

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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №2

Выполнил:

студент группы ИУ5-54

Кравцов А.Н.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Ю.Е. Гапанюк.

Подпись и дата:

Москва, 2020 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. 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»:
 - 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`.

1.

Текст программы

```
from math import sqrt
import re

def main():
    solveBiquard(getExpressionVar())

def getExpressionVar():
    var = []
    for i in range(3):
        unCorrectVar = True
        while unCorrectVar:
            var.append(input())
            if re.match(r'\-*\d+', var[i]):
                unCorrectVar = False
            else:
                var.pop()
    return list(map(float, var))

def solveBiquard(var):

    Flag = 5
    x1,x2,x3,x4 = None,None,None,None
    disk = var[1]**2 - 4 * var[0] * var[2]
    if disk < 0:
        print("Нет действительных корней")
    if disk == 0:
        Flag = 5
        if 2 * var[0] != 0 and (-1 * var[1]) / (2 * var[0]) >=
0:
            Flag = 1
            x1 = sqrt((-var[1]) / (2 * var[0]))
            x2 = x1 * -1
    if disk > 0:
        x1 = (-var[1] + sqrt(disk)) / (2 * var[0])
```

```

x3 = (-var[1] - sqrt(diskr)) / (2 * var[0])
if x1 >= 0:
    Flag = 1
    x1 = sqrt(x1)
    x2 = x1 * -1
if x3 >= 0:
    if Flag == 1:
        Flag = 3
    else:
        Flag = 2
        x3 = sqrt(x3)
        x4 = x3 * (-1)

printSolution(Flag, x1, x2, x3, x4)

def printSolution(Flag, *args):
    if Flag == 0:
        print("Нет действительных корней")
    elif Flag == 1:
        if args[0] != args[1] and args[0] != None and args[1] !=
None:
            print("Уравнение имеет два корня:")
            print(args[0], args[1])
        else:
            print("Уравнение имеет один корень:")
            print(args[0])

    elif Flag == 2:
        if args[2] != args[3] and args[2] != None and args[3] !=
None:
            print("Уравнение имеет два корня:")
            print(args[2], args[3])
        else:
            print("Уравнение имеет один корень:")
            print(args[2])

    elif Flag == 3:
        if args[0] != args[1] and args[2] != args[3] and args[0]
!= None and args[1] != None and args[2] != None and args[3] !=
None:

```

```

        print("Уравнение имеет четыре корня:")
        print(args[0], args[1], args[2], args[3])

    else:
        if args[0] != args[1] and args[0] != None and
args[1] != None:
            print("Уравнение имеет три корня:")
            print(args[0], args[1], args[2])
        else:
            print("Уравнение имеет три корня:")
            print(args[1], args[2], args[3])
    elif Flag == 4:
        if args[0] != args[1] and args[0] != None and args[1] !=
None:
            print("Уравнение имеет два корня:")
            print(args[0], args[1])
        else:
            print("Уравнение имеет один корень:")
            print(args[0])
    elif Flag == 5:
        print("Нет действительных корней")

main()

```

Экранные формы с примерами выполнения программы

```

bash-3.2$ python3 1lab.py
1
2
-7
Уравнение имеет два корня:
1.3521934494539567 -1.3521934494539567
bash-3.2$

```