

**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 ИУ5-32Б:
Лапотко Ирина Евгеньевна
Подпись и дата:

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

Москва, 2022 г.

Постановка задачи:

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 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]
    except:
        while True:
```

```

try:
    # Вводим с клавиатуры
    print(prompt)
    coef_str = input()
    # Переводим строку в действительное число
    coef = float(coef_str)
    break
except ValueError:
    print("Вы ввели не число. Попробуйте снова. ")
return coef

```

```

def get_roots(a, b, c):

```

```

    """

```

Вычисление корней ,биквадратного уравнения

Args:

a (float): коэффициент A

b (float): коэффициент B

c (float): коэффициент C

Returns:

list[float]: Список корней

```

    """

```

```

try:

```

```

    result = []

```

```

    D = b*b - 4*a*c

```

```

    if D == 0.0:

```

```

        y = -b / (2.0*a)

```

```

        if y >= 0:

```

```

            root_1 = math.sqrt(y)

```

```

            root_2 = -math.sqrt(y)

```

```

            result.append(root_1)

```

```

            result.append(root_2)

```

```

    elif D > 0.0:

```

```

        sqD = math.sqrt(D)

```

```

        y1 = (-b + sqD) / (2.0*a)

```

```

        y2 = (-b - sqD) / (2.0*a)

```

```

        if (y1 >= 0):

```

```

            root1_1 = math.sqrt(y1)

```

```

            root1_2 = -math.sqrt(y1)

```

```

            result.append(root1_1)

```

```

            result.append(root1_2)

```

```

        if (y2 >= 0):

```

```
root2_1 = math.sqrt(y2)
root2_2 = -math.sqrt(y2)
result.append(root2_1)
result.append(root2_2)
```

```
except ZeroDivisionError:
    print("Коэффициент А не может быть равен нулю.")
return result
```

```
def main():
```

```
    """
```

```
    Основная функция
```

```
    """
```

```
    a = get_coef(1, 'Введите коэффициент А:')
```

```
    b = get_coef(2, 'Введите коэффициент В:')
```

```
    c = get_coef(3, 'Введите коэффициент С:')
```

```
    # Вычисление корней
```

```
    roots = get_roots(a,b,c)
```

```
    # Вывод корней
```

```
    len_roots = len(roots)
```

```
    if len_roots == 0 and a != 0:
```

```
        print('Нет корней')
```

```
    elif len_roots == 2:
```

```
        print('Два корня: {:.2f} и {:.2f}'.format(roots[0], roots[1]))
```

```
    elif len_roots == 4:
```

```
        print('Четыре корня: {:.2f}, {:.2f}, {:.2f} и {:.2f}'.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
```

Экранные формы с примерами выполнения программы

Python 3.9.0 Shell

File Edit Shell Debug Options Window Help

Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:\Users\user\Desktop\ИУ5\1_ЛР.py =====

Введите коэффициент А:

1

Введите коэффициент В:

-5

Введите коэффициент С:

4

Четыре корня: 2.00, -2.00, 1.00 и -1.00

>>>

===== RESTART: C:\Users\user\Desktop\ИУ5\1_ЛР.py =====

Введите коэффициент А:

2

Введите коэффициент В:

-5

Введите коэффициент С:

-3

Два корня: 1.73 и -1.73

>>>

===== RESTART: C:\Users\user\Desktop\ИУ5\1_ЛР.py =====

Введите коэффициент А:

g

Вы ввели не число. Попробуйте снова.

Введите коэффициент А:

2

Введите коэффициент В:

h

Вы ввели не число. Попробуйте снова.

Введите коэффициент В:

-5

Введите коэффициент С:

h

Вы ввели не число. Попробуйте снова.

Введите коэффициент С:

-3

Два корня: 1.73 и -1.73

>>> |

===== RESTART: C:\Users\user\Desktop\ИУ5\БКИТ_ЛР\lab1\1_ЛР.py =====

Введите коэффициент А:

0

Введите коэффициент В:

4

Введите коэффициент С:

6

Коэффициент А не может быть равен нулю.

>>>

C:\WINDOWS\system32\cmd.exe

C:\Users\user\Desktop\ИУ5>1_ЛР.py

Введите коэффициент A:

1

Введите коэффициент B:

-5

Введите коэффициент C:

4

Четыре корня: 2.00, -2.00, 1.00 и -1.00

C:\Users\user\Desktop\ИУ5>1_ЛР.py

Введите коэффициент A:

2

Введите коэффициент B:

-5

Введите коэффициент C:

-3

Два корня: 1.73 и -1.73

C:\Users\user\Desktop\ИУ5>1_ЛР.py

Введите коэффициент A:

9

Вы ввели не число. Попробуйте снова.

Введите коэффициент A:

2

Введите коэффициент B:

-5

Введите коэффициент C:

3

Четыре корня: 1.22, -1.22, 1.00 и -1.00

Тестирование:

<i>Ввод/ожидание</i>	<i>Вывод</i>
2 48 / 2 часа 48 минут ночи	Введите показания с часов... Введите часы (от 0 до 23!!): 2 Введите минуты (от 0 до 59!!): 48 2 часа 48 минут ночи
5 00 / 5 часов утра ровно	Введите показания с часов... Введите часы (от 0 до 23!!): 5 Введите минуты (от 0 до 59!!): 00 5 часов утра ровно
12 00 / полдень	Введите показания с часов... Введите часы (от 0 до 23!!): 12 Введите минуты (от 0 до 59!!): 00 Полдень
12 23 / 12 часов 23 минуты дня	Введите показания с часов... Введите часы (от 0 до 23!!): 12 Введите минуты (от 0 до 59!!): 23 12 часов 23 минуты дня
13 12 / 1 час 12 минут дня	Введите показания с часов... Введите часы (от 0 до 23!!): 13 Введите минуты (от 0 до 59!!): 12 1 час 12 минут дня
21 33 / 9 часов 33 минуты вечера	Введите показания с часов... Введите часы (от 0 до 23!!): 21 Введите минуты (от 0 до 59!!): 33 9 часов 33 минуты вечера
00 00 / полночь	Введите показания с часов... Введите часы (от 0 до 23!!): 00 Введите минуты (от 0 до 59!!): 00 Полночь
03 01 / 3 часа 1 минута ночи	Введите показания с часов... Введите часы (от 0 до 23!!): 03 Введите минуты (от 0 до 59!!): 01 3 часа 1 минута ночи

Вывод:

Практика в разработке разветвляющих алгоритмов позволила повысить навык решения подобных задач. Разветвляющие алгоритмы позволяют организовать эффективную работу программы.