Приложение
к ППССЗ специальности 09.02.11 Разработка и управление программным обеспечением

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
РЕСПУБЛИКИ ДАГЕСТАН «ТЕХНИЧЕСКИЙ КОЛЛЕДЖ ИМЕНИ Р.Н. АШУРАЛИЕВА»

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.06 Основы алгоритмизации и программирования

Специальность: 09.02.11 Разработка и управление программным обеспечением

Квалификация выпускника: программист

Махачкала, 2026 г.

ОДОБРЕНО

предметной (цикловой) комиссией УГС 09.00.00. Информатика и вычислительная техника и 10.00.00 Информационная безопасность

Председатель П(Ц)К

 Ш.М. Мусаева

Протокол № 8 от 30 апреля 2026 г.

Рабочая программа учебной дисциплины ОП.06 Основы алгоритмизации и программирования разработана на основе:

- Приказа Минпросвещения России от 24.02.2025 N 138 "Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением" (Зарегистрировано в Минюсте России 31.03.2025 N 81696);

с учетом:

- Примерной образовательной программы «Профессионалитет» по специальности 09.02.11 Разработка и управление программным обеспечением, утвержденная протоколом ФУМО в системе СПО по УГПС 09.00.00 Информатика и вычислительная техника от 29.08.2025 № 6/2025, зарегистрирована в государственном реестре примерных образовательных программ – регистрационный номер 18/2025. Реквизиты утверждающего документа –приказ ФГБОУ ДПО ИРПО № 01-09-581/2025 от 13.10.2025;

в соответствии с рабочим учебным планом.

Разработчик:

- Алиев Эльдар Русланович, преподаватель ГБПОУ РД «Технический колледж имени Р.Н. Ашуралиева»

© Алиев Эльдар Русланович, 2026

© ГБПОУ РД «Технический колледж Р.Н. Ашуралиева», 2026

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.06. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» **.....4**

1.1. Цель и место дисциплины в структуре программы подготовки специалистов среднего звена: 4

1.2. Планируемые результаты освоения дисциплины:.....4

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**.....6**

2.1. Объем учебной дисциплины и виды учебной работы 6

2.2. Тематический план и содержание учебной дисциплины «ОП.06. Основы алгоритмизации и программирования» 7

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.06. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**.....11**

3.1. Материально-техническое обеспечение реализации программы 11

3.2. Учебно-методическое обеспечение 11

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.06. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» **.....11**

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

1.1. Цель и место дисциплины в структуре программы подготовки специалистов среднего звена:

Учебная дисциплина «ОП.06 Основы алгоритмизации и программирования» принадлежит общепрофессиональному циклу ОП.00 обязательной части ФГОС специальности 09.02.11 Разработка и управление программным обеспечением.

Цель учебной дисциплины «ОП.06 Основы алгоритмизации и программирования»: формирование навыков работы в среде программирования, разработки алгоритмов для решения простых задач, реализации готовых и разработанных алгоритмов на языке Python.

1.2. Планируемые результаты освоения дисциплины:

Освоение дисциплины должно способствовать формированию общих 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 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Освоение дисциплины должно способствовать формированию профессиональных компетенций:

- ПК 2.2. Разрабатывать модули программного обеспечения с использованием языков программирования и технологий.
- ПК 2.4. Выполнять отладку и тестирование программного обеспечения.

В результате освоения дисциплины обучающийся должен уметь:

- Разрабатывать простые алгоритмы для учебных и профессионально ориентированных задач.
- Работать в среде программирования Python, запускать и отлаживать простые программы.
- Использовать ввод и вывод данных, переменные, арифметические операции и базовые типы данных.
- Применять условные операторы, циклы, строки, списки и функции при решении простых задач.
- Оформлять код программы в соответствии с базовыми требованиями PEP 8.
- Выполнять проверку, поиск и исправление ошибок в простом программном коде.
- Создавать простые консольные мини-проекты на языке Python.
- Создавать простые классы и объекты, определять их свойства и методы.
- Выполнять чтение данных из текстового файла и запись результатов работы программы в файл.

В результате освоения дисциплины обучающийся должен знать:

- Основные понятия алгоритмизации, свойства алгоритмов и этапы решения задач на компьютере.
- Назначение языка Python, среды программирования, компиляторов и

интерпретаторов.

