**Министерство науки и высшего образования Российской Федерации** ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий Дисциплина:**

«Алгоритмы и структуры данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 0**

«Решение квадратного уравнения»

**Выполнил:**

Чу Ван Доан N3247



(подпись)

**Проверил:**

Ерофеев С. А.



(подпись)

Санкт-Петербург 2024 г.

# СОДЕРЖАНИЕ

[Введение 4](#_heading=h.1fob9te)

1. [Описание функционала программы 5](#_heading=h.2et92p0)
2. [Блок-схема 6](#_heading=h.3dy6vkm)
3. [Код программы 7](#_heading=h.4d34og8)
4. [Результаты тестирования 8](#_heading=h.17dp8vu)

[Заключение 10](#_heading=h.26in1rg)

# ВВЕДЕНИЕ

Задача работы – разработать алгоритм решения квадратного уравнения 𝑎𝑥2 + 𝑏𝑥 +

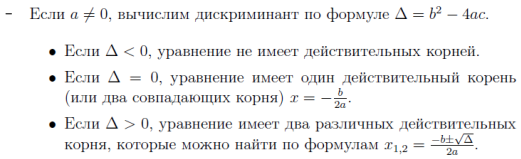
𝑐 = 0 в вещественных числах и реализовать его на одном из языков программирования.

Для выполнения задачи требуется реализовать функцию проверки входных данных, а также функцию решения квадратного уравнения.

Для реализации был выбран язык программирования C++

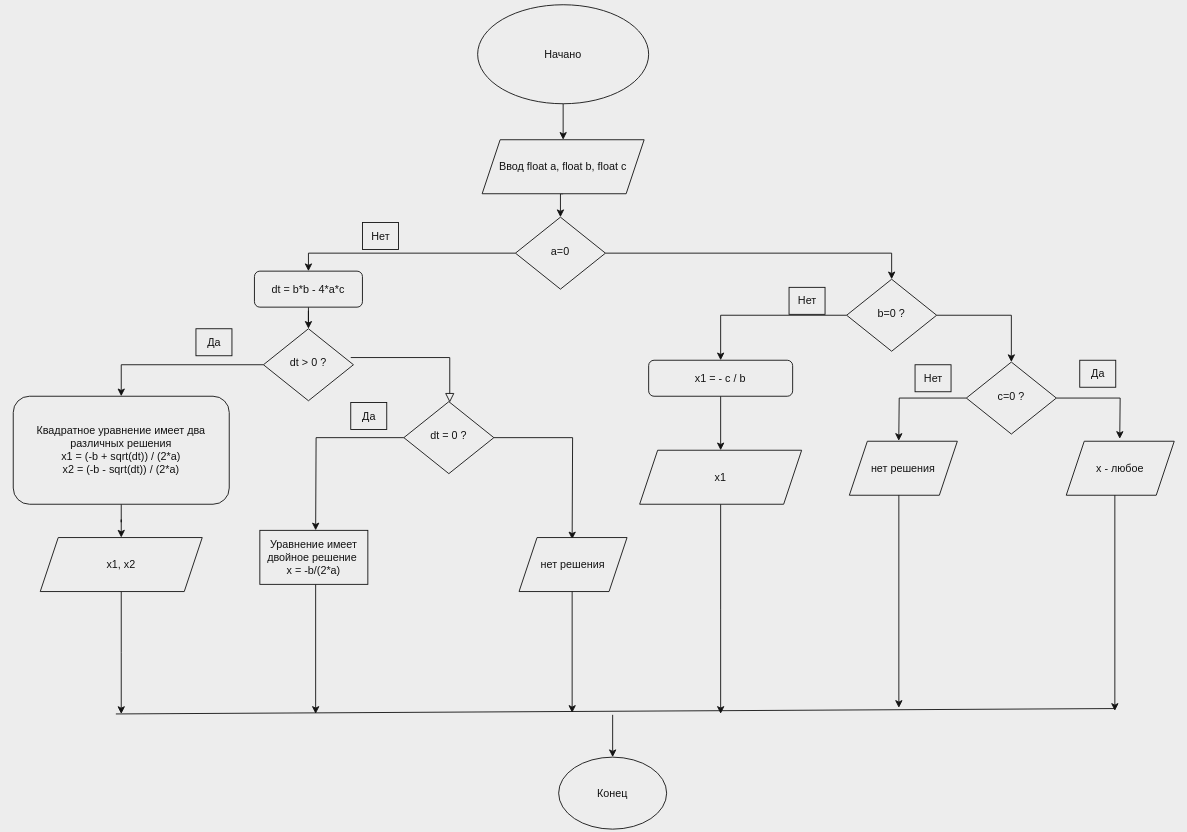
# ОПИСАНИЕ ФУНКЦИОНАЛА ПРОГРАММЫ

- Программа должна вести себя соответствующе при любых входных данных. Ввод коэффициентов квадратного уравнения будет реализован через пробел, соответственно будет реализована функция проверки введенных данных на валидность. В зависимости от коэффициентов будет реализовано 5 возможных выводов. Если все коэффициенты равны нулю, получаем ответ – любое число. Если a и b равны нулю, а c – нет, то решений нет. Если a равно нулю, то решение 𝑥 = − 𝑐 /𝑏 . Если a не равно нулю, производится подсчёт дискриминанта квадратного уравнения и в зависимости от него считаются корни.



