



УТВЕРЖДАЮ
Заместитель директора Ур СЭИ (филиал)
ОУП ВО «АТиСО» по УВР

_____ О.В. Зубкова

« 17» февраля 2026 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)
ОПЦ.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

**СПЕЦИАЛЬНОСТЬ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ**

09.02.11 Разработка и управление программным обеспечением
(код профессии, специальности СПО)

Программист
(наименование квалификации)

Кафедра: Информатики и математики

Разработчик программы:
Зав. кафедрой

Маскайкин Е.П.

17 февраля 2026 г.

Оглавление

1. ОБЛАСТЬ ПРИМЕНЕНИЯ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОГО ПРЕДМЕТА, КУРСА, ДИСЦИПЛИНЫ (МОДУЛЯ)	3
2. ЦЕЛИ И ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	3
3. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ПРОГРАММЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА	3
4. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ:.....	3
5.СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
6. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	9
7. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	10
8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	11
9.ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ СО.02.02. ИНФОРМАТИКА	11

1. ОБЛАСТЬ ПРИМЕНЕНИЯ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОГО ПРЕДМЕТА, КУРСА, ДИСЦИПЛИНЫ (МОДУЛЯ)

Рабочая программа учебной дисциплины «Основы алгоритмизации и программирования» является частью программы подготовки специалистов среднего звена (ППССЗ) в соответствии с ФГОС СПО по специальности 09.02.11 Разработка и управление программным обеспечением, квалификация – программист.

Программа может быть использована в дополнительном профессиональном образовании и профессиональной подготовке специалистов в области информационных технологий.

2. ЦЕЛИ И ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель изучения: формирование навыков работы в среде программирования, разработки алгоритмов для решения конкретных задач, реализации готовых и разработанных алгоритмов на выбранном языке программирования (Python, 1С и др.), а также отладки и тестирования программных модулей.

Задачи:

- изучить основы алгоритмизации, свойства алгоритмов, способы описания;
- освоить базовые конструкции языков программирования (линейные, ветвящиеся, циклические);
- научиться работать с массивами, строками, файлами;
- освоить технологии модульного и объектно-ориентированного программирования;
- сформировать навыки разработки, отладки и тестирования программных модулей;
- развить умения применять современные среды разработки и инструменты отладки.

3. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ПРОГРАММЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА

Учебная дисциплина «Основы алгоритмизации и программирования» относится к общепрофессиональному циклу (ОПЦ.06) специальности 09.02.11 Разработка и управление программным обеспечением.

Дисциплина базируется на знаниях, умениях и навыках, полученных при изучении информатики и математики в школе.

Знания, умения и навыки, полученные при изучении данной дисциплины, являются основой для освоения профессиональных модулей, связанных с разработкой программного обеспечения, баз данных, веб-приложений, а также для практической деятельности программиста.

4. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ:

В результате освоения дисциплины обучающийся должен овладеть следующими общими компетенциями:

Код	Формулировка компетенции
ПК 2.2	Разрабатывать модули программного обеспечения.
ПК 2.4	Выполнять тестирование и отладку программного обеспечения.

Таким образом, в результате освоения дисциплины обучающийся должен:
уметь:

- У1 – распознавать задачу, анализировать и выделять её составные части; определять этапы решения задачи;
- У2 – выявлять и эффективно искать информацию, необходимую для решения задачи;
- У3 – разрабатывать алгоритмы для конкретных задач с использованием блок-схем и псевдокода;
- У4 – применять средства информационных технологий для решения профессиональных задач (среды программирования);
- У5 – разрабатывать модули программного обеспечения с использованием различных языков программирования (Python) и технологий;
- У6 – использовать основные алгоритмические конструкции (линейные, ветвления, циклы);
- У7 – работать с массивами, строками, файлами;
- У8 – применять принципы модульного и объектно-ориентированного программирования;
- У9 – выполнять отладку и тестирование программного обеспечения;
- У10 – оформлять код в соответствии со стандартами.

знать:

- 31 – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- 32 – основные элементы языка программирования (типы данных, операторы, структуры данных);
- 33 – способы создания подпрограмм, библиотек подпрограмм;
- 34 – объектно-ориентированную модель программирования, основные принципы ООП (инкапсуляция, наследование, полиморфизм);
- 35 – методы отладки и тестирования программных модулей;
- 36 – современные среды разработки и инструменты тестирования.

5. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

5.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	3 семестр	4 семестр	Итого
Максимальная учебная нагрузка (всего)	30	42	72
Обязательная аудиторная учебная нагрузка (всего)	10	42	52

Вид учебной работы	3 семестр	4 семестр	Итого
в том числе: лекции	6	10	16
практические занятия	4	32	36
Самостоятельная работа обучающегося (всего)	20	0	20
Промежуточная аттестация	—	—	—
Итоговая аттестация	Дифференцированный зачет (3 семестр)	Экзамен (4 семестр)	Экзамен

5.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, практические занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
Раздел 1. Введение в программирование		30 (3 семестр)	
Тема 1.1. Основы алгоритмизации, языки и системы программирования	Лекция: Алгоритм и его свойства. Виды алгоритмов. Способы записи алгоритмов. Эволюция и классификация языков программирования. Среда программирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Практические занятия: Разработка блок-схем линейных алгоритмов. Самостоятельная работа: изучение среды Python, написание простейших программ.	4 (2 лекции + 1 практика + 1 СР)	1-2
Тема 1.2. Основные элементы языка. Типы данных. Операторы.	Лекция: Основные элементы языка. Типы данных (простые, производные). Операторы ввода/вывода. Обработка исключений. Линейные программы. Практические занятия: Написание линейных программ на Python. Самостоятельная работа: решение задач на линейные алгоритмы.	6 (2 лекции + 1 практика + 3 СР)	2
Тема 1.3. Управляющие структуры. Ветвления и циклы.	Лекция: Операторы разветвляющихся программ (if, elif, else). Циклические программы (while, for). Практические занятия: Написание программ с ветвлениями и циклами. Самостоятельная работа: решение задач на ветвления и циклы.	8 (1 лекция + 1 практика + 6 СР)	2

Наименование разделов и тем	Содержание учебного материала, практические занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
Тема 1.4. Массивы. Строки.	Лекция: Одномерные и двумерные массивы. Сортировка массивов. Строки, операции со строками. Практические занятия: Работа с массивами и строками. Самостоятельная работа: решение задач на обработку массивов и строк.	8 (1 лекция + 1 практика + 6 СР)	2
Тема 1.5. Файлы.	Лекция: Доступ к файлам. Виды файлов. Считывание и запись в файл. Практические занятия: Работа с файлами. Самостоятельная работа: решение задач на работу с файлами.	4 (0 лекций + 1 практика + 3 СР)	2
Раздел 2. Технологии программирования		42 (4 семестр)	
Тема 2.1. Модульное программирование.	Лекция: Локальные и глобальные переменные. Подпрограммы (функции). Передача данных. Рекурсия. Практические занятия: Создание функций, рекурсивных функций. Создание модулей. Самостоятельная работа: (не выделена)	14 (4 лекции + 10 практик)	2-3
Тема 2.2. Основные принципы объектно-ориентированного программирования.	Лекция: Базовые понятия ООП: объект, класс, интерфейс. Инкапсуляция, наследование, полиморфизм. Практические занятия: Создание классов, конструкторов, свойств. Реализация наследования и полиморфизма.	14 (4 лекции + 10 практик)	2-3
Раздел 3. Разработка приложений		14 (4 семестр)	

Наименование разделов и тем	Содержание учебного материала, практические занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
Тема 3.1. Этапы разработки приложений.	Лекция: Визуально-событийное программирование. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя (консольного/графического). Тестирование, отладка, оптимизация. Практические занятия: Разработка консольных приложений с использованием изученных конструкций. Тестирование и отладка кода.	14 (2 лекции + 12 практик)	3
Всего		72	

6. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

6.1. Требования к минимальному материально-техническому обеспечению

Реализация программы требует наличия лаборатории «Алгоритмизации и программирования», оснащённой в соответствии с требованиями ФГОС.

Оборудование лаборатории:

- рабочие места по количеству обучающихся (компьютеры с установленными средами разработки: Python, PyCharm или VS Code);
- рабочее место преподавателя;
- мультимедийное оборудование (проектор, экран).

