Приложение 10
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»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по учебной дисциплине

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.

Фонд оценочных средств учебной дисциплины ОП.06 Основы алгоритмизации и программирования разработан 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);
- Рабочей программы учебной дисциплины ОП.06 Основы алгоритмизации и программирования с учетом:
  - Примерной образовательной программы «Профессионалитет» по специальности 09.02.11 Разработка и управление программным обеспечением, утвержденной протоколом ФУМО в системе СПО по УГПС 09.00.00 Информатика и вычислительная техника от 29.08.2025 № 6/2025, зарегистрированной 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. ПАСПОРТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ 3
УЧЕБНОЙ ДИСЦИПЛИНЫ 3

2. ОСНОВНЫЕ ПОКАЗАТЕЛИ ОЦЕНКИ РЕЗУЛЬТАТОВ 5

3. РАСПРЕДЕЛЕНИЕ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ВИДАМ КОНТРОЛЯ И ЭЛЕМЕНТАМ ЗНАНИЙ И УМЕНИЙ 6

4.1. Вопросы для устного опроса по темам 9

4.2. Варианты контрольных работ 12

4.3. Задания для промежуточной аттестации в форме экзамена 15

4.3.1. Тестовые задания для контрольной работы по дисциплине 15

4.3.2. Перечень экзаменационных вопросов 20

4.3.3. Варианты практических заданий на экзамен 21

1. ПАСПОРТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «ОП.06 Основы алгоритмизации и программирования».

Контрольно-оценочные средства включают контрольные материалы для проведения текущего и рубежного контроля, а также промежуточной аттестации в форме экзамена.

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.

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.
- Базовые понятия объектно-ориентированного программирования: класс, объект, свойства и методы. Основы работы с текстовыми файлами в Python.

2. ОСНОВНЫЕ ПОКАЗАТЕЛИ ОЦЕНКИ РЕЗУЛЬТАТОВ

В результате текущей аттестации по дисциплине осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций.

Результаты обучения (освоенные умения, усвоенные знания)	**Основные показатели оценки результатов**	**Форма контроля и оценивания**
Умения:		
У 01 Разрабатывать простые алгоритмы для учебных и профессионально ориентированных задач	Алгоритм соответствует условию задачи, представлен в словесной форме, в виде блок-схемы или программы	Проверка выполнения и защита лабораторных и самостоятельных работ. Экзамен по дисциплине
У 02 Работать в среде программирования Python	Создана, сохранена, запущена и отлажена простая программа в выбранной среде разработки	
У 03 Использовать ввод и вывод данных, переменные, арифметические операции и базовые типы данных	Программа корректно получает исходные данные, выполняет вычисления и выводит результат в требуемом виде	
У 04 Применять условные операторы, циклы, строки, списки и функции при решении простых задач	Выбраны подходящие управляющие конструкции и структуры данных; решение работает на тестовых наборах	
У 05 Оформлять код программы в соответствии с базовыми требованиями PEP 8	Код читаем, имеет понятные имена, отступы, комментарии при необходимости и не содержит избыточных конструкций	
У 06 Выполнять проверку, поиск и исправление ошибок в простом программном коде	Найдены и исправлены синтаксические и логические ошибки; результат выполнения соответствует эталону	
У 07 Создавать простые консольные мини-проекты на языке Python, использовать классы, объекты и файлы	Мини-проект имеет рабочую структуру, хранит и обрабатывает данные, использует функции, классы или файлы по заданию	
Знания:		
З 01 Основные понятия алгоритмизации, свойства алгоритмов и этапы решения задач на компьютере	Точно раскрывает понятия алгоритма, исполнителя, блок-схемы, этапов разработки и проверки решения	Устный опрос. Тестирование. Контрольные работы. Экзамен по дисциплине
З 02 Назначение языка Python, среды программирования, компиляторов и интерпретаторов	Объясняет назначение Python, среды разработки, интерпретатора, файла программы и консоли вывода	
З 03 Синтаксис базовых конструкций Python: ввод-вывод, переменные, типы данных, условия и циклы	Правильно определяет назначение операторов, функций ввода-вывода, типов данных, условий и циклов	
З 04 Основные приемы работы со строками, списками, функциями, модулями, множествами и словарями	Выбирает подходящие методы обработки данных, объясняет порядок работы со строками, списками и функциями	
З 05 Базовые понятия объектно-ориентированного программирования и основы работы с текстовыми файлами в Python	Объясняет понятия класса, объекта, атрибута, метода, чтения и записи файлов; применяет их в учебной задаче	

