

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

Факультет «Информатика и системы управления»

**Отчёт по лабораторной работе №1
по курсу «Разработка интернет-приложений»
Основные конструкции языка Python**

Выполнил:

студент группы ИУ5-51Б

Бессонова К.С.

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

28.12.2021

Подпись и дата:

28.12.2021

Москва, 2021 г.

Описание задания.

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты A, B, C могут быть заданы в виде параметров командной строки (вариант задания параметров приведен в конце файла с примером кода). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
4. Если коэффициент A, B, C введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.

Текст программы.

```
import sys
import math

def getC(index, prompt):
    try:
        coef_str = sys.argv[index]
    except:
        print(prompt)
        coef_str = input()
    coef = float(coef_str)
    return coef

def getRoots(a, b, c):
    result = []
    D = b*b - 4*a*c
    if D == 0.0:
        root = -b / (2.0*a)
        if root > 0.0:
            result.extend([math.sqrt(root), -math.sqrt(root)])
        elif root == 0.0:
            result.append(0)
    elif D > 0.0:
        sqD = math.sqrt(D)
        root1 = (-b + sqD) / (2.0*a)
        root2 = (-b - sqD) / (2.0*a)
        if root1 > 0.0:
            result.extend([math.sqrt(root1), -math.sqrt(root1)])
        elif root1 == 0.0:
            result.append(0)
        if root2 > 0.0:
            result.extend([math.sqrt(root2), -math.sqrt(root2)])
    return result

def main():
    try:
        a = getC(1, 'A:')
```

```

b = getC(2, 'B:')
c = getC(3, 'C:')
roots = getRoots(a,b,c)
len_roots = len(roots)
if len_roots == 0:
    print('0 roots ;(')
elif len_roots == 1:
    print('1 root: {}'.format(roots[0]))
elif len_roots == 2:
    print('2 roots: {} and {}'.format(roots[0], roots[1]))
elif len_roots == 3:
    print('3 roots: {}, {}, {}'.format(roots[0], roots[1], roots[2]))
elif len_roots == 4:
    print('4 roots: {}, {}, {}, {}'.format(roots[0], roots[1],
roots[2], roots[3]))
except:
    print('Неверный формат')

if __name__ == "__main__":
    main()

```

Экранные формы с примерами выполнения программы.

```

pstri@PC-FOR-GAAGA MINGW64 ~/Desktop/rip5sem/RIP-5-sem/lab1 (master)
$ python3 main.py
A:
4
B:
-6
C:
1
4 roots: 1.1441228056353685, -1.1441228056353685, 0.43701602444882104, -0.43701602444882104

```

```

pstri@PC-FOR-GAAGA MINGW64 ~/Desktop/rip5sem/RIP-5-sem/lab1 (master)
$ python3 main.py
A:
1
B:
2
C:
4
0 roots ;(

pstri@PC-FOR-GAAGA MINGW64 ~/Desktop/rip5sem/RIP-5-sem/lab1 (master)
$ python3 main.py
A:
1
B:
4
C:
-2
2 roots: 0.6704399621018856 and -0.6704399621018856

```