Отчет по лабораторной работе №4

Тонко Полины, гр. 253501

Вариант 24

**Тема**: Работа с файлами, классами, сериализаторами, регулярными выражениями и стандартными библиотеками.

**Цель**: освоить базовый синтаксис языка Python, приобрести навыки работы с файлами, классами, сериализаторами, регулярными выражениями и стандартными библиотеками и закрепить их на примере разработки интерактивных приложений.

**Требования к выполнению**

1. Программа должна быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и название, версию программы, Ф.И.О. разработчика и дату разработки.
2. Программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом
3. Выполнить документирование кода для получения справки по каждой функции
4. Для выполнения каждого задания разработать **пользовательские классы и обеспечить их наследование**. Реализовать примеры использования:

* статических и динамических атрибутов класса,

task5.Matrix

* полиморфизма,

Entities.Figure

* специальных(магических методов),

Entities.Student **str**, \_\_repr\_\_

* super(),

Entities.IsoscelecTrapezoid.\_\_ init \_\_

* геттеров и сеттеров,

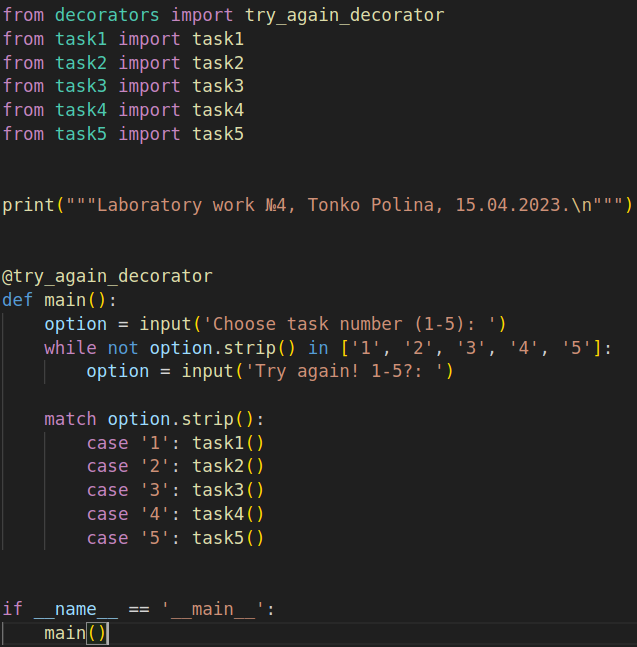
task5.Matrix

* свойств класса,
* примесей

Utilities.DictToStudentMixin

1. При разработке программ рекомендуется придерживаться принципа: за решение одной конкретной задачи должна отвечать одна функция.
2. Все функции необходимо сгруппировать в модулях, согласно их логике их работы.
3. Разработанные основные функции, размещенные в отдельных модулях, нужно подключить в другом модуле, где будет происходить тестирование данных функций.
4. В программах предусмотреть возможность повторного выполнения без выхода из программы и защиту от ввода некорректных пользовательских данных. Для этих целей рекомендуется разработать отдельные функции.
5. Обеспечить обработку конкретных классов исключений

**main.py**

****

**Задание 1.** Исходные данные представляют собой словарь. Необходимо поместить их в файл, используя сериализатор. Организовать считывание данных, поиск, сортировку в соответствии с индивидуальным заданием. Обязательно использовать классы. Реализуйте два варианта:

1)формат файлов CSV;

2)модуль pickle.

О поступивших в вуз студентах собрана информация: фамилия, нуждается ли в общежитии, стаж работы (если есть), что окончил, какой язык изучал. Определите:

a) сколько человек нуждаются в общежитии;

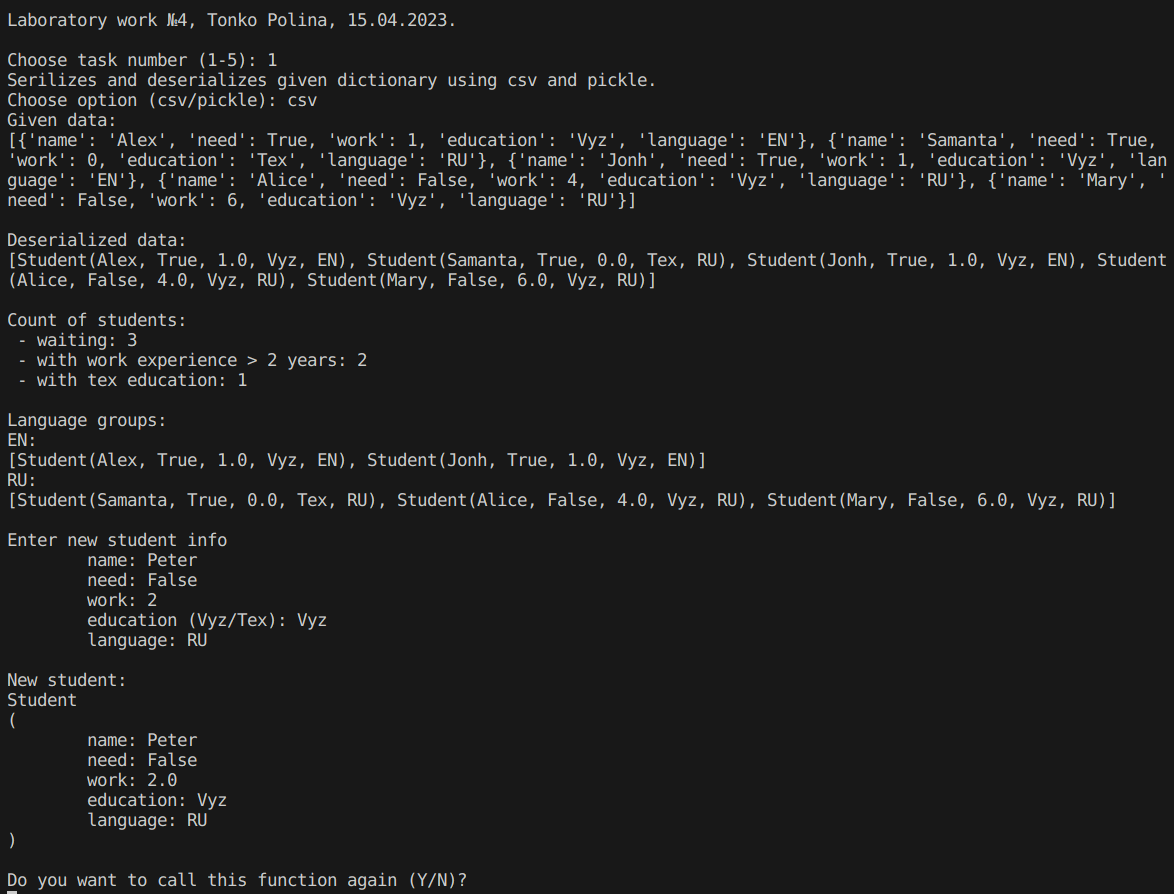
b) списки студентов, имеющих стаж работы более 2 лет;

c) списки окончивших техникум;

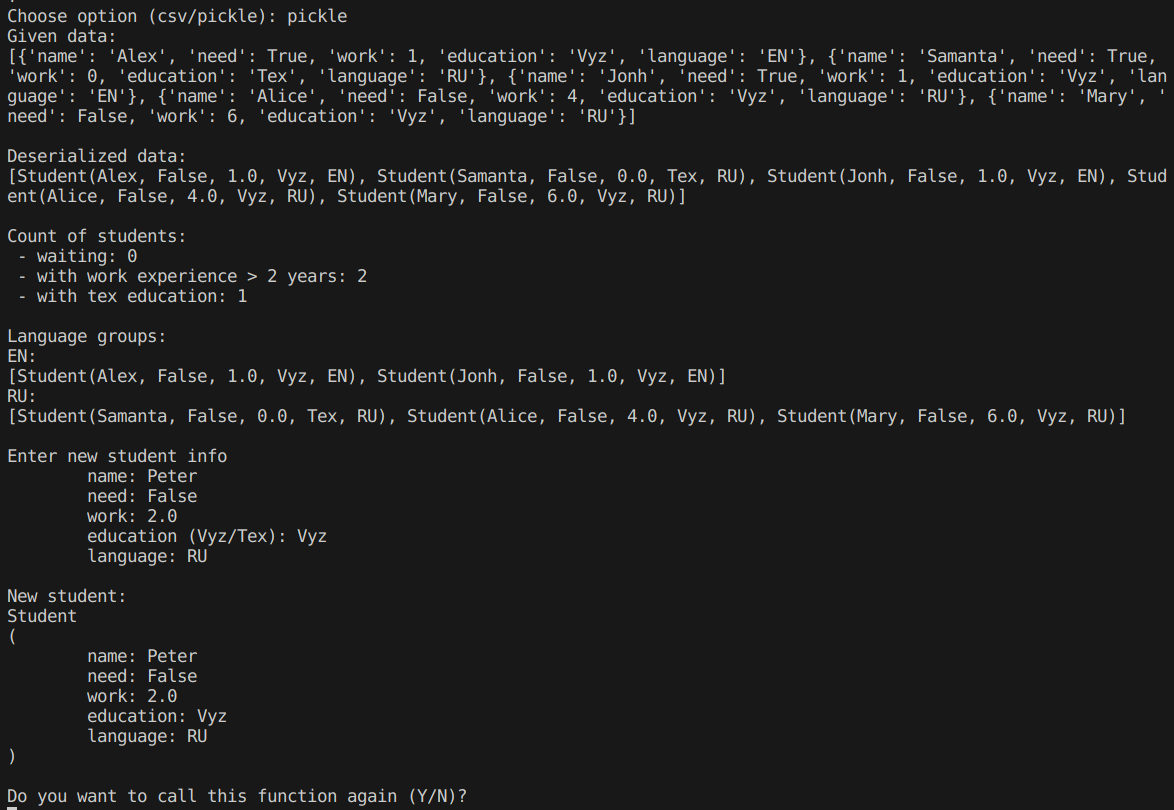
d) списки языковых групп.

Выведите информацию о студенте, введенном с клавиатуры.

