

МИНИСТЕРСТВО ОБРАЗОВАНИЯ КРАСНОЯРСКОГО КРАЯ

**КРАЕВОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«КРАСНОЯРСКИЙ КОЛЛЕДЖ ОТРАСЛЕВЫХ ТЕХНОЛОГИЙ
И ПРЕДПРИНИМАТЕЛЬСТВА»**

РАССМОТРЕНО
методической комиссией
протокол № 10 от 18.06.2026

УТВЕРЖДЕНО
Директор КГАПОУ «ККОТиП»
_____/Н. В. Журова/
Приказ № 01-55-1п от 29.06.2026

**ПРОГРАММА ПОДГОТОВКИ
СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

09.02.07 Информационные системы и программирование

на базе основного общего образования

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

ОП.07 Основы алгоритмизации и программирования

Красноярск 2026

СОСТАВ КОМПЛЕКТА

1. Паспорт фонда оценочных средств
2. Формы контроля и оценивания элементов учебной дисциплины
3. Результаты освоения дисциплины
 - 3.1 Профессиональные компетенции, подлежащие проверке при выполнении задания
 - 3.2 Общие компетенции, подлежащие проверке при выполнении задания
 - 3.3 Основные показатели оценки результатов
4. Оценка освоения курса учебной дисциплины
 - 4.1. Задания для текущего контроля освоения учебной дисциплины
 - 4.2. Задания для проведения промежуточной аттестации по учебной дисциплине

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Общие положения

Контрольно-измерительные материалы предназначены для проверки результатов освоения учебной дисциплины **ОП.07 Основы алгоритмизации и программирования** основной образовательной программы среднего профессионального образования по специальности **09.02.12 Техническая эксплуатация и сопровождение информационных систем**.

Контрольно-измерительные материалы предназначены для текущего и промежуточного контроля, оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины ОП.07 Основы алгоритмизации и программирования.

Формой промежуточной аттестации по учебной дисциплине является дифференцированный зачет, который оценивается по пятибалльной шкале оценок.

2. ФОРМЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ ЭЛЕМЕНТОВ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контролируемые темы (разделы) учебной дисциплины	Форма контроля и оценивания	
	Текущий контроль	Промежуточная аттестация
Раздел 1. Алгоритмизация и основы программирования	Устный опрос	Промежуточная аттестация в форме экзамен
Раздел 2. Программирование		

3. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ

3.1. Профессиональные компетенции, подлежащие проверке при выполнении задания

В результате текущего контроля и оценки результатов освоения умений и знаний по учебной дисциплине ОП.07 Основы алгоритмизации и программирования осуществляется комплексная проверка следующих профессиональных компетенций:

Профессиональные компетенции	Показатели оценки результата
ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	грамотность планирования и проведения необходимых тестовых проверок и профилактических осмотров
ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Обеспечение наличия программно-технических средств сбора данных для анализа показателей использования и функционирования компьютерной сети

3.2. Общие компетенции, подлежащие проверке при выполнении задания

В результате текущего контроля и оценки по учебной дисциплине ОП.07 Основы алгоритмизации и программирования осуществляется комплексная проверка следующих общих компетенций:

Код	Общие компетенции	Показатели оценки результата
-----	-------------------	------------------------------

ОК.01	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	активность студентов при проведении учебно-воспитательных мероприятий профессиональной направленности.
ОК.02	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество	обоснование выбора и применения методов и способов решения профессиональных задач в области подготовки и организации технологических процессов на предприятиях.
ОК.04	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития	оперативность поиска и использования необходимой информации для качественного выполнения профессиональных задач, профессионального и личностного развития.
ОК.05	Использовать информационно-коммуникационные технологии в профессиональной деятельности.	оперативность, точность и широта подготовки и организации технологических процессов с использованием общего и специализированного программного обеспечения.
ОК.09	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	проявление интереса к инновациям в области профессиональной деятельности, участие в проектной, конкурсной деятельности.