3. РАСПРЕДЕЛЕНИЕ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ВИДАМ КОНТРОЛЯ И ЭЛЕМЕНТАМ ЗНАНИЙ И УМЕНИЙ

Элемент учебной дисциплины	Коды ПК, ОК	Результаты обучения (освоенные умения, усвоенные знания)	Текущий контроль	Рубежный контроль
1	2	3	4	5
Раздел 1. Введение в программирование	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4			
Тема 1.1. Языки программирования	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 01, У 02, З 01, З 02	Устный опрос. Лабораторная работа: знакомство со средой Python	Контрольная работа по разделам 1-2
Раздел 2. Введение в язык Python	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4			
Тема 2.1. Ввод-вывод данных	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 02, У 03, У 05, З 02, З 03	Устный опрос. Лабораторные работы по print(), input(), переменным и целочисленной арифметике	Контрольная работа по разделам 1-2
Раздел 3. Основы программирования на языке Python	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4			
Тема 3.1. Условный оператор	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 01, У 03, У 04, У 06, З 03	Устный опрос. Практические задания на if, elif, else и логические операции	Контрольная работа по теме 3.1
Тема 3.2. Типы данных	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 03, У 04, У 05, З 03	Устный опрос. Практические задания на int, float, str, преобразование типов и модуль math	Контрольная работа по темам 3.1-3.2

Тема 3.3. Циклы for и while	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 01, У 04, У 06, З 03	Устный опрос. Практические задания на for, while, range(), break, continue	Контрольная работа по теме 3.3
Тема 3.4. Строковый тип данных	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 03, У 04, У 05, З 04	Устный опрос. Практические задания на индексацию, срезы, методы строк и форматирование вывода	Контрольная работа по теме 3.4
Тема 3.5. Списки	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 03, У 04, У 06, З 04	Устный опрос. Практические задания на списки, вложенные списки, множества и словари	Контрольная работа по теме 3.5
Тема 3.6. Основы структурного программирования. Функции	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 01, У 04, У 05, У 06, З 04	Устный опрос. Практические задания на функции, параметры, возвращаемые значения и модули	Контрольная работа по теме 3.6
Раздел 4. Основы объектно-ориентированного программирования (ООП)	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4			
Тема 4.1. Основные принципы объектно-ориентированного программирования (ООП)	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 07, З 05	Устный опрос. Практические задания на классы, объекты, атрибуты и методы	Контрольная работа по разделу 4
Тема 4.2. Работа над мини-проектом с использованием ООП	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 02, У 05, У 06, У 07, З 05	Лабораторная работа. Разработка, тестирование и защита мини-проекта	Защита мини-проекта. Экзамен
Консультация			Консультация по выполнению заданий и подготовке к экзамену	

Самостоятельная работа обучающихся	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 01 - У 07, З 01 - З 05	Выполнение самостоятельных заданий по темам дисциплины	Проверка самостоятельных работ
Промежуточная аттестация	ОК 01, ОК 02, ОК 09, ПК 2.2, ПК 2.4	У 01 - У 07, З 01 - З 05	Подготовка к экзамену	Экзамен

*Аудиторная самостоятельная работа включает работу студента в аудитории по заданиям из курса лекций и методических указаний для проведения лабораторных работ по учебной дисциплине ОП.06 «Основы алгоритмизации и программирования».
**Внеаудиторная самостоятельная работа включает работу студента по заданиям методических указаний к выполнению самостоятельной работы по учебной дисциплине ОП.06 «Основы алгоритмизации и программирования».

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

4.1. Вопросы для устного опроса по темам

Вопросы по разделу 1 «Введение в программирование»

Тема 1.1. Языки программирования

1. Дайте классификацию языков программирования.
2. Перечислите основные отличия машинного языка, языка ассемблера и языков высокого уровня.
3. Объясните назначение языка Python и области его применения.
4. Что такое среда программирования? Какие элементы входят в состав рабочей среды Python?
5. Что такое синтаксис и семантика языка программирования?
6. Чем компилятор отличается от интерпретатора?
7. Назовите основные этапы решения задачи на компьютере.
8. Что такое алгоритм? Перечислите свойства алгоритмов.
9. Какими способами можно представить алгоритм?
10. Что такое блок-схема? Какие базовые блоки используются при ее построении?
11. Почему важно соблюдать требования к читаемости программного кода?
12. Что означает оценка сложности простого алгоритма?

