

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

Факультет Радиотехнический  
Кафедра РТ5

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №2  
«Объектно-ориентированные возможности языка Python.»

Выполнил:

студент группы РТ5-31Б:  
Коровин К.С.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023

## Описание задания

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](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

### 1. circle.py

```
from geometric_shape import GeometricShape
from color import Color
import math

class Circle(GeometricShape):
    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()
        )
```

### 2. rectangle.py

```
from geometric_shape import GeometricShape
from color import Color

class Rectangle(GeometricShape):
    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()
        )
```

### 3. 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()
        )
```

### 4. geometric\_shape.py

```
from abc import ABC, abstractmethod

class GeometricShape(ABC):
```

```

    @abstractmethod
    def calculate_area(self):
        pass

```

## 5. color.py

```

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

```

## 6. main.py

```

from colorama import init, Fore
from rectangle import Rectangle
from square import Square
from circle import Circle

def main():
    init(autoreset=True)
    blue_rectangle = Rectangle(24, 24, Fore.BLUE)
    green_circle = Circle(24, Fore.GREEN)
    red_square = Square(24, Fore.RED)

    print(blue_rectangle)
    print(green_circle)
    print(red_square)

if __name__ == "__main__":
    main()

```

## Экранные формы с примерами выполнения программы

Rectangle - Width: 24, Height: 24, Color: , Area: 576

Circle - Radius: 24, Color: , Area: 1809.5573684677208

Square - Side Length: 24, Color: , Area: 576