

**РОСЖЕЛДОР**

**Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
"Ростовский государственный университет путей сообщения"  
(ФГБОУ ВО РГУПС)**

---

**УТВЕРЖДАЮ**  
Первый проректор М.А. Кравченко

Кафедра «Вычислительная техника и автоматизированные системы управления»

**РАБОЧАЯ ПРОГРАММА  
ДИСЦИПЛИНЫ**

**ОП.03 «Основы алгоритмизации и программирования»**

**по Учебному плану**

специальности среднего профессионального образования  
09.02.09 Веб-разработка

Квалификация специалиста среднего звена "Разработчик веб-приложений"

Ростов-на-Дону  
2025 г.

Авторы-составители Загика Константин Николаевич и Щербакова Ксения Сергеевна предлагают настоящую Рабочую программу дисциплины ОП.03 «Основы алгоритмизации и программирования» в качестве материала для проектирования Образовательной программы РГУПС и осуществления учебно-воспитательного процесса по федеральному государственному образовательному стандарту среднего профессионального образования.

Рабочая программа дисциплины рассмотрена на кафедре «Вычислительная техника и автоматизированные системы управления».

## **Наименование, цель и задача дисциплины**

**Дисциплина «Основы алгоритмизации и программирования».**

Учебный план по Образовательной программе утвержден на заседании Ученого совета университета от 26.12.2025 г. №4.

Целью дисциплины «Основы алгоритмизации и программирования» является подготовка в составе других дисциплин блока "Дисциплины (модули)" Образовательной программы в соответствии с требованиями, установленными федеральным государственным образовательным стандартом среднего профессионального образования для формирования у выпускника профессиональных компетенций, способствующих решению профессиональных задач в соответствии с типом задач профессиональной деятельности, предусмотренным учебным планом.

Задачи изучения дисциплины:

- анализ и алгоритмизация решаемых задач;
- оформление решения задачи в графическом виде (в виде схем алгоритмов);
- программирование любого алгоритма, задачи, метода;

В результате освоения дисциплины обучающийся должен знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;

В результате освоения дисциплины обучающийся должен уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Рабочая программа учебной дисциплины ОП.03 «Основы алгоритмизации и программирования» предназначена для реализации программы подготовки специалистов среднего звена по специальности 09.02.09 веб-разработка СПО в очной форме обучения на базе основного общего образования. Содержит разделы: общая характеристика рабочей образовательной программы учебной дисциплины, структура и содержание учебной дисциплины, условия реализации учебной дисциплины, контроль и оценка результатов освоения учебной дисциплины. Определяет объем, содержание, порядок изучения учебной дисциплины, а также способы контроля результатов ее изучения.

## **Виды деятельности**

- проектирование и разработка информационных ресурсов;
- техническая поддержка и администрирование информационных ресурсов;
- разработка веб-приложения на стороне клиента (по выбору);
- разработка веб-приложения на стороне сервера (по выбору).

**ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО  
ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ  
РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.**

**В РАМКАХ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ  
ОБУЧАЮЩИМИСЯ ОСВАИВАЮТСЯ УМЕНИЯ И ЗНАНИЯ.**

<b>Код и наименование компетенции выпускника</b>	<b>Формулировка требований к степени сформированности компетенции</b>
OK 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.	<p><b>Умения:</b></p> <p>Анализировать постановку вычислительной задачи и определять наиболее подходящий тип алгоритма;</p> <p>Выбирать оптимальные структуры данных для хранения и обработки информации в зависимости от условий конкретной задачи;</p> <p>Сравнивать эффективность различных алгоритмов решения одной задачи и обосновывать выбор;</p> <p>Выбирать парадигму программирования в зависимости от сложности и масштаба разрабатываемой программы.</p> <p><b>Знания:</b></p> <p>Классификации алгоритмов и их свойств;</p> <p>Базовых принципов оценки сложности алгоритмов;</p> <p>Особенностей и областей применения различных парадигм программирования.</p>
OK 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.	<p><b>Умения:</b></p> <p>Использовать официальную документацию Python, справочные системы IDE и ресурсы для поиска информации о синтаксисе, стандартных библиотеках и методах решения типовых проблем программирования;</p> <p>Применять интегрированные среды разработки со всеми их инструментами для повышения эффективности разработки;</p> <p>Анализировать и интерпретировать сообщения об ошибках для локализации и устранения дефектов в коде.</p> <p><b>Знания:</b></p> <p>Структуры и правил работы с официальной технической документацией по языку программирования;</p> <p>Основных возможностей современных сред разработки для python;</p> <p>Базовых принципов визуализации данных в контексте программной разработки.</p>
OK 09. Пользоваться профессиональной документацией на государственном и иностранном языках.	<p><b>Умения:</b></p> <p>Читать и понимать фрагменты технических заданий, описаний алгоритмов и спецификаций на русском языке для корректной реализации программы;</p> <p>Использовать англоязычную документацию к стандартным библиотекам python;</p> <p>Понимать ключевую профессиональную лексику на английском языке, сообщениях об ошибках, названиях функций и методов.</p> <p><b>Знания:</b></p> <p>Базовой профессиональной терминологии в области</p>

