

**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 лабораторным работам №6
«Объектно-ориентированные возможности языка Python»

Выполнил:
студент группы ИУ5-32Б
Носков Алексей

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий
Евгеньевич

Москва, 2024 г.

Постановка задачи

Задание:

- ⑩ Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием `pip`.
- ⑩ Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
- ⑩ Все файлы проекта (кроме основного файла `main.py`) должны располагаться в пакете `lab_python_oop`.
- ⑩ Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета `lab_python_oop`.
- ⑩ Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).
- ⑩ Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).
- ⑩ 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.
- ⑩ Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа `math.pi` из модуля `math`.
- ⑩ 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/>
- ⑩ 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):
- ⑩ Прямоугольник синего цвета шириной N и высотой N.
- ⑩ Круг зеленого цвета радиусом N.
- ⑩ Квадрат красного цвета со стороной N.
- ⑩ Также вызовите один из методов внешнего пакета, установленного с использованием `pip`.

Текст программы

circle.py

```
import math
from .geometric_figure import GeometricFigure
from .color import Color

class Circle(GeometricFigure):
    def __init__(self, radius, color):
        self.radius = radius
        self.color = Color(color)

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

    def __repr__(self):
        return "Circle: radius={ }, color={ }, area={ }".format(
            self.radius, self.color.color, self.area()
        )

    @classmethod
    def name(cls):
        return "Circle"
```

color.py

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

geometric_figure.py

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

```
class GeometricFigure(ABC):
    @abstractmethod
    def area(self):
        pass
```

rectangle.py

```
from .geometric_figure import GeometricFigure
from .color import Color
```

```
class Rectangle(GeometricFigure):
    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 "Rectangle: width={ }, height={ }, color={ }, area={ }".format(
        self.width, self.height, self.color.color, self.area()
    )

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

```

square.py

```

from .rectangle import Rectangle

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

    def __repr__(self):
        return "Square: side={ }, color={ }, area={ }".format(
            self.width, self.color.color, self.area()
        )

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

```

main_oop.py

```

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

def main():
    N = 14

    rectangle = Rectangle(N, N, "blue")
    circle = Circle(N, "green")
    square = Square(N, "red")

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

```

```
response = requests.get('https://api.github.com')  
print(response.status_code)
```

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

Вывод программы

Rectangle: width=14, height=14, color=blue, area=196
Circle: radius=14, color=green, area=615.7521601035994
Square: side=14, color=red, area=196
200