3.3. Основные показатели оценки результатов

Перечень основных показателей оценки результатов знаний и умений, подлежащих текущему контролю и промежуточной аттестации

Результаты обучения: умения, знания	Показатели оценки результата
Умения	
разрабатывать алгоритмы для конкретных задач.	- оптимально подобрано ПО в соответствии с заданием и конкретной задачи в соответствии с вариантом задания;
использовать программы для графического отображения алгоритмов.	
определять сложность работы алгоритмов.	- оптимально подобрана комплектация основных устройств ПК в соответствии с их характеристиками для решения конкретной задачи в соответствии с вариантом задания;
работать в среде программирования.	
реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.	
оформлять код программы в соответствии со стандартом кодирования.	- оптимально подобраны комплектующие для системного блока в соответствии с их характеристиками для решения конкретной задачи в соответствии с вариантом задания;
выполнять проверку, отладку кода программы	

Знания	
понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.	- правильно подобраны периферийные устройства в соответствии с их характеристиками для решения конкретной задачи в соответствии с вариантом задания;
эволюцию языков программирования, их классификацию, понятие системы программирования.	
основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.	- комплектующие ПК не имеют конфликтов друг с другом в соответствии с их характеристиками для решения конкретной задачи в соответствии с вариантом задания;
подпрограммы, составление библиотек подпрограмм.	
объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.	- верное обоснование выполненной практической части с учетом архитектурных особенностей цифровых вычислительных систем, а так же с указанием классификации вычислительных платформ, принципов вычислений и повышения производительности в многопроцессорных системах и включая в себя принципы устройства периферийного оборудования

4. ОЦЕНКА ОСВОЕНИЯ КУРСА УЧЕБНОЙ ДИСЦИПЛИНЫ

Наименование контроля	Тема	Форма контроля
Текущий контроль	Раздел 1. Алгоритмизация и основы программирования	Устный опрос
	Раздел 2. Программирование	
Промежуточная аттестация		Экзамен

4.2. Задания для проведения промежуточной аттестации по учебной дисциплине ОП.07 Основы алгоритмизации и программирования

4.2.1. Тестовая часть (общее количество вопросов – 100)

Распределение по типам:

- Тип 1 (один правильный ответ) – 25 вопросов (№ 1–25).
- Тип 2 (несколько правильных ответов) – 25 вопросов (№ 26–50).
- Тип 3 (указание последовательности) – 15 вопросов (№ 51–65).
- Тип 4 (установление соответствия) – 15 вопросов (№ 66–80).
- Тип (открытый ответ) – 20 вопросов (№ 81–100).

5

Тип 1. Выбор одного правильного ответа (№ 1–25)

- Какой оператор используется для вывода данных в Python?
 - input()

- b) `print()`
 - c) `output()`
 - d) `write()`
2. Какая функция возвращает длину списка?
- a) `size()`
 - b) `len()`
 - c) `length()`
 - d) `count()`
3. Какой тип данных используется для хранения целых чисел?
- a) `str`
 - b) `float`
 - c) `int`
 - d) `bool`
4. Какой цикл выполняется, пока условие истинно?
- a) `for`
 - b) `repeat`
 - c) `while`
 - d) `loop`
5. Что выведет следующий код? `print(type(3.14))`
- a) `<class 'int'>`
 - b) `<class 'float'>`
 - c) `<class 'str'>`
 - d) `<class 'complex'>`
6. Какое ключевое слово используется для определения функции в Python?
- a) `function`
 - b) `def`
 - c) `func`
 - d) `define`
7. Каким символом обозначается комментарий в Python?
- a) `//`
 - b) `/*`
 - c) `#`
 - d) `--`
8. Какой метод добавляет элемент в конец списка?
- a) `add()`
 - b) `append()`
 - c) `insert()`
 - d) `push()`
9. Какая структура данных хранит пары «ключ-значение»?
- a) `list`
 - b) `tuple`
 - c) `set`
 - d) `dict`
10. Что делает оператор `**` в Python?
- a) побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ
 - b) умножение
 - c) возведение в степень
 - d) конкатенация строк
11. Какой результат у выражения `15 // 4`?
- a) 3.75
 - b) 3
 - c) 4

