



**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 дисциплине «Базовые компоненты интернет-технологий»**

**Выполнила:
студентка группы ИУ5-32Б
Андреева А. А.**

**Проверил:
Канев А.И.**

2021 г.

Содержание

Описание задания:.....	3
Текст программы:.....	4
Экранные формы с примерами выполнения программы	6

Описание задания:

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`.

Текст программы:

main.py

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.figure import Figure
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square

import numpy as np

def main():
    r = Rectangle("white", 4, 5)
    print(r)
    c = Circle("violet", 6)
    print(c)
    s = Square("black", 3)
    print(s)

    a = np.eye(5)
    print(a)
    Figure()

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

circle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import Color
import math

class Circle(Figure):
    figure_type = "circle"
    @classmethod
    def get_figure_type(cls):
        return cls.figure_type
    def __init__(self, color_val, radius):
        self.radius = radius
        self.fcolor = Color
        self.fcolor.color_prop = color_val
    """ def area(self):
        return math.pi * self.radius * self.radius"""
    def __repr__(self):
        return 'A {} {} with a radius of {} and area of {}'.format(

            self.fcolor.color_prop,
            Circle.get_figure_type(),
            self.radius,
            """self.area()"""
        )
```

color.py

```
class Color:
    def __init__(self):
        self._color = None
    @property
    def color(self):
        return self._color
    @color.setter
    def color(self, value):
        self._color = value
```

figure.py

```
from abc import ABC, abstractmethod

class Figure(ABC):
    @abstractmethod
    def area(self):

        pass
```

rectangle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import Color

class Rectangle(Figure):
    figure_type= "rectangle"
    @classmethod
    def get_figure_type(cls):
        return cls.figure_type
    def __init__(self, color_val, Width, Height):
        self.width = Width
        self.height = Height
        self.fcolor= Color()
        self.fcolor.color_prop = color_val
    def area(self):
        return self.width * self.height
    def __repr__(self):
        return 'A {} {} with a width of {}, a height of {} and area of {}'.format(

            self.fcolor.color_prop,
            Rectangle.get_figure_type(),
            self.width,
            self.height,
            self.area()
        )
```

square.py

```
from lab_python_oop.rectangle import Rectangle

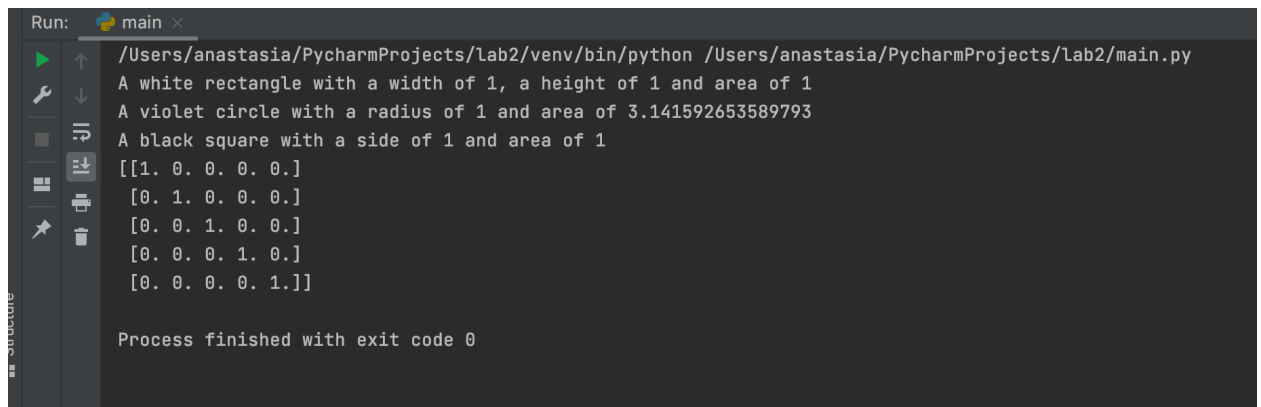
class Square(Rectangle):
    figure_type = "square"

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

    def __init__(self, color_val, side):
        self.side = side
        super().__init__(color_val, self.side, self.side)
    def __repr__(self):
        return 'A {} {} with a side of {} and area of {}'.format(

            self.fcolor.color_prop,
            Square.get_figure_type(),
            self.side,
            self.area()
        )
```

Экранные формы с примерами выполнения программы



The screenshot shows a PyCharm Run console window with a dark theme. The title bar at the top reads 'Run: main x'. The console output is as follows:

```
/Users/anastasia/PycharmProjects/lab2/venv/bin/python /Users/anastasia/PycharmProjects/lab2/main.py  
A white rectangle with a width of 1, a height of 1 and area of 1  
A violet circle with a radius of 1 and area of 3.141592653589793  
A black square with a side of 1 and area of 1  
[[1. 0. 0. 0. 0.]  
 [0. 1. 0. 0. 0.]  
 [0. 0. 1. 0. 0.]  
 [0. 0. 0. 1. 0.]  
 [0. 0. 0. 0. 1.]]  
  
Process finished with exit code 0
```

On the left side of the console, there is a vertical toolbar with icons for running, debugging, and other development actions.