



**Московский государственный технический университет**

**им. Н.Э. Баумана**

**(МГТУ им. Н.Э. Баумана)**

**Радиотехнический факультет (РТ)**

Отчёт по лабораторной работе № 3

По дисциплине

«Разработка интернет-приложений»

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Ю.Е.

Подпись: \_\_\_\_\_

«\_\_» \_\_\_\_\_ 201\_\_ г.

Выполнил:

студент группы РТ5-51Б

Ануров Н.С.

Подпись: \_\_\_\_\_

«\_\_» \_\_\_\_\_ 201\_\_ г.

Москва, 2019

# Задание

## Необходимо разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
  1. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь - <https://docs.python.org/3/library/abc.html>
  2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь - <https://docs.python.org/3/library/functions.html#property>
  3. 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.
  4. 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>
  5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
  6. Для классов «Прямоугольник», «Квадрат», «Круг»:
    1. Определите метод "get", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
    2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
4. 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:
  1. Прямоугольник синего цвета шириной 3 и высотой 2.
  2. Круг зеленого цвета радиусом 5.
  3. Квадрат красного цвета со стороной 5.

## Исходный код

### Main

```
from lab_python_oop.Rectangle import Rectangle
from lab_python_oop.Square import Square
from lab_python_oop.Circle import Circle
if __name__ == "__main__":
    r1 = Rectangle(3, 2, "Синий")
    print(r1)
    s1 = Square(5, "Красный")
    print(s1)
    c1 = Circle(5, "Зелёный")
    print(c1)
```

## Square

```
from lab_python_oop.Figure import Figure
from lab_python_oop.ColorF import ColorF
from lab_python_oop.Rectangle import Rectangle

class Square(Rectangle):
    def __init__(self, d, color, name="квадрат"):
        self.d=d
        self.color=color
        self.name=name
    def __repr__(self):
        return '{} {} со стороной {}'.format(self.color, self.name, self.d)
    def S(self, d):
        return self.d*self.d
```

## Rectangle

```
from lab_python_oop.Figure import Figure
from lab_python_oop.ColorF import ColorF

class Rectangle (Figure):
    def __repr__(self):
        return '{} {} высотой {} и шириной {}'.format(self.color, self.name,
self.height, self.width)
    def __init__(self, width, height, color, name="прямоугольник"):
        self.width = width
        self.height = height
        self.color=color
        self.name=name
    def S(self, width, height):
        return self.width * self.height
```

## Figure

```
from abc import ABCMeta, abstractmethod
class Figure(object):
    __metaclass__ = ABCMeta
    @abstractmethod
    def S(self, x, y):
        pass
```

## Color

```
class ColorF:
    def __init__(self, color):
        self.color = color
    @property
    def svoistv(self):
        return self.color
```

## Circle

```
from lab_python_oop.Figure import Figure
from lab_python_oop.ColorF import ColorF
import math
class Circle(Figure):
    def __init__(self, R, color, name="круг"):
        self.R = R
        self.color=color
        color=ColorF(self.color)
```

```
self.name=name

def S(self,R):
    return self.R * self.R*math.pi
def __repr__(self):
    return '{} {} радиусом {}'.format(self.color, self.name, self.R)
```

## Скриншоты

Синий прямоугольник высотой 2 и шириной 3

Красный квадрат со стороной 5

Зелёный круг радиусом 5