



**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»
Курс «Базовые компоненты интернет-технологий»
Объектно-ориентированные возможности языка
Python**

**Выполнил:
Студент группы ИУ5-34Б
Сергеев Никита
Дата и подпись:**

**Преподаватель:
Гапанюк Ю.Е.
Дата и подпись:**

2021 г.

Постановка задачи

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.

Текст программы

Файл Circle.py

```
from lab_python_oop.geoFigure import geoFigure
from lab_python_oop.ColorFigure import ColorFigure
import math

class Circle (geoFigure):
    figure_type = "Круг"
    def __init__ (self, color, radius):
        self.radius = radius
        self.f_color = ColorFigure()
        self.f_color.color = color

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

    def __repr__(self):
        return '{} {} цвета радиуса {} площадью {}'.format(
            Circle.get_figure_type(),
            self.f_color.color,
            self.radius,
            self.square()
        )
```

Файл ColorFigure.py

```
# Цвет фигур
class ColorFigure:
    def __init__(self):
        self._color = None

# Использование декоратора
@property

    def color(self):
        return self._color

    @color.setter

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

    @color.deleter

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

Файл geoFigure.py

```
from abc import ABC, abstractmethod

class geoFigure(ABC):

    @abstractmethod
    def square(self):
        pass

    @classmethod
    def get_figure_type(cls):
        return cls.figure_type
```

Файл rectangle.py

```
# Класс прямоугольник
from lab_python_oop.geoFigure import geoFigure
from lab_python_oop.ColorFigure import ColorFigure

class rectangle(geoFigure):
    figure_type = "прямоугольник"
# Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В
конструкторе создается объект класса «Цвет фигуры» для хранения цвета
    def __init__(self, color, width, height):
        self.width = width
        self.height = height
        self.f_color = ColorFigure()
        self.f_color.color = color

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

    def __repr__(self):
        return '{} {} цвета шириной {} и высотой {} площадью {}'.format(
            rectangle.get_figure_type(),
            self.f_color.color,
            self.width,
            self.height,
            self.square()
        )
```

Файл square.py

```
from lab_python_oop.rectangle import rectangle
# Класс квадрат
class Square(rectangle):
    figure_type = "Квадрат"
    def __init__(self, color, side):
        super().__init__(color, side, side)
    def __repr__(self):
        return f'{Square.get_figure_type()} {self.f_color.color} цвета со стороной\'
```

```
f" {self.width} и площадью {self.square()}"
```

Файл main.py

```
from lab_python_oop.rectangle import rectangle
from lab_python_oop.Circle import Circle
from lab_python_oop.square import Square
```

```
def main():
    r = rectangle("синего", 6, 2)
    c = Circle("зеленого", 10)
    s = Square("красного", 4)
    print(r)
    print(c)
    print(s)
```

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

Результат выполнения

```
PS D:\лабы\2> & D:/Python/python.exe d:/лабы/2/main.py
прямоугольник синего цвета шириной 6 и высотой 2 площадью 12.
Круг зеленого цвета радиуса 10 площадью 314.1592653589793.
Квадрат красного цвета со стороной 4 и площадью 16
PS D:\лабы\2> █
```