

# СОДЕРЖАНИЕ

<b>СОДЕРЖАНИЕ .....</b>	<b>3</b>
<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ .....</b>	<b>6</b>
1.1 Описание программируемой системы.....	6
1.2 Обзор существующих решений .....	12
1.3 Требования к программируемой системе .....	14
<b>2. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ .....</b>	<b>16</b>
2.1 Диаграмма состояний системы .....	16
2.2 Диаграмма классов системы .....	17
2.3 Диаграмма последовательности системы .....	18
<b>3. ПРАКТИЧЕСКАЯ ЧАСТЬ .....</b>	<b>18</b>
3.1 Реализация требований к системе .....	18
3.2 Функциональное тестирование программного продукта .....	27
3.3 Инструкция по эксплуатации.....	29
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>32</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>33</b>
<b>ПРИЛОЖЕНИЯ .....</b>	<b>35</b>
Приложение А – Исходный код программы .....	35

# ВВЕДЕНИЕ

Современный рынок разработки программного обеспечения обязывает разработчиков программного обеспечения не только быть специалистами своей области, но и ориентироваться в большинстве смежных технологических областей. Данная работа посвящена разработке современного программного обеспечения в соответствии с поставленной задачей и темой работы. Такая деятельность, помимо формирования практического опыта, позволяет сформировать новое видение классических задач по разработке программных комплексов и программного обеспечения.

**Цель работы:** Разработка программного продукта для решения квадратных уравнений с заданными коэффициентами «Решатель» на языке C++.

## **Задачи работы**

1. Описать программируемую систему.
2. Рассмотреть существующие решения-аналоги по данной, либо схожей темам.
3. Сформировать требования к программируемой системе.
4. Спроектировать диаграмму состояний системы.
5. Спроектировать диаграмму классов системы.
6. Спроектировать диаграмму последовательности для системы.
7. Реализовать программный продукт в соответствии с требованиями.
8. Провести функциональное тестирование программного продукта
9. Составить инструкцию по использованию программного продукта.
10. Составить отчет по работе.
11. Сдать отчет и представить его к защите.

**Объектом данного исследования** является автоматизация решения базовых математических задач на основных языках программирования.

**Предметов исследования** в данной работе является автоматизация решения квадратных уравнений с заданными коэффициентами на языке C++.

**В качестве основных методов исследования** применены анализ, синтез, сравнение и моделирование. Практическая реализация поставленной задачи соответствует основным подходам к разработке программного обеспечения.

**Информационной базой** исследования являются открытые источники, в том числе доступные в сети Интернет, а также материалы курса «Технологии индустриального программирования», доступные через систему дистанционного обучения РТУ МИРЭА.

В данном отчете будет представлен процесс разработки программного продукта, в том числе теоретический обзор области и системы, технологическое проектирование и описание системы. а также непосредственно результаты разработки.

# 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Описание программируемой системы

Программный продукт для решения квадратных уравнений с заданными коэффициентами «Решатель» реализован на языке программирования C++. Функционал программы заключается в решении 8 типов квадратных уравнений, а также возможность решения квадратного неравенства методом интервалов, подробное описание всех типов представлено ниже в разделе описания квадратных уравнений.

Старт программы начинается с диалогового окна пользователя, выводящее инструкцию по использованию программы и требующее ввод уравнения или неравенства со стороны пользователя. После получения ввода пользователя программа обрабатывает ввод, определяя один из 8 типов квадратных уравнений. Исходя из типа квадратного уравнения, программа начинает решать уравнение по математическому алгоритму решения квадратных уравнений, представленных ниже. После решения уравнения, программа спросит у пользователя, нужно ли ему решить квадратное неравенство методом интервалов с теми же значениями переменных. Если ответ будет положительным, то программа уточнит, какой именно знак должен стоять между уравнением и 0.

Результатом работы программы является вывод ответа(корней) квадратного уравнения и при надобности интервал решения квадратного неравенства, либо сообщение о том, что решений уравнения нет. После вывода ответа на уравнение, программа приводит подробные пошаговые действия по решению уравнения, чтобы пользователь смог разобраться в решении и проверить себя.

Помимо решения квадратных уравнений, программа может выдать список всех 8 типов данных уравнений для пользователя при выборе соответствующей команды со стороны пользователя. Программа содержит руководство

пользователя и может его вывести при выборе команды “help” со стороны пользователя. При наличии ошибки ввода программа выключается.