Вопросы по разделу 2 «Введение в язык Python»

Тема 2.1. Ввод-вывод данных

13. Для чего предназначена функция print() в Python?
14. Для чего предназначена функция input() в Python?
15. Что такое переменная? Какие правила именования переменных применяются в Python?
16. Как выполнить преобразование введенной строки в целое число?
17. Как выполнить преобразование введенной строки в вещественное число?
18. Назовите основные арифметические операции в Python.
19. Чем отличаются операции /, // и %?
20. Для чего используются параметры sep и end функции print()?
21. Что такое множественное присваивание? Приведите пример.
22. Назовите базовые требования PEP 8 к оформлению кода.
23. Какие ошибки чаще всего возникают при вводе данных пользователем?
24. Как проверить результат выполнения простой программы?

Вопросы по разделу 3 «Основы программирования на языке Python»

Тема 3.1. Условный оператор

25. Для чего используется условный оператор if?
26. Чем отличаются конструкции if, if-else и if-elif-else?
27. Что такое логическое выражение?
28. Назовите операции сравнения в Python.
29. Для чего используются логические операции and, or, not?
30. Что такое цепочка сравнений? Приведите пример.
31. В каких случаях применяются вложенные условные операторы?
32. Как проверить принадлежность числа заданному диапазону?
33. Как организовать выбор одного из нескольких вариантов действий?

34. Какие типичные ошибки возникают при записи условий?

Тема 3.2. Типы данных

35. Назовите основные числовые типы данных Python.
36. Чем отличаются типы int и float?
37. Что такое строковый тип данных str?
38. Для чего выполняется преобразование типов?
39. Как проверить тип значения в Python?
40. Для чего используется модуль math?
41. Какие функции модуля math применяются при простых вычислениях?
42. Как работает оператор in?
43. Что произойдет при попытке выполнить арифметическую операцию над несовместимыми типами?
44. Как выбрать подходящий тип данных для учебной задачи?

Тема 3.3. Циклы for и while

45. Для чего используется цикл for?
46. Для чего используется функция range()?
47. Для чего используется цикл while?
48. Чем цикл for отличается от цикла while?
49. Как организовать подсчет суммы чисел в цикле?
50. Как организовать подсчет количества элементов, удовлетворяющих условию?
51. Для чего используется оператор break?
52. Для чего используется оператор continue?
53. Как работает блок else в циклах?
54. Как выполнить обработку натурального числа по цифрам?
55. Что такое бесконечный цикл и как его избежать?
56. Как проверить корректность программы с циклом на тестовых данных?

Тема 3.4. Строковый тип данных

57. Как получить символ строки по индексу?
58. Что такое срез строки?
59. Как определить длину строки?
60. Назовите основные методы строк Python.
61. Чем отличаются методы lower() и upper()?
62. Для чего используются методы count(), find() и replace()?
63. Как удалить лишние пробелы в начале и конце строки?
64. Как проверить наличие подстроки в строке?
65. Что такое форматирование строк?
66. Как вывести результат программы в заданном формате?

Тема 3.5. Списки

67. Что такое список в Python?
68. Как создать пустой список и список с начальными значениями?
69. Как добавить элемент в список?
70. Как удалить элемент из списка?
71. Как изменить значение элемента списка?
72. Как вывести все элементы списка?
73. Для чего используются методы split() и join()?
74. Что такое списочное выражение?
75. Как выполнить сортировку списка?

76. Что такое вложенный список?
77. Для чего используются множества?
78. Для чего используются словари?

Тема 3.6. Основы структурного программирования. Функции

79. Что такое функция в Python?
80. Чем функция с параметрами отличается от функции без параметров?
81. Для чего используется оператор return?
82. Что такое локальная переменная?
83. Что такое глобальная переменная?
84. Зачем разбивать программу на функции?
85. Что такое модуль Python?
86. Как импортировать стандартный модуль?
87. Как создать простой пользовательский модуль?
88. Что такое рекурсия?
89. В каких случаях рекурсивное решение нужно заменить циклом?
90. Как проверить работу функции на нескольких наборах данных?

