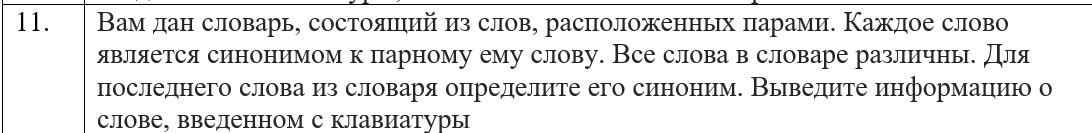
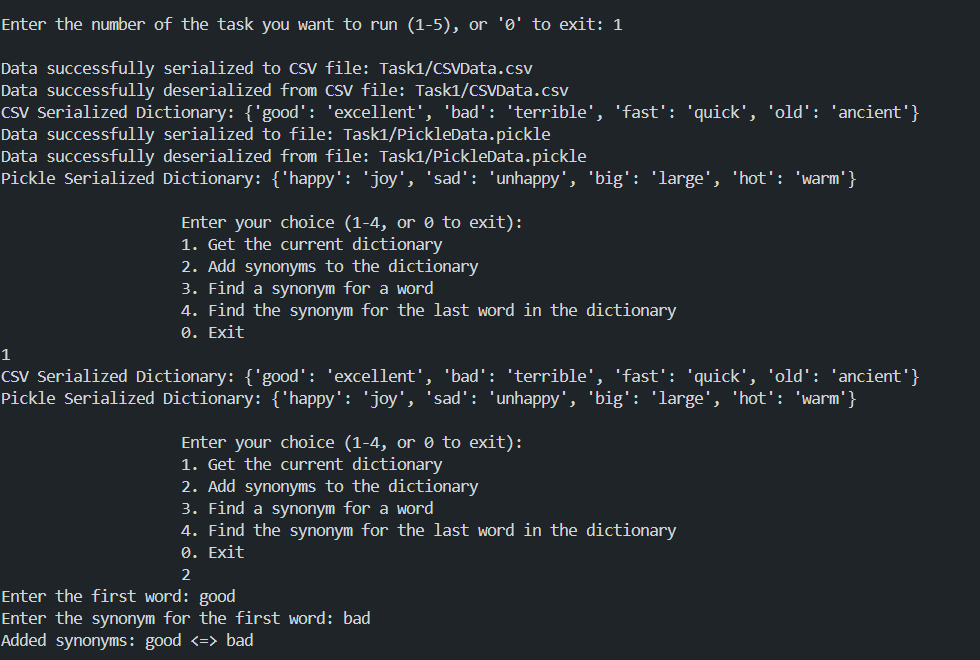
Лабораторная работа №4

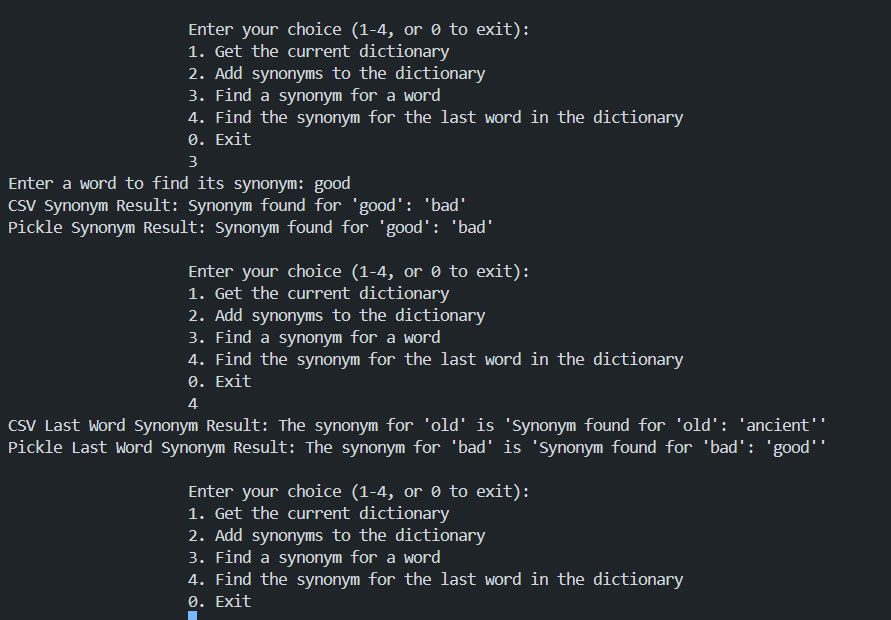
**Тема**: Работа с файлами, классами, сериализаторами, регулярными выражениями и стандартными библиотеками.

**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

**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







**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.

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

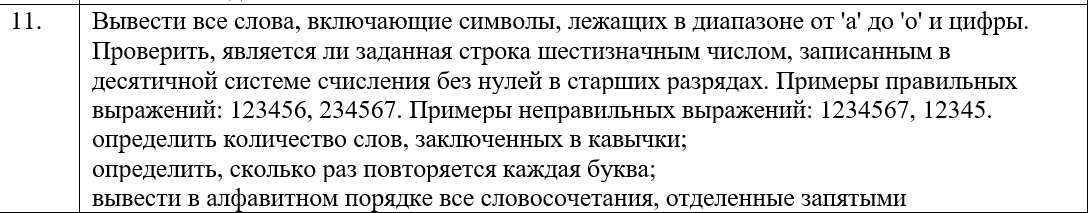
* количество предложений в тексте;
* количество предложений в тексте каждого вида отдельно (повествовательные, вопросительные и побудительные);
* среднюю длину предложения в символах (считаются только слова);
* среднюю длину слова в тексте в символах;
* количество смайликов в заданном тексте. Смайликом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

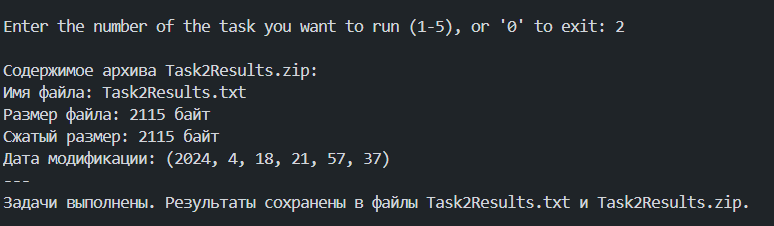
первым символом является либо «;» (точка с запятой) либо «:» (двоеточие) ровно один раз;

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

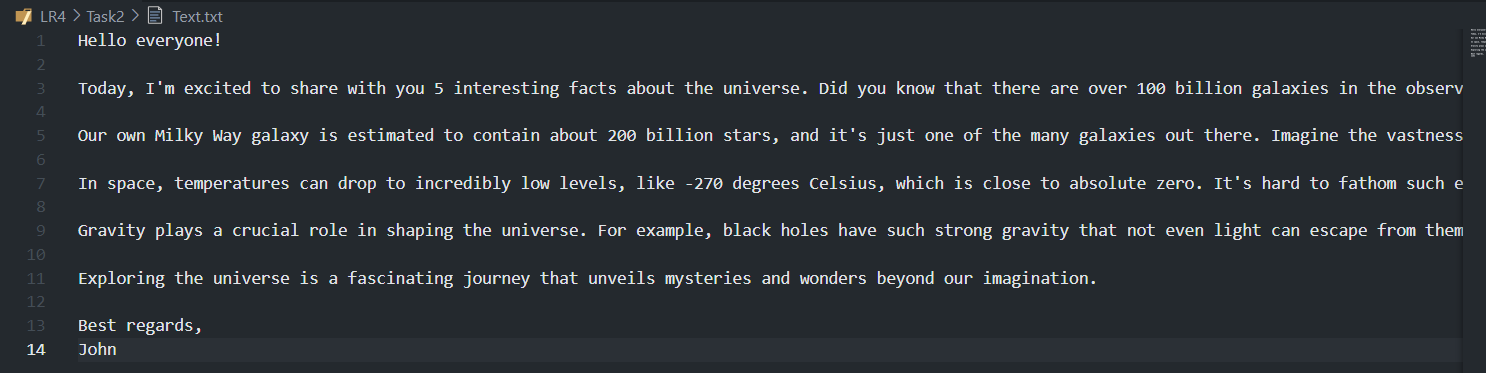
в конце обязательно идет некоторое количество (не меньше одной) одинаковых скобок из следующего набора: «(», «)», «[», «]»;

