



**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 ИУ5-32Б

Еремихин Владислав

Москва, 2021 г.

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 `getr`, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь.
 - 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.
11. Создайте следующие объекты и выведите о них информацию в консоль:

- Прямоугольник синего цвета шириной 5 и высотой 5.
- Круг зеленого цвета радиусом 5.
- Квадрат красного цвета со стороной 5.
- Также вызовите один из методов внешнего пакета, установленного с использованием pip

Текст программы

Файл **main.py**:

```
from lab_python_oop import Rectangle, Circle, Square
from rich import print
from rich.text import Text

def init_r_text(r):
    r_text = Text()
    r_text.append("■")
    r_text.append(repr(r))
    r_text.stylize("bold", 1, 14)
    r_text.stylize("blue", 15, 21)
    r_text.highlight_regex("\d*", style="cyan")
    return r_text

def init_c_text(c):
    c_text = Text()
    c_text.append("●")
    c_text.append(repr(c))
    c_text.stylize("bold", 1, 5)
    c_text.stylize("green", 6, 14)
    c_text.highlight_regex("\d*", style="cyan")
    return c_text

def init_s_text(s):
    s_text = Text()
    s_text.append("■")
    s_text.append(repr(s))
    s_text.stylize("bold", 1, 8)
    s_text.stylize("red", 9, 17)
    s_text.highlight_regex("\d*", style="cyan")
    return s_text

def main():
    r = Rectangle(5, 5, "синего")
    c = Circle(5, "зеленого")
    s = Square(5, "красного")

    print()
    print(init_r_text(r))
```

```

print(init_c_text(c))
print(init_s_text(s))

if __name__ == "__main__":
    main()

```

Пакет **lab_python_oop**:

Файл **__init__.py**:

```

from .color import ShapeColor
from .shape import Shape
from .rectangle import Rectangle
from .square import Square
from .circle import Circle

```

Файл **shape.py**:

```

from abc import ABC, abstractmethod

class Shape(ABC):

    FIGURE_TYPE = None

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

    @abstractmethod
    def area(self):
        pass

```

Файл **color.py**:

```

class ShapeColor:

    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

# def __repr__(self):
#     return self._color

```

Файл **rectangle.py**:

```

from .shape import Shape
from .color import ShapeColor

class Rectangle(Shape):

    FIGURE_TYPE = "Прямоугольник"

    def __init__(self, width, height, color):
        """
        Конструктор

        Args:
            width (int): Ширина прямоугольника
            height (int): Высота прямоугольника
            color (str): Цвет
        """
        self.width = width
        self.height = height
        self.color = ShapeColor()
        self.color.colorproperty = color

    def area(self):
        ''' Возвращает площадь прямоугольника'''
        return self.width * self.height

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

```

Файл **square.py**:

```

from .rectangle import Rectangle

class Square(Rectangle):

    FIGURE_TYPE = "Квадрат"

```

```

def __init__(self, length, color):
    '''
    Конструктор

    Args:
        length (int): Длина стороны квадрата
        color (str): Цвет
    '''
    super().__init__(length, length, color)

def __repr__(self):
    return '{} {} цвета с длиной стороны {} площадью {}'.format(
        Square.get_figure_type(),
        self.color.colorproperty,
        self.height,
        self.area()
    )

```

Файл **circle.py**:

```

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

class Circle(Shape):

    FIGURE_TYPE = "Круг"

    def __init__(self, radius, color):
        '''
        Конструктор

        Args:
            radius (int): Радиус круга
            color (str): Цвет
        '''
        self.name = "Круг"
        self.radius = radius
        self.color = ShapeColor()
        self.color.colorproperty = color

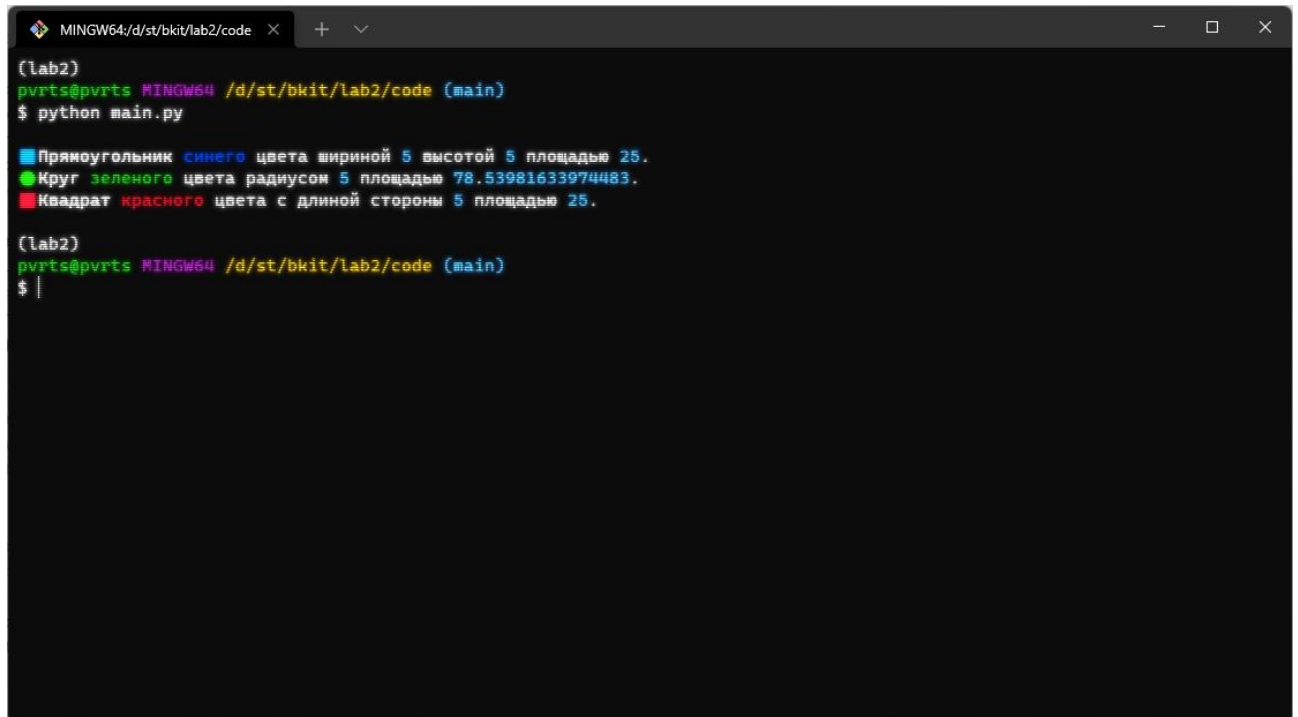
    def area(self):
        ''' Возвращает площадь круга'''
        return pi*(self.radius**2)

    def __repr__(self):
        return '{} {} цвета радиусом {} площадью {}'.format(
            Circle.get_figure_type(),
            self.color.colorproperty,

```

```
        self.radius,  
        self.area()  
    )
```

Пример выполнения программы



```
MINGW64/d/st/bkit/lab2/code × + -  
(lab2)  
pvrtts@pvrtts MINGW64 /d/st/bkit/lab2/code (main)  
$ python main.py  
  
■ Прямоугольник синего цвета шириной 5 высотой 5 площадью 25.  
● Круг зеленого цвета радиусом 5 площадью 78.53981633974483.  
■ Квадрат красного цвета с длиной стороны 5 площадью 25.  
  
(lab2)  
pvrtts@pvrtts MINGW64 /d/st/bkit/lab2/code (main)  
$ |
```