Вопросы по разделу 4 «Основы объектно-ориентированного программирования (ООП)»

Тема 4.1. Основные принципы объектно-ориентированного программирования (ООП)

91. Что такое объектно-ориентированное программирование?
92. Что такое класс?
93. Что такое объект?
94. Чем атрибут объекта отличается от метода?
95. Как создать класс в Python?
96. Как создать объект класса?
97. Для чего используется метод __init__()?
98. Что означает параметр self?
99. Что такое инкапсуляция?
100. Что такое наследование?
101. Что такое полиморфизм?
102. В каких случаях удобно применять классы в учебной программе?

Тема 4.2. Работа над мини-проектом с использованием ООП

103. Какие этапы включает разработка мини-проекта?
104. Как сформулировать требования к простой программе?
105. Как разбить мини-проект на функции и классы?
106. Как организовать хранение данных в текстовом файле?
107. Как выполнить чтение данных из файла?
108. Как выполнить запись результатов работы программы в файл?
109. Как подобрать тестовые данные для мини-проекта?
110. Как оформить пользовательский ввод и вывод в консольной программе?
111. Какие ошибки следует проверить перед защитой мини-проекта?
112. Как представить результаты работы мини-проекта при защите?

Критерии оценки:
оценка «отлично» выставляется студенту, если:
- полно раскрыто содержание вопроса;

- материал изложен логично, грамотно, с использованием профессиональной терминологии;

- приведены корректные примеры алгоритмов, фрагментов кода или результатов выполнения программы;

- показано умение применять теоретические положения при выполнении практического задания;

- ответ дан самостоятельно, без наводящих вопросов преподавателя;

оценка «хорошо», если:

- ответ в основном соответствует требованиям оценки «отлично», но содержит один-два несущественных недочета;

- отдельные формулировки требуют уточнения, но не искажают смысл материала;

- практический пример выполнен с небольшими исправлениями по замечанию преподавателя;

оценка «удовлетворительно», если:

- содержание вопроса раскрыто неполно, но показано общее понимание темы;

- допущены ошибки в терминологии или примере программы, исправленные после наводящих вопросов;

- практическое задание выполнено на обязательном уровне сложности;

оценка «неудовлетворительно», если:

- не раскрыто основное содержание программного материала;

- обнаружено незнание базовых понятий, синтаксиса и способов решения учебных задач;

- допущенные ошибки не исправлены после дополнительных вопросов преподавателя.

4.2. Варианты контрольных работ

4.2.1. Варианты контрольной работы по разделам 1-2 «Введение в программирование» и «Введение в язык Python»

Вариант 1

1. Дайте определение алгоритма и перечислите его свойства.
2. Опишите назначение функций print() и input().
3. Напишите программу, которая запрашивает имя пользователя и выводит приветствие.
4. Составьте программу для вычисления периметра прямоугольника.

Вариант 2

1. Назовите способы представления алгоритмов.
2. Объясните отличие компилятора от интерпретатора.
3. Напишите программу, которая вводит два числа и выводит их сумму.
4. Используйте параметры sep и end при выводе трех слов в одну строку.

Вариант 3

1. Перечислите этапы решения задачи на компьютере.
2. Назовите правила именования переменных в Python.
3. Напишите программу для вычисления площади круга по радиусу.
4. Поясните, зачем соблюдать требования PEP 8.

Вариант 4

1. Что такое среда программирования?
2. Чем отличаются операции /, // и %?
3. Напишите программу для обмена значениями двух переменных.

4. Составьте блок-схему линейного алгоритма вычисления средней оценки.

Вариант 5
1. Что такое синтаксис и семантика языка?
2. Для чего используется преобразование типов?
3. Напишите программу, которая вводит количество минут и выводит часы и минуты.
4. Приведите пример множественного присваивания.

Вариант 6
1. Назовите области применения Python.
2. Что такое переменная?
3. Напишите программу для вычисления стоимости покупки по цене и количеству.
4. Опишите типичные ошибки при вводе числовых данных.

4.2.2. Варианты контрольной работы по теме 3.1 «Условный оператор»

Вариант 1
1. Объясните назначение конструкции if-else.
2. Напишите программу, которая определяет, является ли число четным.
3. Напишите программу выбора большего из двух чисел.

