**Пояснительная записка**

В текущей директории представлено решение поставленной задачи. Было разработано 4 класса, позже преобразованных в отдельные модули, 3 функции для обработки данных и 1 основной файл для запуска всего алгоритма.

**Условие задачи**

Необходимо разработать решение, для работы с плоской геометрической фигурой, размещённой в координатной сетке. Из возможных фигур представлены: Круг, Треугольник, Прямоугольник. У каждой фигуры есть общий параметр – цвет (все возможные значения: {красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый}. Так же, для каждой фигуры должен быть разработан алгоритм вычисления её площади. В последующем, должен быть разработан алгоритм сортировки Шелла, в качестве аргумента берётся площадь фигуры.

**Метрики**

Для запуска программы необходимо переместиться в консоли(командной строке) в необходимую директорию и прописать команду:  
*python3 main.py <input file> <output file>*где input file – это название файла, в котором находятся все исходные данные

output file – это название файла, в который будет выводиться информация о работе программы.

Для корректной работы программы следует придерживаться некоторых правил при создании input file. Каждая строка в таком файле представляет отдельный объект для создания. Первым словом должен идти тип объекта (circle, rectangle, triangle, figure). Второе слово – общий параметр для всех типов объектов, а именно цвет. Стоит помнить, что при несовпадении типа объекта или цвета с допустимыми значениями, объект не будет создан и выведется сообщение об этом в консоль. После этих двух “слов” идут целочисленные аргументы для создания объекта класса (кол-во аргументов тоже стоит учитывать). Пример правильного файла:

Изображение выглядит как текст

Автоматически созданное описание

**Структура программы**

Вся программа поделена на 7 файлов: 6 файлов-модулей и 1 основной файл исполнения.

*figure.py circle.py rectangle.py triangle.py shell\_sorting.py read\_write.py main.py*

Теперь рассмотрим каждый файл по отдельности.

*figure.py*

Присутствующие имена:

* Figure – родительский класс для всех фигур
  + def \_\_init\_\_(color) - функция для инициализации объекта класса
    - color: string – цвет фигуры
  + def get\_area() – функция для получения площади объекта
  + def \_\_str\_\_() – функция преобразования объекта к типу данных string
  + def \_\_repr\_\_() – функция репрезентации объекта в итерируемых типах данных

*circle.py*

Присутствующие имена:

* Circle – класс для создания окружностей
  + def \_\_init\_\_(color, x, y, r) - функция для инициализации объекта класса
    - color: string – цвет фигуры
    - x: int – x-координата центра круга
    - y: int – y-координата центра круга
    - r: int – радиус круга
  + def get\_area() – функция для получения площади треугольника
  + def \_\_str\_\_() – функция преобразования объекта к типу данных string
  + def \_\_repr\_\_() – функция репрезентации объекта в итерируемых типах данных
* math.py – модуль для математических операций
* figure.py – модуль с родительским классом фигуры

*rectangle.py*

Присутствующие имена:

* Rectangle – класс для создания прямоугольников
  + def \_\_init\_\_(color, x1, y1, x2, y2) - функция для инициализации объекта класса
    - color: string – цвет фигуры
    - x1: int – x-координата левого верхнего угла прямоугольника
    - y1: int – y-координата левого верхнего угла прямоугольника
    - x2: int – x-координата правого нижнего угла прямоугольника
    - y2: int – y-координата правого нижнего угла прямоугольника
  + def get\_area() – функция для получения площади прямоугольника
  + def \_\_str\_\_() – функция преобразования объекта к типу данных string
  + def \_\_repr\_\_() – функция репрезентации объекта в итерируемых типах данных
* figure.py – модуль с родительским классом фигуры

*triangle.py*

Присутствующие имена:

* Triangle – класс для создания треугольиков
  + def \_\_init\_\_(color, x1, y1, x2, y2, x3, y3) - функция для инициализации объекта класса
    - color: string – цвет фигуры
    - x1: int – x-координата первого угла треугольника
    - y1: int – y-координата первого угла треугольника
    - x2: int – x-координата второго угла треугольника
    - y2: int – y-координата второго угла треугольника
    - x3: int – x-координата третьего угла треугольника
    - y3: int – y-координата третьего угла треугольника
  + def get\_area() – функция для получения площади прямоугольника
  + def \_\_str\_\_() – функция преобразования объекта к типу данных string
  + def \_\_repr\_\_() – функция репрезентации объекта в итерируемых типах данных
* figure.py – модуль с родительским классом фигуры

*shell\_sorting.py*

Присутствующие имена:

* def shell\_sort\_by\_area(obj\_list, reverse) – функция для сортировки списка объектов плоских фигур по убыванию в зависимости от их площади
  + obj\_list: list – список объектов плоских фигур
  + reverse: boolean – если True, то вернётся отсортированный список по возрастанию, по умолчанию False

*read\_write.py*

Присутствующие имена:

* COLORS: list[…] – список всех возможных цветов для создания плоской фигуры
* def read\_info(filename) – функция считывания и создания объектов плоских фигур из файла
  + filename: string – название файла, в котором хранятся входные данные
* def write\_info(info, data, filename) – функция для записи всей необходимой информации о работе программы в файл
  + info: list[string, string, string] – список аргументов, переданных через командную строку при запуске программы
  + data: list[…] – отсортированный список объектов плоских фигур
  + filename: string – название файла, для записи выходной информации
* figure.py – модуль с родительским классом фигуры
* circle.py – модуль с классом фигуры Круг
* rectangle.py – модуль с классом фигуры Прямоугольник
* triangle.py – модуль с классом фигуры Треугольник

*main.py*

Присутствующие имена:

* args: list[string, string, string] - список аргументов, переданных через командную строку при запуске программы
* input\_file: string – название входного файла
* output\_file: string – название выходного файла
* fig\_list: list[…] – список всех объектов плоских фигур
* read\_write.py – модуль с функциями для записи и чтения информации о плоских фигурах
* shell\_sorting.py – модуль для сортировки списка объектов плоских фигур
* sys.py – модуль для работы с аргументами консоли и различными системными действиями

**Примеры работы программы**

* входной файл

Изображение выглядит как текст

Автоматически созданное описание

* выходной файл

Изображение выглядит как текст

Автоматически созданное описание

* сообщения в консоли

Изображение выглядит как текст

Автоматически созданное описание

**Тестирование программы**

В ходе тестирования программы был разработал алгоритм генерации объёмной выборки данных в файл для дальнейшего использования их в программе под названием:

*big\_data\_generator.py*

Присутствующие имена:

* COLORS: list[…] – список всех возможных цветов для создания плоской фигуры
* CLASSES: dict[string: int …] – словарь всех возможных типов фигур с количеством аргументов, необходимых для её создания
* filename: string – название выходного файла без расширения
* fig\_list: list[…] – список всех объектов плоских фигур
* random.py – модуль с функциями для работы с псевдослучайными числами