**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет информатика и системы управления

Кафедра системы обработки информации и управления

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Основные конструкции языка Python»

Вариант 20

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б: |  | преподаватель каф. ИУ5 |
| Пчелинцева Д.А. |  | Нардид А.Н. |
| Подпись и дата: 29.10.2025г. |  | Подпись и дата: |

Москва, 2025 г.

**Задание**

Разработать программу для решения [биквадратного уравнения.](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B5%D1%82%D0%B2%D1%91%D1%80%D1%82%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8#%D0%91%D0%B8%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5)

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](https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

**Текст программы**

**Файл lab1.py** (решение с применением процедурной парадигмы):  
import sys

import math

def get\_coefficient(prompt):

    while True:

        try:

            value = input(prompt)

            return float(value)

        except ValueError:

            print("Ошибка, введите другое число")

def d\_urav(a, b, c):

    if a == 0:

        if b == 0:

            if c == 0:

                return "inf"

            else:

                return []

        m = -c / b

        return [m]

    discriminat = b \* b - 4 \* a \* c

    if discriminat < 0:

        return []

    elif discriminat == 0:

        y = -b / (2 \* a)

        return [y]

    else:

        sqd = math.sqrt(discriminat)

        y1 = (-b - sqd) / (2 \* a)

        y2 = (-b + sqd) / (2 \* a)

        return [y1, y2]

def solve\_biquadratic\_or\_reduced(a, b, c):

    print(f"Решение уравнения: ({a})x^4 + ({b})x^2 + ({c}) = 0")

    if a == 0 and b == 0 and c == 0:

        print("Уравнение: 0 = 0")

        print("Бесконечное количество решений.")

        return

    if a == 0 and b == 0:

        print(f"Уравнение: {c} = 0")

        print("Нет решений.")

        return

    y\_roots = d\_urav(a, b, c)

    if y\_roots == "inf":

        print("Бесконечное количество решений.")

        return

    x\_roots = []

    for y in y\_roots:

        if y < 0:

            continue

        elif y == 0:

            x\_roots.append(0.0)

        else:

            sqrt\_y = math.sqrt(y)

            x\_roots.append(sqrt\_y)

            x\_roots.append(-sqrt\_y)

    x\_roots = sorted(set(x\_roots))

    if x\_roots:

        print("Действительные корни уравнения:")

        for root in x\_roots:

            print(root)

    else:

        print("Действительных корней нет.")

def main():

    coeff = []

    # Попробуем прочитать коэффициенты из командной строки

    if len(sys.argv) == 4:

        try:

            a = float(sys.argv[1])

            b = float(sys.argv[2])

            c = float(sys.argv[3])

            coeff = [a, b, c]

            print(f"A: {a}, B: {b}, C: {c}")

        except ValueError:

            print("Аргументы должны быть числами. Переход к ручному вводу...")

    if len(coeff) != 3:

        a = get\_coefficient("Введите A: ")

        b = get\_coefficient("Введите B: ")

        c = get\_coefficient("Введите C: ")

        coeff = [a, b, c]

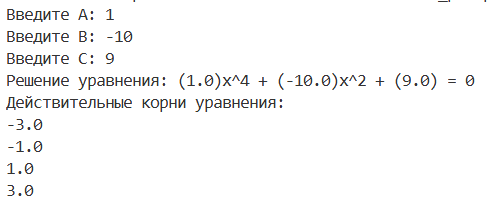
    a, b, c = coeff

    solve\_biquadratic\_or\_reduced(a, b, c)

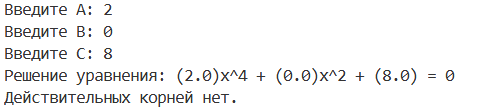
if \_\_name\_\_ == "\_\_main\_\_":

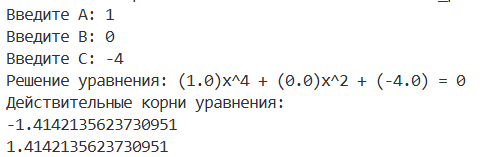
    main()

**Скриншоты работы приложения**

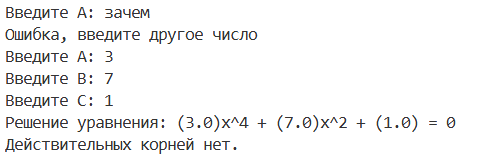


**Рис. 1:** решение уравнения с 4-мя действительными корнями и вводом коэффициентов через консоль

  
**Рис. 2:** решение уравнения, не имеющего действительных корней



**Рис. 3:** решение уравнения с 2-мя действительными корнями и указанием коэффициентов через аргументы командной строки



**Рис. 4:** обработка ошибок ввода

**Ветка с ЛР в моём GitHub:** <https://github.com/daria788/labs_of_pikeap>