	<p>алгоритмизации и программирования на русском и английском языках; Структуры и условных обозначений, принятых в технической документации по программированию.</p>
ПК 1.1. Проектировать информационные ресурсы.	<p><b>Умения:</b></p> <p>Анализировать постановку задачи и выделять основные сущности и процессы предметной области; Разрабатывать алгоритмы решения типовых вычислительных задач с использованием основных алгоритмических конструкций; Выбирать оптимальные структуры данных для хранения и обработки информации в соответствии с условиями задачи; Проектировать структуру программы, выделяя логические блоки и определяя взаимосвязи между ними.</p> <p><b>Знания:</b></p> <p>Основных свойств алгоритмов (дискретность, понятность, определенность, результативность, массовость) и способов их описания; Основных алгоритмических конструкций и принципов структурного программирования; Характеристик, областей применения и ограничений базовых структур данных языка python; Этапов жизненного цикла программного обеспечения.</p>
ПК 1.2. Разрабатывать интерфейсы пользователя.	<p><b>Умения:</b></p> <p>Организовывать консольный ввод данных от пользователя с использованием функции <code>input()</code> и преобразованием типов; Форматировать вывод результатов работы программы для удобства восприятия; Реализовывать простое интерактивное консольное меню выбора действий; Обеспечивать «дружественность» интерфейса путем обработки некорректного ввода пользователя с использованием механизма исключений;</p> <p><b>Знания:</b></p> <p>Принципов организации ввода-вывода в консольных приложениях; Методов форматирования строк в python; Базовых принципов создания удобного интерфейса и обработки ошибок ввода.</p>
ПК 1.3. Интегрировать программный код в соответствующую инфраструктуру.	<p><b>Умения:</b></p> <p>Создавать пользовательские функции и классы для организации многофайлового проекта; Импортировать и использовать собственные и стандартные модули python; Интегрировать отдельные программные модули в единую исполняемую программу; Организовывать чтение исходных данных из внешних файлов и запись итоговых результатов в файлы; Использовать виртуальные окружения для управления зависимостями проекта.</p> <p><b>Знания:</b></p> <p>Принципов модульности и повторного использования кода; Синтаксиса и семантики оператора импорта модулей в python;</p>

	<p>Основных форматов файлов для обмена данными и способов работы с ними;</p> <p>Назначения и основы использования виртуальных окружений.</p>
ПК 1.4. Использовать систему контроля версий в процессе коллективной (параллельной) разработки.	<p><b>Умения:</b></p> <p>Создавать локальный и удаленный репозиторий для учебного программного проекта;</p> <p>Выполнять основные операции с репозиторием;</p> <p>Организовывать работу с удаленным репозиторием;</p> <p>Выполнять слияние веток и разрешать простые конфликты слияния;</p> <p>Анализировать историю коммитов.</p> <p><b>Знания:</b></p> <p>Назначения, преимуществ и базовых принципов работы распределенных систем контроля версий;</p> <p>Инициировать репозиторий для учебного проекта и подключать удаленный репозиторий;</p> <p>Корректно фиксировать изменения в коде с информативными комментариями;</p> <p>Создавать отдельные ветки для разработки новых функций и сливать их с основной веткой.</p>
ПК 1.5. Выполнять процедуры тестирования программного кода.	<p><b>Умения:</b></p> <p>Идентифицировать типы ошибок в программе;</p> <p>Применять основные методы отладки;</p> <p>Разрабатывать и применять простые тестовые наборы данных;</p> <p>Проводить ручное функциональное тестирование программы на соответствие техническому заданию.</p> <p><b>Знания:</b></p> <p>Классификации ошибок программирования;</p> <p>Основных стратегий и методов тестирования программного обеспечения (отладка, модульное тестирование);</p> <p>Основных типов исключений в python и принципов их обработки.</p>

