

**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»  
Отчет по лабораторной работе №2  
«Объектно-ориентированные возможности языка Python»**

Выполнил:  
студент группы ИУ5-36Б  
Ордянец Эрик

Проверил:  
Гапанюк Е.Ю.

2024 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»:
10. Определите метод `getr`, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод `format` - <https://pyformat.info/>
11. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
12. В корневом каталоге проекта создайте файл `main.py` для тестирования Ваших классов (используйте следующую конструкцию - [https://docs.python.org/3/library/\\_\\_main\\_\\_.html](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):
13. Прямоугольник синего цвета шириной N и высотой N.
14. Круг зеленого цвета радиусом N.
15. Квадрат красного цвета со стороной N.
16. Также вызовите один из методов внешнего пакета, установленного с использованием `pip`.

# Текст программы

Структура директорий:

```
.
├── lab_python_oop
│   ├── circle.py
│   ├── __init__.py
│   ├── rectangle.py
│   ├── shape_color.py
│   ├── shape.py
│   └── square.py
└── main.py
```

2 directories, 7 files

## main.py

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

import pandas as pd

if __name__ == "__main__":
    N = 14

    shape1 = Rectangle(N, N, "синий")
    shape2 = Circle(N, "зелёный")
    shape3 = Square(N, "красный")

    print(shape1)
    print(shape2)
    print(shape3)

    print()

    #Тут использовал внешнюю библиотеку
    data = pd.DataFrame({
        'Имя': ['Эрик', 'Илья', 'Гасан'],
        'Возраст': [18, 19, 19],
        'Город': ['Москва', 'Ульяновск', 'Махачкала']
    })

    print(data)
```

## lab\_python\_oop/shape.py

```
from abc import ABC, abstractmethod

class Shape(ABC):
    @abstractmethod
    def area(self):
        """Вычисление площади фигуры"""
        pass

    @classmethod
    @abstractmethod
    def name(cls):
        """Возврат названия фигуры"""
        pass
```

## lab\_python\_oop/shape\_color.py

```
class ShapeColor:
    def __init__(self, color: str):
        self._color = color

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

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

## lab\_python\_oop/rectangle.py

```
from lab_python_oop.shape import Shape
from lab_python_oop.color import ShapeColor

class Rectangle(Shape):
    def __init__(self, width, height, color):
        self.width = width
        self.height = height
```

```

        self.color = ShapeColor(color)

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

    @classmethod
    def name(cls):
        return "Прямоугольник"

    def __repr__(self):
        return "Фигура: {}, Цвет: {}, Ширина: {}, Высота: {}, Площадь: {}".format(
            self.name(), self.color.color, self.width, self.height, self.area()
        )

```

## lab\_python\_oop/circle.py

```

import math
from lab_python_oop.shape import Shape
from lab_python_oop.color import ShapeColor

class Circle(Shape):
    def __init__(self, radius, color):
        self.radius = radius
        self.color = ShapeColor(color)

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

    @classmethod
    def name(cls):
        return "Круг"

    def __repr__(self):
        return "Фигура: {}, Цвет: {}, Радиус: {}, Площадь: {}".format(
            self.name(), self.color.color, self.radius, self.area()
        )

```

## lab\_python\_oop/square.py

```

from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    def __init__(self, lenthSide, color):
        super().__init__(lenthSide, lenthSide, color)

    @classmethod

```

```
def name(cls):
    return "Квадрат"

def __repr__(self):
    return "Фигура: {}, Цвет: {}, Сторона: {}, Площадь: {}".format(
        self.name(), self.color.color, self.width, self.area()
    )
```

## Экранные формы с примерами выполнения программы

```
● → PCPL git:(main) x p lab2/main.py
Фигура: Прямоугольник, Цвет: синий, Ширина: 14, Высота: 14, Площадь: 196
Фигура: Круг, Цвет: зелёный, Радиус: 14, Площадь: 615.7521601035994
Фигура: Квадрат, Цвет: красный, Сторона: 14, Площадь: 196
```

	Имя	Возраст	Город
0	Эрик	18	Москва
1	Илья	19	Ульяновск
2	Гасан	19	Махачкала