

**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 №1
«Основные конструкции языка Python»

Выполнил:
студент группы ИУ5-54Б
Киреев А.А.
Подпись и дата:

Москва, 2021 г.

Цель лабораторной работы: изучение основных конструкций языка Python.

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.

Текст программы.

```
import sys
import math

def get_coef(index, prompt):
    try:
        # Пробуем прочитать коэффициент из командной строки
        coef_str = sys.argv[index]
    except:
        # Вводим с клавиатуры
        print(prompt)
        coef_str = input()
    # Переводим строку в действительное число
    try:
        coef = float(coef_str)
    except:
        error = 'error'
        return error
    return coef

def get_roots(a, b, c):
    results = []
    if a==0:
        if b==0:
```

```

        if c==0:
            print ("x - любое число\n")
            error = 'exception'
            return error
        else:
            print ('Уравнение вида {} = 0 не имеет решения'.format(c))
            error = 'exception'
            return error
    else:

        d=-c/b
        if d>0:
            results.append(-math.sqrt(d))
            results.append(math.sqrt(d))
        if d == 0:
            results.append(d)
    else:
        if b==0:
            if c==0:
                results.append(0)
            else:
                d=-c/a
                if d>0:
                    results.append(math.sqrt(math.sqrt(d)))
                    results.append(-math.sqrt(math.sqrt(d)))
        else:
            d=b*b-4*a*c
            if d==0:
                m=(-b)/(2*a)
                if m==0:
                    results.append(m)
                if m>0:
                    results.append(math.sqrt(m))
                    results.append(-math.sqrt(m))
            if d>0:
                m1=(-b+math.sqrt(d))/(2*a)
                m2=(-b-math.sqrt(d))/(2*a)
                if m1>0:
                    x1 = math.sqrt(m1)
                    x2 = -math.sqrt(m1)
                    if m2>0:
                        x3 = math.sqrt(m2)
                        x4 = -math.sqrt(m2)
                        results.append(x1)
                        results.append(x2)
                        results.append(x3)
                        results.append(x4)
                    if m2==0:
                        x3 = 0
                        results.append(x1)
                        results.append(x2)

```

```

        results.append(x3)
    if m2<0:
        results.append(x1)
        results.append(x2)
    if m1 == 0:
        x1 = math.sqrt(m1)
        if m2>0:
            x3 = math.sqrt(m2)
            x4 = -math.sqrt(m2)
            results.append(x1)
            results.append(x3)
            results.append(x4)
        if m2==0:
            x3 = 0
            results.append(x1)
            results.append(x3)
        if m2<0:
            results.append(x1)
    if m1<0:
        if m2>0:
            x3 = math.sqrt(m2)
            x4 = -math.sqrt(m2)
            results.append(x3)
            results.append(x4)
        if m2==0:
            x3 = 0
            results.append(x3)

    return results

```

```

def main():
    """
    Основная функция
    """
    while True:
        a = get_coef(1, 'Введите коэффициент A:')
        b = get_coef(2, 'Введите коэффициент B:')
        c = get_coef(3, 'Введите коэффициент C:')
        if type(a) == float and type(b) == float and type(c) == float:
            break
        else:
            print('Введен неверный коэффициент. Повторите ввод коэффициентов.')

    # Вычисление корней
    roots = get_roots(a,b,c)
    # Вывод корней
    if roots == 'exception':
        return
    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 {}'.format(roots[0], roots[1], roots[2]))
    elif len_roots == 4:
        print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1],
roots[2], roots[3]))

if __name__ == "__main__":
    main()

```

Результат работы программы:

```

Введите коэффициент A:
12
Введите коэффициент B:
-32
Введите коэффициент C:
11
Четыре корня: 1.5037599743548404, -1.5037599743548404, 0.6366887828405621 и -0.6366887828405621

```

```

Введите коэффициент A:
0
Введите коэффициент B:
0
Введите коэффициент C:
1
Уравнение вида 1.0 = 0 не имеет решения

Process finished with exit code 0
|

```

```

Введите коэффициент A:
-12
Введите коэффициент B:
-53
Введите коэффициент C:
-33
Нет корней

Process finished with exit code 0

```

```

Введите коэффициент A:
5
Введите коэффициент B:
6
Введите коэффициент C:
-8
Два корня: 0.89442719099999159 и -0.89442719099999159

Process finished with exit code 0

```