

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
“Объекто-ориентированное программирование на Python”

Выполнил:
Студент группы ИУ5-34Б
Жеребенков А.Ю.
Преподаватель:
Нардид А.Н.

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.

Код(main.py)

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
from colorama import Fore, Back, Style, init

init(autoreset=True)

def main():
    print(Fore.CYAN + '_' * 20)

    myRectangle = Rectangle('Прямоугольник', 10, 10, "Синий")
    myRectangle.print_shape()
    myRectangle.repr()

    print(Fore.CYAN + "_" * 20)

    myCircle = Circle("Круг", 10, "Зеленый")
    myCircle.print_shape()
    myCircle.repr()

    print(Fore.CYAN + "_" * 20)

    mySquare = Square("Квадрат", 10, "Красный")
    mySquare.print_shape()
    mySquare.repr()

    print(Fore.CYAN + "_" * 20)
```

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

Код circle.py

```
from .shape import Shape
from math import pi

class Circle(Shape):
    def __init__(self, shape, radius, colour):
        super().__init__(shape, colour)
        self.radius = radius

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

    def repr(self):
        print(f"Radius: {self.radius} \nColour: {self.shape.colour.colour}"
              f" \nArea: {self.area()}")

    def print_shape(self):
        super().print_shape()
```

Код rectangle.py

```
from .shape import Shape

class Rectangle(Shape):
    def __init__(self, shape, width, height, colour):
        super().__init__(shape, colour)
        self.width = width
        self.height = height

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

    def repr(self):
        print(f"Width: {self.width} \nHeight: {self.height} \nColour:"
              f" {self.shape.colour.colour} \nArea: {self.area()}")

    def print_shape(self):
        super().print_shape()
```

Код shape.py

```
from .shape import Shape

class Rectangle(Shape):

    def __init__(self, shape, width, height, colour):
        super().__init__(shape, colour)
        self.width = width
        self.height = height

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

    def repr(self):
        print(f"Width: {self.width} \nHeight: {self.height} \nColour: "
              f" {self.shape.colour.colour} \nArea: {self.area()}")

    def print_shape(self):
        super().print_shape()
```

Код shape_colour.py

```
class Shape_colour():

    def __init__(self, colour):
        self._colour = colour

    @property
    def colour(self):
        return self._colour
```

Код square.py

```
from .rectangle import Rectangle

class Square(Rectangle):

    def __init__(self, shape, side, colour):
        super().__init__(shape, side, side, colour)

    # def area(self):
    #     super().area()
```

Работа приложения

```
C:\Users\ADM\PycharmProjects\pikyapLabs\.venv\Scripts\python.exe C:\Users\ADM\PycharmProjects\Zhrebenkov-PkiyapLabs\lab2\main.py
-----
Фигура: Прямоугольник
Width: 10
Height: 10
Colour: Синий
Area: 100

-----
Фигура: Круг
Radius: 10
Colour: Зеленый
Area: 314.1592653589793

-----
Фигура: Квадрат
Width: 10
Height: 10
Colour: Красный
Area: 100

-----
Process finished with exit code 0
```