



**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»

**Отчет по
лабораторной работе №2
по курсу «Разработка интернет-приложений»**

**Выполнил:
студент группы ИУ5-54Б
Сысойкин Е.М.**

**Проверил:
Доцент каф. ИУ5
Гапанюк Ю.Э.**

1. Задание

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`.

2. Текст 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 numpy

def main():
    rect = Rectangle(3, 8, "#0000FF")
    circle = Circle(5, "#00FF00")
    square = Square(10, "#FF0000")
    print("", rect, circle, square)
    print(numpy.linspace(0, 1, 5))

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

lab_python_oop/Figure.py:

```
from abc import ABCMeta, abstractmethod

class Figure(metaclass=ABCMeta):

    @abstractmethod
    def area(self):
        pass
```

lab_python_oop/FigureColor.py:

```
class FigureColor:
    def __init__(self):
        self._color = None

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

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

    @color.deleter
    def color(self):
        del self._color
```

lab_python_oop/Rectangle.py:

```
from lab_python_oop.Figure import Figure
from lab_python_oop.FigureColor import FigureColor

class Rectangle(Figure):
    type="Rectangle"

    @classmethod
    def get_type(cls):
        return cls.type

    def __init__(self, w, h, c):
        self.width = w
        self.height = h
        self.color = FigureColor()
        self.color.color = c

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

    def __repr__(self):
        return "{} width={}, height={}, area={}, color={} \n" \
            .format(
                self.get_type(),
                self.width,
                self.height,
                self.area(),
                self.color.color
            )
```

lab_python_oop/Square.py:

```
from lab_python_oop.Rectangle import Rectangle

class Square(Rectangle):
    type = "Square"

    def __init__(self, s, c):
        super().__init__(s, s, c)

    def __repr__(self):
        return "{} side={}, area={}, color={} \n" \
            .format(
                self.get_type(),
                self.width,
                self.area(),
                self.color.color
            )
```

lab_python_oop/Circle.py:

```
from lab_python_oop.Figure import Figure
from lab_python_oop.FigureColor import FigureColor
from math import pi

class Circle(Figure):
    type = "Circle"

    @classmethod
    def get_type(cls):
        return cls.type
```

```

def __init__(self, r, c):
    self.radius = r
    self.color = FigureColor()
    self.color.color = c

def area(self):
    return 2 * pi * self.radius * self.radius

def __repr__(self):
    return "{} radius={}, area={}, color={} \n" \
        .format(
            self.get_type(),
            self.radius,
            self.area(),
            self.color.color
        )

```

Пример выполнения программы(main.py):

```

(env) [tujh@tujhNotebook env]$ python main.py
Rectangle width=3, height=8, area=24, color=#0000FF
Circle radius=5, area=157.07963267948966, color=#00FF00
Square side=10, area=100, color=#FF0000

[0.    0.25 0.5   0.75 1.   ]
(env) [tujh@tujhNotebook env]$ 

```