Защищено:
Гапанюк Ю.Е.

"__"_____**2025** г.

Демонстрация:
Константинов А.А.

"__"_____**2025** г.

Отчет по лабораторной работе № 2 по курсу Парадигмы и конструкции языков программирования

Тема работы: " Объектно-ориентированные возможности языка Python"

5
(количество листов)

студент группы ИУ5Ц-53Б _____
(подпись)

Константинов А.А. "__"_____**2025** г.

Москва, МГТУ - **2025**

1. **Описание задания**

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.
11.Дополнительное задание. Протестируйте корректность работы Вашей программы с помощью модульного теста.

2. Листинг программы

Main.py

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

# внешний пакет - prettytable (pip install prettytable уже установлен)
from prettytable import PrettyTable

def main():
    N = 5  # замените на ваш номер варианта
    rect = Rectangle(N, N, "синий")
    circle = Circle(N, "зелёный")
    square = Square(N, "красный")

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

    table = PrettyTable(["Фигура", "Цвет", "Площадь"])
    table.add_row([rect.figure_type, rect.color.color, f"{rect.area():.2f}"])
    table.add_row([circle.figure_type, circle.color.color, f"{circle.area():.2f}"])
    table.add_row([square.figure_type, square.color.color, f"{square.area():.2f}"])

    print("\nТаблица фигур:")
    print(table)

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

circle.py

```
import math
from lab_python_oop.figure import GeometricFigure
from lab_python_oop.color import FigureColor

class Circle(GeometricFigure):
    figure_type = "Круг"

    def __init__(self, radius, color):
        self.radius = radius
        self.color = FigureColor(color)

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

    def __repr__(self):
        return "{} цвета {} радиусом {}. Площадь: {:.2f}".format(
            self.figure_type, self.color.color, self.radius, self.area()
        )
```

Color.py

```
class FigureColor:
    """Класс для описания цвета геометрической фигуры"""

    def __init__(self, color):
        self._color = color

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

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

scuare.py

```
from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    figure_type = "Квадрат"

    def __init__(self, side, color):
        super().__init__(side, side, color)

    def __repr__(self):
        return "{} цвета {} со стороной {}. Площадь: {:.2f}".format(
            self.figure_type, self.color.color, self.width, self.area()
        )
```

Rectangle .py

```
from lab_python_oop.figure import GeometricFigure
from lab_python_oop.color import FigureColor

class Rectangle(GeometricFigure):
    figure_type = "Прямоугольник"

    def __init__(self, width, height, color):
        self.width = width
        self.height = height
        self.color = FigureColor(color)

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

    def __repr__(self):
        return "{} цвета {} шириной {} и высотой {}. Площадь: {:.2f}".format(
            self.figure_type, self.color.color, self.width, self.height, self.area()
        )
```

Figure.py

```
from abc import ABC, abstractmethod

class GeometricFigure(ABC):
    """Абстрактный класс Геометрическая фигура"""

    @abstractmethod
    def area(self):
        """Вычисление площади фигуры"""
        Pass
```

3. **Результаты работы программы**

```
(venv) PS C:\Users\anton\OneDrive\Рабочий стол\ЛР2> python .\main.py
Прямоугольник цвета синий шириной 5 и высотой 5. Площадь: 25.00
Круг цвета зелёный радиусом 5. Площадь: 78.54
Квадрат цвета красный со стороной 5. Площадь: 25.00

Таблица фигур:
+---------------+---------+---------+
|     Фигура    |   Цвет  | Площадь |
+---------------+---------+---------+
| Прямоугольник |  синий  |  25.00  |
|      Круг     | зелёный |  78.54  |
|    Квадрат    | красный |  25.00  |
+---------------+---------+---------+
```