



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования «Московский государственный технический  
университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

**Лабораторная работа №2**

**по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнил:**

**студент группы ИУ5-35Б**

**Емельянова Т.И.**

### Задание:

1. Необходимо создать виртуальное окружение и установить в нем хотя бы один внешний пакет с использованием `pip`.
2. Необходимо программу, реализующую работу с классами. Программа должна быть добавлена 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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». Второе 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](https://docs.python.org/3/library/math.html).
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
  - Определите метод "get", 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 классов (используйте следующую конструкцию - <https://docs.python.org/3/library/main.html>). Создайте следующие объекты.
  - 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`.

main.py

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.quadrate import Quadrate
from lab_python_oop.circle import Circle
from PIL import Image
```

```
def main():
    number = 5
    r = Rectangle(number, number, 'синего')
    c = Circle(number, 'зелёного')
    q = Quadrate(number, 'красного')
    print(r, c, q, sep='\n')
    flower_image = Image.open('flower.png')
    flower_image.show()
```

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

circle.py

```
from lab_python_oop.figure import GFigure
from lab_python_oop.color import Color
from math import pi
```

```
class Circle(GFigure):
```

```
    figure_name = 'Круг'
```

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

```
    @classmethod
```

```
    def get_name(cls):
        return cls.figure_name
```

```
    def square(self):
        return pi * self.radius ** 2
```

```
    def __repr__(self):
        return '{} {} цвета, с радиусом {} и площадью
```

```

{: .4f}.'.format(
    Circle.get_name(),
    self.f_color.color,
    self.radius,
    self.square()
)

```

color.py

```
class Color:
```

```

    def __init__(self):
        self._color = None

```

```

@property
def color(self):
    return self._color

```

```

@color.setter
def color(self, color):
    self._color = color

```

```

@color.deleter
def color(self):
    self._color = None

```

figure.py

```
from abc import ABC, abstractmethod
```

```
class GFigure(ABC):
```

```

    @abstractmethod
    def square(self):
        pass

```

quadrade.py

```
from lab_python_oop.rectangle import Rectangle
```

```
class Quadrade(Rectangle):
```

```
    figure_name = 'Квадрат'
```

```

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

@classmethod
def get_name(cls):
    return cls.figure_name

def square(self):
    return self.side ** 2

def __repr__(self):
    return '{} {} цвета, со стороной {} и площадью {}'.format(
        Quadrate.get_name(),
        self.f_color.color,
        self.side,
        self.square()
    )

```

rectangle.py

```

from lab_python_oop.color import Color
from lab_python_oop.figure import GFigure

```

```

class Rectangle(GFigure):

```

```

    figure_name = 'Прямоугольник'

```

```

def __init__(self, width, height, color):
    self.width = width
    self.height = height
    self.f_color = Color()
    self.f_color.color = color

```

```

@classmethod

```

```

def get_name(cls):
    return cls.figure_name

```

```

def square(self):
    return self.height * self.width

```

```

def __repr__(self):
    return '{} {} цвета, шириной {}, высотой {} и площадью {}.'.format(
        Rectangle.get_name(),

```

```
self.f_color.color,  
self.width,  
self.height,  
self.square()  
)
```

flower.png



Результат работы main.py(N=5):

