

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №2  
«Объектно-ориентированные возможности языка Python».

Выполнил:  
Плотников Ф.С. ИУ5-51

Проверил:  
Гапанюк Ю. Е.  
Балашов А. М.

Москва, 2021 г.

## Цель работы:

Изучение объектно-ориентированных возможностей языка Python.

## 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.
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`.
  - 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 классов. Создайте следующие объекты и выведите о них информацию в консоль:
  - Прямоугольник синего цвета шириной 12 и высотой 12.
  - Круг зеленого цвета радиусом 12.
  - Квадрат красного цвета со стороной 12.
  - Также вызовите один из методов внешнего пакета, установленного с использованием `pip`.

## Создание виртуального окружения:

```
D:\BMSTU\5\RIP2021>python -m venv labenv  
  
D:\BMSTU\5\RIP2021>labenv\Scripts\activate.bat  
  
(labenv) D:\BMSTU\5\RIP2021>
```

## Установка пакетов:

Выполнить поиск пакета не удалось из-за нагрузки на сервер PyPI

```
(labenv) D:\BMSTU\5\RIP2021>pip search math  
ERROR: XMLRPC request failed [code: -32500]  
RuntimeError: PyPI's XMLRPC API is currently disabled due to unmanageable load and will be deprecated in the near future. See https://status.pyth  
on.org/ for more information.
```

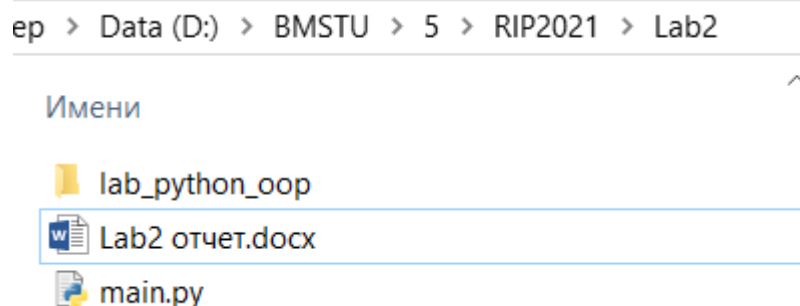
Поэтому была проведена установка уже известного пакета numpy

```
(labenv) D:\BMSTU\5\RIP2021>pip install numpy  
Collecting numpy  
  Downloading numpy-1.21.3-cp310-cp310-win_amd64.whl (14.0 MB)  
    |#####| 14.0 MB 219 kB/s  
Installing collected packages: numpy  
Successfully installed numpy-1.21.3  
  
(labenv) D:\BMSTU\5\RIP2021>pip list  
Package      Version  
-----  
numpy        1.21.3  
pip          21.3.1  
setuptools   57.4.0
```

## Код:

Был создан пакет `lab_python_oop`, который содержит в себе модули классов фигур, причем модули `figure` и `color` подключаются к модулям `circle`, `rectangle` и `square`, так как классы этих фигур наследуются от абстрактного класса `figure` и от класса `color`. `__pycache__` это тот же пакет, просто заэкшированный.

Вышеуказанный пакет подключается к файлу `main.py`, в котором тестируется корректность импорта пакета и его модулей.



ep > Data (D:) > BMSTU > 5 > RIP2021 > Lab2 > lab\_python\_oop >

Имени








 \_\_pycache\_\_  
 \_\_init\_\_.py  
 circle.py  
 color.py  
 figure.py  
 rectangle.py  
 square.py

figure.py:

```
from abc import ABC, abstractmethod

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

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
```

rectangle.py:

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

class Rectangle(Figure):
    """
    Класс «Прямоугольник» наследуется от класса «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 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()
        )
```

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()
        )

```

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 «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 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()
    )

```

main.py:

```

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

def main():
    r = Rectangle("синего", 12, 15)
    c = Circle("зеленого", 12)
    s = Square("красного", 12)
    print(r)
    print(c)
    print(s)

if __name__ == "__main__":
    main()

```

### Пример запуска:

```

(labenv) D:\BMSTU\5\RIP2021>d:/BMSTU/5/RIP2021/labenv/Scripts/python.exe d:/BMSTU/5/RIP2021/Lab2/main.py
Прямоугольник синего цвета шириной 12 и высотой 15 площадью 180.
Круг зеленого цвета радиусом 12 площадью 452.3893421169302.
Квадрат красного цвета со стороной 12 площадью 144.

(labenv) D:\BMSTU\5\RIP2021>

```