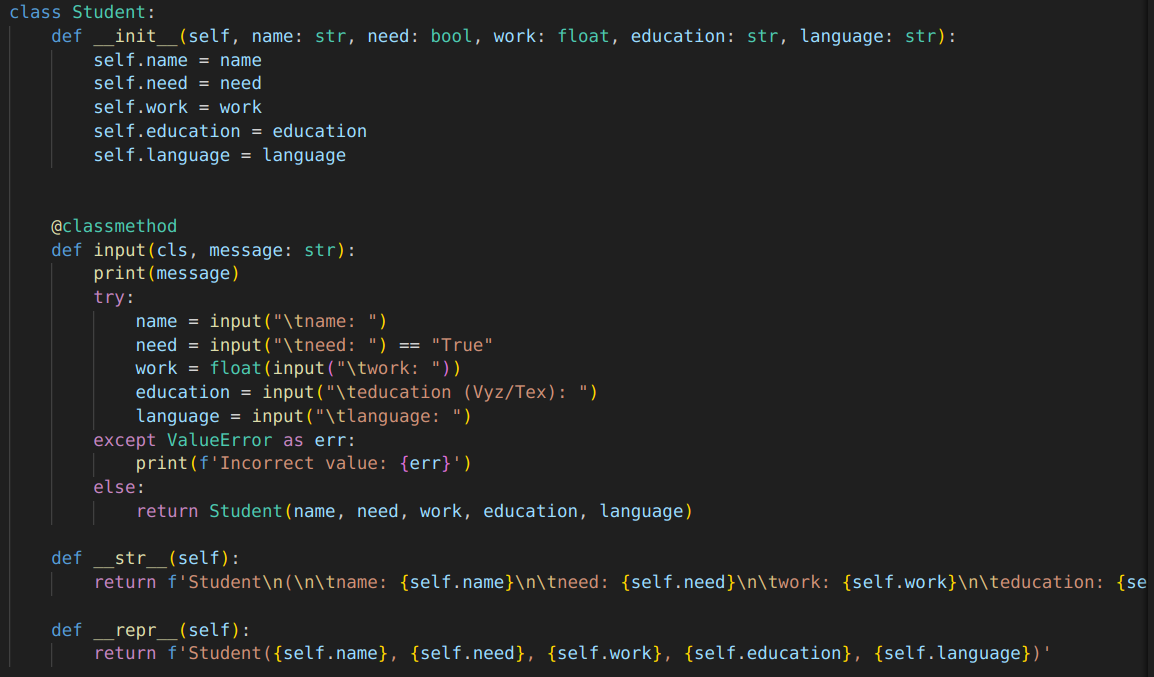
CSV



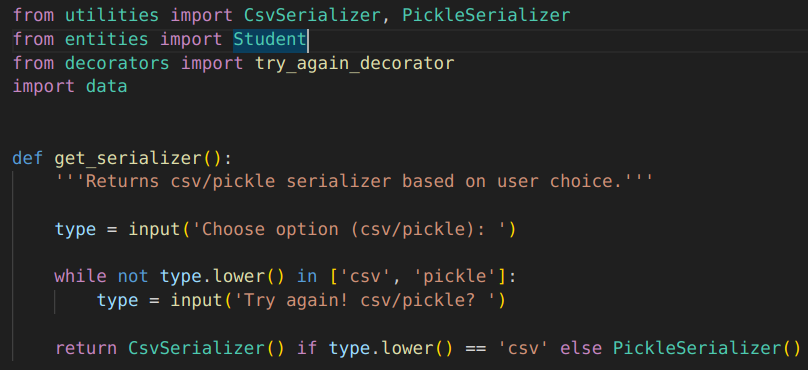
Pickle

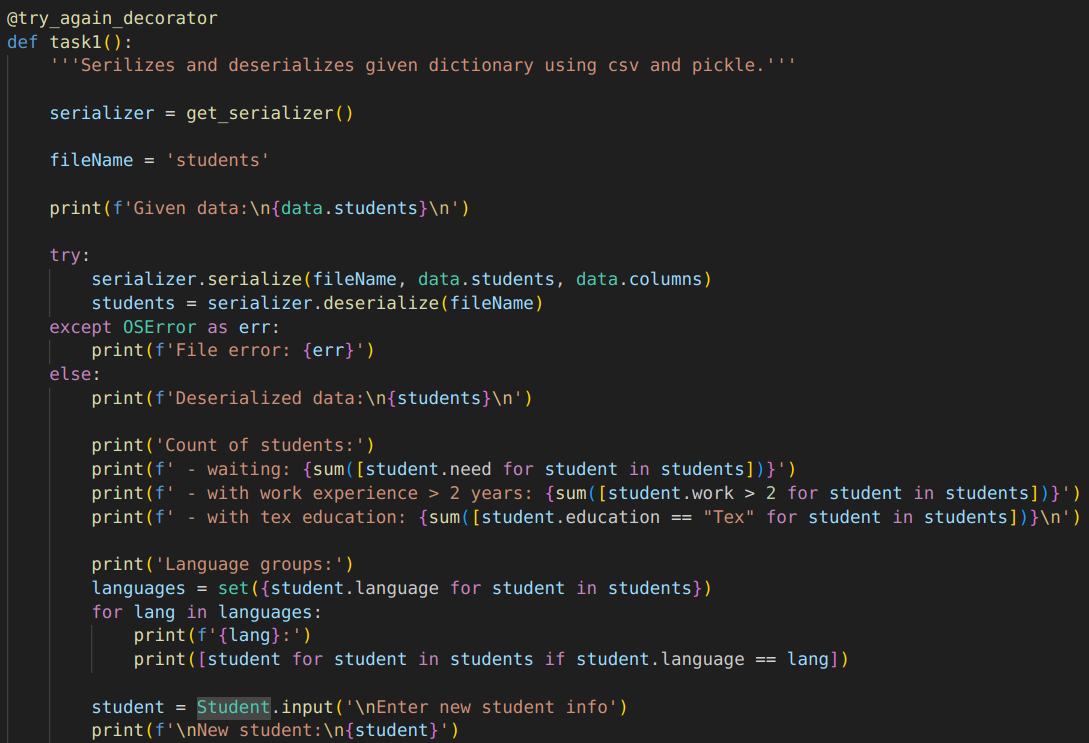


entities.py

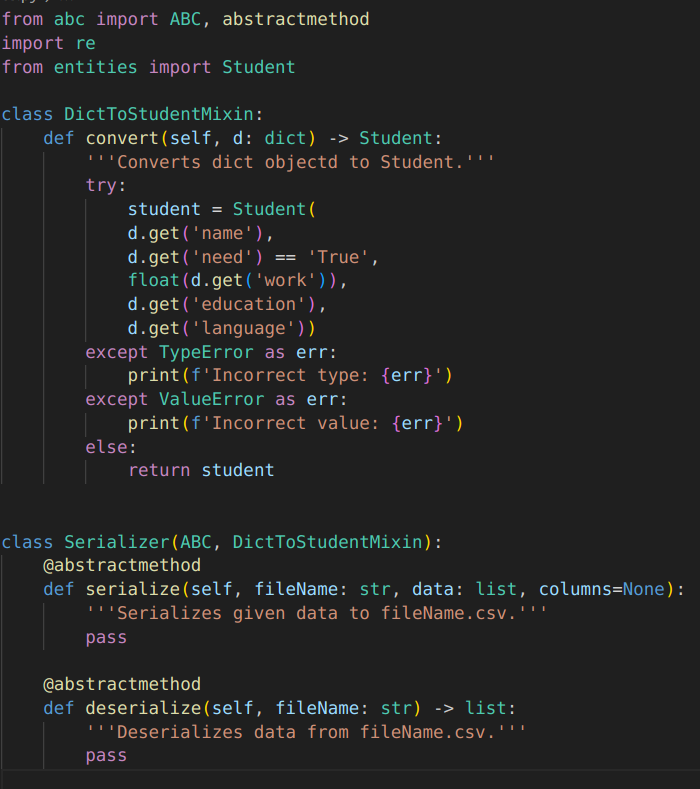
****

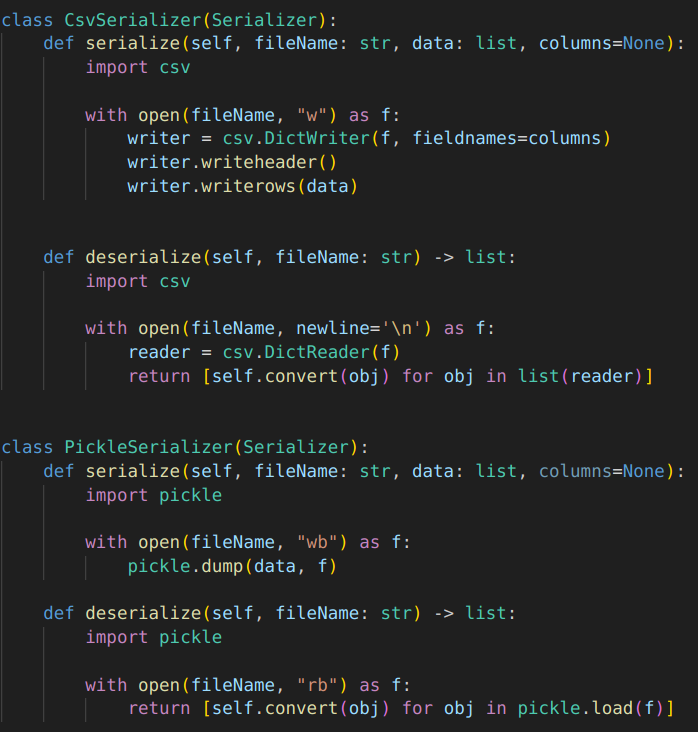
**task1.py**

****

****

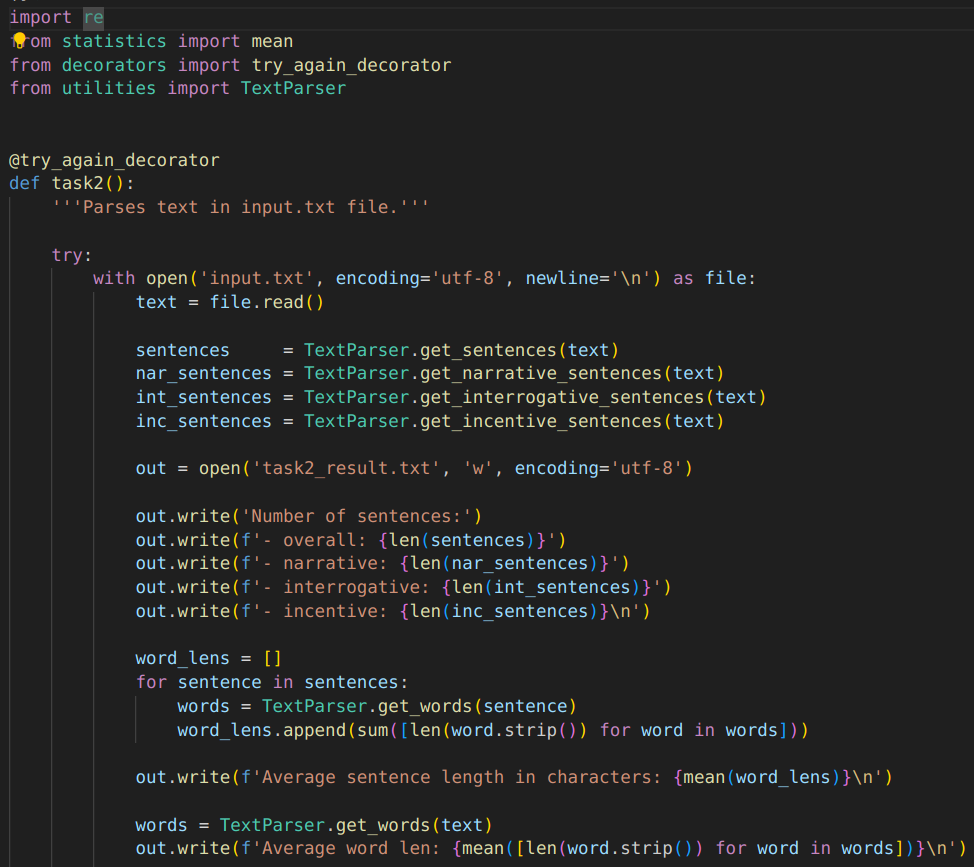
**utilities.py**

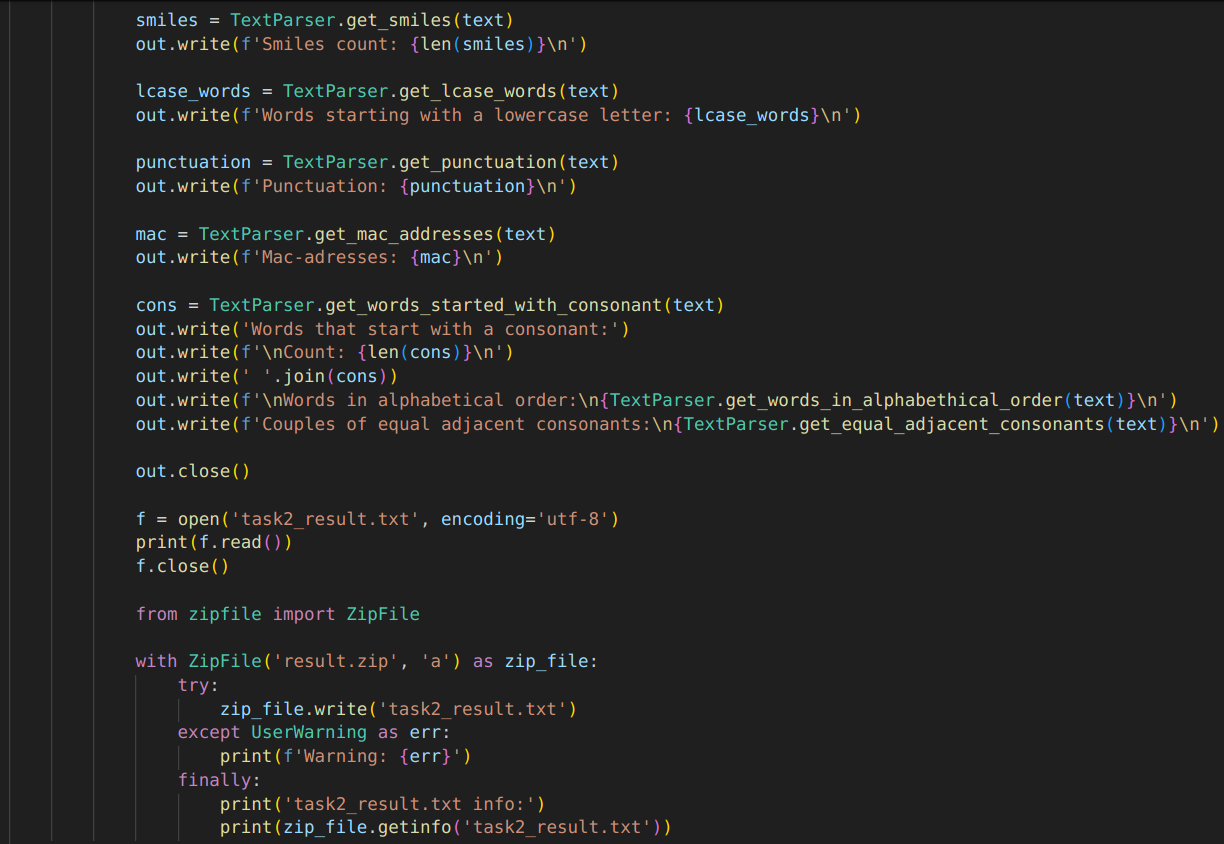
****

****

**Задание 2.** В соответствии с заданием своего варианта составить программу для анализа текста. Считать из исходного файла текст. Используя регулярные выражения получить искомую информацию (см. условие), вывести ее на экран и сохранить в другой файл. Заархивировать файл с результатом с помощью модуля zipfile и обеспечить получение информации о файле в архиве.

