



**«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления (ИУ5)

О т ч е т

по лабораторной работе №2

Дисциплина: Разработка Интернет-Приложений

Студент гр. ИУ5-53Б

(Подпись, дата)

Ваксина И.Р.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Гапанюк Ю.Е.

(И.О. Фамилия)

Москва, 2020

1. Цель работы

Изучение объектно-ориентированных возможностей языка Python.

2. Задание

- 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»:

- a. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
- b. 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):

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

3. Текст программы

3.1. circle.py

```
4. from figure import Figure
5. from color import FigureColor
6. import math
7.
8. class Circle(Figure):
9.     """
10.    Класс «Круг» наследуется от класса «Геометрическая фигура».
11.    """
12.    FIGURE_TYPE = "Круг"
13.
14.    @classmethod
15.    def get_figure_type(cls):
```

```

16.         return cls.FIGURE_TYPE
17.
18.     def __init__(self, color_param, r_param):
19.         """
20.         Класс должен содержать конструктор по параметрам «радиус» и «цвет». В конс
21.         трукторе создается объект класса «Цвет фигуры» для хранения цвета.
22.         """
23.         self.r = r_param
24.         self.fc = FigureColor()
25.         self.fc.colorproperty = color_param
26.
27.     def square(self):
28.         """
29.         Класс должен переопределять метод, вычисляющий площадь фигуры.
30.         """
31.         return math.pi*(self.r**2)
32.
33.     def __repr__(self):
34.         return '{} {} цвета радиусом {} площадью {}'.format(
35.             Circle.get_figure_type(),
36.             self.fc.colorproperty,
37.             self.r,
38.             self.square()
39.         )

```

3.2. color.py

```

4. class FigureColor:
5.     """
6.     Класс «Цвет фигуры»
7.     """
8.
9.     def __init__(self):
10.         self._color = None
11.
12.     @property
13.     def colorproperty(self):
14.         """
15.         Get-аксеccop
16.         """
17.         return self._color
18.
19.     @colorproperty.setter
20.     def colorproperty(self, value):
21.         """
22.         Set-аксеccop
23.         """
24.         self._color = value

```

3.3. figure.py

```

4. from abc import ABC, abstractmethod
5.

```

```

6. class Figure(ABC):
7.     """
8.     Абстрактный класс «Геометрическая фигура»
9.     """
10.    @abstractmethod
11.    def square(self):
12.        """
13.        содержит виртуальный метод для вычисления площади фигуры.
14.        """
15.        pass

```

3.4. rectangle.py

```

4. from figure import Figure
5. from color import FigureColor
6.
7. class Rectangle(Figure):
8.     """
9.     Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».
10.    """
11.    FIGURE_TYPE = "Прямоугольник"
12.
13.    @classmethod
14.    def get_figure_type(cls):
15.        return cls.FIGURE_TYPE
16.
17.    def __init__(self, color_param, width_param, height_param):
18.        """
19.        Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет».
20.        В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.
21.        """
22.        self.width = width_param
23.        self.height = height_param
24.        self.fc = FigureColor()
25.        self.fc.colorproperty = color_param
26.
27.    def square(self):
28.        """
29.        Класс должен переопределять метод, вычисляющий площадь фигуры.
30.        """
31.        return self.width*self.height
32.
33.    def __repr__(self):
34.        return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(
35.            Rectangle.get_figure_type(),
36.            self.fc.colorproperty,
37.            self.width,
38.            self.height,
39.            self.square()

```

3.5. square.py

```

4. from rectangle import Rectangle
5.
6. class Square(Rectangle):
7.     """
8.     Класс «Квадрат» наследуется от класса «Прямоугольник».
9.     """
10.    FIGURE_TYPE = "Квадрат"
11.
12.    @classmethod
13.    def get_figure_type(cls):
14.        return cls.FIGURE_TYPE
15.
16.    def __init__(self, color_param, side_param):
17.        """
18.        Класс должен содержать конструктор по параметрам «сторона» и «цвет».
19.        """
20.        self.side = side_param
21.        super().__init__(color_param, self.side, self.side)
22.
23.    def __repr__(self):
24.        return '{} {} цвета со стороной {} площадью {}'.format(
25.            Square.get_figure_type(),
26.            self.fc.colorproperty,
27.            self.side,
28.            self.square()
29.        )

```

4. Результат выполнения программы

```

Прямоугольник красного цвета, шириной 1, высотой 2 и площадью 2.
Круг зеленого цвета, радиусом 3 и площадью 28.274333882308138.
Квадрат жёлтого цвета, стороной 4 и площадью 16.

```