



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «ГУИМЦ»
Кафедра ИУ5 «Системы обработки информации и управления»

Отчет по лабораторной работе №2
по дисциплине «Базовые компоненты интернет-технологий»
Тема «Объектно-ориентированные возможности языка Python»

Выполнил:
студент группы
ИУ5Ц-42Б Дзауров И.А.
_____, “__” _____ 2022 г.

Проверил:
преподаватель кафедры
ИУ5 - Гапанюк Ю.Е.
_____, “__” _____ 2022 г.

Москва, 2022 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 `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`.

Листинг программы

Структура проекта:

```
main.py
lib/
    figure.py
    color.py
    rectangle.py
    circle.py
    square.py
```

Основной файл - [main.py](#):

```
# This is a sample Python script.

from lib.rectangle import Rectangle
from lib.circle import Circle
from lib.square import Square

def main():
    rect = Rectangle("синий", 12, 4)
    circle = Circle("зеленый", 5)
    square = Square("красный", 5)
    print(rect)
    print(circle)
    print(square)

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

Файл - [lib/figure.py](#):

Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры.

```
from abc import ABC, abstractmethod

class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """
    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
        pass
```

Файл - [lib/color.py](#):

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.

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

    def __init__(self):
        self._color = None

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

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

Файл - [lib/rectangle.py](#):

Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс содержит конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс переопределяет метод, вычисляющий площадь фигуры.

```
from lib.figure import Figure
from lib.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 содержит 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 переопределяет 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()
    )
```

Файл - [lib/circle.py](#):

Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа `math.pi` из модуля [math](#).

```
from lib.figure import Figure
from lib.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 содержит 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 переопределяет 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()
        )

```

Файл - [lib/square.py](#):

Класс «Квадрат» наследуется от класса «Прямоугольник». Класс содержит конструктор по длине стороны.

```

from lib.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 содержит 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()
        )

```

Для классов «Прямоугольник», «Квадрат», «Круг»:

- Определен метод «repr», который возвращает в виде строки основные параметры фигуры, ее цвет и площадь.
- Название фигуры («Прямоугольник», «Квадрат», «Круг») задается в виде поля данных класса и возвращается методом класса.

Экранные формы с примерами выполнения программы

Пример :

Прямоугольник синий цвета шириной 12 и высотой 4 площадью 48.

Круг зеленый цвета радиусом 5 площадью 78.53981633974483.

Квадрат красный цвета со стороной 5 площадью 25.