

**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 ИУ5-34Б:
Такташова Дарья Юрьевна
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е
Подпись и дата:

Москва, 2022 г.

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). 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.

Код программы:

Файл `lab2.py`

```
import cowsay
cowsay.cow("Hello")
```

Файл 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", 3, 2)
    r1 = Rectangle("белого", 4, 4)
    c = Circle("зеленого", 5)
    c1 = Circle("красного", 3)
    s = Square("красного", 5)
    s1 = Square("синего", 6)
    print(r)
    print(c)
    print(s)
    print(r > r1)
    print(c > c1)
    print(s == s1)
    lst = [c,r,s]
    print(lst)
    lst.sort()
    print(lst)
    if __name__ == "__main__":
        main()
```

Файл figure.py

```
from abc import ABC, abstractmethod
class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """
    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
        pass
    # перегрузки операторов сравнения для фигур
    # фигуры сравнивают по площадям
    def __lt__(self, other):
        return self.square() < other.square()

    def __le__(self, other):
        return self.square() <= other.square()

    def __eq__(self, other):
        return self.square() == other.square()

    def __ne__(self, other):
        return self.square() != other.square()
```

```
def __gt__(self, other):  
    return self.square() > other.square()
```

```
def __ge__(self, other):  
    return self.square() >= other.square()
```

Анализ результатов

```
• dasha@dasha-BOHB-WAX9:~/Desktop/BIKT/BIKT/lab2$ cd ..  
⊗ dasha@dasha-BOHB-WAX9:~/Desktop/BIKT/BIKT$ python3 lab2.py  
python3: can't open file '/home/dasha/Desktop/BIKT/BIKT/lab2.py': [Errno 2] No such file or directory  
• dasha@dasha-BOHB-WAX9:~/Desktop/BIKT/BIKT$ python3 main.py  
Прямоугольник синего цвета шириной 3 и высотой 2 площадью 6.  
Круг зеленого цвета радиусом 5 площадью 78.53981633974483.  
Квадрат красного цвета со стороной 5 площадью 25.  
False  
True  
False  
[Круг зеленого цвета радиусом 5 площадью 78.53981633974483., Прямоугольник синего цвета шириной 3 и высотой 2  
площадью 6., Квадрат красного цвета со стороной 5 площадью 25.]  
[Прямоугольник синего цвета шириной 3 и высотой 2 площадью 6., Квадрат красного цвета со стороной 5 площадью 2  
5., Круг зеленого цвета радиусом 5 площадью 78.53981633974483.]
```

```
• dasha@dasha-BOHB-WAX9:~/Desktop/BIKT/BIKT/lab2/env$ source bin/activate  
• (env) dasha@dasha-BOHB-WAX9:~/Desktop/BIKT/BIKT/lab2/env$ python3 lab2.py  
  
| Hello |  
=====  
  \      \  
   ^  ^  
  (oo)\_____  
  (__) \_____)\\  
        ||----w |  
        ||     ||  
  
• (env) dasha@dasha-BOHB-WAX9:~/Desktop/BIKT/BIKT/lab2/env$ deactivate  
⊗ dasha@dasha-BOHB-WAX9:~/Desktop/BIKT/BIKT/lab2/env$ python3 lab2.py  
Traceback (most recent call last):  
  File "/home/dasha/Desktop/BIKT/BIKT/lab2/env/lab2.py", line 1, in <module>  
    import cowsay  
ModuleNotFoundError: No module named 'cowsay'
```