task2.py

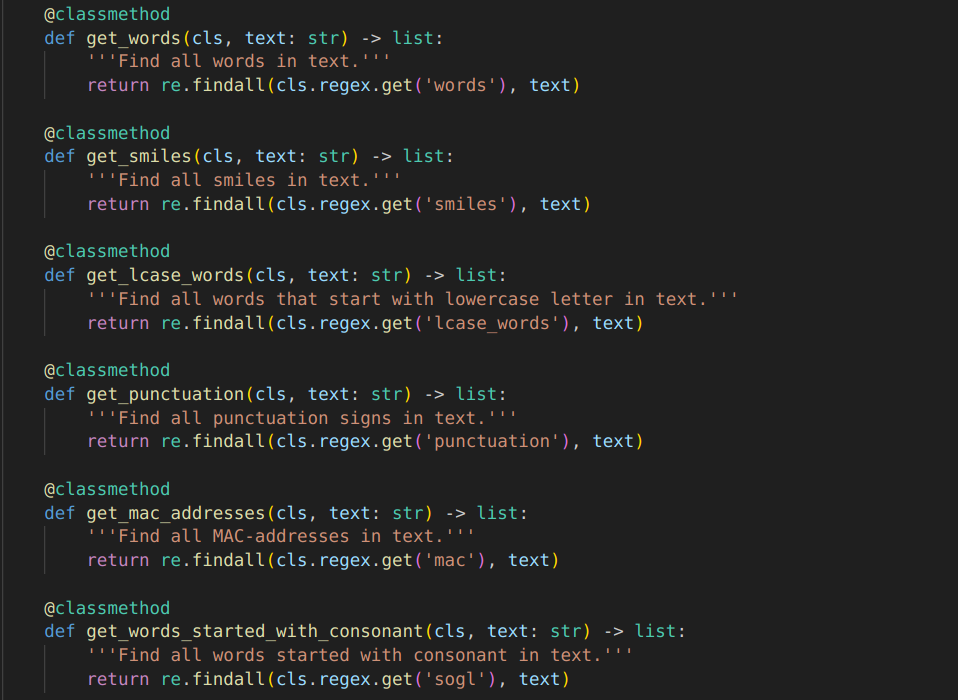


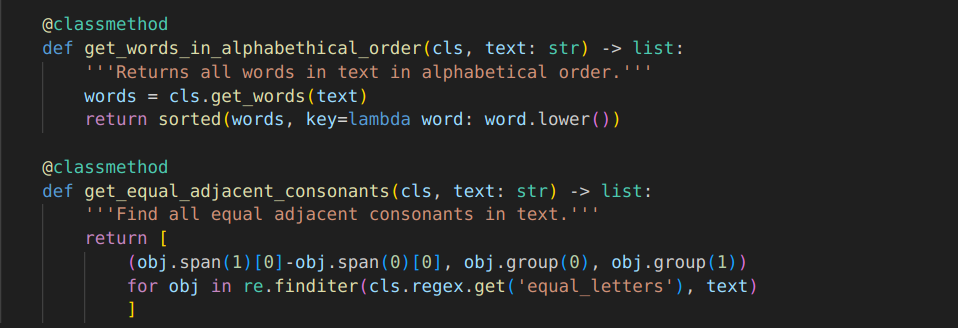


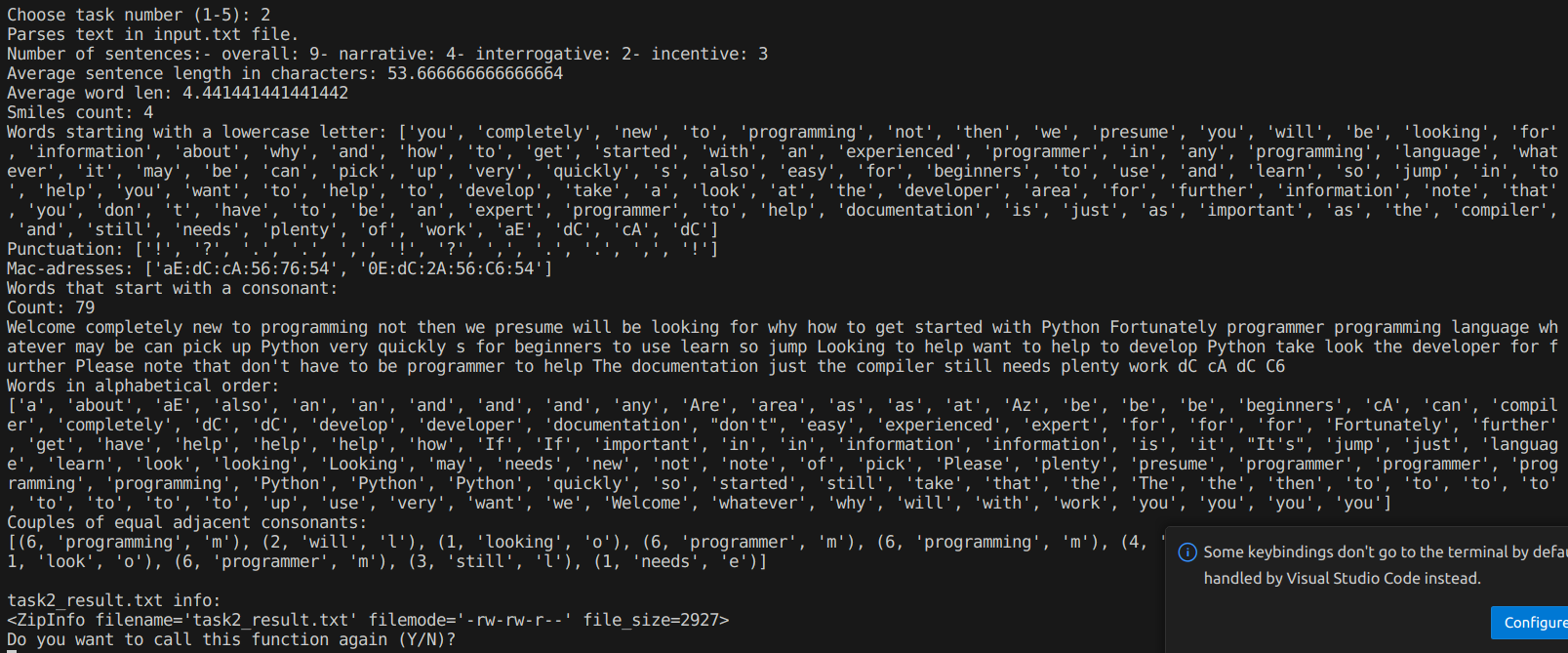


utilities.py









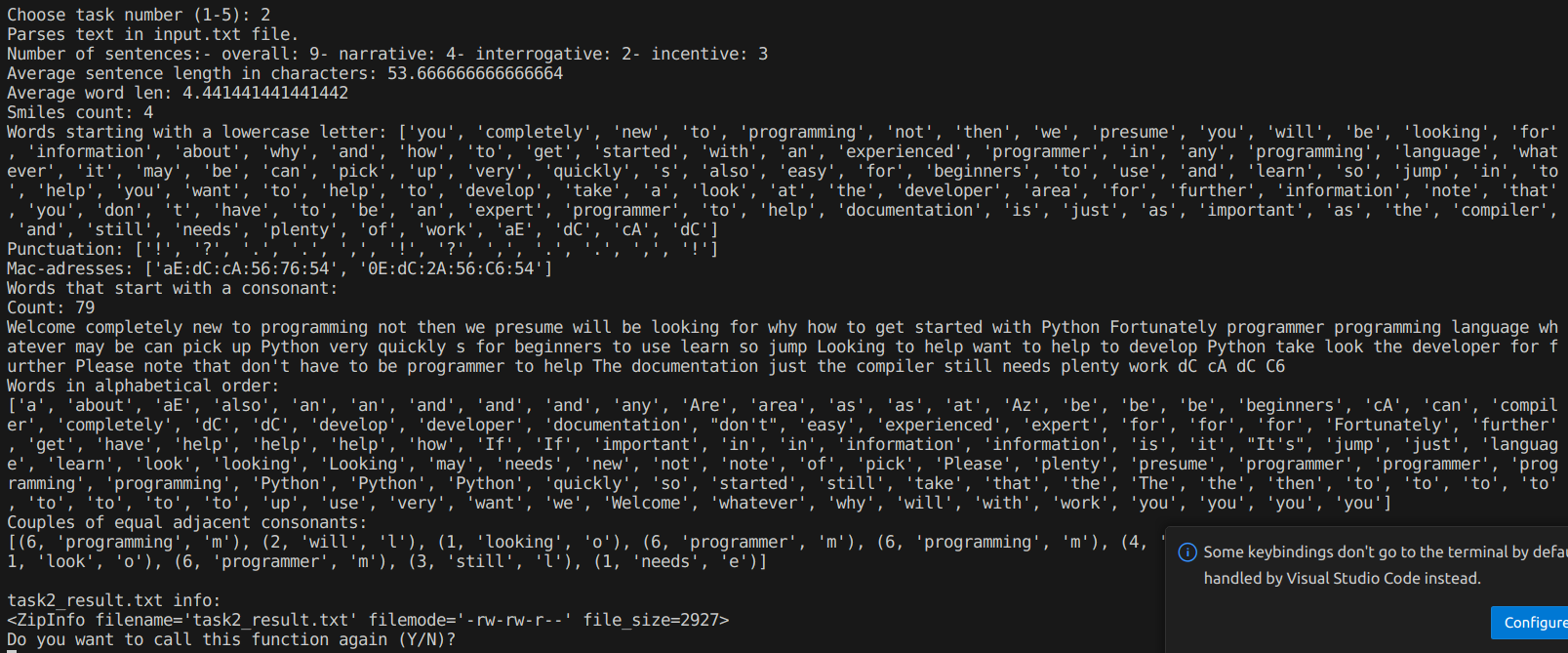
Также выполнить общее задание – определить и сохранить в файл с результатами:

* количество предложений в тексте;
* количество предложений в тексте каждого вида отдельно (повествовательные, вопросительные и побудительные);
* среднюю длину предложения в символах (считаются только слова);
* среднюю длину слова в тексте в символах;
* количество смайликов в заданном тексте. Смайликом будем считать последовательность символов, удовлетворяющую условиям:

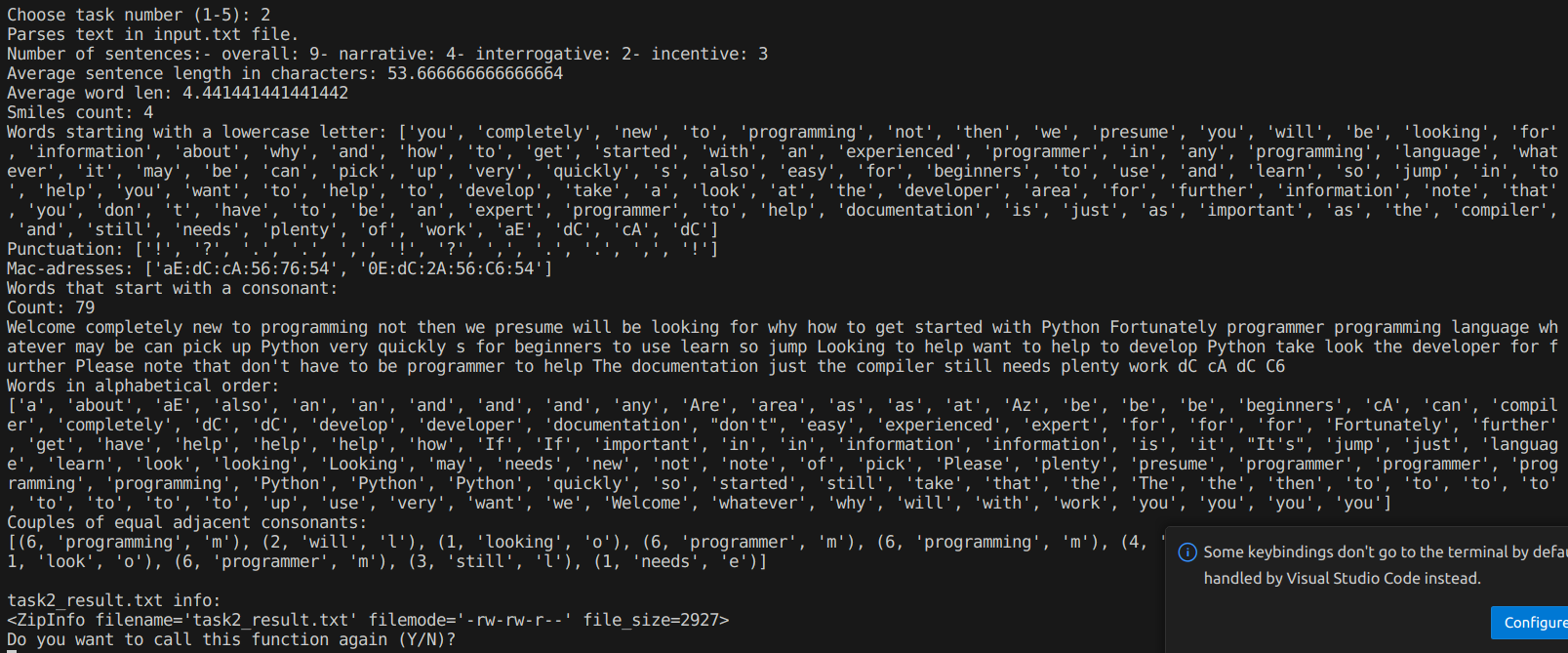
первым символом является либо «;» (точка с запятой) либо «:» (двоеточие) ровно один раз;

далее может идти символ «-» (минус) сколько угодно раз (в том числе символ минус может идти ноль раз);