- Синтаксис базовых конструкций Python: ввод-вывод, переменные, типы данных, условия и циклы.
- Основные приемы работы со строками, списками и функциями.
- Простые способы проверки программы и поиска типичных ошибок начинающего программиста.
- Базовые понятия объектно-ориентированного программирования: класс, объект, свойства и методы.
- Основы работы с текстовыми файлами в Python.

Код ПК, ОК	Умения	Знания
ОК 01 ОК 02 ОК 09 ПК 2.2 ПК 2.4	– Разрабатывать простые алгоритмы для учебных и профессионально ориентированных задач. – Представлять алгоритмы в словесной форме, в виде блок-схем и программ на языке Python. – Работать в среде программирования Python: создавать, сохранять, запускать и отлаживать простые программы. – Использовать ввод и вывод данных, переменные, арифметические операции и базовые типы данных. – Применять условные операторы, логические операции, циклы for и while при решении простых задач. – Обрабатывать строки и списки, использовать основные методы строк и списков. – Разрабатывать простые функции с параметрами и возвращаемым значением. – Создавать простые классы и объекты, определять их свойства и методы. – Выполнять чтение данных из текстового файла и запись результатов работы программы в файл. – Оформлять программный код в соответствии с базовыми требованиями читаемости и стилем PEP 8. – Выполнять проверку, поиск и исправление синтаксических и логических ошибок в программном коде.	– Основные понятия алгоритмизации, свойства алгоритмов, основные алгоритмические конструкции и этапы решения задач на компьютере. – Назначение языков программирования, понятие среды программирования, компилятора и интерпретатора. – Назначение и основные возможности языка программирования Python. – Синтаксис базовых конструкций Python: ввод и вывод данных, переменные, типы данных и арифметические операции. – Синтаксис условных операторов, логических операций, циклов for и while. – Основные приёмы работы со строками, списками и функциями в Python. – Назначение модулей и примеры использования стандартных модулей Python. – Базовые понятия объектно-ориентированного программирования: класс, объект, свойства и методы. – Основы работы с текстовыми файлами в Python. – Основные правила оформления и повышения читаемости программного кода в соответствии со стилем PEP 8. – Типичные синтаксические и логические ошибки в программах и простые способы их поиска и исправления. – Основные этапы разработки, проверки и тестирования простых программ и мини-проектов.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	**Объем в часах**
Объем образовательной программы	**188**
в том числе:	
Теоретическое обучение	68
Лабораторные занятия	70
Консультация	2
Самостоятельная работа	42
Промежуточная аттестация в форме экзамена	6

2.2. Тематический план и содержание учебной дисциплины «ОП.06. Основы алгоритмизации и программирования»

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
Раздел 1. Введение в программирование			**12**	ОК 01 ОК 02 ОК 09 ПК 2.2 ПК 2.4
Тема 1.1. Языки программирования	**Содержание учебного материала**		8	
	1.	Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования.		
	2.	Среда проектирования. Понятие интегрированной среды программирования. Состав языка программирования. Синтаксис и семантика. Компиляторы и интерпретаторы.		
	3.	Жизненный цикл программы. Программа. Программный продукт и его характеристики.		
	4.	Основные этапы решения задач на компьютере. Алгоритмы и блок-схемы. Понятие сложности алгоритма. Оценка сложности простых алгоритмов.		
	Лабораторные занятия		4	
	5.	Знакомство со средой Python. Создание, запуск и сохранение простой программы.		
	6.	Изучение основных элементов среды разработки: окно редактора, консоль вывода, файлы программы, запуск и просмотр результатов выполнения.		
Раздел 2. Введение в язык Python			**20**	ОК 01 ОК 02 ОК 09 ПК 2.2 ПК 2.4
Тема 2.1. Ввод-вывод данных	**Содержание учебного материала**		8	
	7.	Введение. Знакомство с Python.		
	8.	Команды print() и input(), переменные.		
	9.	Параметры sep и end, правила оформления кода по PEP 8.		
	10.	Целочисленная арифметика.		
	Лабораторные занятия		12	
	11.	Создание программ с выводом текста и вводом данных пользователем.		
	12.	Решение простых задач на переменные и целочисленную арифметику.		
	13.	Оформление простых программ по PEP 8.		
	14.	Создание программ-диалогов с пользователем: вывод сообщений, ввод с использованием print() и input().		
	15.	Использование параметров sep и end при выводе данных. Формирование строк вывода по образцу.		
	16.	Множественное присваивание		
Раздел 3. Основы программирования на языке Python			**96**	ОК 01