## **Место дисциплины ОП.03 «Основы алгоритмизации и программирования»**

Дисциплина отнесена к общепрофессиональному циклу Образовательной программы. Дисциплина реализуется в 3 и 4 семестре.

**Объем дисциплины в академических часах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся:**

<b>Вид учебной работы</b>	<b>Всего часов</b>	<b>Число часов в семестре</b>	
		<b>3</b>	<b>4</b>
<b>Объем образовательной программы учебной дисциплины</b>	<b>144</b>	<b>72</b>	<b>72</b>
в том числе:			
Лекции (теоретическое обучение)	52	32	20
Практические занятия			
Лабораторные работы	72	32	40
Самостоятельная работа	12	6	6
<b>Промежуточная аттестация (в форме зачета в 3 семестре и экзамена в 4 семестре)</b>	<b>8</b>	<b>2</b>	<b>6</b>

**Вид обучения: 2 года 10 месяцев очное СПО**

**Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

### ***Содержание дисциплины***

	<b>Раздел дисциплины</b>	<b>Изучаемые компетенции</b>
1	Основы алгоритмизации и технологии программирования.	ОК 01.; ОК 02.; ОК 09.; ПК 1.1.; ПК 1.2.; ПК 1.3.; ПК 1.4.; ПК 1.5.
2	Основы программирования.	ОК 01.; ОК 02.; ОК 09.; ПК 1.1.; ПК 1.2.; ПК 1.3.; ПК 1.4.; ПК 1.5.

**Отведенное количество часов по видам учебных занятий и работы**  
**Лекционные занятия**  
**Семестр № 3**

№ п/п	Наименование лекционных занятий	Трудоемкость аудиторной работы, часы
<b>Раздел № 1</b>		
1	Введение. Понятие алгоритма и его свойства. Типы алгоритмов. Способы описания алгоритмов	2
2	Схемы алгоритмов. Основные алгоритмические конструкции.	2
3	Введение. Элементы технологии программирования.	2
4	Понятие структурного, модульного, объектно-ориентированного программирования.	2
<b>Раздел № 2</b>		
5	Идентификаторы. Ключевые слова и имена. Символы операций и разделители. Литералы.	2
6	Типы данных и объявления переменных.	2
7	Операции и выражения. Операторы присваивания. Операторы ввода- вывода.	2
8	Организация ветвлений и циклов. Составные и пустые операторы.	2
9	Условные операторы. Оператор-переключатель.	2
10	Организация циклических вычислений. Операторы цикла. Вложенные циклы. Операторы перехода и возврата.	2
11	Массивы как структурированный тип данных. Объявление массивов. Ввод-вывод одномерных массивов. Обработка одномерных массивов.	2
12	Двумерные массивы. Ввод-вывод двумерных массивов. Обработка двумерных массивов	2
13	Списки, кортежи, множества, словари. Создание списков, кортежей, множеств и словарей операции над списками и кортежами.	2
14	Строки. Объявление строковых типов данных. Стандартные функции для работы со строками.	2
15	Поиск, удаление, замена символа в строке	2
16	Регулярные выражения	2

**Семестр № 4**

№ п/п	Наименование лекционных занятий	Трудоемкость аудиторной работы, часы
<b>Раздел № 2</b>		
1	Пользовательские типы данных. Действия над пользовательскими тиปами данных.	2
2	Понятие функции, их сущность и назначение. Организация функций.	2
3	Функции, определенные пользователем, передача аргументов. Декораторы	2