- d) 3.0
- 12. Какая функция используется для преобразования строки в целое число? a) `str()`
 - b) `float()`
 - c) `int()`
 - d) `bool()`
- 13. Какой результат у выражения `bool(0)`?
 - a) `True`
 - b) `False`
 - c) `0`
 - d) `None`
- 14. Какой оператор используется для проверки принадлежности элемента коллекции? a) `in`
 - b) `contains`
 - c) `like`
 - d) `within`
- 15. Какой тип исключения возникает при делении на ноль?
 - a) `ValueError`
 - b) `IndexError`
 - c) `ZeroDivisionError`
 - d) `TypeError`
- 16. Какой метод удаляет последний элемент списка?
 - a) `remove()`
 - b) `delete()`
 - c) `pop()`
 - d) `discard()`
- 17. Какое значение вернет `len("Привет")`?
 - a) 5
 - b) 6
 - c) 7
 - d) 12
- 18. Какая функция генерирует последовательность чисел для цикла `for`?
 - a) `range()`
 - b) `xrange()`
 - c) `sequence()`
 - d) `iterate()`
- 19. Что делает метод `strip()` для строки?
 - a) удаляет все пробелы внутри строки
 - b) удаляет пробелы в начале и конце строки
 - c) заменяет пробелы на подчеркивания
 - d) разбивает строку на слова
- 20. Какой символ используется для индексации с конца (отрицательная индексация) в списках?
 - a) `!`
 - b) `^`
 - c) `-`
 - d) `$`
- 21. Что выведет код: `print(2 ** 3 ** 2)`?
 - a) 64
 - b) 512
 - c) 256
 - d) 128
- 22. Какой оператор используется для сравнения «не равно»? a) `<>`

- b) !=
 - c) !==
 - d) /=
23. Какая функция возвращает модуль числа?
- a) mod()
 - b) abs()
 - c) fabs()
 - d) math.mod()
24. Какой результат у выражения "a" * 3?
- a) "aaa"
 - b) "a a a"
 - c) Ошибка
 - d) "a3"
25. Какой метод используется для получения ключей словаря?
- a) keys()
 - b) values()
 - c) items()
 - d) get()

Тип 2. Выбор нескольких правильных ответов (№ 26–50)

26. Какие из следующих типов данных являются неизменяемыми (immutable) в Python? a) list
- b) tuple
 - c) dict
 - d) str
27. Какие способы импорта модуля math являются корректными?
- a) import math
 - b) include math
 - c) from math import sqrt
 - d) use math
28. Какие методы используются для добавления элемента в список?
- a) append()
 - b) add()
 - c) insert()
 - d) extend()
29. Какие из перечисленных структур данных поддерживают хранение пар «ключ–значение»?
- a) list
 - b) tuple
 - c) dict
 - d) set
30. Какие операторы сравнения существуют в Python?
- a) ==
 - b) <>
 - c) !=
 - d) =
31. Какие из следующих значений являются ложными (False) в логическом контексте? a) 0
- b) ""
 - c) None
 - d) []

32. Какие методы можно использовать для удаления элемента из списка по значению? a) `remove()`
b) `pop()`
c) `delete()`
d) `discard()`
33. Какие из перечисленных типов данных являются упорядоченными? a) `list`
b) `tuple`
c) `dict` (Python 3.7+)
d) `set`
34. Какие операции допустимы для строк?
a) `+` (конкатенация)
b) `*` (повторение)
c) `-` (вычитание)
d) `/` (деление)
35. Какие из следующих функций являются встроенными в Python? a) `len()`
b) `print()`
c) `max()`
d) `sort()`
36. Какие ключевые слова используются для обработки исключений? a) `try`
b) `except`
c) `catch`
d) `finally`
37. Какие методы позволяют получить элемент из словаря, не вызывая ошибку, если ключ отсутствует?
a) `get()`
b) `pop()`
c) `setdefault()`
d) `getitem`
38. Какие из перечисленных действий создают новую строку?
a) `s = "hello".upper()`
b) `s += " world"`
c) `s[0] = 'H'`
d) `s.replace('l', 'L')`
39. Какие методы используются для работы с файлами?
a) `open()`
b) `read()`
c) `write()`
d) `close()`
40. Какие аргументы может принимать функция `range()`?
a) `start`
b) `stop`
c) `step`
d) `length`
41. Какие из следующих выражений являются генераторами списков?
a) `[x2 for x in range(10)]`
b) `(x2 for x in range(10))`
c) `list(map(lambda x: x2, range(10)))`
d) `{x: x2 for x in range(10)}`
42. Какие ключевые слова используются в определении класса?
a) `class`
b) `def`
c) `init`