Тема 3.1. Условный оператор	**Содержание учебного материала**		6	ОК 02 ОК 09 ПК 2.2 ПК 2.4
	17.	Выбор из двух: условный оператор if.		
	18.	Логические операции и цепочки сравнений.		
	19.	Вложенные и каскадные условия.		
	Лабораторные занятия		8	
	20.	Решение практических задач с несколькими вариантами выбора		
	21.	Составление программ с каскадными условиями if – elif – else		
	22.	Решение задач с логическими операциями и несколькими условиями.		
	23.	Использование вложенных условных операторов для реализации многоуровневой логики программы.		
Тема 3.2. Типы данных	**Содержание учебного материала**		6	
	24.	Числовые типы данных: int, float.		
	25.	Строковый тип данных str.		
	26.	Использование модуля math для простых вычислений.		
	Лабораторные занятия		6	
	27.	Решение задач на преобразование типов и простые вычисления.		
	28.	Использование базовых функций модуля math в учебных задачах.		
	29.	Оператор in		
Тема 3.3. Циклы for и while	**Содержание учебного материала**		8	
	30.	Цикл for.		
	31.	Цикл for: функция range().		
	32.	Цикл while.		
	33.	Операторы break, continue и else.		
	Лабораторные занятия		10	
	34.	Решение задач с циклами for и while.		
	35.	Обработка чисел по цифрам.		
	36.	Исправление ошибок в простых программах с циклами.		
	37.	Решения задач с прерывания цикла break, continue		
	38.	Блок else в циклах		
Тема 3.4. Строковый тип данных	**Содержание учебного материала**		10	
	39.	Индексация строк.		
	40.	Срезы.		
	41.	Основные методы строк.		
	42.	Форматирование строк.		
	43.	Сравнение строк.		
	Лабораторные занятия		6	
	44.	Решение задач на индексацию и срезы строк.		
	45.	Использование методов строк для обработки пользовательского ввода.		
	46.	Форматирование вывода результатов программы.		
Тема 3.5. Списки	**Содержание учебного материала**		16	
	47.	Введение в списки.		

	48.	Основы работы со списками. Вложенные (двумерные) списки. Множества и словари в Python.		
	49.	Методы списков. Часть 1.		
	50.	Вывод элементов списка.		
	51.	Методы строк split() и join().		
	52.	Методы списков. Часть 2.		
	53.	Списочные выражения.		
	54.	Сортировка списков.		
	Лабораторные занятия		10	
	55.	Создание и обработка списков.		
	56.	Разделение строки на элементы списка и объединение элементов в строку.		
	57.	Сортировка списков и вывод результатов.		
	58.	Работа с вложенными (двумерными) списками.		
	59.	Использование множеств и словарей для хранения и обработки данных.		
Тема 3.6. Основы структурного программирования. Функции	**Содержание учебного материала**		4	
	60.	Функции без параметров и с параметрами. Локальные и глобальные переменные. Основы структурного программирования.		
	61.	Модули Python. Импорт стандартных модулей. Создание простого пользовательского модуля. Понятие рекурсии. Простейшие рекурсивные функции.		
	Лабораторные занятия		6	
	62.	Разработка простых функций для повторяющихся действий.		
	63.	Передача параметров в функцию и получение результата.		
	64.	Разбиение небольшой программы на понятные функции.		
Раздел 4. Основы объектно-ориентированного программирования (ООП)			**10**	
Тема 4.1. Основные принципы объектно-ориентированного программирования (ООП)	**Содержание учебного материала**		2	ОК 01 ОК 02 ОК 09 ПК 2.2 ПК 2.4
	65.	Базовые понятия объектно-ориентированного программирования: класс, объект, свойства (атрибуты) и методы. Инкапсуляция, наследование и полиморфизм на ознакомительном уровне.		
Тема 4.2. Работа над мини-проектом с использованием ООП	**Лабораторные занятия**		8	
	66.	Классы и объекты в Python. Создание класса, определение свойств (атрибутов) и методов.		
	67.	Работа с файлами		
	68.	Разработка, тестирование и доработка мини-проекта с использованием классов, файлов и простого графического интерфейса.		
	69.	Разработка небольших приложений: калькулятор		
Консультация			**2**	
Самостоятельная работа обучающихся:			**42**	
	1) Изучение основных этапов решения задачи на компьютере. Составление алгоритма решения простой бытовой задачи.			