Программное обеспечение:

- Python (версия 3.x);
- интегрированная среда разработки (PyCharm, VS Code или IDLE);
- утилиты для тестирования и отладки (pytest, unittest).

Технические средства обучения:

- компьютер, мультимедийный проектор, экран;
- доступ к сети Интернет для использования электронных образовательных ресурсов.

6.2. Информационное обеспечение обучения

Основная литература

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>.
2. 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>.
3. 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>.

Дополнительная литература

4. 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>.

5. Черпаков, И. В. Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 196 с. — (Высшее образование). — ISBN 978-5-534-18759-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/583636>.
6. Нагаева, И. А. Основы алгоритмизации и программирования : практикум : учебное пособие / И. А. Нагаева, И. А. Кузнецов. — 2-е изд., стер. — Москва : Директ-Медиа, 2025. — 168 с. : схем. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=722929>. — Библиогр.: с. 162-163. — ISBN 978-5-4499-5220-2. — DOI 10.23681/722929. — Текст : электронный..

Ресурсы информационно-телекоммуникационной сети «Интернет», необходимые для освоения дисциплины

<https://metanit.com> – сайт по программированию;
<https://docs.python.org/3/> – официальная документация Python;
<https://stepik.org> – онлайн-курсы по Python.

7. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Рабочая программа учебной дисциплины обеспечена оценочными материалами для проведения текущего контроля и промежуточной аттестации.

Оценочные материалы включают в себя тестовые задания, практические задания, контрольные работы, вопросы к дифференцированному зачёту и экзамену.

Текущий контроль осуществляется в форме проверки выполнения практических работ, устных опросов, тестирования, защиты отчётов по самостоятельной работе.

Промежуточная аттестация:

- в 3 семестре – дифференцированный зачёт (защита выполненных практических заданий или письменная работа);
- в 4 семестре – экзамен (письменная работа по билетам, включающая теоретические вопросы и задачи на программирование).

Примерный перечень вопросов к дифзачету и экзамену:

(Включает вопросы по всем разделам)

1. Алгоритм, его свойства, способы записи.
2. Линейный алгоритм, примеры.
3. Условный оператор if-else, примеры.
4. Циклы while и for, примеры.
5. Одномерные массивы, операции.
6. Двумерные массивы.
7. Строки, основные операции.
8. Работа с файлами: чтение, запись.
9. Функции: определение, параметры, возврат значения.
10. Рекурсия, примеры.
11. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
12. Классы и объекты, конструкторы, методы.
13. Наследование, переопределение методов.
14. Полиморфизм, примеры.
15. Тестирование и отладка программ.
16. Сортировка массивов (пузырьковая, выбором).
17. Обработка исключений.
18. Модули и библиотеки в Python.

19. Разработка консольных приложений.
20. Жизненный цикл программного продукта.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

По выполнению лекционных занятий:

- вести конспект, фиксируя основные определения, синтаксис, примеры;
- после лекции прорабатывать материал по учебнику;
- выделять неясные вопросы для обсуждения с преподавателем.

По выполнению практических занятий:

- выполнять задания в среде программирования, следуя инструкциям;
- комментировать код, использовать отладчик;
- фиксировать результаты, делать выводы.

По самостоятельной работе:

- выполнять домашние задания, решать задачи на программирование;
- использовать онлайн-тренажёры (например, Stepik);
- готовиться к контрольным работам и зачёту/экзамену.

По подготовке к дифзачету и экзамену:

- повторить теоретические положения по разделам;
- прорешать типовые задачи;
- использовать вопросы для самопроверки.

9. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Фонд оценочных средств включает задания разных типов, проверяющих как знание теории, так и умение применять математические методы.

Тип задания	Инструкция	Пример задания
Задание закрытого типа на установление соответствия (Знать)	Прочитайте текст и установите соответствие между конструкциями языка Python и их описанием.	Установите соответствие между конструкцией и её описанием: А) if; Б) for; В) while; Г) def. 1) Определение функции; 2) Цикл с предусловием; 3) Цикл с параметром; 4) Условный оператор.