- d) `self`
43. Какие из перечисленных методов являются «магическими» (dunder-методами)? a) `init`
b) `str`
c) `add`
d) `call`
44. Какие операторы используются для проверки тождественности объектов? a) `is`
b) `==`
c) `is not`
d) `!=`
45. Какие из следующих вариантов являются способами создания множества (set)?
a) `{1, 2, 3}`
b) `set([1, 2, 3])`
c) `(1, 2, 3)`
d) `{1:1, 2:2}`
46. Какие функции позволяют получить случайное число?
a) `random.random()`
b) `random.randint()`
c) `random.choice()`
d) `random.sample()`
47. Какие из следующих типов данных являются итерируемыми? a) `list`
b) `str`
c) `int`
d) `dict`
48. Какие операторы могут использоваться для работы с множествами?
a) `|` (объединение)
b) `&` (пересечение)
c) `-` (разность)
d) `^` (симметрическая разность)
49. Какие способы позволяют скопировать список?
a) `new_list = old_list[:]`
b) `new_list = list(old_list)`
c) `new_list = old_list.copy()`
d) `new_list = old_list`
50. Какие методы строки позволяют разбить строку на части?
a) `split()`
b) `partition()`
c) `join()`
d) `splitlines()`

Тип 3. Указание правильной последовательности (№ 51–65)

51. Установите правильный порядок операций для открытия файла, чтения строки и закрытия файла (с использованием менеджера контекста):
A) `file = open("data.txt", "r")`
B) `line = file.readline()`
C) `file.close()`
D) `with open("data.txt", "r") as file: line = file.readline()`
52. Расположите этапы решения задачи программирования в правильном порядке:
A) Написание кода
B) Тестирование
C) Построение алгоритма
D) Анализ условия

53. Установите порядок выполнения операций в выражении: `result = 2 + 3 * 4 ** 2`
- A) умножение
 - B) возведение в степень
 - C) сложение
54. Расположите в правильной последовательности шаги для создания класса и его использования:
- A) Создание объекта
 - B) Определение метода
 - C) Вызов метода
 - D) Определение класса
55. Установите порядок обработки исключений в конструкции `try-except-else-finally`:
- A) `except`
 - B) `else`
 - C) `try`
 - D) `finally`
56. Расположите стадии жизненного цикла программы на Python:
- A) Выполнение (runtime)
 - B) Написание исходного кода
 - C) Интерпретация байт-кода
 - D) Компиляция в байт-код
57. Установите порядок преобразования типов для безопасного ввода целого числа:
- A) вызов `int()`
 - B) вызов `input()`
 - C) обработка исключения `ValueError`
 - D) присваивание переменной
58. Расположите в правильном порядке шаги для создания и использования модуля:
- A) импорт модуля
 - B) вызов функции из модуля
 - C) создание файла с расширением `.py`
 - D) определение функций в модуле
59. Установите порядок работы с контекстным менеджером при записи в файл:
- A) выполнение блока `with`
 - B) автоматический вызов `exit`
 - C) вызов `enter`
 - D) открытие файла
60. Расположите этапы поиска ошибки (отладки) в программе:
- A) локализация места ошибки
 - B) исправление кода
 - C) воспроизведение ошибки
 - D) повторное тестирование
61. Установите последовательность, в которой Python вычисляет логическое выражение `not a and b or c`:
- A) `and`
 - B) `not`
 - C) `or`
62. Расположите в правильном порядке действия при создании виртуального окружения:
- A) активация окружения
 - B) установка пакетов
 - C) создание окружения
 - D) выход из окружения