Предполагаемая сложность алгоритма решения уравнений  $O(1)$ . Предполагаемая выделяемая память для работы с уравнениями  $O(N)$ , где  $N$  – сумма всех строк и букв, включая инструкции, вывод решений, ввод пользователя и т.д. Таким образом, программа является одновременно быстрой и занимает небольшое количество памяти. С методом решений квадратных уравнений предлагается ознакомиться в следующем абзаце текста.

Квадратное уравнение — алгебраическое уравнение второй степени с общим видом

$$ax^2 + bx + c = 0, a \neq 0. \quad (1.1)$$

где  $a, b, c$  — вещественные числа

$x$  — неизвестное

Элементы квадратного уравнения имеют собственные названия, в дальнейшем в описании решений квадратных уравнений будут использоваться эти обозначения:

- $a$  называют *старшим* коэффициентом;
- $b$  называют *средним* коэффициентом;
- $c$  называют *свободным членом*.

В зависимости от значений, которые могут принимать коэффициенты квадратного уравнения, можно выделить восемь типов квадратных уравнений:

1.  $a = 0, b = 0, c = 0$ :

в таком случае уравнение (1.1) принимает следующий вид:

$$0 * x^2 + 0 * x + 0 = 0$$

решением уравнения будут являться все  $x \in (-\infty; +\infty)$ .

2.  $a \neq 0, b = 0, c = 0$ :

средний коэффициент  $b$  равен нулю, свободный член  $c$  равен 0, старший коэффициент  $a$  не равен нулю, уравнение (1.1) принимает вид:

$$a * x^2 = 0$$

уравнение имеет единственный корень  $x = 0$ .

3.  $c = 0$ :

уравнение (1.1) является неверным и не имеет решений, так как задан только свободный член  $c$ .

4.  $a \neq 0, c \neq 0, b = 0$ :

уравнение (1.1) принимает следующий вид:

$$a * x^2 + c = 0$$

перенесем свободный член  $c$  в правую часть уравнения:

$$a * x^2 = -c \quad (1)$$

следовательно:

$$x^2 = -\frac{c}{a} \quad (2)$$

выделим два случая:

1. при  $-\frac{c}{a} < 0$  уравнение не имеет решений.

2. при  $-\frac{c}{a} \geq 0$  уравнение имеет следующие корни:

$$x_1 = \sqrt{\frac{c}{a}},$$

$$x_2 = -\sqrt{\frac{c}{a}}$$

5.  $a = 0, b \neq 0, c \neq 0$ :

уравнение (1.1) принимает следующий вид:

$$bx + c = 0.$$

единственным корнем уравнения будет являться  $x = -\frac{c}{b}$ .

6.  $a \neq 0, b \neq 0, c \neq 0$ :  
уравнение (1.1) принимает следующий вид:

$$ax^2 + bx + c = 0$$

данное уравнение является классическим квадратным уравнением, методы его решения будут приведены ниже.

7.  $a = 0, b \neq 0, c = 0$ :  
уравнение (1.1) принимает следующий вид:

$$bx = 0$$

единственным решением уравнения является  $x = 0$ .

8.  $a \neq 0, b \neq 0, c = 0$ :  
уравнение (1.1) принимает следующий вид:

$$ax^2 + bx = 0$$

данное уравнение раскладывается на уравнение вида:

$$x(ax + b) = 0$$

уравнение имеет два корня:

$$x_1 = 0$$

$$x_2 = -\frac{b}{a}$$

Для решения квадратных уравнений используются различные методы [1].  
Выбор метода зависит от типа квадратного уравнения.

1. Общая формула для вычисления корней с помощью дискриминанта.

Дискриминантом квадратного уравнения  $ax^2 + bx + c = 0$  называется величина  $D = b^2 - 4ac$ . От значения дискриминанта зависит количество решений для квадратного уравнения:

- $D > 0$ :

уравнение (1.1) имеет два корня, которые находятся по следующей формуле:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (1.2)$$

- $D = 0$ :

уравнение (1.1) имеет один единственный корень:

$$x = -\frac{b}{2a}$$

- $D < 0$ :

уравнение (1.1) не имеет действительных корней.

2. Корни квадратного уравнения при чётном коэффициенте  $b$ .

Для уравнений вида  $ax^2 + 2kx + c = 0$ , то есть при четном  $b$ , где

$$k = \frac{1}{2}b,$$

вместо формулы (1.2) для нахождения корней существует возможность использования более простых выражений:

$$\frac{D}{4} = k^2 - ac.$$

далее в зависимости от значения дискриминанта выделяют два случая:

- $D > 0$ :

$$x = \frac{-k \pm \sqrt{k^2 - ac}}{a}$$

- $D = 0$ :

$$x = -\frac{k}{a}$$

3. Разложение квадратного трехчлена на линейные множители.



