

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет “Информатика и системы управления” Кафедра “Системы
обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования” Отчет по
Лабораторное работе №2

Вариант 14

Выполнил:

Студент группы ИУ5-35Б

Костылев М.С.

Преподаватель:

Гапанюк Ю.Е.

Москва 2025

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

Код:

main.py:

```
from lab_python_oop.Rectangle import Rectangle  
from lab_python_oop.Circle import Circle  
from lab_python_oop.Square import Square
```

N = 14

```
def main():
```

```
rectangle = Rectangle(N, N, "синий")
```

```
circle = Circle(N, "зеленый")
```

```
square = Square(N, "красный")
```

```
print(rectangle)
```

```
print(circle)
```

```
print(square)
```

```
figures = [rectangle, circle, square]
```

```
print("\nВсе фигуры:")
```

```
for figure in figures:
```

```
    print(f' - {figure.getName()}: площадь = {figure.area():.2f}')
```

```
s = [431, 398, 423, 401, 423, 404, 389, 428, 402, 404, 427, 398, 422, 409, 420, 422, 397, 458, 403, 411, 398, 408, 438, 414, 413, 404, 426, 434, 430, 397, 383, 415, 418, 438, 394, 417, 412, 404, 389, 398, 431, 423, 401, 423, 435, 427, 428, 405, 414, 415, 439, 409, 391, 416, 419, 401, 372, 395, 418, 413, 407, 445, 428, 420, 429, 395, 433, 406, 402, 398, 399, 432, 405, 412, 425, 417, 424, 416, 396, 403, 432, 402, 431, 419, 423, 441, 424, 410, 424, 413, 393, 412, 302, 408, 437, 416, 436, 415, 421, 407, 404, 404, 403, 434, 412, 419, 405, 402, 394, 423, 398, 415, 401, 398, 428, 416, 453, 371, 424, 417]
```

```
print(sorted(s))
```

```
print(len(set(s)))
```

```
if __name__ == "__main__":
```

```
    main()
```

Circle.py:

```
import math
```

```
from lab_python_oop.GeometricFigure import GeometricFigure
from lab_python_oop.FigureColor import Color

class Circle(GeometricFigure):
    name = "Круг"

    def __init__(self, radius, color):
        self.radius = radius
        self.colorObj = Color(color)

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

    def __repr__(self):
        return "Фигура: {}, радиус: {}, цвет: {}, площадь: {:.2f}".format(
            self.getName(), self.radius, self.colorObj, self.area())
```

FigureColor.py:

```
class Color:

    def __init__(self, color):
        self.color = color

    def __repr__(self):
        return self.color

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

GeometricFigure.py:

```
from abc import ABC, abstractmethod
```

```
class GeometricFigure(ABC):
```

```
    @abstractmethod
```

```
    def area(self):
```

```
        pass
```

```
    @classmethod
```

```
    def getName(cls):
```

```
        return cls.name
```

Rectangle.py:

```
from lab_python_oop.GeometricFigure import GeometricFigure
```

```
from lab_python_oop.FigureColor import Color
```

```
class Rectangle(GeometricFigure):
```

```
    name = "Прямоугольник"
```

```
    def __init__(self, width, height, color):
```

```
        self.width = width
```

```
        self.height = height
```

```
        self.colorObj = Color(color)
```

```
    def area(self):
```

```
        return self.width * self.height
```

```
    def __repr__(self):
```

```
    return "Фигура: {0}, ширина: {1}, высота: {2}, цвет: {3}, площадь:  
{4}".format(  
        self.getName(), self.width, self.height, self.colorObj, self.area())
```

Square.py:

```
from lab_python_oop.Rectangle import Rectangle
```

```
class Square(Rectangle):
```

name = "Квадрат"

```
def __init__(self, side, color):
```

```
super().__init__(side, side, color)
```

```
def __repr__(self):
```

```
return "Фигура: {0}, сторона: {1}, цвет: {2}, площадь: {3}".format(  
    self.getName(), self.width, self.colorObj, self.area())
```

Работа кода: