

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Лабораторная работа №2

“Объектно-ориентированные возможности языка Python.”

Вариант №19

Выполнил:

Студент ИУ5-34Б

Флоринский В. А.

Подпись и дата:

Проверил:

Преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023 г.

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 "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 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`.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

Текст программ.

Данная лабораторная работа была создана в виртуальном окружении Python. Программы с классами фигур и цветами должны быть помещены в папку под названием «`lab_python_oop`», а основная программа помещены в папку выше уровнем под названием `main.py`.

`figure.py`:

```
from abc import ABC, abstractmethod

class figuresh(ABC):
    @abstractmethod
    def __repr__(self):
        pass
```

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

colorcl.py:

```
class colorofshape:
    def __init__(self, color):
        self.__color = color

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

    @color.setter
    def color(self, color):
        self.__color = color
```

circlecl.py:

```
from lab_python_oop.figure import figuresh
from lab_python_oop.colorcl import colorofshape
from math import pi

class circle(figuresh):
    name = "circle"

    def __init__(self, radius, color):
        self.__radius = radius
        self.__color = colorofshape(color).color

    def __repr__(self):
        return (f"name: {self.name} | radius : {self.__radius} | color : {self.__color} \
                | square : {int(pi**2 * self.__radius)}")

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

rectanglecl.py:

```
from lab_python_oop.figure import figuresh
from lab_python_oop.colorcl import colorofshape

class rectangle(figuresh):
    name = "rectangle"

    def __init__(self, width, height, color):
        self.__width = width
        self.__height = height
        self.__color = colorofshape(color).color

    def square(self):
```

```

        return self.__width * self.__height

    def __repr__(self):
        return (f"name: {self.name} | width : {self.__width} | height: {self.__height} \
                | color : {self.__color} | square : {self.__width * self.__height}")

```

squaresp.py:

```

from lab_python_oop.rectangle1 import rectangle
from lab_python_oop.color1 import colorofshape

class squareshape(rectangle):

    name = "square"

    def __init__(self, width, color):
        self.__width = width
        self.__color = colorofshape(color).color

    def __repr__(self):
        return (f"name: {self.name} | width : {self.__width} | color : {self.__color} \
                | square : {self.__width**2}")

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

```

main.py:

```

if __name__ == "__main__":
    from lab_python_oop.rectangle1 import rectangle
    from lab_python_oop.circle1 import circle
    from lab_python_oop.squaresp import squareshape
    from termcolor import colored

    rec = rectangle(19, 19, "blue")
    cir = circle(19, "green")
    sq = squareshape(19, "red")

    print(colored(rec, "blue"))
    print(colored(cir, "green"))
    print(colored(sq, "red"))

```

Примеры работы.

```

C:\Users\vladi\OneDrive\Рабочий стол\LR3thSem\LR2_3\myenv\Scripts>python main.py
name: rectangle | width : 19 | height: 19 | color : blue | square : 361
name: circle | radius : 19 | color : green | square : 187
name: square | width : 19 | color : red | square : 361

```