63. Установите порядок наследования в множественном наследовании (поиск метода):
- родительский класс, указанный первым
 - текущий класс
 - родительский класс, указанный вторым
 - object
64. Расположите этапы разбора строки (парсинга) с помощью регулярных выражений:
- применение шаблона
 - импорт модуля re
 - получение результатов
 - компиляция шаблона
65. Установите порядок действий при использовании Git (системы контроля версий) для проекта:
- commit
 - add
 - clone
 - push

Тип 4. Установление соответствия (№ 66–80)

66. Установите соответствие между типом данных и его описанием:

Тип данных	Описание
1. list	А. Неизменяемая упорядоченная коллекция
2. tuple	В. Изменяемая упорядоченная коллекция
3. dict	С. Коллекция уникальных элементов
4. set	Д. Коллекция пар «ключ–значение»

67. Установите соответствие между методом строки и его действием:

Метод	Действие
1. upper()	А. Преобразует строку в нижний регистр
2. lower()	В. Удаляет пробелы в начале и конце строки
3. strip()	С. Преобразует строку в верхний регистр
4. replace()	Д. Заменяет подстроку на другую

68. Установите соответствие между понятием ООП и его определением:

Понятие	Определение
1. Инкапсуляция	А. Возможность объектов разных классов реагировать на один и тот же метод по-своему
2. Наследование	В. Скрытие внутреннего состояния объекта и предоставление доступа через методы
3. Полиморфизм	С. Создание нового класса на основе существующего

69. Установите соответствие между оператором или функцией и ее результатом (при a=5, b=3):

Оператор	Результат
1. a // b	А. 2
2. a % b	В. 125
3. a ** b	С. 1
4. a / b	Д. 1.666...

70. Установите соответствие между функцией стандартной библиотеки и её назначением:

Функция	Назначение
---------	------------

1. math.sqrt()	А. Генерация случайного числа
2. random.randint()	В. Получение текущей даты и времени
3. datetime.now()	С. Вычисление квадратного корня
4. os.getcwd()	Д. Получение текущей рабочей директории

71. Установите соответствие между оператором цикла и его характеристикой:

Цикл	Характеристика
1. for	А. Использует счетчик (итератор)
2. while	В. Выполняется, пока условие истинно
3. break	С. Прерывает выполнение цикла
4. continue	Д. Переходит к следующей итерации

72. Установите соответствие между функцией и её описанием:

Функция	Описание
1. map()	А. Применяет функцию ко всем элементам последовательности
2. filter()	В. Фильтрует элементы по условию
3. zip()	С. Объединяет элементы нескольких последовательностей
4. enumerate()	Д. Возвращает пары (индекс, значение)

73. Установите соответствие между типом исключения и ситуацией:

Исключение	Ситуация
1. IndexError	А. Обращение к несуществующему атрибуту
2. KeyError	В. Выход за границы списка
3. AttributeError	С. Отсутствие ключа в словаре
4. TypeError	Д. Операция с несовместимыми типами

74. Установите соответствие между методом списка и его действием:

Метод	Действие
1. sort()	А. Переворачивает список
2. reverse()	В. Сортирует список
3. index()	С. Возвращает количество вхождений элемента
4. count()	Д. Возвращает индекс первого вхождения

75. Установите соответствие между способом импорта и синтаксисом:

Способ	Синтаксис
1. Импорт всего модуля	А. from module import func
2. Импорт конкретной функции	В. import module as alias
3. Импорт с псевдонимом	С. import module
4. Импорт всех функций	Д. from module import *

76. Установите соответствие между понятием ООП и примером кода (фрагмент):

