

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 лабораторной работе №2
"Объектно-ориентированные возможности языка Python"

Выполнил:
Студент группы ИУ5-35Б
Королев М.О.
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 `"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`.

<https://github.com/koromax/Korolev-PCPL-Labs-2025/tree/main/lab2>

Листинг кода:

```
# lab_python_oop/circle.py
```

```
from math import pi
```

```
from lab_python_oop.shape import Shape
```

```
class Circle(Shape):
```

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

```
        self._radius = radius
```

```
        self._color = color
```

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

```
        return pi * self._radius * self._radius
```

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

```
        return f"{self._color} круг радиусом {self._radius} и площадью {self.area()}"
```

```
# lab_python_oop/color.py
```

```
class Color:
```

```
    def __init__(self, value):
```

```
        self._value = value
```

```
    @property
```

```
    def value(self):
```

```
        return self._value
```

```
    @value.setter
```

```
    def value(self, value):
```

```
        self._value = value
```

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

```
        return self._value
```

```
# lab_python_oop/rectangle.py
```

```
from lab_python_oop.shape import Shape
```

```
from lab_python_oop.color import Color
```

```
class Rectangle(Shape):
```

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

```
        self._width = width
```

```
        self._height = height
```

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

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

```
        return self._width * self._height
```

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

```
        return f"{self._color} прямоугольник шириной {self._width}, высотой {self._height} и площадью {self.area()}"
```

```
# lab_python_oop/shape.py
```

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

```
class Shape(ABC):
```

```
    @abstractmethod
```

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

```
pass
```

```
# lab_python_oop/square.py
from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    def __init__(self, length, color):
        super().__init__(length, length, color)
        self._length = length

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

    def __repr__(self):
        return f"{self._color} квадрат со стороной {self._length} и площадью {self.area()}"
```

```
# main.py
from lab_python_oop.circle import Circle
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.square import Square

from art import text2art
print(text2art("Shapes"))

rectangle = Rectangle(12, 12, "синий")
circle = Circle(12, "зеленый")
square = Square(12, "красный")

print(rectangle)
print(circle)
print(square)
```

Результат выполнения программы:

```
● (venv) 10:lab2 koromax$ python3 main.py
```



```
синий прямоугольник шириной 12, высотой 12 и площадью 144
зеленый круг радиусом 12 и площадью 452.3893421169302
красный квадрат со стороной 12 и площадью 144
```

```
○ (venv) 10:lab2 koromax$ █
```