпрограммы. Процедуры и функции. Локальные и глобальные переменные. Модификаторы доступа. Передача параметров: по значению, по ссылке, по имени. <i>Функции-обработчики событий в вебе. Колбэки.</i>	2	
	12	Рекурсия Прямая и косвенная рекурсия. Базовый случай и шаг рекурсии. Примеры: факториал, числа Фибоначчи, обход дерева. Преимущества и недостатки. <i>Рекурсивный обход DOM-дерева. Рекурсивные компоненты в вебе (дерева, меню).</i>	2	
	13	Модульное программирование Разделение кода на модули. Интерфейс модуля. Скрытие реализации. Создание и подключение модулей. Преимущества модульного подхода. <i>ES6-модули (import/export). Сборка модулей веб-приложения.</i>	2	
	14	Введение в ООП: классы и объекты Базовые понятия: объект, свойства, методы, класс, интерфейс. Создание класса. Конструкторы и	2	

		деструкторы. <i>Классы для работы с API-клиентами. Объекты для хранения состояния веб-приложения.</i>		
	15	Принципы ООП: инкапсуляция, наследование, полиморфизм Инкапсуляция: сокрытие данных, геттеры/сеттеры. Наследование: база и производный класс. Полиморфизм: перегрузка методов, виртуальные функции. <i>Веб-компоненты и Shadow DOM как проявление инкапсуляции.</i>	2	
	16	Системы контроля версий: Git <i>Назначение Git. Основные команды: init, add, commit, push, pull, clone. Понятие ветки (branch). Слияние (merge). Разрешение конфликтов.</i>	2	
	17	Отладка и профилирование <i>Инструменты отладки. Точки останова (breakpoints). Пошаговое выполнение. Просмотр переменных. Логирование (уровни логирования). Профилирование производительности.</i>	2	
		в том числе, практических занятий	6	
		*ПЗ №14. Использование подпрограмм (функции без параметров и с параметрами). <i>Создание функции-обработчика для кнопки</i>	2	
		*ПЗ №15. Использование подпрограмм (возврат значений, переменные). <i>Передача данных между функциями в веб-приложении</i>	2	
		*ПЗ №16. Работа с рекурсивными функциями. <i>Рекурсивный рендер дерева комментариев</i>	2	
Итого за семестр			66	
СЕМЕСТР 4				
Тема 2.1 Модульное программирование	Содержание учебного материала		24	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
	18	Асинхронное программирование <i>Синхронный и асинхронный подход. Колбэки (callbacks). Промисы (Promises). Async/await. Событийный цикл (event loop). Неблокирующие операции.</i>	2	
	19	Событийно-ориентированное программирование <i>Понятие события. Обработчики событий. Регистрация и удаление обработчиков. Всплытие и погружение событий. Примеры в веб-разработке.</i>	2	
		в том числе, практических занятий	20	
		*ПЗ №17. Создание и подключение модулей. <i>Разделение кода на ES6-модули</i>	2	
		*ПЗ №18. Работа с классами. Создание конструкторов	2	
		*ПЗ №19. Применение свойств (геттеры/сеттеры). <i>Валидация данных через сеттеры</i>	2	
		*ПЗ №20. Работа с наследованием классов	2	
		*ПЗ №21. Работа с переопределением методов (полиморфизм). <i>Переопределение метода render для разных компонентов</i>	2	
		*ПЗ №22. <i>Работа с Git: инициализация, коммиты, ветки</i>	2	
		*ПЗ №23. <i>Разрешение конфликтов при слиянии в Git</i>	2	
		*ПЗ №24. <i>Выполнение отладки: поиск и исправление ошибок</i>	2	
		*ПЗ №25. <i>Асинхронная загрузка данных (асинхронные запросы к API)</i>	2	
	*ПЗ №26. <i>Создание обработчиков событий (клики, ввод текста)</i>	2		
Раздел 3. Разработка приложений			36	

Тема 3.1 Этапы разработки приложений	Содержание учебного материала		36	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
	20	Визуально-событийное программирование Разработка приложений с графическим интерфейсом. Событийная модель. Компоненты интерфейса. Проектирование форм. <i>Браузерная событийная модель, DOM-элементы</i>	2	
	21	Жизненный цикл разработки приложения Этапы: анализ требований, проектирование, реализация, тестирование, отладка, оптимизация, сопровождение. Документирование. <i>Жизненный цикл веб-приложения (MVP, итерации, CI/CD).</i>	2	
	22	Проектирование объектно-ориентированного приложения Диаграммы классов. Диаграммы последовательностей. Выделение сущностей. Проектирование интерфейсов взаимодействия модулей. <i>Проектирование веб-приложения с компонентами.</i>	2	
	23	Тестирование и отладка приложений Виды тестирования: модульное, интеграционное, системное. Тест-кейсы. Отладка. Оптимизация программы по скорости и памяти. <i>Unit-тесты для веб-модулей (Jest). Интеграционное тестирование API.</i>	2	
	24	REST API: принципы проектирования Ресурсы и их идентификация. Методы HTTP для CRUD операций. Статус-коды. Версионирование API. Пагинация. Фильтрация.	2	
	25	Принципы масштабируемости и безопасности Горизонтальное и вертикальное масштабирование. Кэширование. Аутентификация и авторизация (JWT, OAuth). Валидация входных данных. Защита от XSS, CSRF, SQL-инъекций.	2	
	26	Интеграция модулей и компонентов Методы интеграции. API-иллюзы. Очереди сообщений. Форматы обмена данными. Протоколы (REST, SOAP, GraphQL).	2	
	27	Многопоточное программирование Потоки (threads). Состояния потоков. Синхронизация (мьютексы, семафоры). Проблемы: гонки данных, дедлоки. Многопоточность в веб-серверах.	2	
	в том числе, практических занятий		20	
	*ПЗ №27. Создание проекта с использованием компонентов для работы с текстом. <i>Создание формы ввода и отображение текста на веб-странице</i>		2	
	*ПЗ №28. Создание проекта с использованием кнопочных компонентов		2	
	*ПЗ №29. Создание проекта с использованием переключателей (radio, checkbox)		2	
	*ПЗ №30. Создание проекта с использованием компонентов для отображения таблиц. <i>Отображение данных из массива в HTML-таблице</i>		2	
	*ПЗ №31. Создание проекта с использованием компонентов для отображения дат и времени		2	
*ПЗ №32. Разработка интерфейса приложения. <i>Создание макета веб-приложения</i>		2		
*ПЗ №33. Разработка интерфейса приложения (привязка событий и логики)		2		
*ПЗ №34. Тестирование приложения (написание и выполнение тест-кейсов). <i>Написание и запуск unit-тестов (Jest)</i>		2		
*ПЗ №35. Создание клиента для REST API (CRUD операции)		2		

