

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

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1
«Основные конструкции языка Python»

Выполнил:

Каженец Д.Н.
ИУ5-31Б

Подпись и дата:

Проверил:

Гапанюк Ю.Е.

Подпись и дата:

Постановка задачи

Разработка алгоритма

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.

Код программы

```
import argparse
import math

def get_coefficient(n):
    while True:
        try:
            return float(input(n))
        except ValueError:
            print("Пожалуйста, введите корректное число.")

def calculate_roots(a, b, c):
    if a == 0:
        print("Уравнение не является биквадратным.")
        return []

    discriminant = b ** 2 - 4 * a * c

    if discriminant < 0:
```

```

    return []

y1 = (-b + math.sqrt(discriminant)) / (2 * a)
y2 = (-b - math.sqrt(discriminant)) / (2 * a)
roots = set()
if y1 >= 0:
    roots.add(math.sqrt(y1))
    roots.add(-math.sqrt(y1))
if y2 >= 0 and y1 != y2:
    roots.add(math.sqrt(y2))
    roots.add(-math.sqrt(y2))
return list(roots)

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('a', nargs='?', type=str)
    parser.add_argument('b', nargs='?', type=str)
    parser.add_argument('c', nargs='?', type=str)
    args = parser.parse_args()
    a = float(args.a if args.a is not None else get_coefficient("Введите коэффициент А: "))
    b = float(args.b if args.b is not None else get_coefficient("Введите коэффициент В: "))
    c = float(args.c if args.c is not None else get_coefficient("Введите коэффициент С: "))
    if a == 0:
        print("Коэффициент А не может быть равен нулю для квадратного уравнения.")
        return roots = calculate_roots(a, b, c)
    if len(roots) == 4:
        print(f"Уравнение имеет четыре действительных корня: {roots[0]}, {roots[1]}, {roots[2]} и {roots[3]}")
    elif len(roots) == 3:
        print(f"Уравнение имеет три действительных корня: {roots[0]}, {roots[1]} и {roots[2]}")
    elif len(roots) == 2:
        print(f"Уравнение имеет два действительных корня: {roots[0]} и {roots[1]}")
    elif len(roots) == 1:
        print(f"Уравнение имеет один действительный корень: {roots[0]}")
    else:

```

```
print("Уравнение не имеет действительных корней.")
```

```
if __name__ == "__main__":  
    main()
```

Анализ результатов

Попробуйте новую кроссплатформенную оболочку PowerShell (<https://aka.ms/pscore6>)

```
PS D:\3_sem> python Lab_01.py 0 0 0  
Уравнение не является биквадратным.  
Уравнение не имеет действительных корней.  
PS D:\3_sem> python Lab_01.py q 12 3  
Введите три числа  
Уравнение не является биквадратным.  
Уравнение не имеет действительных корней.  
PS D:\3_sem> █
```

```
D:\3_sem\venv\Scripts\python.exe D:/3_sem/Lab_01.py  
Введите коэффициент A:  
-1  
Введите коэффициент B:  
5  
Введите коэффициент C:  
6  
Уравнение имеет два действительных корня: 2.449489742783178 и -2.449489742783178
```

```
Process finished with exit code 0  
|
```