Вариант 2
1. Что такое логическое выражение?
2. Напишите программу, которая проверяет принадлежность числа диапазону от 10 до 99.
3. Составьте программу определения положительного, отрицательного или нулевого числа.

Вариант 3
1. Для чего используются операции and, or, not?
2. Напишите программу проверки возможности существования треугольника по трем сторонам.
3. Составьте программу, которая определяет сезон по номеру месяца.

Вариант 4
1. Что такое вложенный условный оператор?
2. Напишите программу, которая определяет максимальное из трех чисел.
3. Составьте программу расчета скидки по сумме покупки.

Вариант 5
1. Чем отличается if-elif-else от нескольких независимых if?
2. Напишите программу, которая проверяет пароль по заданным условиям.
3. Составьте программу определения оценки по количеству баллов.

Вариант 6
1. Что такое цепочка сравнений?
2. Напишите программу, которая проверяет, является ли год високосным.
3. Составьте программу проверки корректности введенного возраста.

4.2.3. Варианты контрольной работы по теме 3.3 «Циклы for и while»

Вариант 1
1. Объясните отличие циклов for и while.
2. Напишите программу вывода чисел от 1 до n.
3. Найдите сумму чисел от 1 до n.

Вариант 2
1. Для чего используется функция range()?
2. Напишите программу подсчета количества четных чисел в диапазоне.
3. Найдите произведение цифр натурального числа.

Вариант 3
1. Для чего используются break и continue?
2. Напишите программу поиска первого числа, кратного 7, в заданном диапазоне.
3. Составьте программу проверки числа на простоту.

Вариант 4
1. Что такое бесконечный цикл?
2. Напишите программу, которая запрашивает числа до ввода нуля и выводит их сумму.
3. Составьте таблицу квадратов чисел от 1 до 10.

Вариант 5
1. Как работает блок else в цикле?
2. Напишите программу поиска максимальной цифры натурального числа.
3. Составьте программу подсчета количества цифр числа.

Вариант 6
1. Какие ошибки характерны для программ с циклами?
2. Напишите программу вывода всех делителей числа.
3. Составьте программу вычисления факториала числа.

4.2.4. Варианты контрольной работы по темам 3.4-3.5 «Строки и списки»

Вариант 1
1. Что такое индекс строки?
2. Напишите программу подсчета количества гласных букв в строке.
3. Создайте список чисел и найдите его максимальный элемент.

Вариант 2
1. Что такое срез строки?
2. Напишите программу, которая проверяет, является ли строка палиндромом.
3. Создайте список и выведите только положительные элементы.

Вариант 3
1. Назовите основные методы строк.
2. Напишите программу замены пробелов в строке на символ подчеркивания.
3. Отсортируйте список чисел по возрастанию.

Вариант 4
1. Для чего используются split() и join()?
2. Напишите программу, которая подсчитывает количество слов в строке.
3. Создайте список квадратов чисел от 1 до n.

Вариант 5
1. Что такое вложенный список?
2. Напишите программу обработки строки с помощью lower() и strip().
3. Составьте программу поиска среднего арифметического элементов списка.

Вариант 6
1. Для чего используются множества и словари?

2. Напишите программу подсчета частоты символов в строке.
3. Создайте словарь товаров и выведите общую стоимость покупки.

4.2.5. Варианты контрольной работы по темам 3.6 и 4.1-4.2 «Функции, файлы и ООП»

Вариант 1
1. Что такое функция в Python?
2. Напишите функцию, возвращающую большее из двух чисел.
3. Создайте класс Student с полями имя и оценка.

Вариант 2
1. Для чего используется return?
2. Напишите функцию проверки числа на четность.
3. Напишите программу записи списка строк в текстовый файл.

Вариант 3
1. Что такое локальная переменная?
2. Напишите функцию подсчета суммы элементов списка.
3. Создайте класс Rectangle с методами вычисления площади и периметра.

Вариант 4
1. Что такое модуль Python?
2. Напишите функцию, которая принимает строку и возвращает количество слов.
3. Напишите программу чтения чисел из файла и подсчета их суммы.

Вариант 5
1. Что такое класс и объект?
2. Напишите функцию сортировки списка без изменения исходного списка.
3. Создайте мини-проект «Телефонный справочник» с хранением данных в файле.