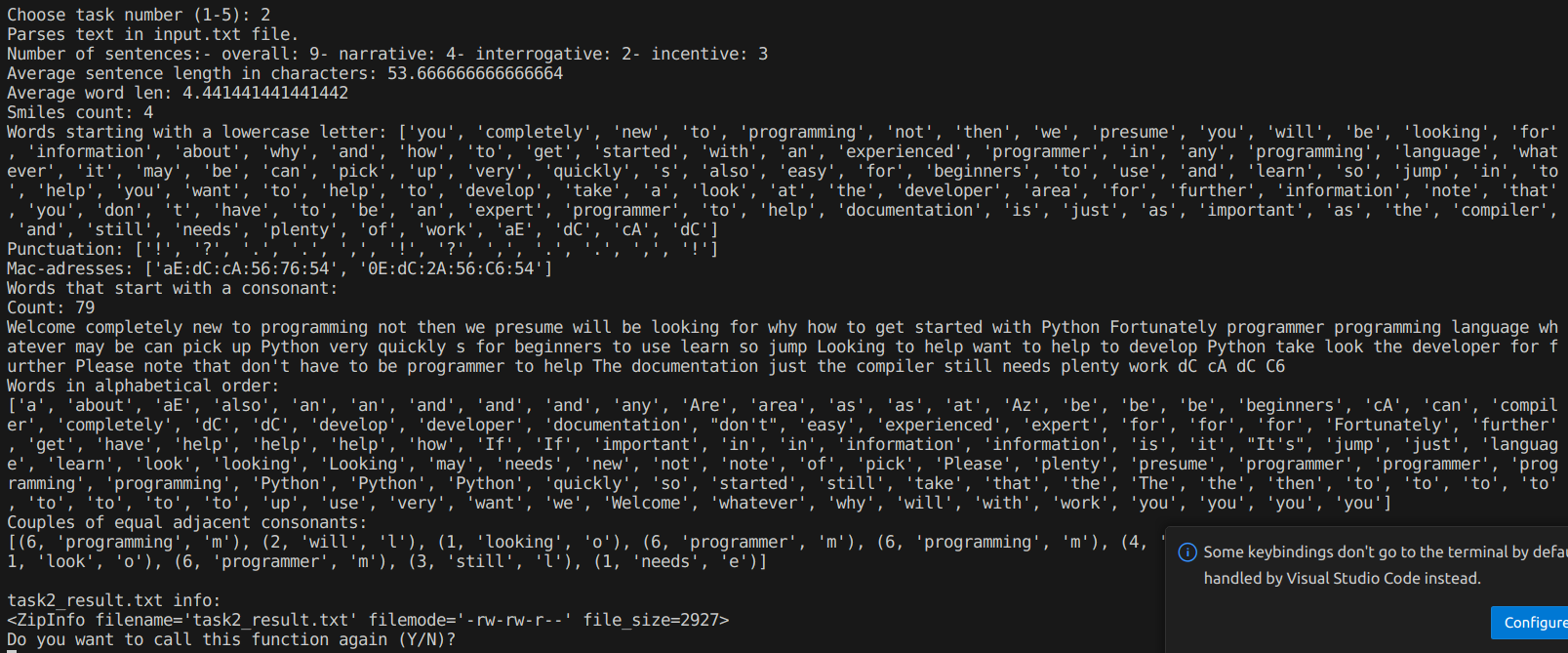
в конце обязательно идет некоторое количество (не меньше одной) одинаковых скобок из следующего набора: «(», «)», «[», «]»;

внутри смайлика не может встречаться никаких других символов. Например, эта последовательность является смайликом: «;---------[[[[[[[[». Эти последовательности смайликами не являются: «]», «;--»,«:»,«)».

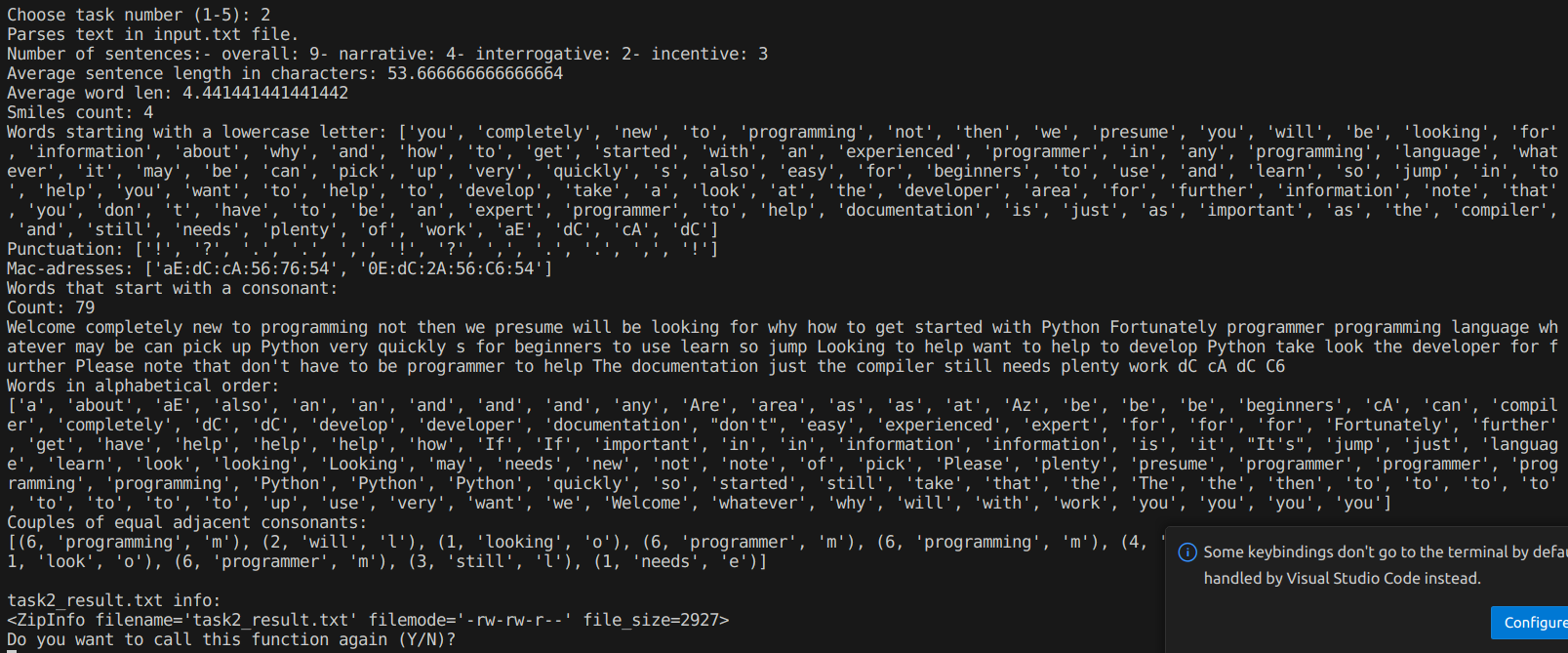
Вывести все слова, начинающиеся со строчной буквы и знаки препинания.



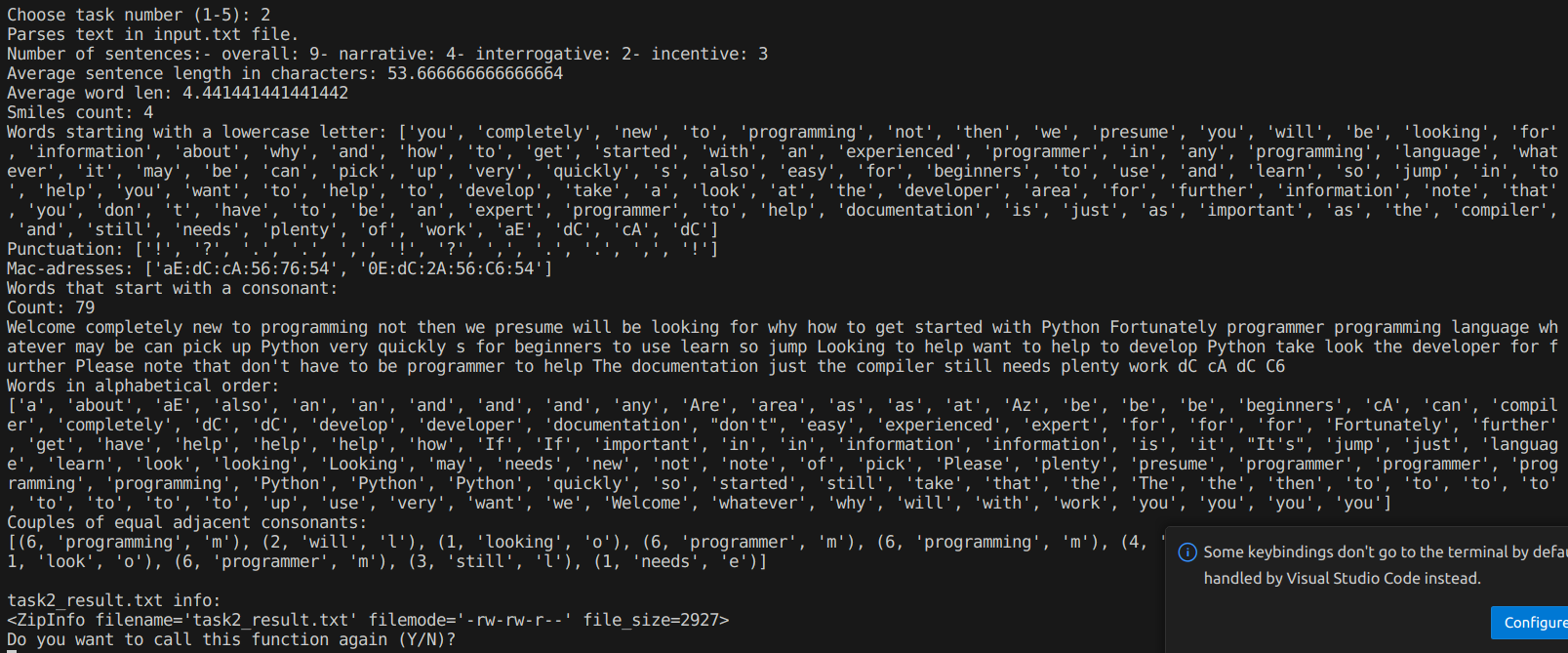
Определить, является ли заданная строка правильным MAC-адресом. Пример правильного выражения: aE:dC:cA:56:76:54. Пример неправильного выражения: 01:23:45:67:89:Az.



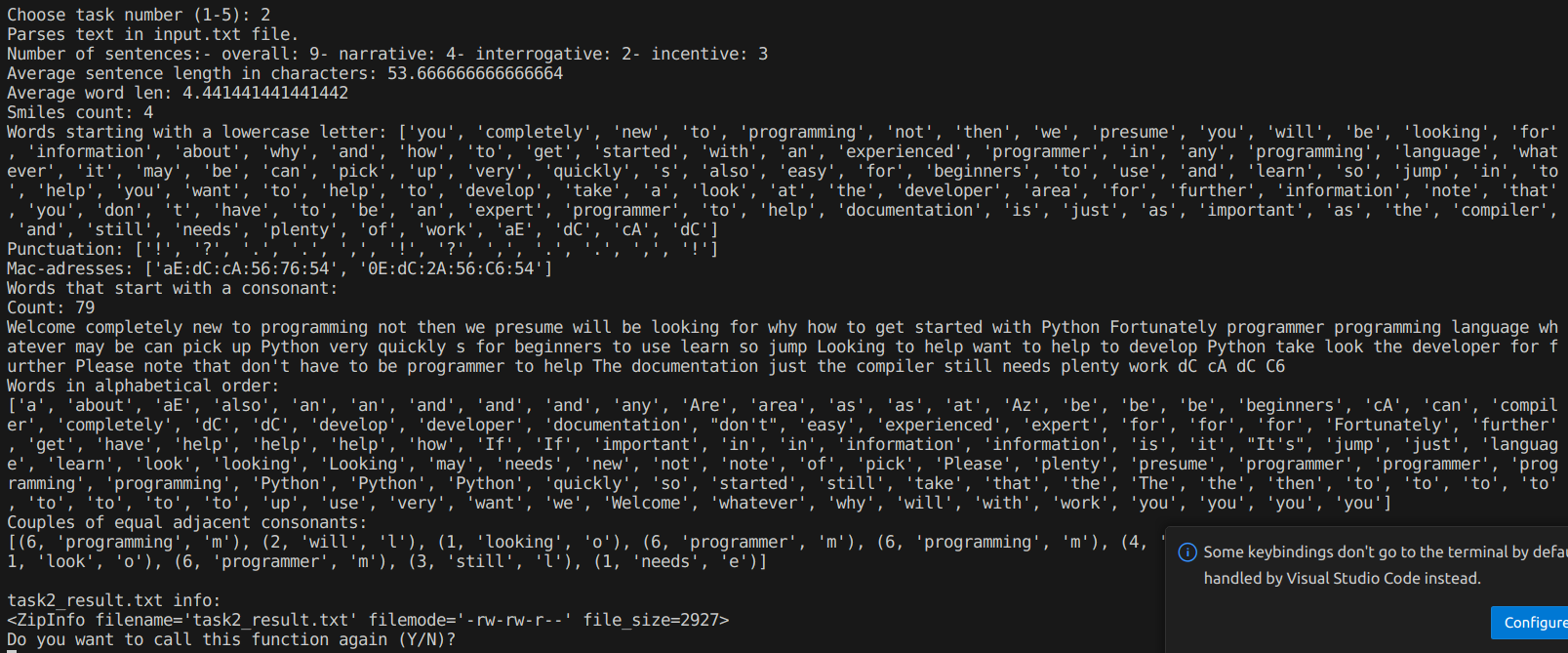
определить число слов, которые начинаются с согласной;



найти слова, содержащие две одинаковые буквы подряд и их порядковые номера;



вывести слова в алфавитном порядке

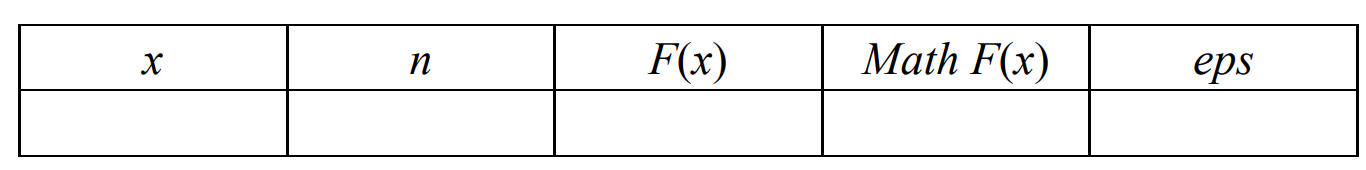
****

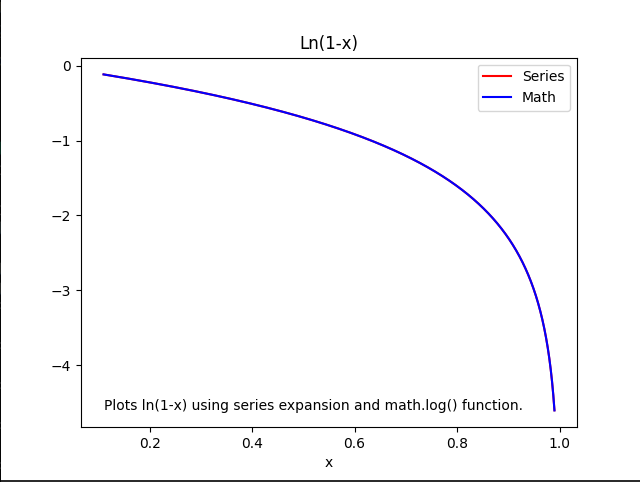
**Задание 3.**  В соответствии с заданием своего варианта доработать программу из ЛР3, использовав класс и обеспечить:

а) определение дополнительных параметров среднее арифметическое элементов последовательности, медиана, мода, дисперсия, СКО последовательности;

б) с помощью библиотеки matplotlib нарисовать графики разных цветов в одной координатной оси:

* график по полученным данным разложения функции в ряд, представленным в таблице,
* график соответствующей функции, представленной с помощью модуля math. Обеспечить отображение координатных осей, легенды, текста и аннотации.