	2) Составление линейных алгоритмов и представление их в словесной форме, в виде блок-схем и программы. 3) Выполнение заданий на ввод и вывод данных в программах на языке Python. Решение задач на использование переменных, арифметических операций и преобразование типов данных. 4) Решение вариативных задач на целочисленную арифметику: нахождение последней цифры числа, суммы цифр, количества десятков. 5) Оформление программ в соответствии с основными правилами читаемости кода и стилем PEP 8. 6) Решение задач с условным оператором if для выбора одного из двух вариантов действий. 7) Решение задач с разветвляющимися алгоритмами с использованием конструкции if – else. 8) Решение задач с несколькими вариантами выбора с использованием конструкции if – elif – else. 9) Решение задач с логическими операциями and, or, not и цепочками сравнений. Составление программ с вложенными условными операторами. 10) Решение задач с циклом for и функцией range(). 11) Решение задач с циклом while, выполняемым до наступления заданного условия. 12) Решение задач на обработку натуральных чисел по цифрам с использованием циклов.Решение задач с использованием операторов break и continue. 13) Решение задач на обработку строк: определение длины строки, получение символов по индексу, использование срезов.Решение задач с использованием основных методов строк: lower(), upper(), count(), find(), replace(), strip(). 14) Решение задач на форматирование строк и вывод результатов программы в требуемом виде. 15) Создание и обработка списков: добавление, удаление, изменение и вывод элементов. 16) Решение задач на сортировку списков, поиск максимального и минимального элементов, подсчёт количества элементов. 17) Решение задач с методами split() и join() для преобразования строки в список и объединения элементов списка в строку. Решение задач с использованием списковых выражений. 18) Решение задач с использованием вложенных списков, множеств и словарей. 19) Разработка функций с параметрами и возвращаемым значением. Использование модулей и простых рекурсивных функций. 20) Создание и использование классов, объектов, свойств и методов в программах на Python. 21) Разработка и тестирование мини-проекта на языке Python с использованием условий, циклов, списков, функций, файлов и классов.		
	Промежуточная аттестация в форме экзамена	**6**	
	Всего:	**188**	

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.06. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

3.1. Материально-техническое обеспечение

Для реализации программы учебной дисциплины предусмотрена лаборатория «Проектирование и разработка информационных систем», оснащенная оборудованием и техническими средствами обучения.

- Рабочие места на обучающихся.
- Автоматизированные рабочие места на обучающихся: персональный компьютер, монитор, клавиатура, мышь.
- Автоматизированное рабочее место преподавателя: ноутбук.
- Интерактивная доска.
- МФУ.
- Программное обеспечение общего и профессионального назначения: операционная система, офисный пакет, браузер, среда программирования Python, редактор кода Visual Studio Code или аналогичный.
- Комплект учебно-методической документации;
- Фонд оценочных средств по дисциплине.

3.2. Учебно-методическое обеспечение

3.2.1. Основные печатные и/или электронные издания

1. 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 (дата обращения: 15.05.2025).

2. Якимов, С. П. Алгоритмизация и программирование: учебник для среднего профессионального образования / С. П. Якимов. — Москва: Издательство Юрайт, 2025. — 342 с. — (Профессиональное образование). — ISBN 978-5-534-19661-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/569196 (дата обращения: 15.05.2025).

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.06. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Результаты обучения	Показатели освоенности компетенций	Методы оценки
Знает: понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; классификация языков программирования; понятие системы программирования;	Разработан и оформлен алгоритм для решения поставленной задачи и выполнена оценка его сложности; предложенный алгоритм реализован в среде программирования на одном из актуальных языков программирования; код разработанной программы	Экспертное наблюдение выполнения практических работ и видов работ по практике Диагностика (тестирование,

основные элементы языка, структура программы; методы реализации типовых алгоритмов; операторы и операции, управляющие структуры, структуры данных, классы памяти; понятие подпрограммы, библиотеки подпрограмм; объектно-ориентированная модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования	отлажен, оформлен в соответствии со стандартами кодирования и соответствует алгоритму (результат выполнения соответствует эталонному).	контрольные работы)
Умеет: разрабатывать и анализировать алгоритмы для решения поставленных задач; определять сложность алгоритмов; реализовывать типовые алгоритмы в виде программ на актуальных языках программирования; использовать средства проектирования для создания и графического отображения алгоритмов; оформлять код программ в соответствии со стандартом кодирования; выполнять проверку, отладку кода программы		