

**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  
Жданова Яна  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.  
Подпись и дата:

### 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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
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`.
  - 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. 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.

### Текст программы:

**main.py**

```

from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
from lab_python_oop.triangle import Triangle
import cowsay

```

```

def main():
    r = Rectangle("синего", 5, 5)
    c = Circle("зеленого", 5)
    s = Square("красного", 5)
    t = Triangle ("желтого", 2, 3, 4)
    print(r)
    print(c)
    print(s)
    print(t)
    cowsay.stegosaurus('Hello!')

```

```

if __name__ == "__main__":
    main()

```

### circle.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

```

```

class Circle(Figure):

```

```

    """

```

```

    Класс «Круг» наследуется от класса «Геометрическая фигура».

```

```

    """

```

```

    FIGURE_TYPE = "Круг"

```

```

    @classmethod

```

```

    def get_figure_type(cls):
        return cls.FIGURE_TYPE

```

```

    def __init__(self, color_param, r_param):

```

```

        """

```

Класс должен содержать конструктор по параметрам «радиус» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.

```

        """

```

```

        self.r = r_param
        self.fc = FigureColor()
        self.fc.colorproperty = color_param

```

```

    def square(self):

```

```

        """

```

Класс должен переопределять метод, вычисляющий площадь фигуры.

```

        """

```

```

        return math.pi*(self.r**2)

```

```

    def __repr__(self):

```

```

        return '{} {} цвета радиусом {} площадью {}'.format(
            Circle.get_figure_type(),
            self.fc.colorproperty,
            self.r,
            self.square()
        )

```

### color.py

```

class FigureColor:
    """
    Класс «Цвет фигуры»
    """

    def __init__(self):
        self._color = None

    @property
    def colorproperty(self):
        """
        Get-аксесор
        """
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        """
        Set-аксесор
        """
        self._color = value

```

### **figure.py**

```

from abc import ABC, abstractmethod

class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """

    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
        pass

```

### **rectangle.py**

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor

class Rectangle(Figure):
    """
    Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».
    """

    FIGURE_TYPE = "Прямоугольник"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, width_param, height_param):
        """
        Класс должен содержать конструктор по параметрам «ширина», «высота» и
        «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения
        цвета.
        """
        self.width = width_param
        self.height = height_param

```

```

        self.fc = FigureColor()
        self.fc.colorproperty = color_param

    def square(self):
        """
        Класс должен переопределять метод, вычисляющий площадь фигуры.
        """
        return self.width*self.height

    def __repr__(self):
        return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(
            Rectangle.get_figure_type(),
            self.fc.colorproperty,
            self.width,
            self.height,
            self.square()
        )

```

### **square.py**

```

from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    """
    Класс «Квадрат» наследуется от класса «Прямоугольник».
    """
    FIGURE_TYPE = "Квадрат"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, side_param):
        """
        Класс должен содержать конструктор по параметрам «сторона» и «цвет».
        """
        self.side = side_param
        super().__init__(color_param, self.side, self.side)

    def __repr__(self):
        return '{} {} цвета со стороной {} площадью {}.'.format(
            Square.get_figure_type(),
            self.fc.colorproperty,
            self.side,
            self.square()
        )

```

### **triangle.py (дополнительное задание)**

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

class Triangle(Figure):
    FIGURE_TYPE = "Треугольник"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, side1_param, side2_param, side3_param):
        self.side1 = side1_param

```

```

        self.side2 = side2_param
        self.side3 = side3_param
        self.fc = FigureColor()
        self.fc.colorproperty = color_param

    def square (self):
        p = (self.side1 + self.side2 + self.side3)
        return math.sqrt(p * (p - self.side1) * (p - self.side2) * (p -
self.side3))

    def __repr__(self):
        return '{} {} цвета со сторонами {}, {} и {} площадью {}'.format(
Triangle.get_figure_type(),
self.fc.colorproperty,
self.side1,
self.side2,
self.side3,
self.square()
)

```

### Пример выполнение программы:

The screenshot shows a Python IDE window titled 'main'. The command bar displays the execution command: `C:\Users\user\Desktop\lab2\venv\Scripts\python.exe G:\БКИТ\lab2\main.py`. The output area contains the following text:

```

Прямоугольник синего цвета шириной 5 и высотой 5 площадью 25.
Круг зеленого цвета радиусом 5 площадью 78.53981633974483.
Квадрат красного цвета со стороной 5 площадью 25.
Треугольник желтого цвета со сторонами 2, 3 и 4 площадью 43.474130238568314

-----
| Hello! |
=====

```

Below the text is an ASCII art drawing of a cat. The cat is composed of various symbols including backslashes, forward slashes, dots, and dashes, arranged to form its shape. The cat is facing right. At the bottom of the output area, it says 'Process finished with exit code 0'.