Тип задания	Инструкция	Пример задания
Задание закрытого типа на установление соответствия (Уметь)	Прочитайте текст и установите соответствие между типами данных и их примерами.	Установите соответствие между типом данных и примером значения: А) int; Б) float; В) str; Г) bool. 1) "Hello"; 2) 3.14; 3) True; 4) 42.
Задание закрытого типа на установление последовательности (Знать)	Прочитайте текст и установите правильную последовательность действий при разработке программы.	Расположите в правильном порядке этапы разработки программы: 1) Тестирование; 2) Кодирование; 3) Проектирование; 4) Анализ требований.
Задание закрытого типа на установление последовательности (Уметь)	Прочитайте текст и установите правильную последовательность выполнения операций при сортировке пузырьком.	Расположите шаги сортировки пузырьком для массива [5, 3, 8, 1] в правильном порядке: 1) Сравнение 5 и 3, замена; 2) Сравнение 5 и 8, без замены; 3) Сравнение 8 и 1, замена; 4) Повтор проходов до полной сортировки.
Задание с выбором одного верного ответа (Знать)	Прочитайте текст, выберите правильный ответ и запишите аргументы.	Какой оператор используется для выхода из цикла досрочно? 1) break; 2) continue; 3) return; 4) exit.
Задание с выбором одного верного ответа (Уметь)	Прочитайте текст, выберите правильный ответ и запишите аргументы.	Какая функция в Python используется для преобразования строки в целое число? 1) int(); 2) str(); 3) float(); 4) input().
Задание с выбором нескольких вариантов (Знать)	Прочитайте текст, выберите все правильные ответы и запишите обоснование.	Какие из перечисленных конструкций являются циклами в Python? 1) for; 2) while; 3) if; 4) repeat.

Тип задания	Инструкция	Пример задания
Задание с выбором нескольких вариантов (Уметь)	Прочитайте текст, выберите все правильные ответы и запишите обоснование.	Какие из перечисленных методов используются для работы со строками в Python? 1) .upper(); 2) .lower(); 3) .split(); 4) .append().
Задание комбинированное с выбором одного верного ответа и обоснованием (Знать)	Прочитайте текст, выберите правильный ответ и запишите развёрнутое обоснование.	Что такое рекурсия? 1) Вызов функцией самой себя; 2) Цикл с предусловием; 3) Обработка исключений; 4) Работа с файлами.
Задание комбинированное с выбором одного верного ответа и обоснованием (Уметь)	Прочитайте текст, выберите правильный ответ и запишите развёрнутое обоснование.	Какой оператор используется для открытия файла в Python? 1) open(); 2) file(); 3) read(); 4) write().
Задание комбинированное с выбором нескольких вариантов и обоснованием (Знать)	Прочитайте текст, выберите все правильные ответы и запишите развёрнутое обоснование.	Какие принципы относятся к ООП? 1) Инкапсуляция; 2) Наследование; 3) Полиморфизм; 4) Абстракция.
Задание комбинированное с выбором нескольких вариантов и обоснованием (Уметь)	Прочитайте текст, выберите все правильные ответы и запишите развёрнутое обоснование.	Какие способы отладки программ существуют? 1) Использование отладчика (pdb); 2) Вывод отладочной информации (print); 3) Сборка проекта; 4) Тестирование с использованием assert.
Задание открытого типа с развёрнутым ответом (Знать)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Опишите свойства алгоритма и приведите пример линейного алгоритма.
Задание открытого типа с развёрнутым ответом (Уметь)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Напишите программу на Python для нахождения суммы элементов массива.

Тип задания	Инструкция	Пример задания
Задание открытого типа с развёрнутым ответом (Знать)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Объясните разницу между функциями и методами в Python.
Задание открытого типа с развёрнутым ответом (Уметь)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Напишите программу, которая читает числа из файла и выводит их сумму.
Практическое задание (Уметь)	Выполните задание и запишите результат.	Напишите функцию для вычисления факториала числа с использованием рекурсии.
Практическое задание (Уметь)	Выполните задание и запишите результат.	Создайте класс «Студент» с полями имя, возраст, оценки. Реализуйте метод для вычисления среднего балла.
Практическое задание (Уметь)	Выполните задание и запишите результат.	Напишите программу для сортировки списка чисел.
Практическое задание (Уметь)	Выполните задание и запишите результат.	Создайте модуль с функциями для работы с геометрическими фигурами (площадь, периметр) и импортируйте его в основную программу.