

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

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

**Отчет по лабораторной работе №1
«Решение биквадратного уравнения»**

Выполнил:
студент группы ИУ5-35Б
Шакиров Тимур
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий Евгеньевич
Подпись и дата:

Постановка задачи

Научиться решать биквадратные уравнения на языке Python. Привести два решения. Первый основан на процедурном подходе, второй - на объектно-ориентированном подходе.

Разработать возможность ввода коэффициентов биквадратного уравнения через командную строку. Если коэффициенты не были переданы на вход программе в качестве аргументов командной строки, их следует вводить через консоль.

Обработать возможные виды исключений: ввод вместо числа строки, недостаточный ввод коэффициентов, деление на 0, ввод первого коэффициента равным нулю, что сводит биквадратное уравнение к линейному и др.

Разработка алгоритма

Описание алгоритма

- 1) Запускаем программу через командную строку и передаем ей коэффициенты биквадратного уравнения
- 2) Обрабатываем коэффициенты или вызываем исключения
- 3) Сохраняем вычисленные корни биквадратного уравнения в списке
- 4) Выводим их на экран

Входные данные

- int a, b, c – коэффициенты биквадратного уравнения

Результат:

- list results – список из корней биквадратного уравнения

Текст программы

main.py – решение процедурным подходом

```
import sys
import math
from itertools import groupby

def input_coefficient(prompt):
    while True:
        try:
            coefficient = float(input(prompt))
            return coefficient
        except ValueError:
            print("Ошибка ввода. Пожалуйста, введите корректное числовое значение.")

def get_coefs():
    if len(sys.argv) == 4:
        try:
            a, b, c = map(float, sys.argv[1:])
        except ValueError:
            print("Ошибка: коэффициенты в командной строке заданы некорректно.")
            return
    else:
        a = input_coefficient("Введите коэффициент A: ")
        b = input_coefficient("Введите коэффициент B: ")
        c = input_coefficient("Введите коэффициент C: ")
    return a, b, c
```

```

def calculate_roots(a, b, c):
    D = b*b - 4*a*c
    f1, f2 = False, False
    x1, x2, x3, x4 = 0, 0, 0, 0
    roots = []
    if D >= 0.0:
        k = math.sqrt(D)
        if -b + k >= 0:
            x1 = math.sqrt((-b + k) / (2*a))
            x2 = -x1
            f1 = True
        if -b - k >= 0:
            x3 = math.sqrt((-b - k) / (2*a))
            x4 = -x3
            f2 = True
        if f1 == True and f2 == True:
            roots.extend([x1, x2, x3, x4])
        elif f1 == True and f2 == False:
            roots.extend([x1, x2])
        elif f1 == False and f2 == True:
            roots.extend([x3, x4])
    return roots

def print_roots(roots):
    if len(roots) == 0:
        print("Нет действительных корней")
    else:
        roots = list(set(roots))
        print("Действительные корни уравнения:", ', '.join(map(str, roots)))

def main():
    a, b, c = get_coefs()
    roots = calculate_roots(a, b, c)
    print_roots(roots)

if __name__ == "__main__":
    main()

```

main_oop.py – решение объектно-ориентированным подходом

```
import sys
import math
from itertools import groupby

class SquareRoots:

    def __init__(self):
        self.coef_A = 0.0
        self.coef_B = 0.0
        self.coef_C = 0.0
        self.roots = []

    def input_coefficient(self, prompt):
        while True:
            try:
                coefficient = float(input(prompt))
                return coefficient
            except ValueError:
                print("Ошибка ввода. Пожалуйста, введите корректное числовое значение.")

    def get_coefs(self):
        if len(sys.argv) == 4:
            try:
                self.coef_A, self.coef_B, self.coef_C = map(float, sys.argv[1:])
            except ValueError:
                print("Ошибка: коэффициенты в командной строке заданы некорректно.")
                return
        else:
            self.coef_A = self.input_coefficient("Введите коэффициент A: ")
            self.coef_B = self.input_coefficient("Введите коэффициент B: ")
            self.coef_C = self.input_coefficient("Введите коэффициент C: ")

    def calculate_roots(self):
        a = self.coef_A
        b = self.coef_B
        c = self.coef_C
        D = b*b - 4*a*c
        f1, f2 = False, False
        x1, x2, x3, x4 = 0, 0, 0, 0
        roots = []
        if D >= 0.0:
            k = math.sqrt(D)
            if -b + k >= 0:
                x1 = math.sqrt((-b + k) / (2*a))
                x2 = -x1
                f1 = True
            if -b - k >= 0:
                x3 = math.sqrt((-b - k) / (2*a))
                x4 = -x3
```

```

        f2 = True
        if f1 == True and f2 == True:
            roots.extend([x1, x2, x3, x4])
        elif f1 == True and f2 == False:
            roots.extend([x1, x2])
        elif f1 == False and f2 == True:
            roots.extend([x3, x4])
    return roots

def print_roots(self):
    self.roots = self.calculate_roots()
    if len(self.roots) == 0:
        print("Нет действительных корней")
    else:
        self.roots = list(set(self.roots))
        print("Действительные корни уравнения:", ', '.join(map(str,
self.roots)))

def main():
    r = SquareRoots()
    r.get_coefs()
    r.calculate_roots()
    r.print_roots()

if __name__ == "__main__":
    main()

```

Анализ результатов

1) Процедурный подход

```

PS C:\Users\shaki\OneDrive\Programming\Python> python 1.py
Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 6
Действительные корни уравнения: 1.7320508075688772, -1.4142135623730951, 1.4142135623730951, -1.7320508075688772
PS C:\Users\shaki\OneDrive\Programming\Python>

```

2) ООП

```

PS C:\Users\shaki\OneDrive\Programming\Python> python 2.py
Введите коэффициент A: 1
Введите коэффициент B: -12
Введите коэффициент C: 41
Нет действительных корней
PS C:\Users\shaki\OneDrive\Programming\Python>

```

Вывод

Я изучил основы языка Python, разработав решения биквадратного уравнения, пользуясь двумя подходами – процедурным и объектно-ориентированным