Вариант 6
1. Для чего используется метод __init__()?
2. Напишите функцию проверки строки на палиндром.
3. Создайте класс Product и программу подсчета общей стоимости товаров.

4.3. Задания для промежуточной аттестации в форме экзамена

Критерии оценки:

Процент выполнения	Оценка
86 – 100 %	отлично
61 – 85 %	хорошо
41 – 60 %	удовлетворительно
0 – 40 %	неудовлетворительно

4.3.1. Тестовые задания для контрольной работы по дисциплине

Вопрос 1. Что выведет команда print('A', 'B', sep='-')?
Ответы:
1) A B
2) A-B
3) AB
4) A, B

Эталон ответа: 2

Вопрос 2. Какой тип данных возвращает функция input()?
Ответы:
1) int
2) float
3) str
4) bool

Эталон ответа: 3

Вопрос 3. Как записывается однострочный комментарий в Python?
Ответы:
1) // комментарий
2) # комментарий
3) /* комментарий */
4) <!-- комментарий -->

Эталон ответа: 2

Вопрос 4. Какой оператор используется для целочисленного деления?
Ответы:
1) /
2) //
3) %
4) **

Эталон ответа: 2

Вопрос 5. Какой оператор возвращает остаток от деления?
Ответы:
1) /
2) //
3) %
4) **

Эталон ответа: 3

Вопрос 6. Как правильно преобразовать строку s в целое число?
Ответы:
1) str(s)
2) int(s)
3) float(s)
4) input(s)

Эталон ответа: 2

Вопрос 7. Какая конструкция используется для выбора из нескольких вариантов?
Ответы:
1) for
2) while
3) if-elif-else
4) def

Эталон ответа: 3

Вопрос 8. Какая операция означает логическое И?
Ответы:
1) or
2) and
3) not
4) in

Эталон ответа: 2

Вопрос 9. Что выведет выражение 5 > 3 and 2 > 4?
Ответы:
1) True
2) False
3) 0
4) Ошибка

Эталон ответа: 2

Вопрос 10. Для чего используется функция range()?
Ответы:
1) для создания диапазона значений
2) для преобразования строки
3) для вывода данных
4) для чтения файла

Эталон ответа: 1

Вопрос 11. Какой цикл чаще применяется, когда известно количество повторений?
Ответы:
1) while
2) for
3) if
4) try

Эталон ответа: 2

Вопрос 12. Какой оператор досрочно завершает цикл?
Ответы:
1) continue
2) pass
3) break
4) return

Эталон ответа: 3

Вопрос 13. Какой оператор пропускает текущую итерацию цикла?
Ответы:
1) continue
2) break
3) pass
4) elif

Эталон ответа: 1

Вопрос 14. Как получить длину строки s?
Ответы:
1) size(s)
2) len(s)
3) length(s)
4) count(s)

Эталон ответа: 2

Вопрос 15. Как получить первый символ строки s?
Ответы:
1) s[1]
2) s[0]
3) s[first]
4) s[-0]

Эталон ответа: 2

Вопрос 16. Что делает метод strip()?
Ответы:
1) сортирует строку
2) удаляет пробелы в начале и конце
3) заменяет символы
4) делит строку на части

Эталон ответа: 2

Вопрос 17. Какой метод строки выполняет замену подстроки?
Ответы:
1) find()
2) count()
3) replace()
4) join()

Эталон ответа: 3

Вопрос 18. Как создать пустой список?

Ответы:
1) {}
2) []
3) ()
4) list{}

Эталон ответа: 2

Вопрос 19. Как добавить элемент x в конец списка a?
Ответы:
1) a.add(x)
2) a.append(x)
3) append(a, x)
4) a.push(x)

Эталон ответа: 2

Вопрос 20. Как отсортировать список a по возрастанию на месте?
Ответы:
1) sorted(a)
2) a.sort()
3) sort(a)
4) a.order()

Эталон ответа: 2

Вопрос 21. Для чего используется split()?
Ответы:
1) для объединения списка
2) для разделения строки на список
3) для сортировки
4) для поиска максимума

Эталон ответа: 2

Вопрос 22. Как объявить функцию в Python?
Ответы:
1) function f():
2) def f():
3) func f():
4) procedure f():

Эталон ответа: 2

Вопрос 23. Как вернуть значение из функции?
Ответы:
1) print
2) return
3) yield only
4) break

