Московский государственный технический

университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python.»

Выполнил: Тянутов Александр Дмитриевич

студент группы : ИУ5-31Б

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

.

Подпись и дата:

Москва, 2022 г.

Постановка задачи

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.

2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.

3. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab_python_oop.

4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab_python_oop.

5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь.

6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь.

7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.

8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math.

9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:

    - Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - https://pyformat.info/

- Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

10. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - https://docs.python.org/3/library/__main__.html). Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):

    - Прямоугольник синего цвета шириной N и высотой N.
    - Круг зеленого цвета радиусом N.
    - Квадрат красного цвета со стороной N.
    - Также вызовите один из методов внешнего пакета, установленного с использованием pip.

Текст программы

circle.py

```
from lab_python_oop.geom_figure import GeometricFigure
from lab_python_oop.color import Color
from math import pi

class Circle(GeometricFigure):
    FIGURE_TYPE='Circle'

    def __init__(self, radius, color):
        self.col = Color()
        self.col.set_x(color)
        self.rad = radius
    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE
    def virtual_calculate_figure_square(self):
        return pi*(self.rad**2)
    def __repr__(self) -> str:
        return '{} | Color {} | Radius {} | Area {}'.format(
            Circle.get_figure_type(),
            self.col.get_x(),
            self.rad,
            round(self.virtual_calculate_figure_square(), 2)
        )
```

color.py

```
class Color:
    def __init__(self):
        self._x = None
    def get_x(self):
        return self._x
    def set_x(self, val):
        self._x = val
    def del_x(self):
        del self._x

    x = property(get_x, set_x, del_x, 'Color')
```

geom_figure.py

```
from abc import ABC, abstractmethod
class GeometricFigure(ABC):
    @abstractmethod
    def virtual_calculate_figure_square(self):
        raise NotImplementedError('virtual_calculate_figure_square not implemented!')
```

rectangle.py

```
from lab_python_oop.geom_figure import GeometricFigure
from lab_python_oop.color import Color

class Rectangle(GeometricFigure):
    FIGURE_TYPE='Rectangle'
    def __init__(self, width, height, color):
        self.w = width
        self.h = height
        self.col = Color()
        self.col.set_x(color)

    def virtual_calculate_figure_square(self):
        return self.w*self.h

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __repr__(self):
        return '{} | Color {} | Width {} | Height {} | Area {}'.format(
            self.get_figure_type(),
            self.col.get_x(),
            self.w,
            self.h,
            self.virtual_calculate_figure_square()
        )
```

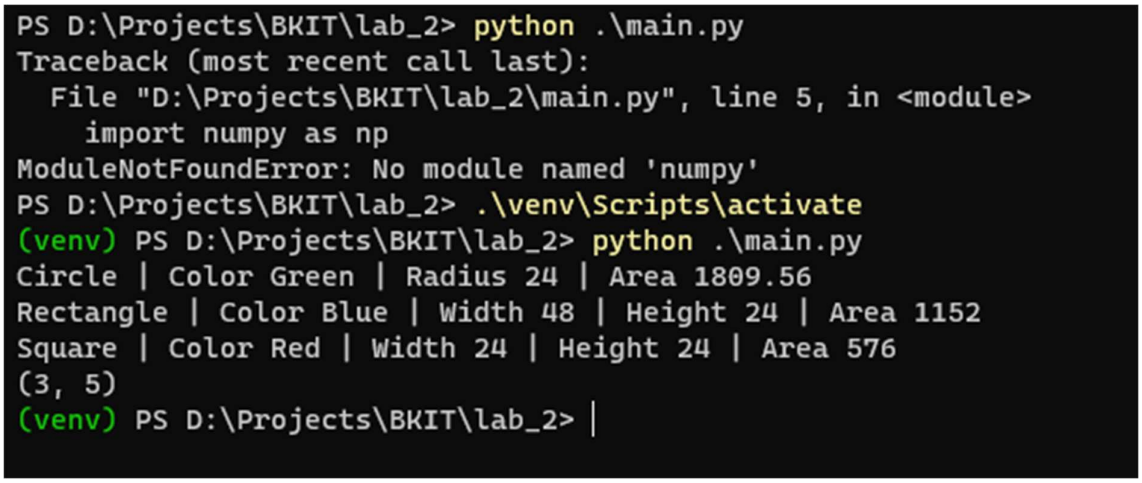
square.py

```
from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    FIGURE_TYPE='Square'

    def __init__(self, side, color):
        super().__init__(side, side, color)

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE
    def __repr__(self):
        return super().__repr__()
```

main.py

```
from ftplib import CRLF
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
import numpy as np

if __name__ == '__main__':
    crcl = Circle(24, 'Green')
    rect = Rectangle(48, 24, 'Blue')
    sq = Square(24, 'Red')
    a = np.arange(15).reshape(3, 5)
    print(crcl)
    print(rect)
    print(sq)
    print(a.shape)
```

Анализ результатов



По результатам анализа видим, что когда virtual environment не активирована библиотека numpy не может быть запущена в отличии от варианта когда venv активирована

Вывод

Освоил объектно-ориентированные возможности языка Python. Научился создавать virtual environment и загружать туда библиотеки, а так же начальную работу с классами и абстрактными классами еще с наследованиями и super классами.