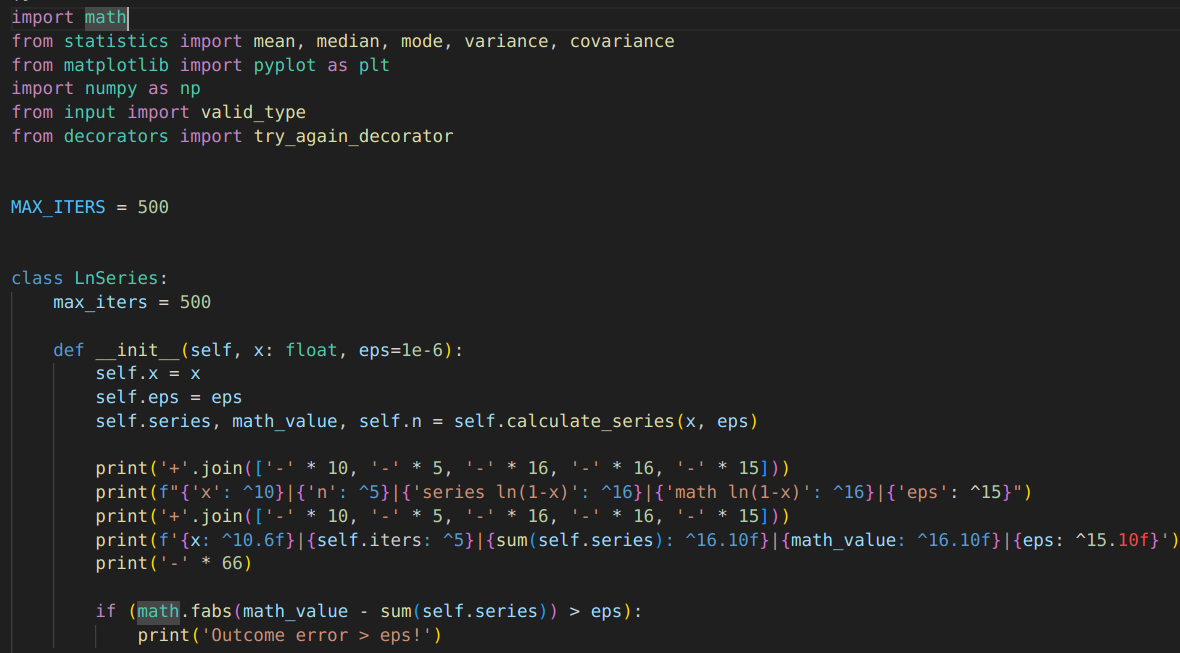
Здесь x – значение аргумента, F(x) – значение функции, n – количество просуммированных членов ряда, Math F(x) – значение функции, вычисленное с помощью модуля math.

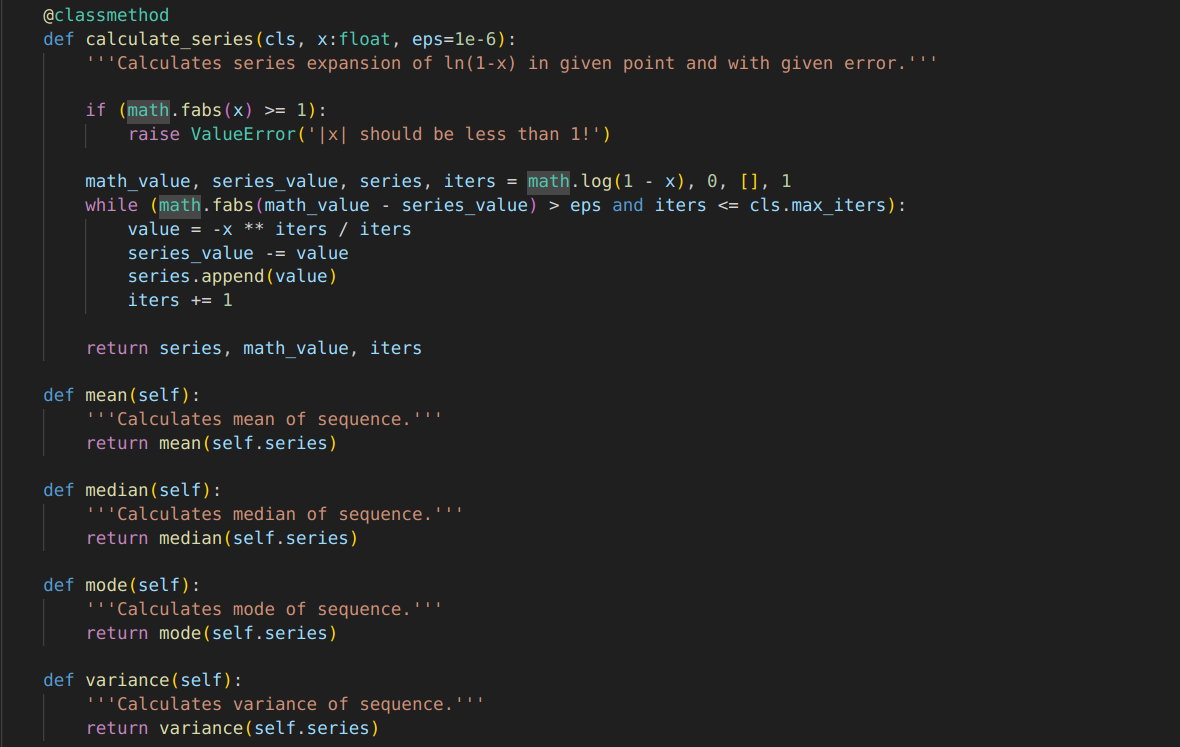
в) сохранить графики в файл

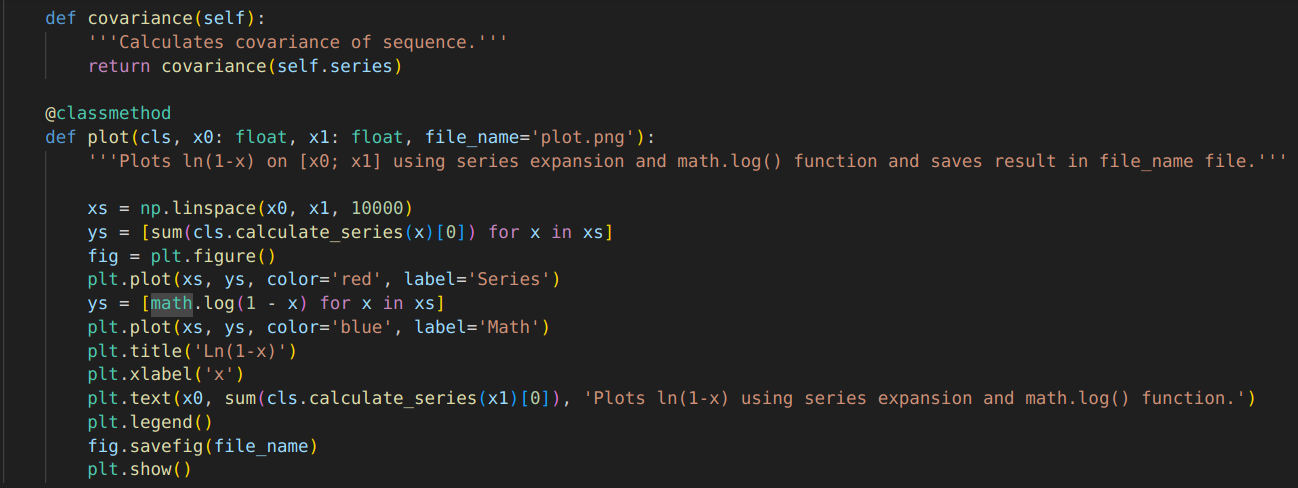


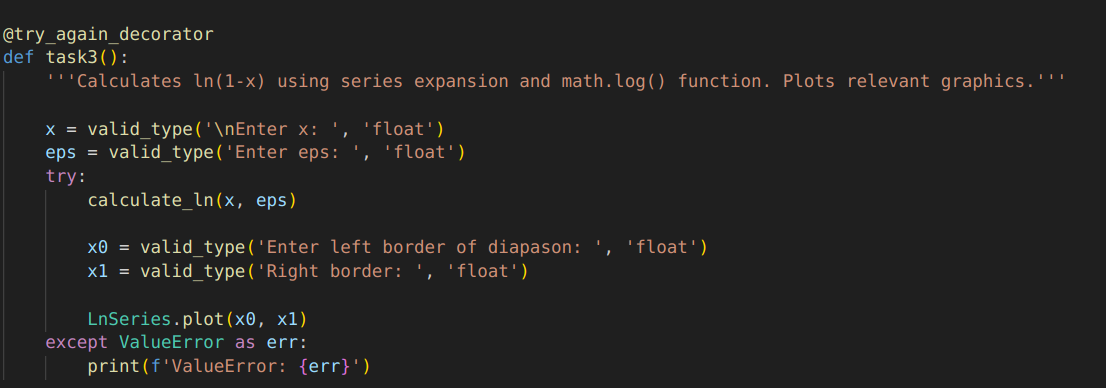


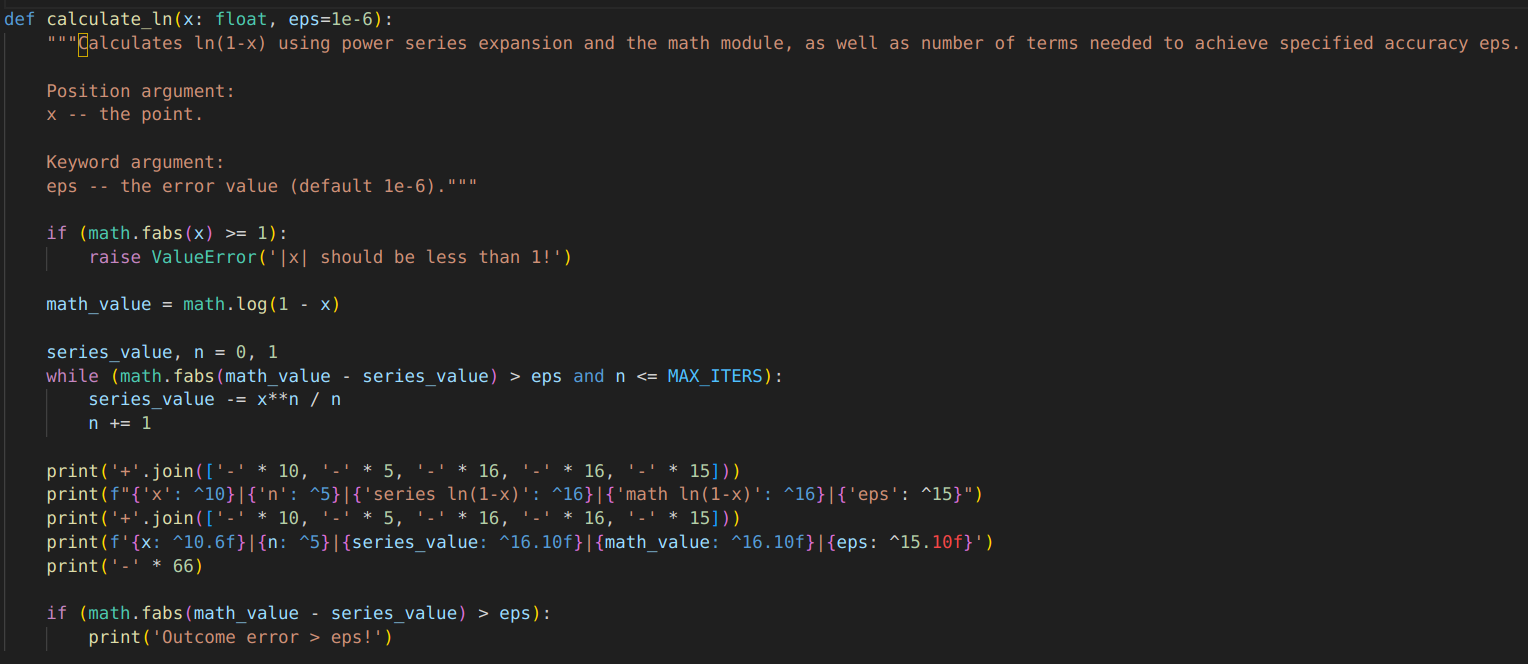
task3.py

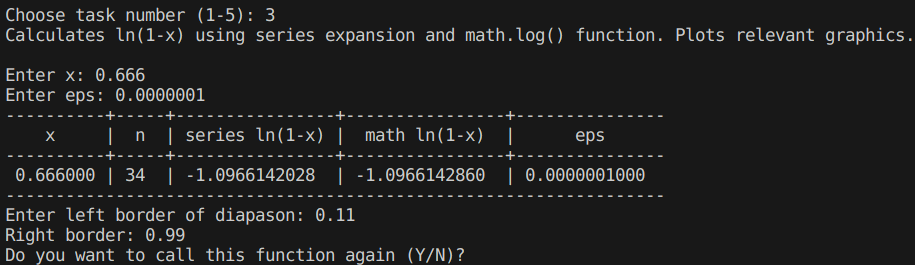










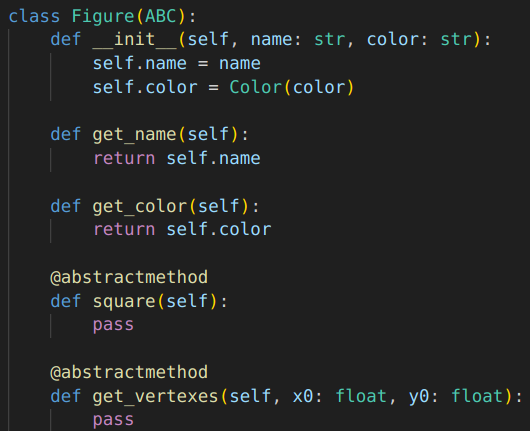


**Задание 4.** В соответствии с заданием своего варианта разработать базовые классы и классы наследники.

Требования по использованию классов:

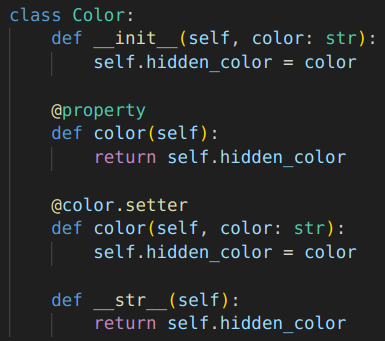
Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры (<https://docs.python.org/3/library/abc.html> )

entities.py



Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры (<https://docs.python.org/3/library/functions.html#property> )

entities.py



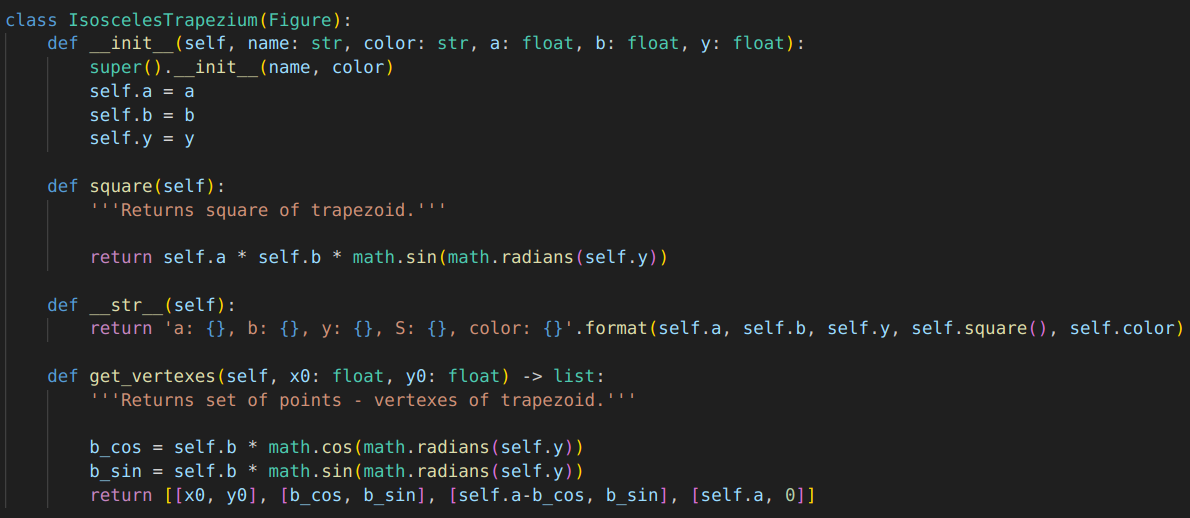
Класс «Прямоугольник» (Круг, Ромб, Квадрат, Треугольник и т.д.) наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» (для другого типа фигуры соответствующие параметры, например, для круга задаем «радиус») и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры <https://docs.python.org/3/library/math.html> .

Для класса «Прямоугольник»(тип фигуры в инд. задании)

определить метод, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Использовать метод format (<https://pyformat.info/> )

название фигуры должно задаваться в виде поля данных класса и возвращаться методом класса.

entities.py



В корневом каталоге проекта создайте файл main.py для тестирования классов. Используйте конструкцию, описанную в <https://docs.python.org/3/library/__main__.html>

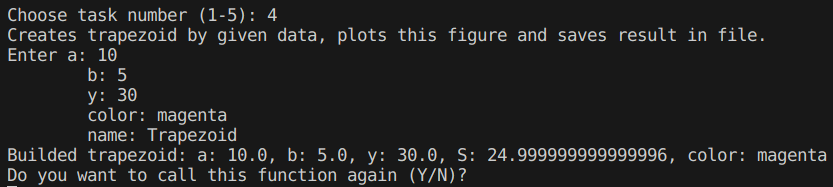
Пример объекта: Прямоугольник синего цвета шириной 5 и высотой 8.

Программа должна содержать следующие базовые функции:

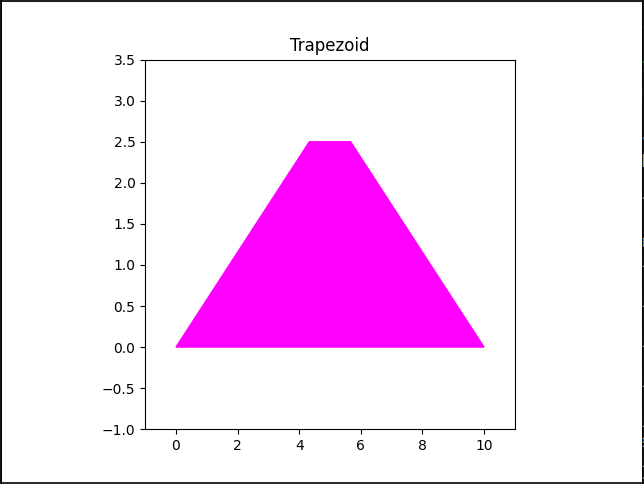
1. ввод значений параметров пользователем;

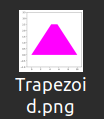
2) проверка корректности вводимых данных;

3) построение, закрашивание фигуры в выбранный цвет, введенный с клавиатуры, и подпись фигуры текстом, введенным с клавиатуры;



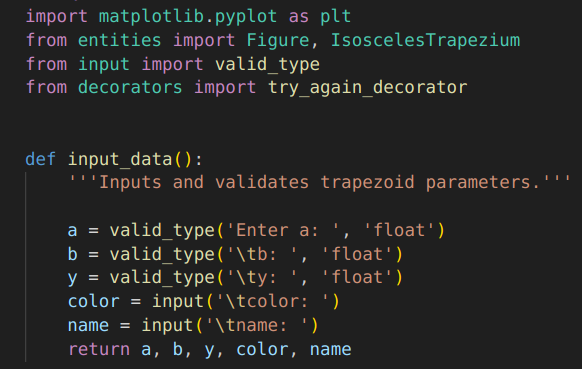
4) вывод фигуры на экран и в файл.

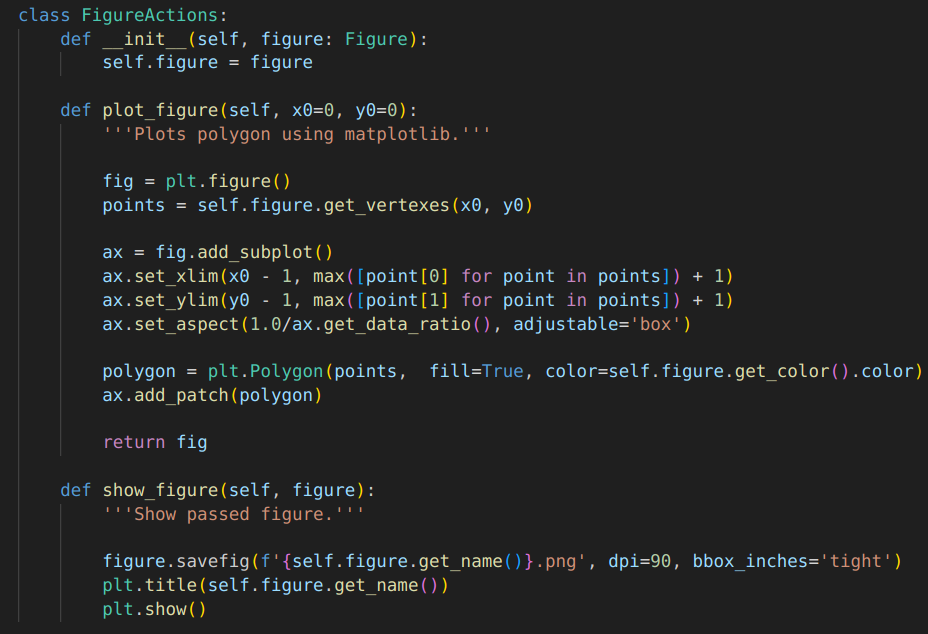


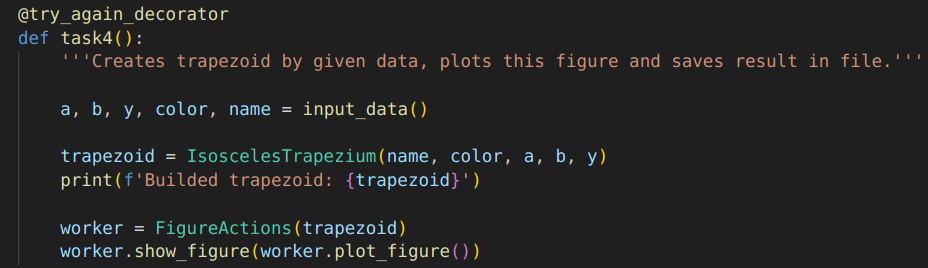


Построить равнобедренную трапецию по основанию a, боковой стороне b и углу между ними Y (в градусах).

task4.py







**Задание 5.** В соответствии с заданием своего варианта исследовать возможности библиотека NumPy при работе с массивами и математическими и статическими операциями. Сформировать целочисленную матрицу А[n,m] с помощью генератора случайных чисел (random).

а) Библиотека NumPy.

1. Создание массива. Функции array() и values().

2. Функции создания массива заданного вида.

3. Индексирование массивов NumPy. Индекс и срез.

4. Операции с массивами. Универсальные (поэлементные) функции.

б) Математические и статистические операции.

1. Функция mean()

2. Функция median()

3. Функция corrcoef()

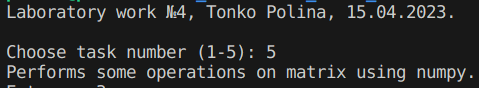
4. Дисперсия var().

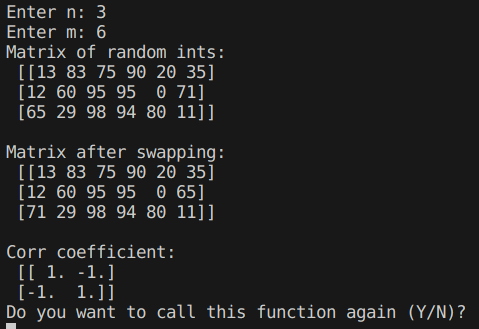
5. Стандартное отклонение std()

В матрице поменять местами наибольшие элементы в первом и последнем столбцах.

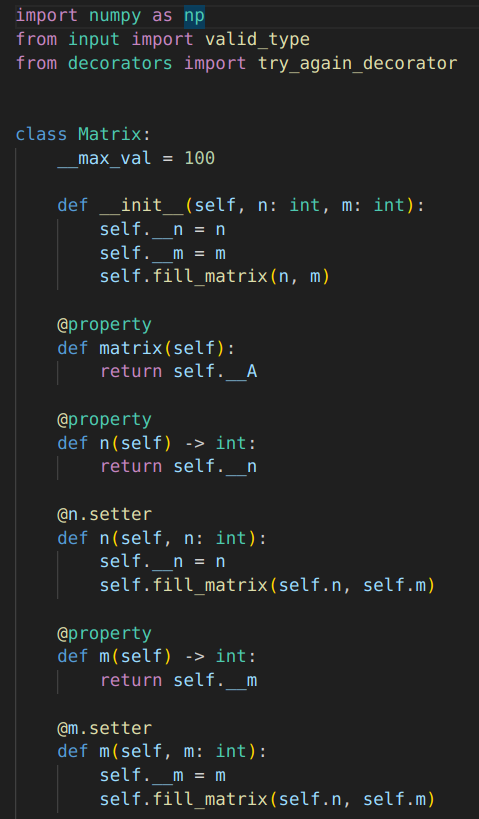
Вычислить значение коэффициента корреляции между элементами

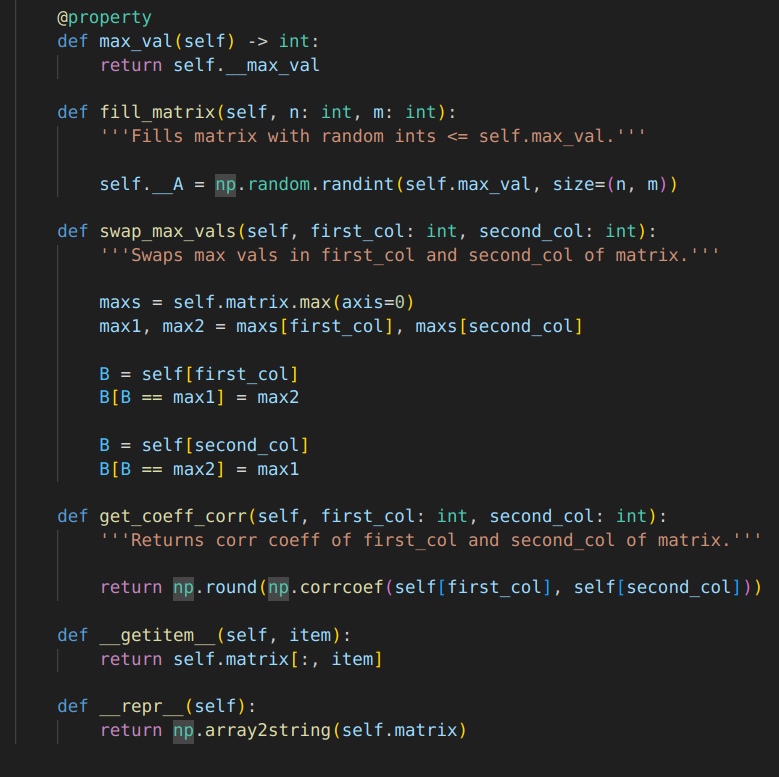
первого и последнего столбца. Ответ округлите до сотых.