<b>№ п/п</b>	<b>Наименование лекционных занятий</b>	<b>Трудоемкость аудиторной работы, часы</b>
4	Рекурсия. Примеры классических рекурсивных задач.	2
5	Типы файлов. Открытие и закрытие файла. Запись в файл, чтение данных из файла. Функции работы с файлами.	2
6	Стеки и очереди. Программирование алгоритмов с использованием стеков и очередей.	2
7	Двунаправленные списки. Программирование алгоритмов с использованием двунаправленных списков.	2
8	Основы объектно-ориентированного программирования: объекты, классы и инкапсуляция.	2
9	Полиморфизм и наследование: расширение и специализация поведения классов	2
10	Деревья. Программирование алгоритмов с использованием деревьев.	2

**Лабораторный практикум**  
**Семестр 3**

<b>№ п/п</b>	<b>Наименование лабораторных работ</b>	<b>Трудоемкость аудиторной работы, часы</b>
<b>Раздел № 1</b>		
1	Лабораторная работа № 1. Разработка и анализ простых алгоритмов с использованием словесного описания и пошаговой детализации. Исследование свойств алгоритмов (детерминированность, конечностъ, массовость и др.) на примере конкретных задач.	2
2	Лабораторная работа № 2. Разработка алгоритмов линейной, разветвляющейся, циклической и рекурсивной структуры. Сравнение различных типов алгоритмов (линейный, разветвляющийся, циклический, рекурсивный) на основе решения одной задачи разными способами.	4
3	Лабораторная работа № 3. Установка интерпретатора Python. Знакомство со штатной средой разработки IDLE.	2
4	Лабораторная работа № 4. Сравнительный анализ: решение одной и той же задачи средствами структурного, модульного и объектно-ориентированного программирования.	2
<b>Раздел № 2</b>		
5	Лабораторная работа № 5. Правила именования и использование идентификаторов. Работа с литералами и основными символами языка.	2
6	Лабораторная работа № 6. Правила объявления и использования переменных с различными типами данных.	2
7	Лабораторная работа № 7. Вычисления с использованием арифметических, логических и сравнительных операций. Ввод данных, присваивание и форматированный вывод.	2
8	Лабораторная работа № 8. Разработка программ линейной структуры	2

<b>№ п/п</b>	<b>Наименование лабораторных работ</b>	<b>Трудоемкость аудиторной работы, часы</b>
9	Лабораторная работа № 9. Разработка программ разветвляющей структуры	2
10	Лабораторная работа № 10. Разработка программ циклической структуры	2
11	Лабораторная работа № 11. Разработка программ с использованием одномерных массивов	2
12	Лабораторная работа № 12. Разработка программ с использованием двумерных массивов	2
13	Лабораторная работа № 13. Разработка программ с использованием списков, кортежей, множеств и словарей	2
14	Лабораторная работа № 14. Разработка программ с использованием стандартных функций для работы со строками.	2
15	Лабораторная работа № 15. Разработка программ с использованием регулярных выражений для работы со строками.	2

***Семестр 4***

<b>№ п/п</b>	<b>Наименование лабораторных работ</b>	<b>Трудоемкость аудиторной работы, часы</b>
<i><b>Раздел № 2</b></i>		
1	Лабораторная работа № 16. Разработка программ с использованием пользовательских типов данных, действия над пользовательскими типами данных.	4
2	Лабораторная работа № 17. Разделение программы на функции: выделение логических блоков и оформление их как отдельных функций.	2
3	Лабораторная работа № 18. Разработка функций с передачей аргументов разных типов, с использованием изменяемых и неизменяемых аргументов.	4
4	Лабораторная работа № 19. Разработка функций с передачей произвольного числа аргументов.	2
5	Лабораторная работа № 20. Разработка функций с реализацией классических рекурсивных алгоритмов: факториал, числа Фибоначчи, НОД (алгоритм Евклида).	2
6	Лабораторная работа № 21. Разработка программ с использованием рекурсивной обработки структур данных: обход вложенных списков или словарей, вычисление суммы/глубины/количества элементов.	2
7	Лабораторная работа № 22. Создание текстового и двоичного файла. Чтение из файла. Изменение данных в файле	2
8	Лабораторная работа № 23. Создание форматированного файла (CSV, JSON). Чтение из файла. Изменение данных в файле.	2
9	Лабораторная работа № 24. Разработка программ с использованием однонаправленных списков типа стек и очередь.	4

