

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Отчет по лабораторной работе № 1 по курсу
Базовые компоненты интернет-технологий
“Основные конструкции языка Python”

ПРЕПОДАВАТЕЛЬ

Гапанюк Ю. Е.

(подпись)

ИСПОЛНИТЕЛЬ:

Студент группы ИУ5-35Б

Ханунов Г.И

(подпись)

" " 2021 г.

Москва - 2021

Задание:

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 А, В, С, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. 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.
4. 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.

Текст программы:

```
import sys
import math

choice = 10

def get_coef(index, prompt):

    try:

        coef_str = sys.argv[index]
    except:

        print(prompt)
        try:
            choice = float(raw_input(""))
            if choice == 0 :
                print ("Вы ввели 0 -> ввод будет начат заново")
                main()
        except ValueError:
            print ("Вы ввели неправильный символ -> ввод будет начат заново")
            main()

    return choice

def get_coef1(index, prompt):
```

```

try:
    coef_str = sys.argv[index]
except:
    # Вводим с клавиатуры
    print(prompt)
    try:
        choice = float(raw_input(""))
    except ValueError:
        print ("Вы ввели неправильный символ -> ввод будет начат заново")
    main()

```

```

return choice

```

```

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 = []
    D = b*b - 4*a*c
    if D == 0.0:
        root = -b / (2.0*a)
        sqr=root
        root1=math.sqrt(sqr)
        root2=-math.sqrt(sqr)
        result.append(root1)
        result.append(root2)

    elif (D > 0 and c!=0 ):
        sqD = math.sqrt(D)
        sq1 = (-b + sqD) / (2.0*a)
        sq2 = (-b - sqD) / (2.0*a)
        if sq1>0:
            root1=math.sqrt(sq1)
            root2=-math.sqrt(sq1)
            result.append(root1)
            result.append(root2)
        if sq2>0:
            root3=math.sqrt(sq2)
            root4=-math.sqrt(sq2)
            result.append(root3)
            result.append(root4)

    elif (b<0 and c==0):
        root3=0
        root1=math.sqrt(abs(a*b))

```

```

        root2=-math.sqrt(abs(a*b))
        result.append(root1)
        result.append(root2)
        result.append(root3)

    return result

def get_ds(a, b, c):
    """
    Дискриминант
    """
    D = b*b - 4*a*c
    return D

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
    len_roots = len(roots)
    d=get_ds(a,b,c)
    print('D = {}'.format(d))
    if len_roots == 0:
        print('Нет корней')
        main()
    elif len_roots == 1:
        print('Один корень: {}'.format(roots[0]))
    elif len_roots == 2:
        if (roots[0]==0.0 and roots[1]==-0.0):
            print('Один корень: {}'.format(0))
        else :
            print('Два корня: {} и {}'.format(roots[0], roots[1]))
    elif len_roots == 3:
        print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))
    elif len_roots == 4:
        print('Четыре корня: {}, {}, {} 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

```

Введите коэффициент A:
0
Вы ввели 0 -> ввод будет начат заново
Введите коэффициент A:
w
Вы ввели неправильный символ -> ввод будет начат заново
Введите коэффициент A:
1
Введите коэффициент B:
2
Введите коэффициент C:
3
 $D = -8.0$
Нет корней
Введите коэффициент A:
3
Введите коэффициент B:
4
Введите коэффициент C:
5
 $D = -44.0$
Нет корней
Введите коэффициент A:
0.0
Вы ввели 0 -> ввод будет начат заново
Введите коэффициент A:
1
Введите коэффициент B:
0
Введите коэффициент C:
3
 $D = -12.0$
Нет корней
Введите коэффициент A:
2
Введите коэффициент B:
0
Введите коэффициент C:
0
 $D = 0.0$
Один корень: 0

Пример: