

**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  
Пермяков Семён  
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 объектно-ориентированном подходе, второй - на процедурном 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. Также реализовать вывод уравнения, если были введены отрицательные коэффициенты.

## Разработка алгоритма

### *Описание алгоритма*

- 1) Запускаем программу 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 объектно-ориентированным подходом

```
import sys
import math
import time

coefficients = {1: 'a', 2: 'b', 3: 'c'}

class SquareEquation:
    def __init__(self):
        self.kA = 0.0
        self.kB = 0.0
        self.kC = 0.0
        self.roots = []

    def get_coefficient(self, index, line):
        try:
            coefficient = int(sys.argv[index])
        except:
            print(f"Enter the coefficient {line.upper()}: ", end="")
            coefficient = ""
            while type(coefficient) != int:
                coefficient = input()
                try:
                    coefficient = int(coefficient)
                except:
                    print("Incorrect input, try again: ", end="")
            return coefficient

    def get_coefficients(self):
        self.kA, self.kB, self.kC = [self.get_coefficient(i, coefficients[i]) for i in range(1, 4)]

    def calculation(self):
```

```

a = self.kA
b = self.kB
c = self.kC
if a != 0.0:
    D = b ** 2 - 4 * a * c
    if D > 0.0:
        rt_1 = (-b + math.sqrt(D)) / (2 * a)
        rt_2 = (-b - math.sqrt(D)) / (2 * a)
        self.roots.extend([rt_1, rt_2])
    elif D == 0.0:
        rt_1 = (-b) / (2 * a)
        self.roots.append(rt_1)
elif b != 0.0:
    self.roots.append(-c / b)

def print_roots(self):
    if self.kB < 0 and self.kC >= 0:
        print(f"Entered equation: {self.kA}x^2{self.kB}x+{self.kC} = 0")
    elif self.kC < 0 and self.kB >= 0:
        print(f"Entered equation: {self.kA}x^2+{self.kB}x{self.kC} = 0")
    elif self.kB < 0 and self.kC < 0:
        print(f"Entered equation: {self.kA}x^2{self.kB}x{self.kC} = 0")
    else:
        print(f"Entered equation: {self.kA}x^2+{self.kB}x+{self.kC} = 0")
    if len(self.roots) != 0:
        if len(self.roots) == 2:
            print("Two real roots")
            print(f"The first root: {self.roots[0]}")
            print(f"The second root: {self.roots[1]}")
        else:
            print(f"One real root: {self.roots[0]}")
    elif self.kC != 0.0:
        print("No roots")
    else:
        print("x - any number")

def main():
    r = SquareEquation()
    r.get_coefficients()
    r.calculation()
    r.print_roots()
    time.sleep(10)

if __name__ == "__main__":
    main()

```

## main1.py – решение процедурным подходом

```

import sys
import math
import time

coefficients = {1: 'a', 2: 'b', 3: 'c'}

def get_coefficient(index, line):
    try:
        coefficient = int(sys.argv[index])
    except:
        print(f"Enter the coefficient {line.upper()}: ", end="")
        coefficient = ""
        while type(coefficient) != int:
            coefficient = input()
        try:

```

```

        coefficient = int(coefficient)
    except:
        print("Incorrect input, try again: ", end="")
    return coefficient

def calculation(a, b, c):
    roots = []
    D = float(b ** 2 - 4 * a * c)
    if D > 0.0:
        rt_1 = (-b + math.sqrt(D)) / (2 * a)
        rt_2 = (-b - math.sqrt(D)) / (2 * a)
        roots.extend([rt_1, rt_2])
    elif D == 0.0:
        rt_1 = (-b) / (2 * a)
        roots.append(rt_1)
    return roots

def linear(b, c):
    roots = [(-c) / b]
    return roots

def main():
    a, b, c = [get_coefficient(i, coefficients[i]) for i in range(1, 4)]
    if a != 0:
        roots = calculation(a, b, c)
    elif b != 0:
        roots = linear(b, c)
    elif c != 0:
        roots = []
    else:
        print("x - any number")
        time.sleep(10)
        return
    if b < 0 and c >= 0:
        print(f"Entered equation: {a}x^2{b}x+{c} = 0")
    elif c < 0 and b >= 0:
        print(f"Entered equation: {a}x^2+{b}x{c} = 0")
    elif b < 0 and c < 0:
        print(f"Entered equation: {a}x^2{b}x{c} = 0")
    else:
        print(f"Entered equation: {a}x^2+{b}x+{c} = 0")
    if len(roots) != 0:
        if len(roots) == 2:
            print("Two real roots")
            print(f"The first root: {roots[0]}")
            print(f"The second root: {roots[1]}")
        else:
            print(f"One real root: {roots[0]}")
    else:
        print("No roots")

    time.sleep(10)

if __name__ == "__main__":
    main()

```

## Анализ результатов

```
Run  main ×  
↺ | ⋮  
↑ /usr/local/bin/python3.10 /Users/semec/PLP-D/lab1/main.py  
↓ Enter the coefficient A: 1  
⇐ Enter the coefficient B: 2  
⇐ Enter the coefficient C: 1  
⇓ Entered equation: 1x^2+2x+1 = 0  
🖨 One real root: -1.0  
🗑  
Process finished with exit code 0
```

```
Run  main ×  
↺ | ⋮  
↑ Enter the coefficient A: 1  
↓ Enter the coefficient B: 4  
⇐ Enter the coefficient C: 3  
⇓ Entered equation: 1x^2+4x+3 = 0  
⇓ Two real roots  
🖨 The first root: -1.0  
🗑 The second root: -3.0  
Process finished with exit code 0
```

```
Run  python main x
Enter the coefficient A: wfe
Incorrect input, try again: 2
Enter the coefficient B: -5
Enter the coefficient C: 0.5
Incorrect input, try again: -3
Entered equation: 2x^2-5x-3 = 0
Two real roots
The first root: 3.0
The second root: -0.5
```

```
Run  python main x
/usr/local/bin/python3.10 /Users/semec/PLP-D/lab1/main.py
Enter the coefficient A: 1
Enter the coefficient B: 2
Enter the coefficient C: 3
Entered equation: 1x^2+2x+3 = 0
No roots
Process finished with exit code 0
```

## Вывод

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