№ п/п	Наименование лабораторных работ	Трудоемкость аудиторной работы, часы
10	Лабораторная работа № 25. Разработка программ с использованием двусвязных списков	4
11	Лабораторная работа № 26. Создание классов и объектов. Инкапсуляция и управление доступом к данным.	4
12	Лабораторная работа № 27. Разработка программы с реализацией иерархии классов с использованием наследования.	2
13	Лабораторная работа № 28. Разработка программы с применением полиморфизма и переопределением методов.	2
14	Лабораторная работа № 29. Разработка программ с использованием деревьев и бинарных деревьев.	4

***Самостоятельное изучение учебного материала (самоподготовка)***

Номер раздела данной дисциплины	Наименование тем, вопросов, вынесенных для самостоятельного изучения	Трудоемкость внеаудиторной работы, часы
<b>Семестр № 2</b>		
1	Работа с внешними библиотеками: установка и использование. Установка пакетов через pip, виртуальные окружения (venv), примеры использования популярных библиотек (например, requests для HTTP-запросов или tabulate для красивого вывода таблиц).	2
2	Работа с числами: точные вычисления и специальные типы. Использование модулей decimal и fractions для точной арифметики, сравнение с float, применение в финансовых или научных задачах, где важна точность.	2
<b>Семестр № 4</b>		
2	Работа с датой и временем в Python. Использование модулей datetime, time, calendar: получение текущей даты/времени, форматирование, вычисление разницы между датами, работа с часовыми поясами.	2
2	Обработка исключений и отладка программ. Использование конструкций try, except, else, finally; создание пользовательских исключений; применение отладчика (например, в IDLE или VS Code); логирование с помощью модуля logging.	2

<b>Номер раздела данной дисциплины</b>	<b>Наименование тем, вопросов, вынесенных для самостоятельного изучения</b>	<b>Трудоемкость внеаудиторной работы, часы</b>
2	Тестирование программ: написание unit-тестов. Использование модуля unittest или pytest для проверки корректности функций, написание тест-кейсов, проверка граничных значений и ошибочных входных данных.	2
2	Создание документации и аннотаций типов. Написание docstring в стиле Google или NumPy, использование type hints (typing), проверка типов с помощью mypy, улучшение читаемости и поддержки кода.	2

**Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

**Перечень компетенций с указанием этапов их формирования в процессе освоения Образовательной программы**

<b>Компетенция</b>	<b>Указание (+) этапа формирования в процессе освоения ОП (3 семестр)</b>
ПК 1.1	+
ПК 1.2	+
ПК 1.3	+
ПК 1.4	+
ПК 1.5	+

<b>Компетенция</b>	<b>Указание (+) этапа формирования в процессе освоения ОП (4 семестр)</b>
ПК 1.1	+
ПК 1.2	+
ПК 1.3	+
ПК 1.4	+
ПК 1.5	+

**Описание показателей и критериев оценивания компетенций на различных этапах их формирования**

<b>Компетенция</b>	<b>Этап формирования ОП (семестр)</b>	<b>Показатель оценивания</b>	<b>Критерий оценивания</b>
ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	3	Дуальная оценка на зачете	- полнота усвоения материала, - качество изложения материала, - правильность выполнения заданий, - аргументированность решений.
ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	4	Балльная оценка на экзамене	- полнота усвоения материала, - качество изложения материала, - правильность выполнения заданий, - аргументированность решений.

<b>Компетенция</b>	<b>Этап формирования ОП (семестр)</b>	<b>Показатель оценивания</b>	<b>Критерий оценивания</b>
ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	4	Процент верных на тестировании	- правильность выполнения заданий.
ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	4	Выполненная лабораторная работа	- правильность выполнения заданий.