Если трехчлен вида  $ax^2 + bx + c$  удастся представить в качестве произведения линейных множителей  $(kx + m)(lx + n) = 0$ , то можно найти корни уравнения (1.1) — ими будут  $-\frac{m}{k}$  и  $-\frac{n}{l}$ . Рассмотрим некоторые случаи для обобщения:

- Использование формулы квадрата суммы (разности):  
Если квадратный трехчлен имеет вид  $(ax)^2 + 2abx + b^2 = (ax + b)^2$ , то

$$x = -\frac{b}{a}$$

- Выделение полного квадрата суммы (разности):

$$1. \quad x^2 + px + \left(\frac{p}{2}\right)^2 - \left(\frac{p}{2}\right)^2 + q = 0$$

$$2. \quad \left(x + \frac{p}{2}\right)^2 = \frac{p^2}{4} - q$$

$$x = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$$

#### 4. Обратная теорема Виета.

Обратная теорема Виета позволяет решать квадратные уравнения (1.1) при коэффициенте  $a = 1$  устно. Согласно данной теореме, всякая пара чисел(число),  $x_1, x_2$  будучи решением системы уравнений

$$\{x_1 + x_2 = -p \quad x_1 x_2 = q\}$$

являются корнями уравнения  $x^2 + px + q = 0$ .

**Квадратное неравенство** – это такое неравенство, которое имеет вид  $ax^2 + bx + c < 0$ , где  $a, b$ , и  $c$  – некоторые числа, причем  $a$  не равно нулю.  $x$  – это переменная, а на месте знака “<” может стоять любой другой знак неравенства. В данной курсовой работе будет рассмотрен такой способ решения квадратных неравенств, как метод решения интервалами. В основу данного метода входит решения квадратного уравнения и приведение его к виду  $(x - x_1) * (x - x_2) > 0$ , где  $x_1$  и  $x_2$  – корни уравнения, а “>” - любой знак

неравенства. Разберем отдельные случаи так же, как и в квадратных уравнениях.

1.  $a=0, b=0, c=0$ . В данном случае в случаях, когда знак неравенства строгим (" $>$ ", " $<$ "), то решения у данного уравнения не будет. В двух других же случаях  $x \in (-\infty; +\infty)$ .

2.  $a \neq 0, b = 0, c = 0$ . В случае

После, на основе данного неравенства строится график параболы на прямой абсцисс, где отображены корни уравнения. Далее расставляются знаки, справа налево ставятся  $+$ ,  $-$  и т.д.

## 1.2 Обзор существующих решений

С появлением и развитием средств вычислительной техники решались различные математические задачи, в том числе и задача по нахождению корней квадратного уравнения. Рассмотрено три программных продукта, функционал которых включает в себя решений квадратных уравнений.

### Mathcad

Mathcad — это инженерное математическое программное обеспечение для выполнения, анализа инженерных расчётов и обмена ими [3].

Mathcad имеет много преимуществ, среди которых [4]:

- удобный многооконный интерфейс;
- простота использования и подробная справка;
- возможность работы без доступа к интернету;
- сильное сходство записи задач с тем, как они решаются на бумаге;
- возможность решать задачи не только по математике, но и в других сферах, таких как физика, экономика, строительство.

Среди недостатков Mathcad [4]:

- отсутствие онлайн-версии;

- возможность возникновения сложностей при установке и настройке программы;
- запрет внесения изменений в программу и работа строго по шаблону;
- отсутствие доступа для обычных пользователей (при скачивании требуется указать данные по учёбе или работе).

### **Photomath**

Photomath — это приложение, которое помогает учиться решать математические задания. Задания можно вводить как вручную, так и с помощью камеры [5].

Достоинствами приложения Photomath являются [5]:

- удобный интерфейс;
- возможность ввода заданий с помощью камеры;
- возможность работы без доступа к интернету;
- поэтапное решение заданий;
- сохранение истории при вводе заданий;
- способность решать огромное множество заданий за доли секунды;
- практически полное отсутствие ошибок при решении.

Из недостатков можно выделить следующие [5]:

- приложение доступно только как мобильное;
- приложение не всегда распознаёт написанное задание;
- есть определённые ограничения в решениях (например, приложение не воспринимает текстовые условия).

### **Microsoft Math Solver**

Microsoft Math Solver — это ещё один решатель математических задач, позволяющий получать пошаговые объяснения, создавать графическое представление математических задач, получать различные справки и пояснения по математике на своём языке [6].

Достоинствами Microsoft Math Solver являются [7]:

- пошаговые объяснения, интерактивные графики и рабочие листы;
- сохранение истории решения примеров;
- доступность как в интернете, так и в виде мобильного приложения;
- возможность ввода заданий с помощью камеры.