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

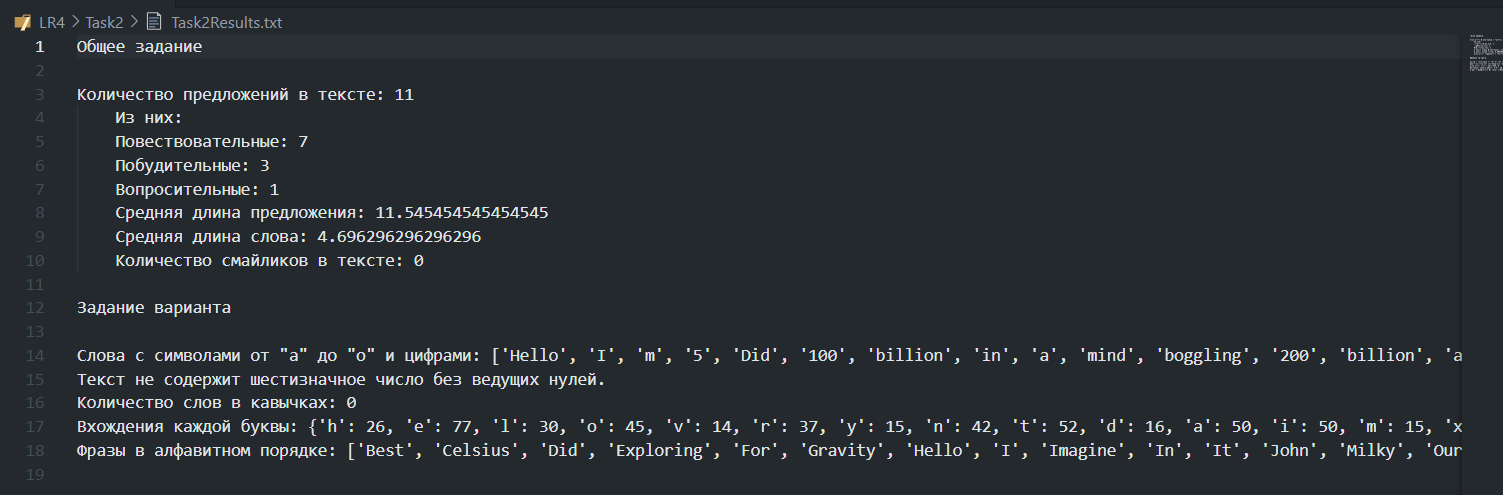




Файл с исходным текстом:



Файл с результатами:



**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;

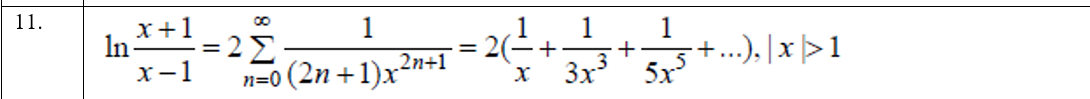
б) с помощью библиотеки matplotlib нарисовать графики разных цветов в одной координатной оси:

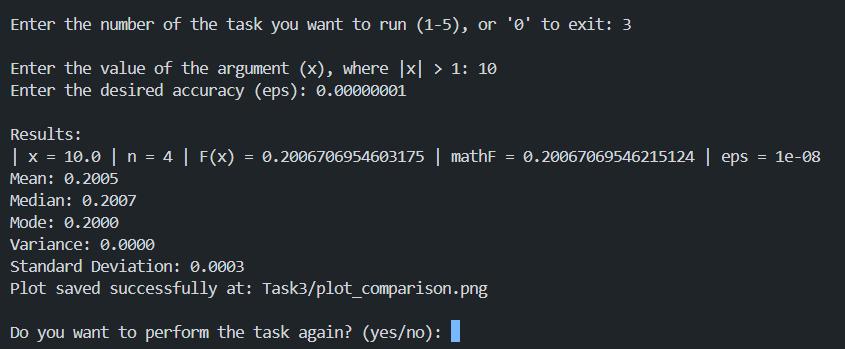
* график по полученным данным разложения функции в ряд, представленным в таблице,
* график соответствующей функции, представленной с помощью модуля math. Обеспечить отображение координатных осей, легенды, текста и аннотации.

