**ИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Сибирский государственный университет науки и технологий   
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Языки программирования

|  |
| --- |
| Классы в Python |

Руководитель А.В. Проскурин

подпись, дата инициалы, фамилия

Обучающийся БПИ22-02, 221219040 К.В. Трифонов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2023 г.

# ЦЕЛЬ РАБОТЫ

Получение практических навыков разработки и отладки программ, использующих экземпляры собственного класса.

# порядок выполнения работы

1. Ознакомиться с общей постановкой задачи.

2. Ознакомится с вариантом задания – соответствует вашему номеру в списке группы (при нехватке заданий вариант задания вычисляется как номер\_в\_списке\_группы - количество\_заданий).

3. Разработать класс согласно варианту задания.

4. Написать и отладить программу на подготовленных наборах тестовых данных.

5. Подготовить отчет по лабораторной работе. Отчет должен включать в себя:

• титульный лист;

• цель лабораторной работы;

• постановку задачи;

• текст программы с комментариями;

• демонстрацию работы программы (Снимки экрана при выполнении действий программы с описанием).

• краткие ответы на контрольные вопросы;

• выводы по лабораторной работе.

6. Защитить лабораторную работу перед преподавателем.

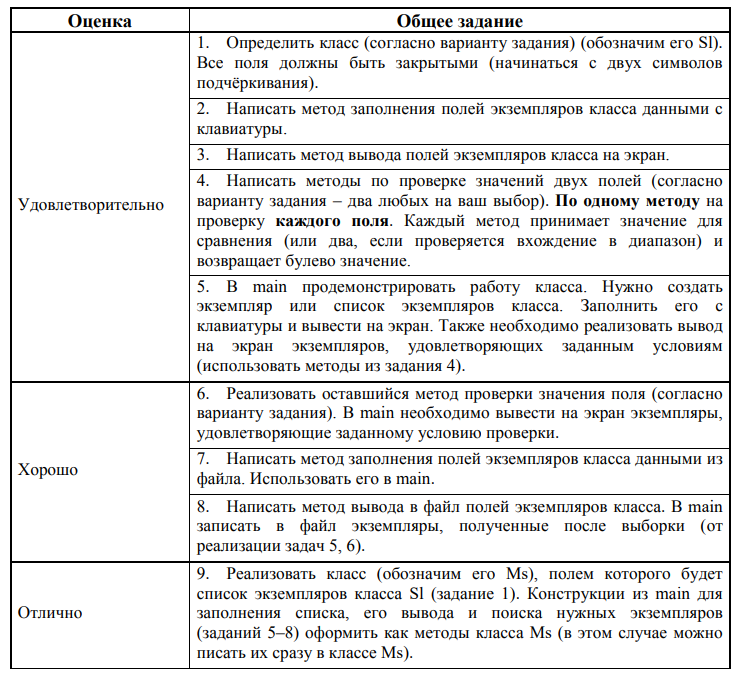
# Задания

Необходимо разработать программу на языке Python, реализующую задачу ранее решенную на языке С++ в лабораторной работе №1.

Программ должна формировать и обрабатывать список экземпляров класса (согласно варианту задания).

В реализуемой программе необходимо предусмотреть возможность демонстрации методов созданного класса.

В зависимости от оценки, на которую вы претендуете, необходимо выполнить следующие задания (Для каждой следующей оценки нужно выполнить ВСЕ предыдущие задания, если обратное не указано явно):



**Вариант 21**. Класс Aeroflot (Рейс): Поля: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Реализовать выборки: а) списка рейсов для заданного пункта назначения; б) списка рейсов самолетов указанного типа; в) списка рейсов для заданного дня недели, время вылета которых находится в заданном интервале. Реализовать сортировку массива объектов класса на основе: - времени вылета.