Понятие	Пример
1. Наследование	А. class Cat(Animal): pass
2. Полиморфизм	В. def speak(self): print("Meow")
3. Инкапсуляция	С. self.__age = 0
4. Абстракция	Д. class Animal: def speak(self): pass

77. Установите соответствие между операцией и её результатом (для строки s = "Python"):

Операция	Результат
1. s[0]	А. "P"
2. s[-1]	В. "n"
3. s[1:4]	С. "yth"
4. s[::2]	Д. "Pto"

78. Установите соответствие между режимом открытия файла и его описанием:

Режим	Описание
1. 'r'	A. Чтение
2. 'w'	B. Запись (перезапись)
3. 'a'	C. Дозапись (append)
4. 'x'	D. Эксклюзивное создание

79. Установите соответствие между встроенной функцией и её назначением:

Функция	Назначение
1. all()	A. Возвращает True, если все элементы итерации истинны
2. any()	B. Возвращает True, если хотя бы один элемент истинен
3. sum()	C. Возвращает сумму элементов
4. sorted()	D. Возвращает отсортированный список

80. Установите соответствие между понятием и его определением:

Понятие	Определение
1. Модуль	A. Файл с расширением .py, содержащий код
2. Пакет	B. Каталог с файлом init.py
3. Библиотека	C. Набор модулей, объединенных функционально
4. Фреймворк	D. Инфраструктура для разработки приложений

Тип 5. Открытый ответ (№ 81–100)

81. Напишите команду импорта модуля random, чтобы использовать функцию randint.

82. Какой оператор используется для определения бесконечного цикла с условием, всегда истинным?

83. Вставьте пропущенное ключевое слово: def my_function(x):

_____ x + 5

84. Как называется метод, который автоматически вызывается при создании объекта класса?

85. Напишите выражение, которое создает список из чисел от 1 до 10 с помощью генератора списков.

86. Какой метод используется для добавления элемента в конец списка?

87. Что выведет код: print("Hello, world!".upper())

88. Как записать условие: «если переменная a больше 5 и не равна 10»?

89. Как называется принцип ООП, при котором один класс может наследовать свойства и методы другого?

90. Какую функцию нужно вызвать, чтобы получить длину строки?

91. Какой оператор позволяет проверить, является ли объект экземпляром класса?

92. Напишите код, который открывает файл "data.txt" для чтения и выводит его содержимое построчно.

93. Как создать пустое множество в Python?

94. Что выведет print(3 * 'abc')?

95. Какой метод строки позволяет проверить, состоит ли строка только из цифр?

96. Напишите код для создания класса Car с атрибутами brand и year.

97. Как получить список всех ключей словаря d?

98. Какая функция используется для получения случайного элемента из списка?

99. Напишите лямбда-функцию, которая возвращает квадрат числа.

100. Как в Python создать кортеж из одного элемента?

4.2.2. Практические задания (4 варианта) Каждое практическое задание состоит из:

- **Построения алгоритма** (блок-схема или словесное описание / псевдокод).

- **Написания программного кода** на Python, реализующего данный алгоритм.

Вариант 1 (структуры данных и функции)

Задание:

Дана последовательность из N целых чисел (N вводится с клавиатуры, затем вводятся сами числа). Написать программу, которая:

1. Формирует список из введенных чисел.
2. Находит и выводит максимальный элемент списка и его индекс.
3. Выводит сумму всех элементов, кратных 3.
4. Сортирует список по возрастанию и выводит его. *Требования:*
 - Использовать функции для ввода, обработки и вывода.
 - Оформить код с комментариями.
 - Предварительно описать алгоритм (блок-схема или псевдокод).

Вариант 2 (работа со строками и файлами)

Задание:

Написать программу, которая читает текстовый файл input.txt (содержит несколько строк), подсчитывает количество слов в каждой строке, а затем записывает в файл output.txt для каждой строки её номер и количество слов. Слова разделены пробелами. Знаки препинания не учитываются (можно считать, что слова состоят только из букв). *Требования:*

- Реализовать функцию для подсчета слов в строке.
- Обработать возможные ошибки (отсутствие файла).
- Привести алгоритм (блок-схема) решения.

Вариант 3 (объектно-ориентированное программирование)

Задание:

Создайте класс Student, который имеет атрибуты: name (имя), scores (список оценок). Реализуйте методы:

- add_score(score) – добавляет оценку в список.
- average() – возвращает средний балл.
- is_excellent() – возвращает True, если средний балл ≥ 4.5 .

Создайте список из нескольких студентов, заполните их данными, выведите информацию о студентах, имеющих средний балл ≥ 4.5 . *Требования:*

- Описать алгоритм (блок-схема или псевдокод).
- Код должен содержать класс и демонстрацию его работы.

Вариант 4 (циклы и матрицы)

Задание:

Дана квадратная матрица размером $N \times N$ (N вводится). Заполнить её случайными целыми числами от 0 до 9. Найти сумму элементов главной диагонали и количество нулевых элементов в матрице. Вывести матрицу и результаты. *Требования:*

- Использовать генератор списков для создания матрицы.
- Написать функцию для вывода матрицы в удобном виде.
- Привести алгоритм.

4.4 Критерии оценок

Экзамен оценивается по 5-балльной шкале. Итоговая оценка складывается из суммы баллов за тестовую часть и практическое задание.

Шкала оценивания тестовой части

Тестовая часть содержит 20 заданий (случайная выборка из общего количества).

Максимальный балл за тест – 100. Баллы распределены следующим образом:

Тип задания	Количество заданий в билете	Балл за одно задание	Максимальный балл за тип
Тип 1 (один ответ)	5	4	20
Тип 2 (несколько ответов)	5	6	30
Тип 3 (последовательность)	3	8	24
Тип 4 (соответствие)	3	8	24
Тип 5 (открытый ответ)	4	5	20
Итого	20		100

Пояснение:

- Для типов 1 и 5 балл присуждается только при полностью правильном ответе.
- Для типа 2 (несколько ответов) за каждый правильно выбранный вариант начисляется 1,5 балла (максимум 6).
- Для типа 3 (последовательность) ответ считается верным, если полностью совпадает с эталоном (8 баллов).
- Для типа 4 (соответствие) каждое правильное сопоставление оценивается в 2 балла (всего 8).

Критерии перевода баллов теста в оценку:

Баллы	Оценка
85–100	5 (отлично)
70–84	4 (хорошо)
50–69	3 (удовлетворительно)
менее 50	2 (неудовлетворительно)

Шкала оценивания практического задания

Практическое задание оценивается по **20-балльной** шкале.

Критерий	Максимальный балл
Алгоритм (блок-схема или словесное описание)	5
– алгоритм верный, логически полный	5
– алгоритм в основном верный, но есть незначительные недочеты	3–4
– алгоритм содержит ошибки, неполный	1–2
– алгоритм отсутствует	0
Реализация кода	10
– код полностью соответствует алгоритму, работает на всех тестах, соблюдены стандарты оформления (PEP 8)	10
– код работает на основных тестах, но есть недочеты (стиль, обработка краевых случаев)	7–9
– код работает частично, содержит логические ошибки, не проходит некоторые тесты	4–6
– код неработоспособен или не соответствует заданию	0–3
Тестирование и отладка (представлены результаты тестирования, пояснения)	5
– продемонстрирована работа программы на нескольких примерах, объяснены результаты	5
– тестирование выполнено не полностью, но основные случаи проверены	3–4
– тестирование отсутствует или выполнено формально	0–2

Критерии перевода баллов практического задания в оценку:

Баллы	Оценка
18–20	5 (отлично)
14–17	4 (хорошо)
10–13	3 (удовлетворительно)
менее 10	2 (неудовлетворительно)

Итоговая оценка за экзамен

Итоговая оценка выставляется на основе среднего арифметического оценок за тест и практическое задание, округленного до целого по правилам математического округления. Если одна из частей оценена на «2» (неудовлетворительно), итоговая оценка – «2» независимо от другой части.