### ***Описание шкал оценивания компетенций***

<b>Значение оценки</b>	<b>Уровень освоения компетенции</b>	<b>Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)</b>	<b>Шкала оценивания (процент верных при проведении тестирования)</b>
Балльная оценка - "удовлетворительно".	Пороговый	Оценка «удовлетворительно» выставляется обучающемуся, который имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения последовательности изложения программного материала и испытывает трудности в выполнении практических навыков.	От 40% до 59%
Балльная оценка - "хорошо".	Базовый	Оценка «хорошо» выставляется обучающемуся, твердо знающему программный материал, грамотно и по существу его излагающему, который не допускает существенных неточностей в ответе, правильно применяет теоретические положения при решении практических работ и задач, владеет необходимыми навыками и приемами их выполнения.	От 60% до 84%
Балльная оценка - "отлично".	Высокий	Оценка «отлично» выставляется обучающемуся, глубоко и прочно усвоившему программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагающему, в ответе которого тесно увязываются теория с практикой. При этом обучающийся не затрудняется с ответом при видоизменении задания, показывает знакомство с литературой, правильно обосновывает ответ, владеет разносторонними навыками и приемами практического выполнения практических работ.	От 85% до 100%

**Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

**Типовые контрольные задания**

**Курсовые проекты (работы)**

Проектирование и разработка программы на выбранную тематику.

**Контрольные работы, расчетно-графические работы, рефераты**

Не предусмотрено.

**Перечни сопоставленных с ожидаемыми результатами освоения дисциплины вопросов (задач):**

Зачет. Семестр. № 3

**Вопросы для оценки результата освоения "Знать":**

- 1) Основные свойства алгоритмов и способы их описания. Сравнение типов алгоритмов: линейных, разветвляющихся, циклических, рекурсивных.
- 2) Основные алгоритмические конструкции и их графическое представление в виде блок-схем.
- 3) Принципы структурного, модульного и объектно-ориентированного программирования и их отличия.
- 4) Правила именования идентификаторов в Python. Ключевые слова языка и основные типы данных.
- 5) Классификация операторов Python: арифметические, сравнения, логические, присваивания, ввода-вывода.
- 6) Синтаксис и семантика условных операторов и операторов циклов. Понятие вложенных циклов.
- 7) Понятие структурированного типа данных. Организация работы с одномерными и двумерными массивами.
- 8) Характеристики и области применения основных структур данных Python: списков, кортежей, множеств и словарей.
- 9) Методы работы со строками. Базовые понятия регулярных выражений.

**Вопросы для оценки результата освоения "Уметь":**

- 1) Анализировать условие задачи и разрабатывать для нее алгоритм в виде блок-схемы или словесного описания.
- 2) Устанавливать интерпретатор Python и настраивать среду разработки для выполнения учебных проектов.
- 3) Разрабатывать программы линейной структуры с использованием операторов ввода, присваивания и вывода.
- 4) Разрабатывать программы разветвляющейся структуры для реализации логики принятия решений.
- 5) Разрабатывать программы циклической структуры для обработки последовательностей данных и реализации итеративных алгоритмов.
- 6) Создавать и обрабатывать одномерные и двумерные массивы (списки): выполнять поиск, сортировку, нахождение суммы, минимума/максимума.
- 7) Применять различные структуры данных для эффективного решения типовых задач.
- 8) Выполнять обработку строк: поиск и замена подстрок, работа с регулярными выражениями для валидации данных.

Экзамен. Семестр № 4

**Вопросы для оценки результата освоения "Знать":**

- 1) Понятие пользовательского типа данных. Основные действия: создание, инкапсуляция данных, определение методов.
- 2) Сущность и назначение функции в программе. Принципы организации кода с использованием функций.
- 3) Способы передачи аргументов в пользовательские. Различия между изменяемыми и неизменяемыми аргументами.
- 4) Понятие рекурсии. Условия корректного завершения рекурсивного алгоритма. Примеры классических рекурсивных задач.
- 5) Типы файлов. Основные операции при работе с файлами.
- 6) Принципы организации и основные операции для абстрактных типов данных.
- 7) Основные принципы объектно-ориентированного программирования.