	*ПЗ №36. Создание проекта с интеграцией всех изученных тем	2	
Консультации к экзамену	Содержание	12	
	1. Алгоритмизация и базовые конструкции Свойства алгоритмов, способы записи, оценка сложности (O-нотация). Визуализация алгоритмов с помощью блок-схем и псевдокода. Реализация линейных, разветвляющихся и циклических алгоритмов на языке программирования. Типичные ошибки при работе с циклами и условиями (бесконечные циклы, неучтённые ветви switch). <i>Оценка сложности алгоритма обработки данных, полученных от веб-API.</i>	2	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
	2. Структуры данных: массивы, строки, коллекции, файлы Одномерные и двумерные массивы: обход, поиск, вставка, удаление, сортировка. Строки: основные операции (конкатенация, поиск подстроки, замена, разбиение). Коллекции: List, Dictionary (Map), Set — когда что выбирать. Работа с текстовыми и бинарными файлами: чтение, запись, обработка ошибок. <i>Парсинг JSON-файлов и строк, полученных от сервера; преобразование JSON в массивы/коллекции и обратно.</i>	2	
	3. Модульное программирование, рекурсия, отладка и Git Подпрограммы (функции/процедуры): параметры, возвращаемые значения, область видимости. Рекурсия: базовый случай, шаг рекурсии, стек вызовов. Примеры (факториал, Фибоначчи, обход дерева). Модульное программирование: разделение кода, экспорт/импорт модулей. Отладка: точки останова, пошаговое выполнение, логирование. Git: инициализация, коммиты, ветки, слияние, разрешение конфликтов. <i>Отладка асинхронного кода; использование Git для сохранения разных версий экзаменационного проекта.</i>	2	
	4. Объектно-ориентированное программирование: классы, объекты, принципы ООП Классы и объекты: поля, методы, конструкторы. Инкапсуляция: модификаторы доступа, геттеры/сеттеры. Наследование: базовый и производный класс, переопределение методов. Полиморфизм: перегрузка методов, виртуальные/абстрактные методы. <i>Проектирование классов для веб-приложения (например, классы User, Product, Cart); использование классов для работы с API-клиентом.</i>	2	
	5. Веб-взаимодействие: HTTP, REST API, асинхронность, события Основы HTTP: методы (GET, POST, PUT, DELETE), коды ответа, заголовки. Форматы данных: JSON (парсинг, генерация), сравнение с XML. REST API: ресурсы, CRUD, версионирование, пагинация. Асинхронное программирование: колбэки, промисы, async/await. Событийно-ориентированное программирование: регистрация обработчиков, всплытие событий. <i>Отправка запросов к реальному API (например, GitHub API или публичное тестовое API), обработка ответов и отображение на веб-странице.</i>	2	
6. Проектирование, тестирование, безопасность и интеграция веб-приложений Этапы разработки приложений: анализ, проектирование, реализация, тестирование, оптимизация. Диаграммы классов и последовательностей (UML) для визуализации модулей. Тестирование: модульное (unit), интеграционное, системное. Написание тест-кейсов. Безопасность: XSS, CSRF, SQL-инъекции, валидация данных, JWT. Масштабируемость и производительность: кэширование, многопоточность, Web Workers. Интеграция модулей: API-шлюзы, очереди, форматы обмена.	2		
Экзамен		6	
Итого за семестр		78	
ВСЕГО:		144	

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Для реализации программы учебной дисциплины предусмотрены следующие специальные помещения:

Лаборатория «Алгоритмизации и программирования»

оснащена оборудованием:

– рабочее место педагога (преподавательский стол (1 шт.), стул (1 шт.));
– рабочие места обучающихся (парты ученические (13 шт.), стулья ученические (25 шт.));

– доска учебная (меловая трех-секционная) (1 шт.);

– книжный шкаф – 1 шт.;

– учебно-методическая литература по дисциплине;

– комплект учебно-наглядных пособий;

техническими средствами обучения:

– персональный компьютер, подключение к сети Интернет с модулем контентной фильтрации Traffic Inspector, NetPolice и YandexDNS, возможность трансляции на экран аудио и видео информации (1 шт.)

– программное обеспечение на ПК – Microsoft Windows 10 или аналог, Microsoft Office (Word, Excel, PowerPoint) или аналог, 7Zip, 24PDF, Яндекс Браузер (1 шт.)

– программное обеспечение Android Studio, Visual Studio, Visual Studio Code или аналоги

– монитор (1 шт.)

– клавиатура (1 шт.)

– мышь (1 шт.)

– телевизор (1 шт.)

– кабель для подключения HDMI (1 шт.)

3.2 Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд института имеет печатные и электронные образовательные и информационные ресурсы, в том числе рекомендованные ФУМО, для использования в образовательном процессе. Список дополнен новыми изданиями.

3.2.1 Основные источники

1. Веб-разработка : учебник для среднего профессионального образования / под общей редакцией О. В. Ратановой, Н. А. Ребус, А. Ю. Анисимова. — Москва : Издательство Юрайт, 2026. — 217 с. — (Профессиональное образование). — ISBN 978-5-534-21227-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590627>

2. Полуэктова, Н. Р. Разработка веб-приложений : учебник для среднего профессионального образования / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-18644-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588469>

3. Семакин И.Г. Основы алгоритмизации и программирования. Практикум: учебное издание / Семакин И.Г., Шестаков А. П. - Москва : Академия, 2023. - 144 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow»

4. Семакин И.Г. Основы алгоритмизации и программирования: учебное издание / Семакин И.Г., Шестаков А. П. - Москва : Академия, 2024. - 304 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow».

5. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Профессиональное образование). — ISBN 978-5-534-20429-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563861>

6. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 196 с. — (Профессиональное образование). — ISBN 978-5-534-18760-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584552>

7. Якимов, С. П. Алгоритмизация и программирование : учебник для среднего профессионального образования / С. П. Якимов. — Москва : Издательство Юрайт, 2026. — 342 с. — (Профессиональное образование). — ISBN 978-5-534-19661-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589849>

3.2.2 Дополнительные источники

8. Абдрахманов, М. И. Основы языка программирования Python : учебное пособие для СПО / М. И. Абдрахманов. — Москва : Ай Пи Ар Медиа, 2023. — 142 с. — ISBN 978-5-4497-2310-9. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROОбразование : [сайт]. — URL: <https://profspo.ru/books/132567>

9. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1927269>

10. Гуриков, С. Р. Основы алгоритмизации и программирования на Visual C++ : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 515 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1039154. - ISBN 978-5-16-015500-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1039154>

11. Дорогов, В. Г. Основы программирования на языке C : учебное пособие / В.Г. Дорогов, Е.Г. Дорогова ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 224 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0809-9. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2010597>

12. Дорохова, Т. Ю. Основы алгоритмизации и программирования : учебное пособие для СПО / Т. Ю. Дорохова, И. Е. Ильина. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2022. — 139 с. — ISBN 978-5-4488-1531-7, 978-5-4497-1718-4. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROОбразование : [сайт]. — URL: <https://profspo.ru/books/122426>

13. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1735805>

14. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебник для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2026. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/587050>

15. Федоров, Д. Ю. Программирование на Python : учебник для среднего профессионального образования / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 166 с. — (Профессиональное образование). — ISBN 978-5-534-22180-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/600873>

16. Черпаков, И. В. Алгоритмизация и программирование в Python : учебник для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2026. — 159 с. — (Профессиональное образование). — ISBN 978-5-534-21911-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/582413>

3.2.3. Электронные источники:

17. Сайт по программированию <https://metanit.com>

4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Методы оценки
<p>Уметь:</p> <ul style="list-style-type: none"> – распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части; – определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации; – взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности; – понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; – разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий; – анализировать требования и определять функциональность модуля; – создавать интерфейсы для взаимодействия с другими модулями и системами; – обеспечивать безопасность, производительность и масштабируемость при разработке модулей; – оптимизировать проектируемые модули для повышения их эффективности и качества; – определять структуру и интерфейсы модулей; – анализировать требования к модулю и определять его функциональность; – создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля; – проектировать интерфейсы программного обеспечения для взаимодействия с другими 	<p>Оценка «5»:</p> <p>глубокий анализ, выделяет скрытые составляющие, строит причинно-следственные связи, формулирует проблему точно и полно.</p> <p>составляет детальный план с приоритизацией, критически оценивает источники, адаптирует стратегию поиска под контекст.</p> <p>инициирует продуктивное общение, управляет коммуникацией, разрешает разногласия, адаптирует стиль общения под аудиторию.</p> <p>понимает детали, нюансы и подтекст; может выделить имплицитную информацию; делает обоснованные выводы.</p> <p>уверенно применяет разные языки и технологии под задачу, модули хорошо структурированы, сопровождаемы, расширяемы.</p> <p>проводит системный анализ, выявляет противоречия и пробелы в требованиях, предлагает оптимальную функциональность. создаёт гибкий, документированный интерфейс с учётом версионирования, обратной совместимости, обработки ошибок и валидации.</p> <p>комплексно проектирует с балансом трёх аспектов, применяет профилирование, защитные механизмы, архитектурные паттерны масштабирования.</p> <p>выполняет комплексную оптимизацию (алгоритмическую, по памяти, по I/O), рефакторинг с повышением качества, измеряет результат.</p> <p>чётко спроектированная иерархическая/сетевая структура, интерфейсы минимальны, полны, удобны и документированы.</p> <p>комплексный набор диаграмм, отражающий структуру и поведение; диаграммы наглядны, непротиворечивы, соответствуют реализации.</p> <p>проектирует контракты с учётом версионирования, семантического версионирования, совместимости, безопасности и производительности.</p> <p>проектирует с полным учётом всех трёх групп требований, выбирает архитектурные паттерны под каждый аспект.</p> <p>выполняет сложную интеграцию множества компонентов с разными протоколами/форматами, обеспечивает надёжность, логирование, восстановление после сбоев.</p> <p>работает с 3+ форматами и 2+ протоколами</p>	<p>Текущий контроль:</p> <p>Оценка результатов выполнения практической работы</p> <p>Экспертное наблюдение за ходом выполнения практической работы (деятельностью студента)</p> <p>Оценка выполнения практического задания (работы)</p> <p>Промежуточная аттестация: экзамен</p>