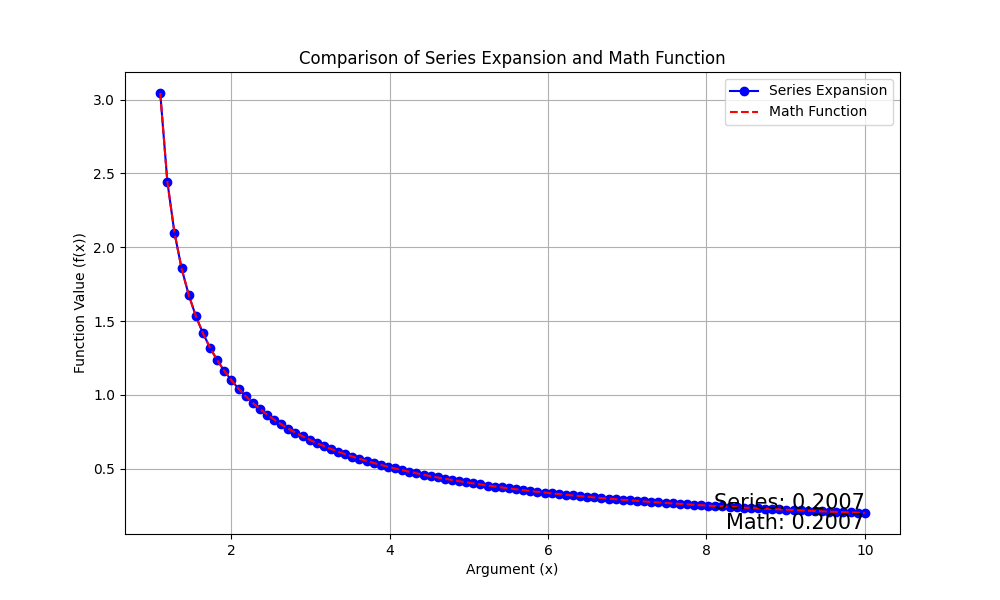


Здесь x – значение аргумента, F(x) – значение функции, n – количество просуммированных членов ряда, Math F(x) – значение функции, вычисленное с помощью модуля math.

в) сохранить графики в файл







**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> )

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

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.

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.

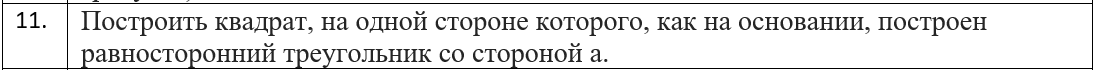
Программа должна содержать следующие базовые функции:

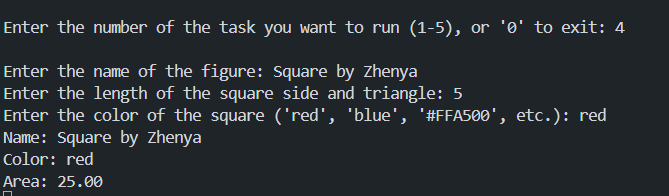
1. ввод значений параметров пользователем;

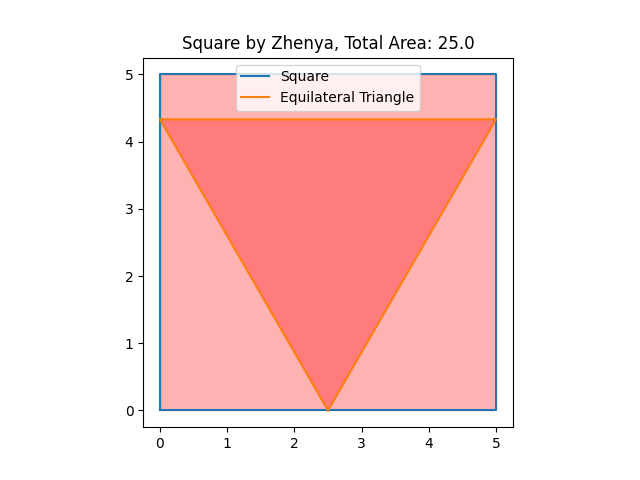
2) проверка корректности вводимых данных;

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.







**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()

