

**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  
«Изучение основных конструкций языка Python»

Выполнил:  
студент группы ИУ5-33Б  
Дувакин А.В.

Проверил:  
Нардид А.Н.

2023 г.

# Задание

Задание:

Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием `pip`.

Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.

Все файлы проекта (кроме основного файла `main.py`) должны располагаться в пакете `lab_python_oop`.

Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета `lab_python_oop`.

Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).

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.

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

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. Используйте метод `format` - <https://pyformat.info/>  
Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

В корневом каталоге проекта создайте файл `main.py` для тестирования Ваших классов (используйте следующую конструкцию - [https://docs.python.org/3/library/\\_\\_main\\_\\_.html](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`.

Дополнительное задание. Протестируйте корректность работы Вашей программы с помощью модульного теста.

## Текст программы

Файл `main.py`:

```
from abc import abstractmethod, ABCMeta
import math
class Color:
    def __init__(self, value: str):
        self._value = value
    @property
    def value(self) -> str:
        return self._value
    def __repr__(self):
        return self.value
class Figure(metaclass=ABCMeta):
    @abstractmethod
    def get_area(self) -> float:
        pass
class Circle(Figure):
    def __init__(self, radius: float, color: Color):
        self._name = "Круг"
        self._radius = radius
        self._color = color

    def get_area(self) -> float:
        return self._radius * math.pi

    def __repr__(self):
        return "{}: радиус: {}, цвет: {}".format(self._name, self._radius,
self._color)

class Rectangle(Figure):
    def __init__(self, width: float, height: float, color: Color):
        self._name = "Прямоугольник"
        self.width = width
        self.height = height
        self.color = color

    def get_area(self) -> float:
        return self.height * self.width
```

```
def __repr__(self):
    return "{}: ширина: {}, высота: {}, цвет: {}".format(self._name, self.width,
self.height, self.color)

class Square(Rectangle):
    def __init__(self, side: float, color: Color):
        super().__init__(side, side, color)
        self._name = "Квадрат"

    def __repr__(self):
        return "{}: сторона: {}, цвет: {}".format(self._name, self.width,
self.color)

def main():
    print(Rectangle(15, 10, Color("синий")))
    print(Circle(15, Color("зеленый")))
    print(Square(15, Color("красный")))

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

## Результаты тестов:

```
Прямоугольник: ширина: 15, высота: 10, цвет: синий
Круг: радиус: 15, цвет: зеленый
Квадрат: сторона: 15, цвет: красный
```