<p>модулями и системами;</p> <ul style="list-style-type: none"> – учитывать требования к масштабируемости, производительности и безопасности при проектировании модуля; – интегрировать модули и компоненты, обеспечивая их взаимодействие; – работать с различными форматами данных и протоколами передачи данных. – <i>разрабатывать простые веб-интерфейсы (HTML и базовый JS) для демонстрации программ;</i> – <i>отправлять HTTP-запросы из программы и обрабатывать ответы в формате JSON;</i> – <i>использовать Git для контроля версий учебных проектов;</i> – <i>отлаживать код с помощью встроенных инструментов среды разработки;</i> – <i>писать асинхронный код для неблокирующих операций (например, загрузка данных);</i> – <i>обрабатывать события мыши, клавиатуры и формы в веб-сценариях;</i> <i>создавать простые модули с экспортом/импортом (ES6-модули или аналоги).</i> 	<p>(HTTP, WebSocket, файловый), выполняет сериализацию/десериализацию, конвертацию между форматами.</p> <p>создаёт аккуратный, адаптивный интерфейс, интуитивно понятный, с динамическим обновлением данных.</p> <p><i>работает с заголовками, параметрами, телом запроса; обрабатывает все типы ошибок; преобразует JSON в объекты/модели.</i></p> <p><i>применяет rebase, cherry-pick, stash; разрешает сложные конфликты; ведёт понятную историю коммитов.</i></p> <p><i>применяет условные точки останова, логирование, watch-выражения; находит и исправляет сложные логические ошибки.</i></p> <p><i>уверенно использует async/await, обрабатывает ошибки, понимает и объясняет событийный цикл.</i></p> <p><i>использует делегирование событий, предотвращает дефолтное поведение, корректно удаляет обработчики, оптимизирует частые события (debounce/throttle).</i></p> <p><i>проектирует модульную архитектуру без циклических зависимостей, минимизирует глобальное состояние, реэкспортирует при необходимости.</i></p> <p>Оценка «4»:</p> <p>структурирует задачу, выделяет ключевые и второстепенные части, видит простые взаимосвязи.</p> <p>разбивает поиск на этапы, обосновывает выбор источников, использует разные типы источников.</p> <p>активно взаимодействует, аргументирует свою позицию, учитывает мнение других, задаёт уточняющие вопросы.</p> <p>понимает ключевую информацию, может пересказать основное содержание своими словами.</p> <p>использует 2+ языка/технологии, модули имеют чёткую структуру, корректно работают.</p> <p>выявляет неявные требования, определяет полный набор функций, группирует их по смыслу.</p> <p>проектирует стабильный интерфейс с несколькими точками входа, обрабатывает типовые ошибки.</p> <p>учитывает все три аспекта: применяет валидацию, избегает очевидных проблем с производительностью, закладывает возможность масштабирования.</p> <p>проводит локальную оптимизацию алгоритмов и структур данных, улучшает читаемость кода.</p> <p>структура модульная с разделением ответственности, интерфейс покрывает</p>	
---	--	--