− Используемые переменные: a, b, c – коэффициенты квадратного уравнения (тип float, диапазон от - 1.7\*10^308 до 1.7\*10^308) disc – дискриминант квадратного уравнения (тип float) x, x\_1, x\_2 – корни уравнения (тип float).

# БЛОК-СХЕМА

****

# КОД ПРОГРАММЫ

#include<iostream>

#include<cmath>

using namespace std;

int sloveEquation(float a, float b, float c, float& x1, float& x2) {

float delta = b\*b - 4\*a\*c;

cout << " delta = " << delta << endl;

cout << endl;

if (delta > 0) {

double sqpr\_delta = sqrt(delta);

x1 = (-b + sqpr\_delta) / (2 \* a);

x2 = (-b - sqpr\_delta) / (2 \* a);

cout << "Квадратное уравнение имеет два различных решения! " << endl;

cout << " x1 = " << x1 << "\t" << " x2 = " << x2;

return 0;

} else if (delta == 0) {

float x;

x = -b / (2 \* a);

cout << "Уравнение имеет двойное решение ! " << endl;

cout << "x = " << x;

return 0;

} else {

cout << "нет решение ";

return 0;

}

}

int main() {

float a, b, c, x1, x2;

cout << "Введите коэффициенты уравнения :" << endl;

cout << " a= "; cin >> a;

cout << " b= "; cin >> b;

cout << " c= "; cin >> c;

cout << a << "x^2 + " << b << "x + " << c << " = 0" << endl;

if (a != 0) {

sloveEquation(a, b, c, x1, x2);

} else {

if (b == 0) {

if (c == 0) {

cout << "x - любое";

} else {

cout << "нет решения";

}

} else {

float x1 = -c / b;

cout << "x1 = " << x1;

}

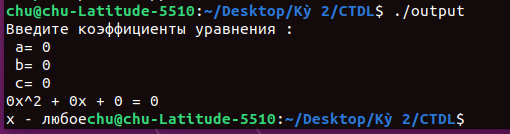
}

return 0;

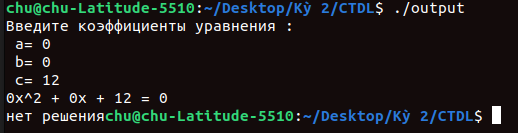
}

# РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

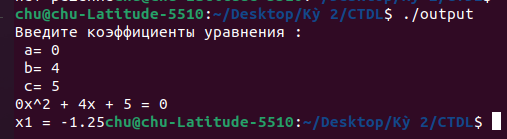
Все коэффициенты равны нулю:



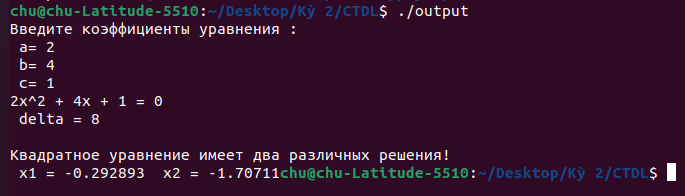
a и b равны нулю, c ≠ 0:



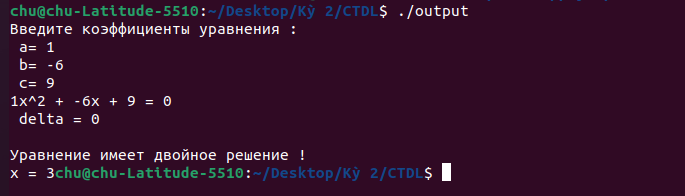
a = 0, b ≠ 0, c ≠ 0:



a ≠ 0, b ≠ 0, c ≠ 0, delta ≠ 0:



a ≠ 0, b ≠ 0, c ≠ 0, disc = 0:



# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной лабораторной работы была успешно достигнута поставленная задача. Я разработал и реализовал алгоритм, способный решать квадратные уравнения во всех возможных случаях. Этот алгоритм позволяет получать точные решения для различных значений коэффициентов квадратного уравнения.

В коде также были реализованы функции ввода и вывода данных, а также обработка исключительных ситуаций.

В итоге была получена полнофункциональная программа, решающая квадратные уравнения при любых значениях коэффициентов, и продемонстрирована возможность разработки алгоритмов решения квадратных уравнений на языке программирования C++.