

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет "Информатика и системы управления"
Кафедра "Системы обработки информации и управления"



Дисциплина "Парадигмы и конструкции языков программирования"

Отчет по лабораторной работе №1
"Основные конструкции языка Python"

Выполнил:
Студент группы ИУ5-35Б
Королев М.О.
Преподаватель:
Гапанюк Ю.Е.

Москва 2025

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 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.

<https://github.com/koromax/Korolev-PCPL-Labs-2025/tree/main/lab1>

Листинг кода:

```
# BiquadraticEquationSolver.py
import sys

def getCoefficient(index):
    names = {1 : 'A', 2 : 'B', 3 : 'C'}
    try:
        coefficient = float(sys.argv[index])
    except:
        while True:
            try:
                coefficient = float(input(f"Введите коэффициент {names[index]}: "))
                break
            except ValueError:
                print("Введите действительное число")
    return coefficient

def solveBiquadraticEquation(a, b, c):
    discriminant = b * b - 4.0 * a * c
    roots = set()

    # let z = x^2

    if discriminant == 0:
        z = (-b / (2.0 * a))
        if z >= 0:
            roots.add(z ** 0.5)
            roots.add(-1 * z ** 0.5)
    elif discriminant > 0:
        z1 = (-b + discriminant ** 0.5) / (2.0 * a)
        z2 = (-b - discriminant ** 0.5) / (2.0 * a)

        if z1 >= 0:
            roots.add(z1 ** 0.5)
            roots.add(-1 * z1 ** 0.5)
        if z2 >= 0:
            roots.add(z2 ** 0.5)
            roots.add(-1 * z2 ** 0.5)

    return roots

roots = solveBiquadraticEquation(getCoefficient(1), getCoefficient(2), getCoefficient(3))
if len(roots) == 0:
    print("Действительных корней уравнения нет")
else:
    print(f"Корни уравнения: ")
    for root in sorted(roots):
        print(f"{root:.5f}", sep=' ')
```

Результат выполнения программы:

- 10:Korolev-PCPL-Labs-2025 koromax\$ python3 ./lab1/BiquadraticEquationSolver.py
Введите коэффициент A: а
Введите действительное число
Введите коэффициент A: 10
Введите коэффициент B: 4
Введите коэффициент C: 3
Действительных корней уравнения нет
- 10:Korolev-PCPL-Labs-2025 koromax\$ python3 ./lab1/BiquadraticEquationSolver.py 1 -3 2
Корни уравнения:
-1.41421
-1.00000
1.00000
1.41421
- 10:Korolev-PCPL-Labs-2025 koromax\$ █