**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:**

Арендаренко М. М., студент группы N3247

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Ерофеев С. А.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Санкт-Петербург

2024 г.

**Содержани**е

1.Формулировка задания

2.Описание использованных типов данных

3.Описание функционала программы

4.Блок-схема алгоритма

5.Исходные коды разработанных программ

6.Результаты тестирования

7.Заключение

**1. Формулировка индивидуального задания**

Задача работы: разработать алгоритм решения квадратного уравнения 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 Python

**2. Описание использованных типов данных**

Число с плавающей точкой одинарной точности — float в диапазоне от -3,4 \* 1038 до 3,4 \* 1038

**3. Описание функционала программы**

Программа должна корректно работать при любых входных данных. Также предусмотрена проверка введенных данных на корректность. В зависимости от коэффициентов возможно семь различных результатов. Если все коэффициенты равны нулю, результатом будет любое число. Если a и b равны нулю, а c не равен нулю, решений не существует. Если a равно нулю, то решением будет. Если a не равно нулю, вычисляется дискриминант квадратного уравнения, и в зависимости от него находятся корни.

Используемые переменные:

a, b, c — коэффициенты квадратного уравнения (тип float)

d — дискриминант квадратного уравнения (тип float);

x1, x2 — корни уравнения (тип float)

**4. Блок-схема алгоритма**

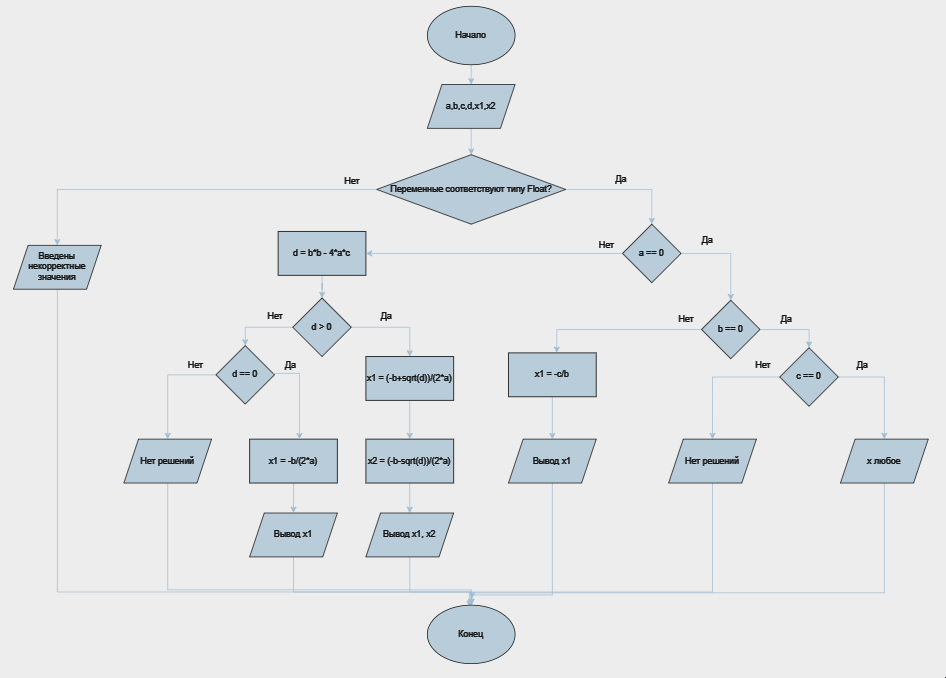


Рис. Блок схема алгоритма

**5. Исходные коды разработанных программ**

|  |
| --- |
| **def** check\_inputs(a, b, c):     # Проверка входных данных  **try**:         a = float(a)         b = float(b)         c = float(c)  **except** ValueError:         print("Ошибка ввода. Пожалуйста, введите действительные числа.")  **return** **False**  **return** **True**  ​  **def** solve\_quadratic\_equation(a, b, c):     # Решение квадратичного уравнения  **if** **not** check\_inputs(a, b, c):  **return**  ​     # Обработка особых случаев  **if** a == b == c == 0:         print("Уравнение имеет бесконечное количество решений ")  **elif** a == b == 0 **and** c != 0:         print("Уравнение не имеет решений ")  **elif** a == 0:         # Решение линейного уравнения  **if** b == 0:  **if** c == 0:                 print("Уравнение имеет бесконечное количество решений")  **else**:                 print("Уравнение не имеет решений")  **else**:             print("Уравнение является линейным:")             print(f"{b} x + {c} = 0")             print("Имеет следующее решение:")             x = -c / b             print(f"{-c} / {b} = 0")             print(x)  **else**:         # Решение квадратного уравнения         D = b\*\*2 - 4\*a\*c  **if** D > 0:             x1 = (-b + D\*\*0.5) / (2\*a)             x2 = (-b - D\*\*0.5) / (2\*a)             print("Уравнение имеет вид:")             print(f"{a}x^2 + {b}x + {c} = 0")             print("Дискриминант больше 0")             print("Уравнение имеет следующее решение:")             print(f"({-b} + {round(D\*\*0.5,2)}) / (2 \* {a})")             print(f"({-b} - {round(D\*\*0.5,2)}) / (2 \* {a})")             print("Уравнение имеет 2 корня: ")             print("x1 =", round(x1, 2))             print("x2 =", round(x2, 2))  **elif** D == 0:             x = -b / (2\*a)             print("x =", round(x, 2))             print("Уравнение имеет вид:")             print(f"{a}x^2 + {b}x + {c} = 0")             print("Дискриминант равен 0")             print("Уравнение имеет следующее решение:")             print(f"({-b}) / 2 \* {a}")             print("Уравнение имеет один корень: ", x)  **else**:             print("Уравнение не имеет решений")  ​  # Получение входных данных от пользователя  **while** **True**:     a = input("Введите коэффициент a: ")  # Проверка введенных данных на некорректные значения  **if** a == "":         print("Пожалуйста, введите коэффициент a.")  **continue**  **elif** a.isdigit() **or** a[0] == '-':             a = float(a)     b = input("Введите коэффициент b: ")  **if** b == "":         print("Пожалуйста, введите коэффициент b.")  **continue**  **elif** b.isdigit() **or** b[0] == '-':             b = float(b)     c = input("Введите свободный коэффициент: ")  **if** c == "":         print("Пожалуйста, введите коэффициент c.")  **continue**  **elif** c.isdigit() **or** c[0] == '-':             c = float(c)  **if** check\_inputs(a, b, c):  **break**  ​  # Вызов функции для решения уравнения  solve\_quadratic\_equation(a, b, c)  ​ |

