

**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 №1
«Основные конструкции 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:

Разработать программу для решения биквадратного уравнения.

Программа должна быть разработана в виде консольного приложения на языке Python.

Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).

Коэффициенты А, В, С 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. Если коэффициент А, В, С 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.

Код программы:

```
import sys
import math
"""
ax^4 + bx^2 + c = 0
at^2 + bt + c = 0
t1, t2
Проверка, что больше или равно 0
x1, x2 = +-sqrt(t1)
x3, x4 = +-sqrt(t2)
"""
def get_coef(index, prompt):
    """
    Читаем коэффициент из командной строки или вводим с клавиатуры

    Args:
        index (int): Номер параметра в командной строке
        prompt (str): Приглашение для ввода коэффициента

    Returns:
        float: Коэффициент квадратного уравнения
    """
    try:
        # Пробуем прочитать коэффициент из командной строки
        coef_str = sys.argv[index]
        # Переводим строку в действительное число
        coef = float(coef_str)
        flag = True
    except:
        flag = False
        while flag == False:
            try:
                # просим пользователя заново ввести коэффициент
                print(prompt)
```

```

        # заново считываем коэффициент
coef_str = input()
        # Переводим строку в действительное число
coef = float(coef_str)
flag = True
except:
    flag = False

```

```

return coef

```

```

def get_roots(a, b, c):

```

```

    """

```

Вычисление корней квадратного уравнения

Args:

a (float): коэффициент A

b (float): коэффициент B

c (float): коэффициент C

Returns:

list[float]: Список корней

```

    """

```

```

    result = []

```

```

    if a == 0:

```

```

        if b != 0 and -c/b >= 0:

```

```

            result.append(-c/b)

```

```

        return result

```

```

    D = b*b - 4*a*c

```

```

    if D == 0.0:

```

```

        root = -b / (2.0*a)

```

```

        if root >= 0:

```

```

            result.append(root)

```

```

    elif D > 0.0:

```

```

        sqD = math.sqrt(D)

```

```

        root1 = (-b + sqD) / (2.0*a)

```

```

        if root1 >= 0:

```

```

            result.append(root1)

```

```

        root2 = (-b - sqD) / (2.0*a)

```

```

        if root2 >= 0:

```

```

            result.append(root2)

```

```

    return result

```

```

def count_roots (roots):

```

```

    """

```

Определение корней для первоначального уравнения на основе полученных корней при замене и переходе к квадратному уравнению

```

"""
result = []
for i in range (0,len(roots), 1):
    if (roots[i] != 0):
        result.append(math.sqrt(roots[i]))
        result.append(-math.sqrt(roots[i]))
    else: result.append(0.0)
return result

def main():
    """
    Основная функция
    """
    a = get_coef(1, 'Введите коэффициент A:')
    b = get_coef(2, 'Введите коэффициент B:')
    c = get_coef(3, 'Введите коэффициент C:')
    # Вычисление корней
    roots = get_roots(a,b,c)

    # Вывод корней

    roots = count_roots(roots)
    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней')
    elif len_roots == 1:
        print('Один корень: {}'.format(roots[0]))
    elif len_roots == 2:
        print('Два корня: {} и {}'.format(roots[0], roots[1]))
    elif len_roots == 3:
        print('Три корня: {} и {} и {}'.format(roots[0], roots[1], roots[2]))
    elif len_roots == 4:
        print('Четыре корня: {} и {} и {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))

# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()

# Пример запуска
# qr.py 1 0 -4

```

Анализ результатов

```

● dasha@dasha-B0NB-WAX9:~/Desktop/BIKT/BIKT$ python3 my_lab.py 1 -4 4
Два корня: 1.4142135623730951 и -1.4142135623730951
dasha@dasha-B0NB-WAX9:~/Desktop/BIKT/BIKT$

```

```
• dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT$ python3 my_lab.py
Введите коэффициент A:
1
Введите коэффициент B:
4
Введите коэффициент C:
4
Нет корней
```

```
• dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT$ python3 my_lab.py 0 0 0
Нет корней
```

```
• dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT$ python3 my_lab.py
Введите коэффициент A:
hello
Введите коэффициент A:
3
Введите коэффициент B:
f
Введите коэффициент B:
5
Введите коэффициент C:
u
Введите коэффициент C:
u
Введите коэффициент C:
6
Нет корней
```

```
• dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT$ python3 my_lab.py 1 -6 5
Четыре корня: 2.23606797749979 и -2.23606797749979 и 1.0 и -1.0
```