	<p>основные сценарии использования. диаграмма классов и одна динамическая диаграмма (последовательности/активности) корректны, нотация соблюдена. проектирует интерфейс с методами, форматами данных, обработкой ошибок, документацией. принимает проектные решения, учитывающие хотя бы два аспекта (например, горизонтальное масштабирование и кэширование). интегрирует несколько модулей через согласованные интерфейсы, обрабатывает базовые ошибки взаимодействия. уверенно работает с 2–3 форматами (JSON, XML, текстовые), использует один протокол (HTTP).</p> <p><i>создаёт функциональную страницу с элементами форм, базовой стилизацией, обработкой действий пользователя. отправляет разные типы запросов (GET, POST), парсит JSON, обрабатывает основные ошибки сети. использует ветки, делает push/pull, разрешает простые конфликты. использует пошаговое выполнение, просмотр и изменение переменных, стек вызовов. применяет промисы с цепочками .then/.catch. обрабатывает несколько типов событий, получает данные события (координаты, клавишу, значение поля). использует именованные и дефолтные экспорты/импорты, разделяет код на логические модули.</i></p> <p>Оценка «3»: выделяет наличие задачи/проблемы, называет 1–2 очевидных составляющих. формулирует запрос, называет 1–2 источника. участвует в общении, отвечает на вопросы, следует заданной роли. улавливает основную тему, понимает отдельные знакомые слова и фразы. пишет простые модули на одном языке, код работает для базовых случаев. выделяет явно указанные требования, называет основные функции. реализует простейший интерфейс (например, одну функцию или endpoint). учитывает хотя бы один аспект (безопасность ИЛИ производительность) на базовом уровне. исправляет очевидные неэффективности (например, повторяющиеся вычисления). предлагает простую линейную структуру, интерфейс — 1–2 функции/метода. создаёт простую диаграмму классов с 2–3 сущностями, есть ошибки нотации.</p>	
--	---	--

	<p>определяет название и базовые параметры интерфейса. упоминает требования, но не закладывает их явно в архитектуру. соединяет два модуля простейшим способом (прямой вызов). читает/пишет один формат (например, JSON или CSV). <i>создаёт статическую HTML-страницу с минимальным JS (например, вывод в консоль).</i> <i>отправляет GET-запрос, выводит сырой ответ.</i> <i>выполняет init, add, commit.</i> <i>ставит точки останова (breakpoints).</i> <i>использует колбэки.</i> <i>вешает обработчик на одно событие (например, onclick).</i> <i>создаёт файл с экспортом одной функции и импортирует её.</i></p>	
<p>Знать: – актуальный профессиональный и социальный контекст, в котором приходится работать и жить; – номенклатура информационных источников, применяемых в профессиональной деятельности; – язык программирования, основные конструкции, синтаксис; – структуры данных; – принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP; – работа с инструментальным программным обеспечением; – многопоточность в программных модулях; – основные принципы проектирования модулей программного обеспечения; – языки программирования и технологии для реализации модулей; – принципы создания интерфейсов для взаимодействия с другими модулями и системами; – принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей; – методы и подходы к</p>	<p>Оценка «5»: ориентируется в текущей ситуации отрасли (востребованные технологии, зарплаты, карьерные траектории, этические и правовые аспекты). системно классифицирует источники, знает авторитетные ресурсы для разных задач (MDN, GitHub, Habr, Tproger, курсы), умеет оценивать достоверность. глубокое знание синтаксиса и семантики, включая современные возможности (стрелочные функции, деструктуризация, spread/rest, шаблонные строки и др.). знает очередь, стек, связные списки, деревья, хеш-таблицы; понимает их сложность и выбор под задачу. сравнивает REST и SOAP, знает альтернативы (GraphQL, gRPC), понимает версионирование, документацию (Swagger/OpenAPI). ориентируется в спектре инструментов (сборщики, линтеры, тестовые фреймворки, профилировщики, CI/CD). знает модели многопоточности в разных языках, понимает асинхронность как альтернативу, знает Web Workers. объясняет каждый принцип SOLID с примерами, знает принципы связности и сцепления, паттерны проектирования. ориентируется в экосистеме языков, знает сильные и слабые стороны каждого для разных задач, следит за новыми технологиями. объясняет OWASP Top 10, профилирование производительности, паттерны масштабирования (репликация, шардирование, микросервисы). знает подходы: точка-точка, шина данных, сервисная сетка (service mesh), очереди сообщений (RabbitMQ, Kafka).</p>	<p>Текущий контроль: Компьютерное тестирование по основным разделам программы. Решение задач по основным разделам программы Промежуточная аттестация: экзамен</p>

<p>интеграции модулей и компонентов;</p> <ul style="list-style-type: none"> – принципы безопасности при интеграции модулей и компонентов. – принципы работы HTTP/HTTPS, методы запросов (GET, POST, PUT, DELETE); – форматы передачи данных: JSON, XML, их парсинг и генерация; – базовые принципы REST API; – основы Git: инициализация репозитория, commit, push, pull, ветки; – отладка (debugging) с использованием breakpoints и логирования; – основы асинхронного программирования (колбэки, промисы, async/await); – принципы событийно-ориентированного программирования (браузерные события, обработчики). 	<p>объясняет безопасность межмодульного взаимодействия.</p> <p>понимает заголовки, cookies, сессии, работу с кешем, идемпотентность методов.</p> <p>сравнивает форматы (скорость, читаемость, поддерживаемость), знает инструменты валидации и преобразования.</p> <p>знает принципы HATEOAS, версионирование, пагинацию, фильтрацию, идемпотентность.</p> <p>понимает внутреннее устройство Git (дерево коммитов, HEAD, merge vs rebase, разрешение конфликтов).</p> <p>знает условные точки останова, watch-выражения, возможности отладки асинхронного кода, профилирование.</p> <p>понимает событийный цикл, микро- и макрозадачи, обработку ошибок в асинхронном коде, преимущества и недостатки каждого подхода.</p> <p>понимает делегирование событий, отмену всплытия, предотвращение дефолтного поведения, создание пользовательских событий, EventEmitter.</p> <p>Оценка «4»:</p> <p>понимает современные тренды разработки, требования рынка, особенности удалённой/офисной работы.</p> <p>называет основные типы источников (официальная документация, профильные форумы, блоги, видеоуроки, книги).</p> <p>уверенно называет все ключевые конструкции (функции, массивы, объекты, обработку ошибок, модули).</p> <p>знает список, словарь, множество, понимает различия и области применения.</p> <p>называет основные инструменты (редактор кода, отладчик, система контроля версий, терминал).</p> <p>объясняет проблемы многопоточности (гонки данных, дедлоки), базовые механизмы синхронизации.</p> <p>называет SOLID, DRY, KISS.</p> <p>называет 2–3 языка и их типичные применения (бэкенд, фронтенд, скрипты).</p> <p>называет конкретные меры: валидация, кэширование, горизонтальное масштабирование.</p> <p>называет событийно-ориентированную интеграцию, API-шлюзы.</p> <p>называет аутентификацию, авторизацию, валидацию входящих данных, защиту от подделки запросов.</p> <p>объясняет методы запросов и их назначение, знает основные коды ответа.</p> <p>объясняет структуру обоих форматов, знает базовые функции парсинга.</p> <p>объясняет ресурсы, методы, stateless, коды ответа.</p>	
--	--	--

	<p><i>объясняет init, clone, push, pull, создание веток.</i></p> <p><i>объясняет пошаговое выполнение, просмотр переменных, стека вызовов, уровни логирования.</i></p> <p><i>объясняет промисы (resolve, reject, then, catch), знает async/await.</i></p> <p><i>объясняет регистрацию обработчиков, объект события, всплытие и погружение.</i></p> <p>Оценка «3»:</p> <p>знает общие черты профессии (разработка ПО, работа в команде).</p> <p>знает 1–2 источника (например, Stack Overflow, документацию).</p> <p>знает основные операторы, типы данных, базовый синтаксис (условия, циклы).</p> <p>знает массивы и строки.</p> <p>знает, что такое API, называет REST.</p> <p>знает среду разработки (IDE) и базовые возможности.</p> <p>знает понятие потока/процесса.</p> <p>знает принцип разделения на модули.</p> <p>знает один язык (например, Python).</p> <p>знает термины «безопасность», «производительность», «масштабируемость».</p> <p>знает прямой вызов одного модуля другим.</p> <p>знает про аутентификацию.</p> <p><i>знает разницу между HTTP и HTTPS.</i></p> <p><i>знает, что JSON похож на объекты JS, XML — на теги.</i></p> <p><i>знает, что REST работает через HTTP.</i></p> <p><i>знает команды add, commit.</i></p> <p><i>знает, что такое точка останова и console.log.</i></p> <p><i>знает понятие колбэка.</i></p> <p><i>знает, что можно повесить обработчик на клик.</i></p>	
--	---	--