# ХОД РАБОТЫ

**Текст программы:**

Содержимое файла module.py:

*#Чтение*

*def* read\_file(*file\_path*):

    with open(file\_path, "r") as file:

        elements = file.read().split()

        return [*int*(element) if element.isdigit() else *float*(element) for element in elements]

Чтение происходит при помощи метода open(), затем они записываются в массив elements

*#Обработка*

*def* process\_list(*original\_list*):

    nZ = 1

    aboveAvg = 0

    total\_sum = 0

    for element in original\_list:

        total\_sum += element

        if element != 0:

            nZ \*= element

        if element > sum(original\_list) / len(original\_list):

            aboveAvg += 1

    return (nZ, aboveAvg)

Обработка списка original\_list с помощью цикла, в котором, проверяется каждый элемент на то, является ли он больше среднего и умножается к общему произведению не нулевых чисел.

*#Запись*

*def* save\_result(*result*, *output\_file\_path*):

    with open(output\_file\_path, "w") as file:

        file.write(*f*"Произведение ненулевых элементов: {result[0]}\n")

        file.write(*f*"Количество элементов, превышающих среднее: {result[1]}")

Запись в файл с помощью метода write().

# Тестирование

**Пример работы программы с исходным файлом, содержащим числа 1 4 9:**

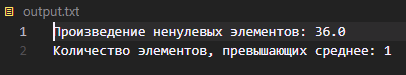
Содержимое файла input.txt:



Результат работы программы:



Содержимое файла output.txt:



**Пример работы программы с исходным файлом, содержащим числа 4 -1 6 -9 -2 -50 23 -3 8:**

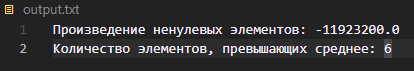
Содержимое файла input.txt:



Результат работы программы:



Содержимое файла output.txt:



Произведение отрицательное т.к. в последовательности нечетное количество отрицательных элементов.

# ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. **В чем состоит сущность методологии объектно-ориентированного программирования?**
   * Методология объектно-ориентированного программирования (ООП) основывается на концепции объектов, которые объединяют данные и методы для их обработки. Основные принципы включают инкапсуляцию, наследование и полиморфизм для упрощения проектирования и повышения переиспользования кода.
2. **Что такое класс? Как выполнить описание класса?**
   * Класс - это шаблон или чертёж для создания объектов. Он определяет атрибуты (поля) и методы, которые будут у объектов этого класса. Описание класса включает в себя ключевое слово **class**, за которым следует имя класса и блок, содержащий его атрибуты и методы.
3. **Что такое экземпляр класса и как его создать в программе?**
   * Экземпляр класса - это конкретный объект, созданный на основе определения класса. Для создания экземпляра используется вызов конструктора класса, например: **my\_instance = MyClass()**.
4. **Что такое поле класса?**
   * Поле класса - это переменная, которая хранит данные для каждого экземпляра класса. Они определяются внутри класса и представляют характеристики объекта.
5. **Что такое методы класса?**
   * Методы класса - это функции, связанные с классом, которые могут выполнять операции с данными класса. Они определяются внутри класса и могут вызываться для экземпляров класса.
6. **Как выполняется вызов метода класса?**
   * Вызов метода класса осуществляется через экземпляр класса, используя точечную нотацию: **instance.method()**.
7. **Какие методы называются конструкторами?**
   * Конструкторы - это методы, которые выполняют инициализацию объекта. В Python, конструктор имеет имя **\_\_init\_\_**. Он вызывается автоматически при создании экземпляра класса.
8. **Как выполняется инициализация экземпляров класса?**
   * Инициализация экземпляра класса происходит в конструкторе класса (**\_\_init\_\_**), где устанавливаются начальные значения атрибутов объекта.
9. **Как в классе реализуются инкапсуляция?**
   * Инкапсуляция в классе реализуется с помощью использования двойных подчеркиваний перед именами атрибутов (например, **self.\_\_attribute**). Это делает атрибуты приватными и ограничивает доступ к ним извне класса.

# ВЫВОД

Работа с примером кода по созданию классов Aeroflot и Flights позволяет лучше понять применение концепций ООП, включая создание объектов, работу с методами, инициализацию и использование конструкторов, а также сортировку данных в объектах класса.