

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

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

Выполнил:
студент группы ИУ5-34Б

Новиков Богдан

Проверил:
преподаватель каф.
ИУ5

Гапанюк Ю. Е.

2022 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»:

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 import Rect, Square, Circle
import emoji
```

```
if __name__ == '__main__':
```

```
n = 15
rect = Rect(n, n, 'blue')
circle = Circle(n, 'green')
square = Square(n, 'red')

print(rect)
print(circle)
print(square)

print(emoji.emojize('Love :red_heart:'))
```

circle.py

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

class Circle(Figure):
    def __init__(self, radius, color):
        self.name = "Circle"
        self.radius = radius
        self.color = Color(color)

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

    def __repr__(self):
        return "Figure: {}, Radius: {}, Color: {}".format(
            self.getName(),
            self.color,
            self.radius,
        )
```

rect.py

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

class Rect(Figure):
    def __init__(self, width, height, color):
        self.name = "Rectangle"
```

```
self.width = width
self.height = height
self.color = Color(color)
```

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

```
def __repr__(self):
    return "Figure: {}, Weight: {}, Height: {}, Color: {}".format(
        self.getName(),
        self.width,
        self.height,
        self.color
    )
```

square.py

```
from .rect import Rect
```

```
class Square(Rect):
    def __init__(self, side, color):
        super().__init__(side, side, color)
        self.name = "Square"
```

color.py

```
class Color():
    def __init__(self, color):
        self.color = color
```

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

figure.py

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

```
class Figure(ABC):
    @abstractmethod
    def area():
        pass

    def getName(self):
        return self.name
```

Результаты выполнения программы:

```
(emoji) → lab2 git:(lab2) x python main.py  
Figure: Rectangle, Weight: 15, Height: 15, Color: blue  
Figure: Circle, Radius: green, Color: 15  
Figure: Square, Weight: 15, Height: 15, Color: red  
Love 🍷🥰
```