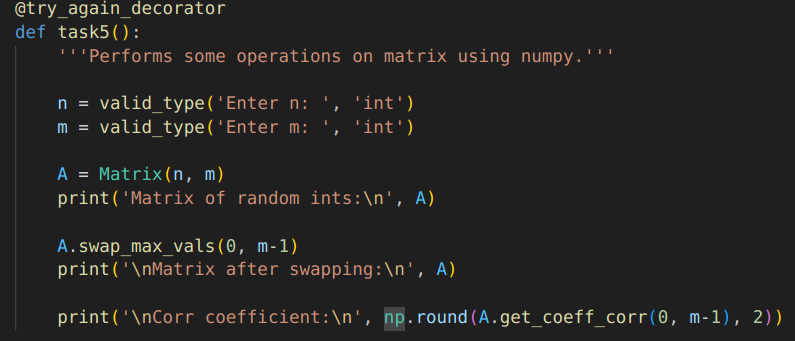




task5.py







Дополнительное задание для тех, кто стремится к большему (не обязательное):

Используя любой из наборов данных (datasets) на ресурсе: <https://www.kaggle.com/datasets> исследовать основные возможности библиотеки Pandas. См. Анализ Данных Python методичка.pdf

*Задание а. Библиотека Pandas. Структуры Series и DataFrame*

1. Библиотека Pandas. Импорт библиотеки.

2. Структура Series.

3. Создание Series .

4. Функция display.

5. Доступ к элементам Series с использованием .loc или .iloc.

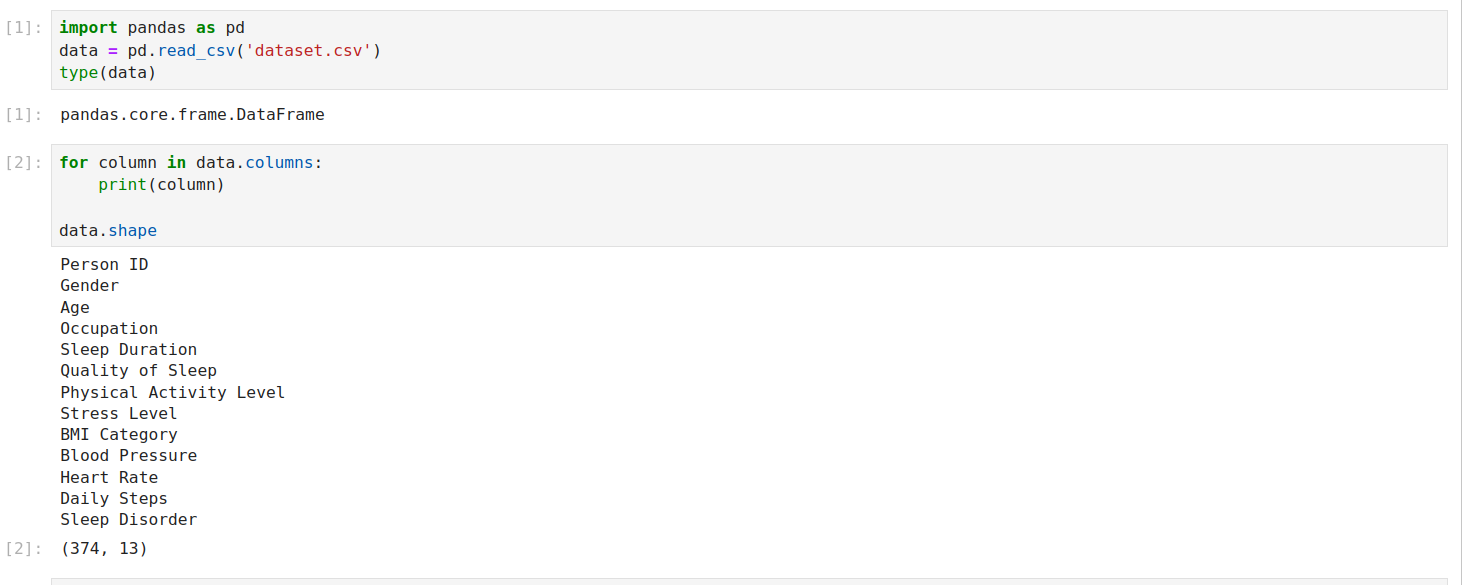
6. Объект DataFrame. Создание.

*Задание б. Основные операции*

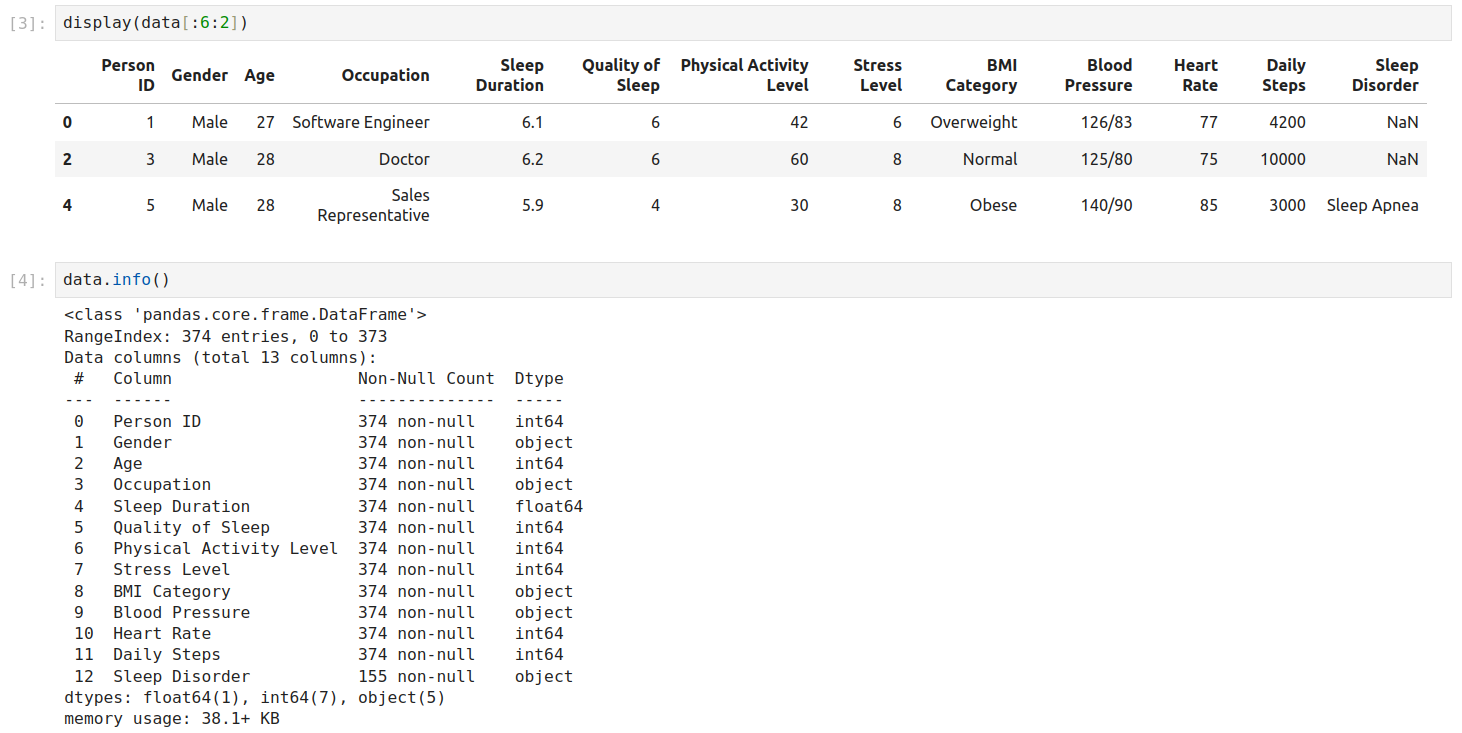
2. Получение информации о датафрейме (по каждому параметру)

5. Индексация и извлечение данных: статистические методы (во сколько раз среднее значение выбранного показателя с максимальным значением другого параметра (например, самый молодой, самый быстрый и т.д.) выше среднего выбранного показателя с минимальным значением )

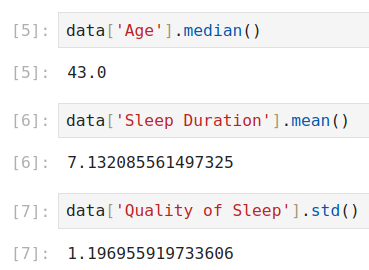
Чтение данных, вывод названий столбцов и размеров таблицы:



Методы display и info:



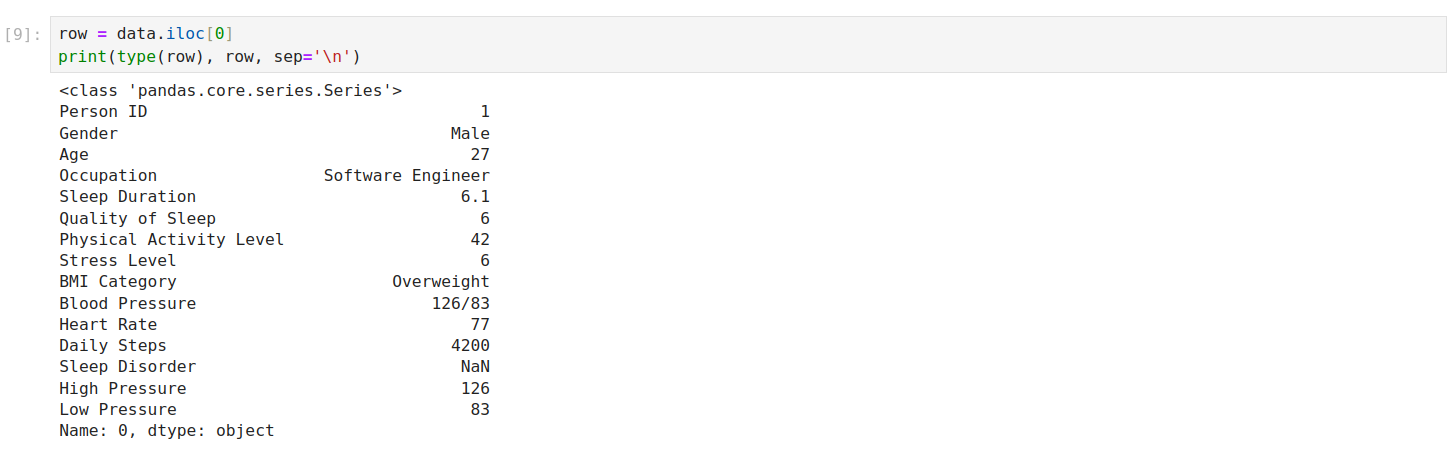
Расчет статистических показателей по данным из определенных столбцов:



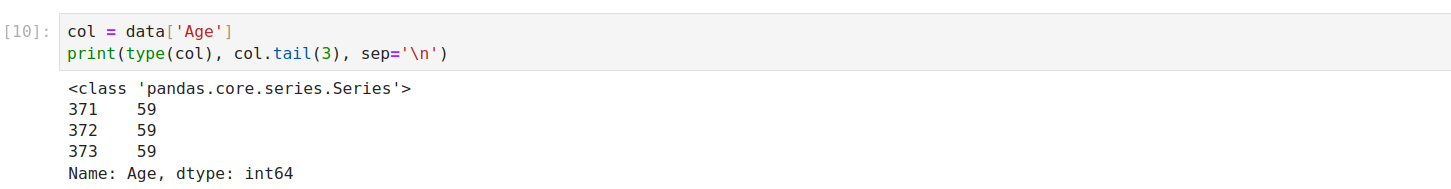
Замена столбца с давлением хх/хх на столбцы High Pressure и Low Pressure:



Извлечение данных из строки по ее номеру:

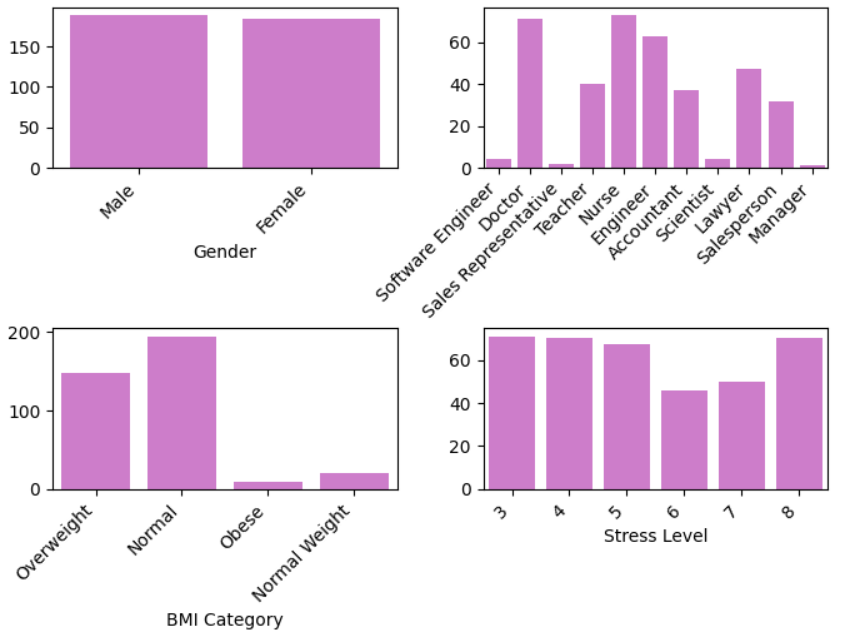


Извлечение данных из столбца по его названию, вывод последних 3 значений:

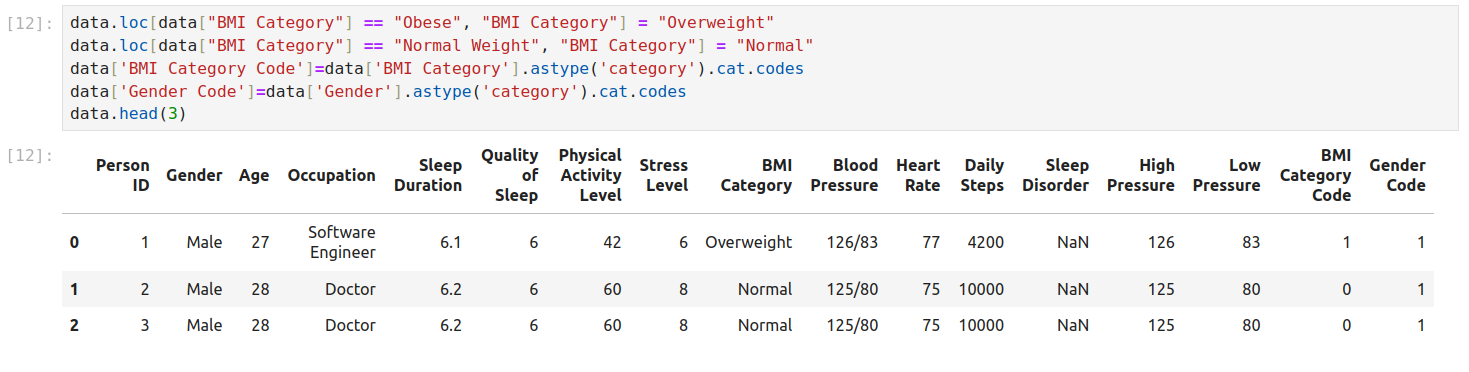


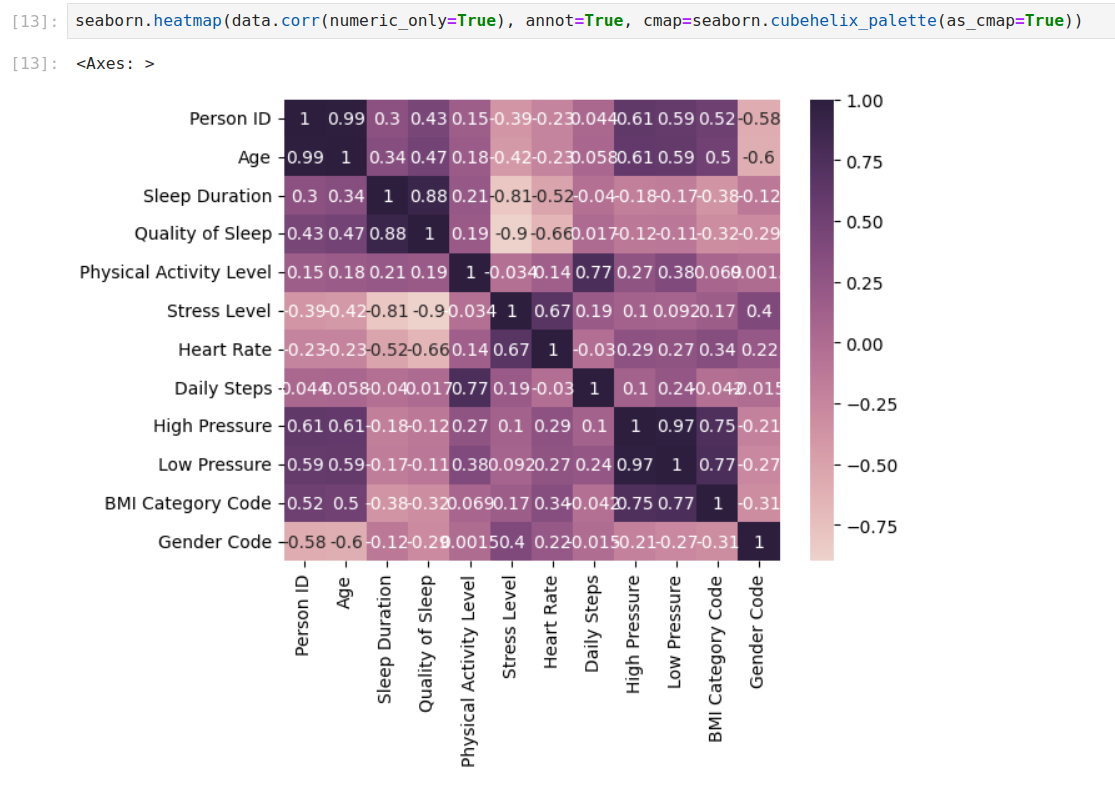
Построить столбчатые диаграммы:





Заменим значения Obese на Overweight и Normal Weight на Normal. Создадим категории для гендера и индекса массы тела:

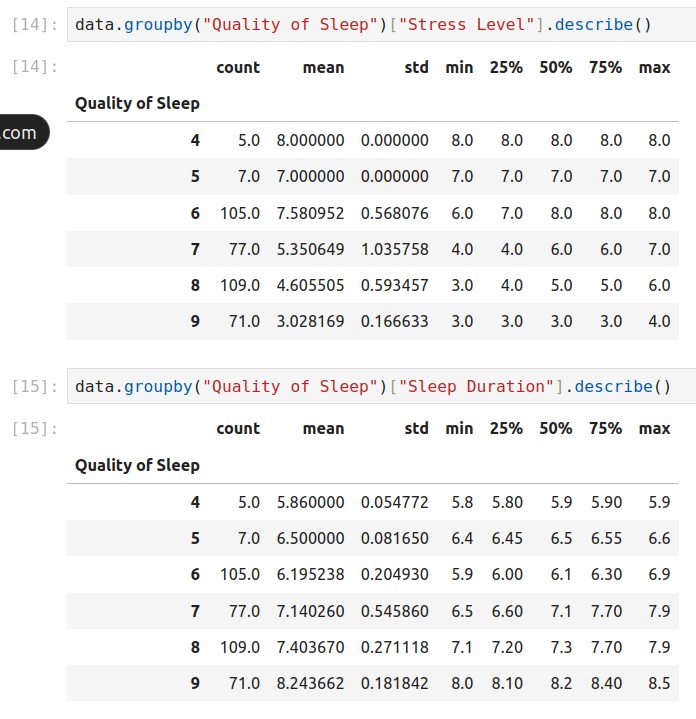
Построим матрицу корреляции, используя столбцы с численными данными:



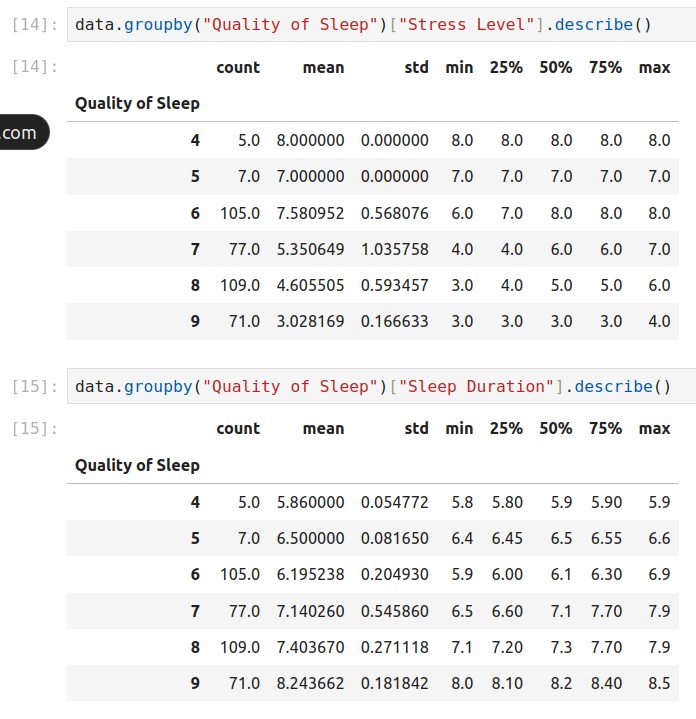
Можно предположить о существовании сильной зависимости между следующими парами столбцов:

* Quality of Sleep, Sleep Duration
* Quality of Sleep, Stress Level

Сгруппируем данные по Quality of Sleep и и выведем статистические показатели по Stress Level. Заметим обратную зависимость между Quality of Sleep и средним значением Stress Level.



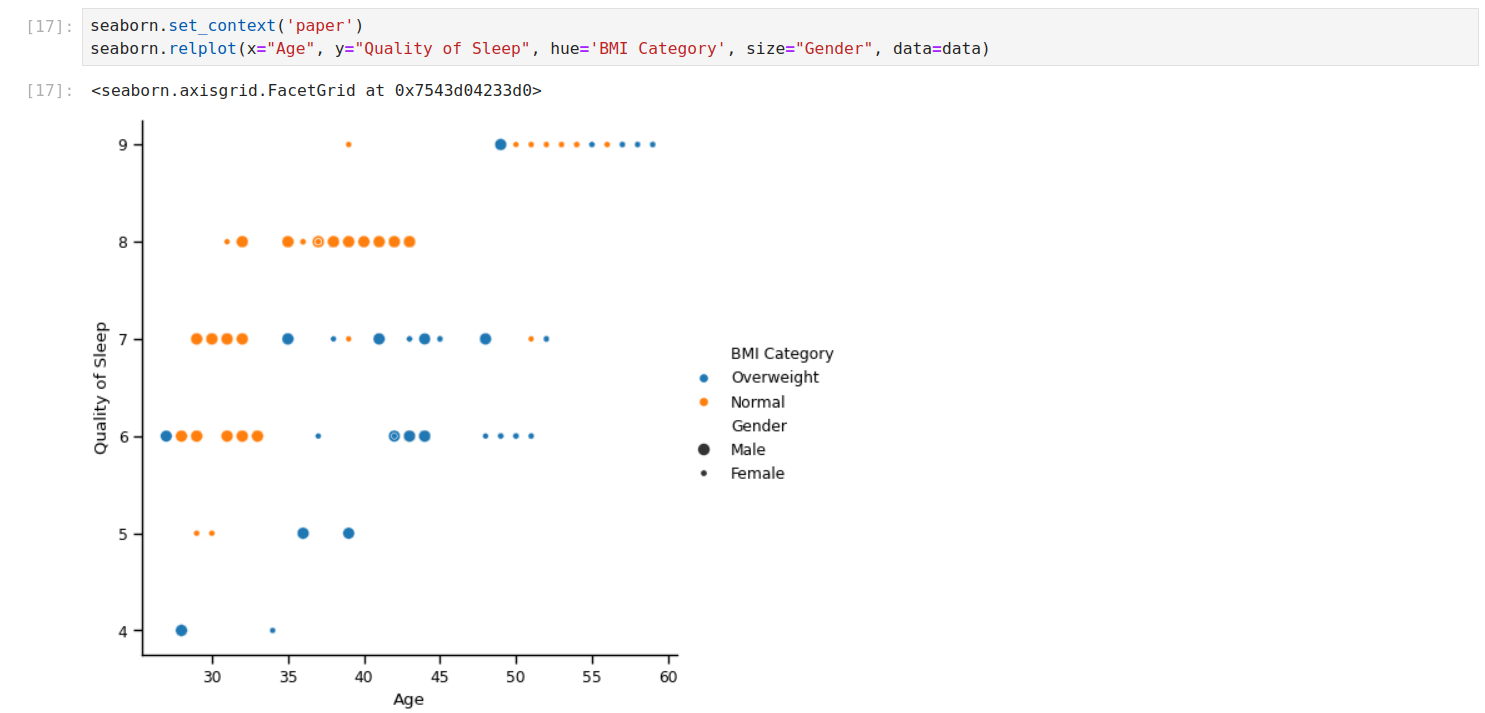
Аналогично заметим прямую зависимость между Quality of Sleep и средним значением Sleep Duration.



Вычислим среднее значение Quality of Sleep для людей со Stress Level >= 8 и лишним весом, для людей со Stress Level < 6 и нормальным весом, их отношение. Можно сделать вывод, что высокий уровень стресса в совокупности с лишним весом могут значительно сокращать Quality of Sleep.



Используем диаграмму рассеивания:



Среди людей в возрасте 40+ гораздо больше людей с лишним весом, чем в категории 30-40 лет.

