

**Московский государственный технический
университет им. Н.Э. Баумана**

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

Курс «Парадигмы и конструкции языков программирования»
Отчет по лабораторной работе №3
«Объектно-ориентированные возможности языка Python»

Выполнил:
студент группы ИУ5-31Б

Сомов Кирилл

Подпись и дата:

Проверил:
преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023 г.

Задание

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

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

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

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

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

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).

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.

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

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.

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.

Код программы

Main.py

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
import requests

def main():
    width = 16
    height = 16
    radius = 16
    side = 16

    blue_rectangle = Rectangle(width, height, "blue")
    green_circle = Circle(radius, "green")
    red_square = Square(side, "red")

    print(blue_rectangle.width)
    # Выводим информацию о фигурах
    print(blue_rectangle)
    print(green_circle)
    print(red_square)

    url = 'https://www.wikipedia.org'

    # Выполнение GET-запроса
    response = requests.get(url)

    if response.status_code == 200:
        print(response.text)
    else:
        print(f"Ошибка: Статус-код {response.status_code}")
```

```
if __name__ == "__main__":
    main()
```

geometrical_figure.py

```
from abc import ABC, abstractmethod

class GeometricalFigure(ABC):
    @abstractmethod
    def calculate_area(self):
        pass
```

color.py

```
class Color:
    def __init__(self, color):
        self.color = color
```

circle.py

```
import math
from .geometrical_figure import GeometricalFigure
from .color import Color

class Circle(GeometricalFigure):
    def __init__(self, radius, color):
        self.radius = radius
        self.color = Color(color)

    def calculate_area(self):
        return math.pi * self.radius ** 2

    def __repr__(self):
        return "Circle: radius={}, color={}, area={}".format(
            self.radius, self.color.color, self.calculate_area()
        )
```

square.py

```
from .rectangle import Rectangle

class Square(Rectangle):
    def __init__(self, side_length, color):
        super().__init__(side_length, side_length, color)

    def __repr__(self):
        return "Square: side_length={}, color={}, area={}".format(
            self.width, self.color.color, self.calculate_area()
        )
```

rectangle.py

```
from .geometrical_figure import GeometricalFigure
from .color import Color

class Rectangle(GeometricalFigure):
    def __init__(self, width, height, color):
        self.width = width
        self.height = height
        self.color = Color(color)
```

```
def calculate_area(self):
    return self.width * self.height

def __repr__(self):
    return "Rectangle: width={}, height={}, color={}, area={}".format(
        self.width, self.height, self.color.color, self.calculate_area()
    )
```

Вывод

```
warnings.warn(
16
Rectangle: width=16, height=16, color=blue, area=256
Circle: radius=16, color=green, area=804.247719318987
Square: side_length=16, color=red, area=256
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
```