Эталон ответа: 2

Вопрос 24. Что такое локальная переменная?
Ответы:
1) переменная, доступная только внутри функции
2) переменная файла
3) переменная класса
4) любая строковая переменная

Эталон ответа: 1

Вопрос 25. Как подключить модуль math?
Ответы:
1) use math
2) include math
3) import math
4) open math

Эталон ответа: 3

Вопрос 26. Как открыть файл для чтения?
Ответы:

1) open('file.txt', 'r')
2) open('file.txt', 'w')
3) read('file.txt')
4) file.open('r')

Эталон ответа: 1

Вопрос 27. Что означает режим 'w' при открытии файла?
Ответы:
1) чтение
2) запись с перезаписью
3) добавление
4) проверка

Эталон ответа: 2

Вопрос 28. Как объявить класс в Python?
Ответы:
1) class Student:
2) def Student:
3) object Student:
4) new Student:

Эталон ответа: 1

Вопрос 29. Для чего используется метод __init__()?
Ответы:
1) для вывода текста
2) для инициализации объекта
3) для сортировки списка
4) для завершения программы

Эталон ответа: 2

Вопрос 30. Что такое объект?
Ответы:
1) экземпляр класса
2) только переменная
3) только функция
4) имя файла

Эталон ответа: 1

Вопрос 31. Какой стиль оформления кода рекомендуется для Python?
Ответы:
1) PEP 8
2) HTML5
3) SQL-92
4) правильных ответов нет

Эталон ответа: 1

Вопрос 32. Что необходимо сделать при отладке программы?
Ответы:
1) проверить программу на тестовых данных
2) удалить все условия
3) заменить все переменные
4) выключить интерпретатор

Эталон ответа: 1

Вопрос 33. Какая структура данных хранит пары ключ-значение?
Ответы:
1) список
2) строка
3) словарь
4) кортеж без ключей

Эталон ответа: 3

Вопрос 34. Какая структура данных хранит уникальные элементы?
Ответы:
1) список

2) множество
3) строка
4) файл

Эталон ответа: 2

Вопрос 35. Какой результат у выражения 'py' in 'python'?
Ответы:
1) True
2) False
3) 0
4) Ошибка

Эталон ответа: 1

Вопрос 36. Какой оператор возводит число в степень?
Ответы:
1) ^
2) **
3) pow=
4) //

Эталон ответа: 2

Вопрос 37. Что делает функция len([1, 2, 3])?
Ответы:
1) возвращает сумму
2) возвращает количество элементов
3) сортирует список
4) удаляет элемент

Эталон ответа: 2

Вопрос 38. Какой метод списков удаляет элемент по значению?
Ответы:
1) remove()
2) delete()
3) clear(value)
4) strip()

Эталон ответа: 1

Вопрос 39. Что проверяется при защите мини-проекта?
Ответы:
1) только название файла
2) работоспособность, структура кода и соответствие заданию
3) только объем текста
4) только наличие комментариев

Эталон ответа: 2

Вопрос 40. Какой результат считается корректным при тестировании?
Ответы:
1) совпадающий с ожидаемым для заданных входных данных
2) любой вывод программы
3) ошибка интерпретатора
4) пустой экран

Эталон ответа: 1

4.3.2. Перечень экзаменационных вопросов

1. Понятие алгоритма. Свойства алгоритмов. Способы представления алгоритмов.
2. Этапы решения задачи на компьютере. Постановка задачи и формализация.
3. Блок-схемы алгоритмов. Основные графические элементы блок-схем.
4. Классификация языков программирования. Назначение языка Python.
5. Среда программирования Python. Создание, сохранение и запуск программы.
6. Интерпретатор Python. Отличие интерпретатора от компилятора.
7. Правила именования переменных. Базовые требования PEP 8.