Среди недостатков данного решения:

- отсутствие настольной версии;
- необходимость подключения к интернету;
- плохая реализация функции рисования формулы.

### 1.3 Требования к программируемой системе

В таблице 1.1 представлены требования к программируемой системе.

Таблица 1.1 – Требования к программируемой системе

№	Требование	Значение
1	Язык программирования	C++
2	Корректность работы	Приложение запускается и поддерживает стабильный цикл работы от момента старта до завершения
3	Применение принципов объектно-ориентированного программирования	При написании приложения, как минимум, были использованы классы в C++, объектный подход к проектированию системы, а также инкапсуляция.
4	Интерфейс пользователя	Создан интерфейс пользователя, поддерживающий корректный пользовательский опыт и содержащий все необходимые пояснения к работе и эксплуатации
5	Инструкция по эксплуатации	Написана инструкция по эксплуатации, содержащая, в том числе, основные рекомендации по использованию и пояснения к возможным ошибкам в программе
6	Решение квадратных уравнений всех типов	Программа должна уметь решать все типы квадратных уравнений безошибочно.
7	Поддержка различных типов данных	Программа поддерживает различные типы данных, такие как целые, отрицательные, дробные числа для универсальности ее использования.

8	Обработка ошибок и исключений	Реализована обработка ошибок и исключительных ситуаций, например когда уравнение имеет всего лишь один корень или не имеет корней вообще.
9	Кроссплатформенность	Программа должна быть способна работать на различных платформах, включая Windows и Linux.
10	Пример решения квадратного уравнения	Программа должна будет уметь выводить пример решения квадратного уравнения, чтобы пользователь мог понять, как программа будет решать квадратные уравнения

## 2. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

### 2.1 Диаграмма состояний системы

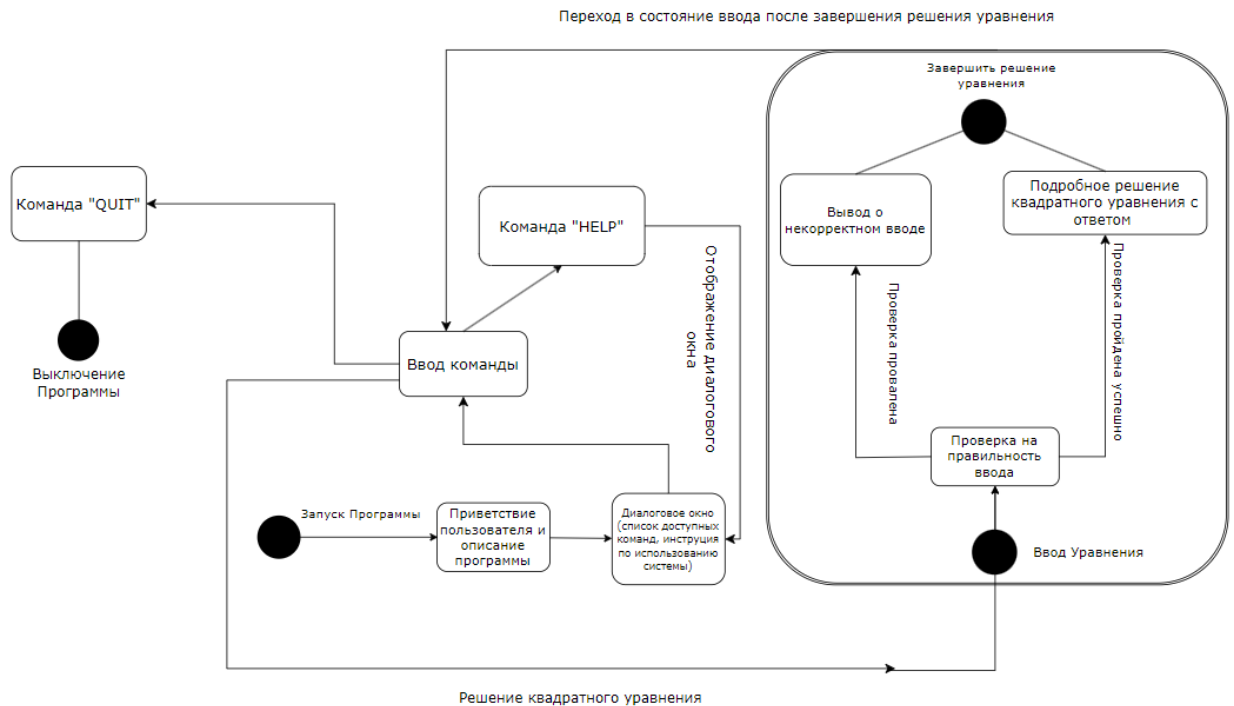


Рисунок 2.1 – Диаграмма состояний системы UML

На рисунке 2.1 представлена диаграмма состояний системы программы. Рассмотрим её устройство.

