



**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 БКИТ

“Объектно-ориентированные возможности языка Python.”

Выполнил:
студент группы ИУ5-31Б
Ашуров Г. В.

Проверил:
Гапанюк Юрий Евгеньевич

2022г.

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 "getr", 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.

Файл main.py:

```
from lab_python_oop import Circle
from lab_python_oop import Square
from lab_python_oop import Rectangle
import cowsay

if __name__ == "__main__":

    a = Rectangle.Rectangle(6, 6, 'Синий', 'Прямоугольник')
    b = Circle.Circle(6, 'Зеленый', 'Круг')
    c = Square.Square(6, 'Красный', 'Квадрат')

    cowsay.trex(str(a)+'\n'+str(b)+'\n'+str(c))
```

Файл Circle.py:

```
import math
from .Shape import Shape
class Circle(Shape):
    def __init__(self, radius=0, color=None, name=''):
        super().__init__(name=name)
        self.color.value = color
        self.radius = radius

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

    def get_name(self):
        return self.name
```

Файл Rectangle.py:

```
from .Shape import Shape

class Rectangle(Shape):
    def __init__(self, width=0, height=0, color=None, name=''):
```

```

    super().__init__(name=name)
    self.width = width
    self.height = height
    self.color.value = color

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

    def get_name(self):
        return self.name

```

Файл Square.py:

```

from .Rectangle import Rectangle

class Square(Rectangle):
    def __init__(self, width=0, color=None, name=''):
        super().__init__(name=name)
        self.width = width
        self.color.value = color

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

    def get_name(self):
        return self.name

```

Файл Shape.py:

```

from abc import ABC
from abc import abstractmethod
from .ShapeColor import ShapeColor

class Shape(ABC):
    def __init__(self, name):
        self.color = ShapeColor()
        self.name = name

    @abstractmethod
    def area(self):
        pass

    @abstractmethod
    def get_name(self):
        return self.name

    def __repr__(self):
        return "{}'Площадь {:.3f} и цвет {}'.format(self.get_name(), self.area(),
self.color.value)

```

Файл Shape_color.py:

```
class ShapeColor():
    def __init__(self):
        self.value = None

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

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

    @color.deleter
    def color(self):
        del self.value

    def color(self):
        return self.value
```

Выводимый результат:

[illegible]