**Вопросы для оценки результата освоения "Уметь":**

- 1) Создавать пользовательские типы данных (классы) для описания сущностей предметной области, инкапсулируя данные и методы работы с ними.
- 2) Разделять программу на логические блоки, оформляя их в виде отдельных функций с четко определенным интерфейсом.
- 3) Разрабатывать функции с передачей аргументов различных типов, в том числе произвольного количества.
- 4) Разрабатывать и применять рекурсивные алгоритмы для решения задач.
- 5) Организовывать работу с файлами различных форматов: чтение данных, их обработка и запись результатов.
- 6) Реализовывать базовые операции над динамическими структурами данных.
- 7) Реализовывать алгоритмы работы с бинарными деревьями.
- 8) Создавать иерархии классов, используя механизм наследования для расширения и специализации поведения.
- 9) Применять полиморфизм через переопределение методов и создание функций, работающих с объектами разных классов.
- 10) Выполнять процедуры тестирования и отладки программного кода: выявлять синтаксические и логические ошибки, использовать отладчик и инструменты трассировки.
- 11) Интегрировать отдельные программные модули (функции, классы) в единую работающую программу, решающую учебную задачу.

***Иные контрольные материалы для автоматизированной технологии оценки имеются в Центре мониторинга качества образования***

***Методические материалы, определяющие процедуру оценивания знаний, умений, характеризующих этапы формирования компетенций***

<b>№ п/п</b>	<b>Библиографическое описание</b>
1	Методические указания, определяющие процедуру оценивания знаний, умений, навыков, характеризующих этапы формирования компетенций: учебно-методическое пособие / М.С. Тимофеева; ФГБОУ ВО РГУПС. - 4-е изд., перераб. и доп. - Ростов н/Д, 2025. - 50 с.: ил. - Библиогр.: с. 14 (ЭБС РГУПС)
2	Разработка фондов оценочных средств в условиях цифровой трансформации высшего образования : учебное пособие/ М.С. Тимофеева, Г.С. Мизюков, В.Н. Семенов [и др.]; под ред. М.С. Тимофеевой; ФГБОУ ВО РГУПС. - Ростов-на-Дону : РГУПС, 2022. - 94 с.

**Для каждого результата обучения по дисциплине определены  
Показатели и критерии оценивания сформированности компетенций  
на различных этапах их формирования**

Резуль-тат обуче-ния	Компетенция	Этап формиро-вания в процессе освоения ОП (семестр)	Этапы формирования компетенции при изучении дисциплины (раздел дисциплины)	Показатель сформиро-ванности компетенции	Критерий оценивания
Знает, Умеет	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	3	1	Дуальная оценка на зачете	- полнота усвоения материала, - качество изложения материала, - правильность выполнения заданий, - аргументированность решений.
Знает, Умеет	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	4	2	Балльная оценка на экзамене	- полнота усвоения материала, - качество изложения материала, - правильность выполнения заданий, - аргументированность решений.
Знает, Умеет	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	3,4	1, 2	Процент верных на тестировании	- правильность выполнения заданий.
Знает, Умеет	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5	3,4	1, 2	Выполненная лабораторная работа	- правильность выполнения заданий.

**Шкалы и процедуры оценивания**

Значение оценки	Уровень освоения компетенции	Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)	Процедура оценивания
Балльная оценка - "отлично", "хорошо", "удовлетворительно". Дуальная оценка - "зачтено".	Пороговый, Базовый, Высокий	В соответствии со шкалой оценивания в разделе РПД "Описание шкал оценивания компетенций"	Экзамен (письменно-устный). Автоматизированное тестирование. Выполнение лабораторной работы

<b>Значение оценки</b>	<b>Уровень освоения компетенции</b>	<b>Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)</b>	<b>Процедура оценивания</b>
Балльная оценка - "неудовлетворительно". Дуальная оценка - "не зачтено".	Не достигнут		(подготовка отчета). Защита курсовой работы.