**6. Результаты тестирования**

Были проведены тесты программы с различными входными данными. Все реализованные случаи вывода корректно обрабатываются и отображаются. Тестирование производилось в среде разработки PyCharm Community Edition 2021.2.3

*Результаты тестирования:*

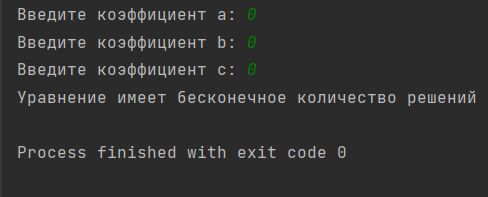


Рис. Результат тестирования при коэффициентах равных 0

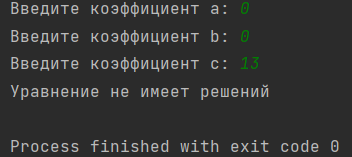


Рис. Результат тестирования при коэффициентах a и b равных 0

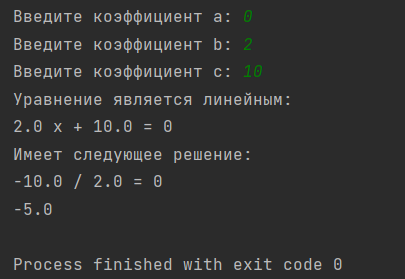


Рис. Результат тестирования при коэффициенте a равном 0

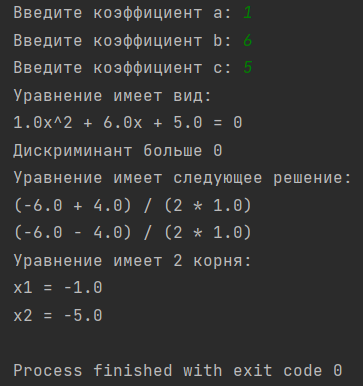


Рис. Результат тестирования при дискриминанте больше 0

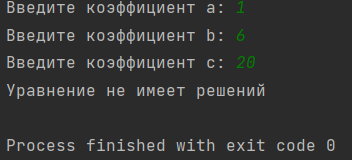


Рис. Результат тестирования при дискриминанте меньше 0

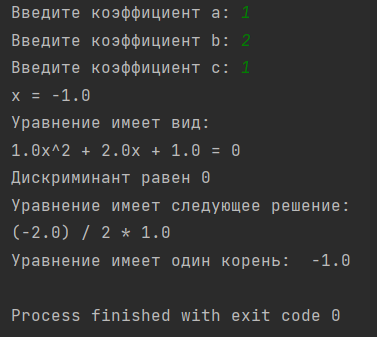


Рис. Результат тестирования при дискриминанте равном 0

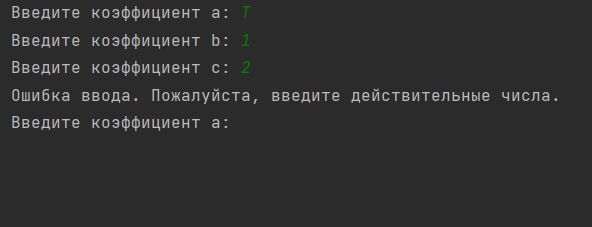


Рис. Результат тестирования при некорректном вводе

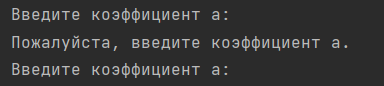


Рис. Результат тестирования при пустном значении

**7. Заключение**

В результате выполнения лабораторной работы была достигнута поставленная цель: реализован алгоритм для решения квадратного уравнения во всех возможных случаях. Было проведено тестирование программы на языке Python с использованием среды разработки PyCharm Community Edition 2021.2.3. В результате тестирования было установлено, что все разработанные выводы успешно работают.

Программа решает квадратные уравнения с действительными коэффициентами и выводит результаты в виде корней уравнения или сообщения о том, что уравнение не имеет действительных корней. Программа обрабатывает все возможные случаи, включая уравнения с нулевыми коэффициентами и уравнения, которые не являются квадратными уравнениями.

Программа была разработана с использованием следующих принципов:

\* Проверка корректности входных данных

\* Обработка особых случаев

\* Вывод результатов