8. Функции print() и input(). Параметры sep и end.
9. Типы данных int, float, str. Преобразование типов.
10. Арифметические операции в Python. Операции /, //, %, **.
11. Логические выражения и операции сравнения.
12. Условный оператор if. Конструкции if-else и if-elif-else.
13. Логические операции and, or, not. Цепочки сравнений.
14. Вложенные условные операторы. Типичные ошибки при записи условий.
15. Цикл for и функция range(). Примеры применения.
16. Цикл while. Условия завершения цикла.
17. Операторы break, continue и блок else в циклах.
18. Обработка натурального числа по цифрам.
19. Строковый тип данных. Индексация строк.
20. Срезы строк. Определение длины строки.
21. Методы строк lower(), upper(), count(), find(), replace(), strip().
22. Форматирование строк и вывод результатов программы.
23. Списки в Python. Создание, изменение, добавление и удаление элементов.
24. Методы списков. Сортировка и поиск элементов.
25. Методы split() и join(). Преобразование строки в список и списка в строку.
26. Списочные выражения. Примеры использования.
27. Вложенные списки. Простейшая обработка двумерных данных.
28. Множества и словари в Python. Области применения.
29. Функции в Python. Параметры и возвращаемые значения.
30. Локальные и глобальные переменные. Структурное программирование.
31. Модули Python. Импорт стандартных модулей и создание пользовательского модуля.
32. Понятие рекурсии. Простейшие рекурсивные функции.
33. Классы и объекты в Python. Атрибуты и методы.
34. Метод __init__() и параметр self. Создание объектов класса.
35. Основы работы с текстовыми файлами. Чтение и запись данных.
36. Отладка и тестирование простых программ. Подбор тестовых данных.
37. Разработка мини-проекта на Python. Структура, проверка и защита.

4.3.3. Варианты практических заданий на экзамен

1. Составить алгоритм и программу на Python для вычисления площади и периметра прямоугольника по введенным сторонам.
2. Составить программу, которая вводит два числа и выводит их сумму, разность, произведение и частное.
3. Составить программу перевода количества минут в часы и минуты.
4. Составить программу вычисления стоимости покупки по цене товара, количеству и проценту скидки.
5. Составить программу, которая определяет, является ли введенное число четным.
6. Составить программу выбора наибольшего из трех введенных чисел.
7. Составить программу проверки возможности существования треугольника по трем сторонам.
8. Составить программу определения оценки по количеству набранных баллов.
9. Составить программу проверки, принадлежит ли число заданному диапазону.
10. Составить программу определения високосного года.
11. Составить программу вывода всех чисел от 1 до n.
12. Составить программу вычисления суммы чисел от 1 до n.

13. Составить программу вычисления факториала числа.

14. Составить программу подсчета количества четных чисел в диапазоне от a до b.

15. Составить программу поиска всех делителей натурального числа.

16. Составить программу проверки числа на простоту.

17. Составить программу обработки натурального числа по цифрам: найти сумму и количество цифр.

18. Составить программу поиска максимальной цифры натурального числа.

19. Составить программу, которая запрашивает числа до ввода нуля и выводит сумму введенных чисел.

20. Составить программу вывода таблицы квадратов чисел от 1 до n.

21. Составить программу подсчета количества гласных букв в строке.

22. Составить программу проверки строки на палиндром.

23. Составить программу замены всех пробелов в строке на символ подчеркивания.

24. Составить программу подсчета количества слов в строке.

25. Составить программу форматирования введенных фамилии, имени и отчества в одну строку по образцу.

26. Составить программу создания списка чисел и поиска максимального и минимального элементов.

27. Составить программу подсчета суммы положительных элементов списка.

28. Составить программу сортировки списка чисел и вывода результата.

29. Составить программу, которая удаляет из списка все отрицательные элементы.

30. Составить программу преобразования строки чисел в список и вычисления среднего арифметического.

31. Составить программу обработки вложенного списка: найти сумму элементов каждой строки.

32. Составить программу подсчета количества уникальных слов в строке с использованием множества.

33. Составить программу создания словаря товаров и подсчета общей стоимости покупки.

34. Написать функцию, которая принимает два числа и возвращает большее из них.

35. Написать функцию, которая принимает список чисел и возвращает сумму его элементов.

36. Написать функцию проверки строки на палиндром и использовать ее в основной программе.

37. Создать модуль с двумя функциями для вычисления площади прямоугольника и круга, подключить его в основной программе.

38. Составить программу чтения чисел из текстового файла и записи их суммы в другой файл.

39. Создать класс Student с полями имя и оценка, методами вывода информации и проверки успешной сдачи.

40. Разработать мини-проект «Телефонный справочник»: добавление, поиск и сохранение записей в текстовом файле.

41. Разработать мини-проект «Калькулятор расходов»: ввод операций, подсчет общей суммы и сохранение отчета в файл.