**Ресурсы электронной информационно-образовательной среды, электронной библиотечной системы и иные ресурсы, необходимые для осуществления образовательного процесса по дисциплине**

**Перечень учебной литературы для освоения дисциплины**

<b>№ п/п</b>	<b>Библиографическое описание</b>	<b>Ресурс</b>
1	Кудрявцева, И. А. Программирование: комбинаторная логика : учебник для среднего профессионального образования / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва: издательство Юрайт, 2025. — 524 с. — (профессиональное образование). — isbn 978-5-534-15128-2. — текст : электронный // образовательная платформа юрайт [сайт].	ЭБС Юрайт
2	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].	ЭБС Юрайт
3	Чернышев, С. А. Основы программирования на Python : учебник для среднего профессионального образования / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Профессиональное образование). — ISBN 978-5-534-17056-6. — Текст : электронный // Образовательная платформа Юрайт	ЭБС Юрайт

**Перечень учебно-методического обеспечения**

<b>№ п/п</b>	<b>Библиографическое описание</b>	<b>Ресурс</b>
1	Загика, К.Н. Основы алгоритмизации и программирования: учеб.-метод пособие для лаб.работ и сам.работы/ К. Н. Загика, К.С. Щербакова К. С. ; ФГБОУ ВО РГУПС. - Ростов н/Д:РГУПС, 2025. - 153 с. Библиогр.- Текст : электронный	ЭБС РГУПС

**Электронные образовательные ресурсы в сети "Интернет"**

<b>№ п/п</b>	<b>Адрес в Интернете, наименование</b>
1	<a href="http://rgups.ru/">http://rgups.ru/</a> . Официальный сайт РГУПС
2	<a href="http://www.iprbookshop.ru/">http://www.iprbookshop.ru/</a> . Электронно-библиотечная система "IPR SMART"
3	<a href="https://urait.ru/">https://urait.ru/</a> . Электронно-библиотечная система "Юрайт"
4	<a href="http://cmko.rgups.ru/">http://cmko.rgups.ru/</a> . Центр мониторинга качества образования РГУПС
5	<a href="https://portal.rgups.ru/">https://portal.rgups.ru/</a> . Система личных кабинетов НПР и обучающихся в ЭИОС
6	<a href="http://www.umczdt.ru/">http://www.umczdt.ru/</a> . Электронная библиотека "УМЦ ЖДТ"
7	<a href="https://webirbis.rgups.ru/">https://webirbis.rgups.ru/</a> . Электронно-библиотечная система РГУПС
8	<a href="https://eivis.ru/">https://eivis.ru/</a> . Универсальная база данных "ИВИС"

**Профессиональные базы данных и информационно-справочные системы**

№ п/п	Адрес в Интернете, наименование
1	http://www.glossary.ru/. Глоссарий.ру (служба тематических толковых словарей)
2	http://www.consultant.ru/. КонсультантПлюс

**Лицензионное и свободно распространяемое программное обеспечение**

№ п/п	Наименование	Произ-во
1	Debian, Simply Linux, Microsoft Windows. Системное программное обеспечение.	И
2	LibreOffice. Программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др.	И
3	Python-IDLE - среда разработки для написания кода на языке Python.	И

О - программное обеспечение отечественного производства

И - импортное программное обеспечение

**Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Аудитория компьютерного практикума:

Основное оборудование:

Стол на два посадочных места – 16 шт.

Стул – 26 шт.

Стул преподавателя – 1 шт.

Стол на одно посадочное место – 1 шт.

Доска меловая – 1 шт.

Полка навесная – 3 шт.

Панель с 7 крючками – 3 шт.

Технические средства обучения:

Телевизор – 1 шт.

Компьютерная техника:

Компьютер с необходимым комплектом лицензионного программного обеспечения и выходом в интернет – 17 шт.

Оборудование:

Кондиционер – 1 шт.

Помещение для самостоятельной работы обучающихся:

Основное оборудование:

Стол на одно посадочное место – 24 шт.

Стул – 24 шт.

Стол преподавателя – 1 шт.

Стул преподавателя – 1 шт.

Иное оборудование с указанием количества:

Проектор – 1 шт.

Роутер – 1 шт.

Шкаф – 2 шт.

Телевизор – 1 шт.

Технические средства обучения:

Компьютерная техника:

Ноутбук с необходимым комплектом лицензионного программного обеспечения и выходом в интернет – 15 шт.

Иное оборудование (при наличии)

Кондиционер – 1 шт.

**Авторы-составители:**

Ассистент

Кафедра "Вычислительная техника и  
автоматизированные системы управления"

\_\_\_\_\_ Загика К.Н.

Ассистент

Кафедра "Вычислительная техника и  
автоматизированные системы управления"

\_\_\_\_\_ Щербакова К.С.