После запуска программы в окне терминала появляется приветствие пользователя, после чего появляется диалоговое окно, которое содержит в себе описание программы, список доступных команд, и инструкция по использованию системой. Диалоговое окно отображается на русском языке.

Далее пользователю высвечивается сообщение с требованием выбора команды. В зависимости от выбранной команды система перейдет в три различных состояния. Рассмотрим каждое из них.

При выборе команды “QUIT” система завершает свою работу. После завершения работы программа прощается с пользователем, после чего функция ввода команд будет недоступна.

При выборе команды “HELP” система выводит диалоговое окно, о котором было написано в самом начале описании системы.

При выборе решения квадратного уравнения (инструкция по вводу представлена в диалоговом окне) программа переходит в составное состояние системы “ввод уравнения”. Программа переходит в состояние проверки правильности ввода квадратного уравнения. При неправильном вводе уравнения программа переходит в состояние о выводе ошибки ввода. После вывода ошибки программа попросит у пользователя ввести коэффициенты еще раз.

При правильном вводе уравнения программа запускает алгоритм решения уравнения и выводит ответ с подробными описаниями решения уравнения. После вывода решения, составное состояние “ввод уравнения” завершается и переходит в состояние отображения окна.

## 2.2 Диаграмма классов системы

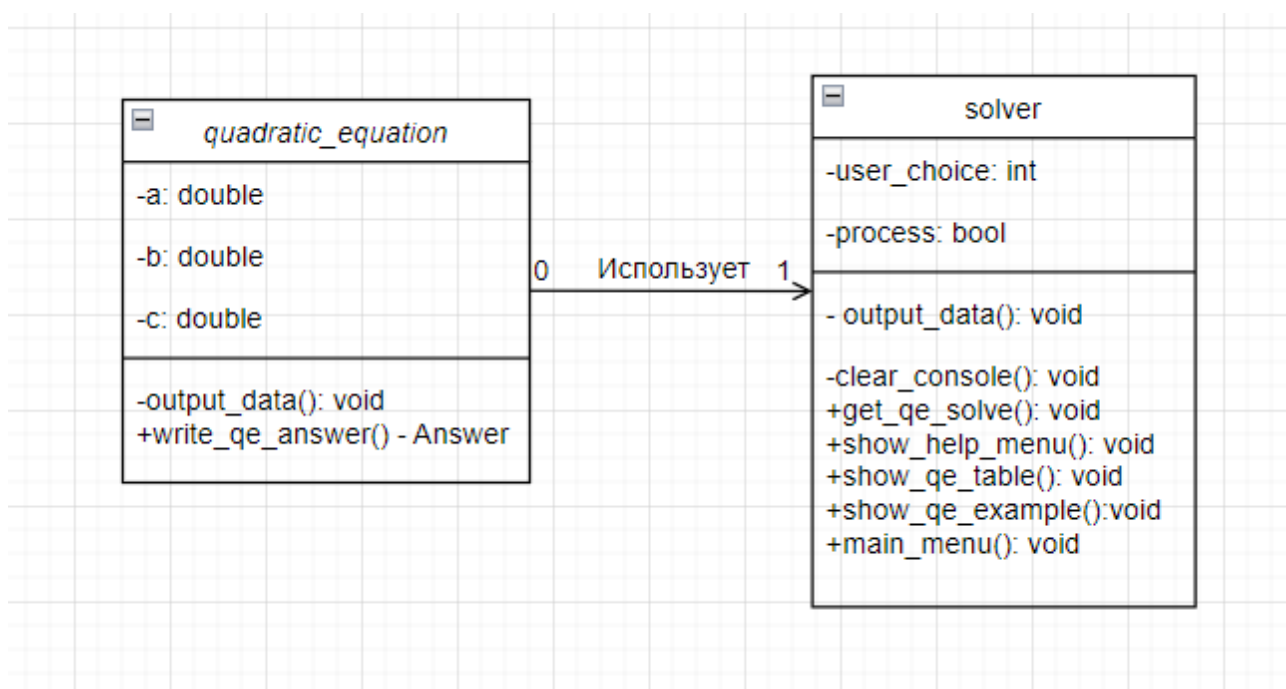


Рисунок 2.2 – Диаграмма классов системы

## 2.3 Диаграмма последовательности системы

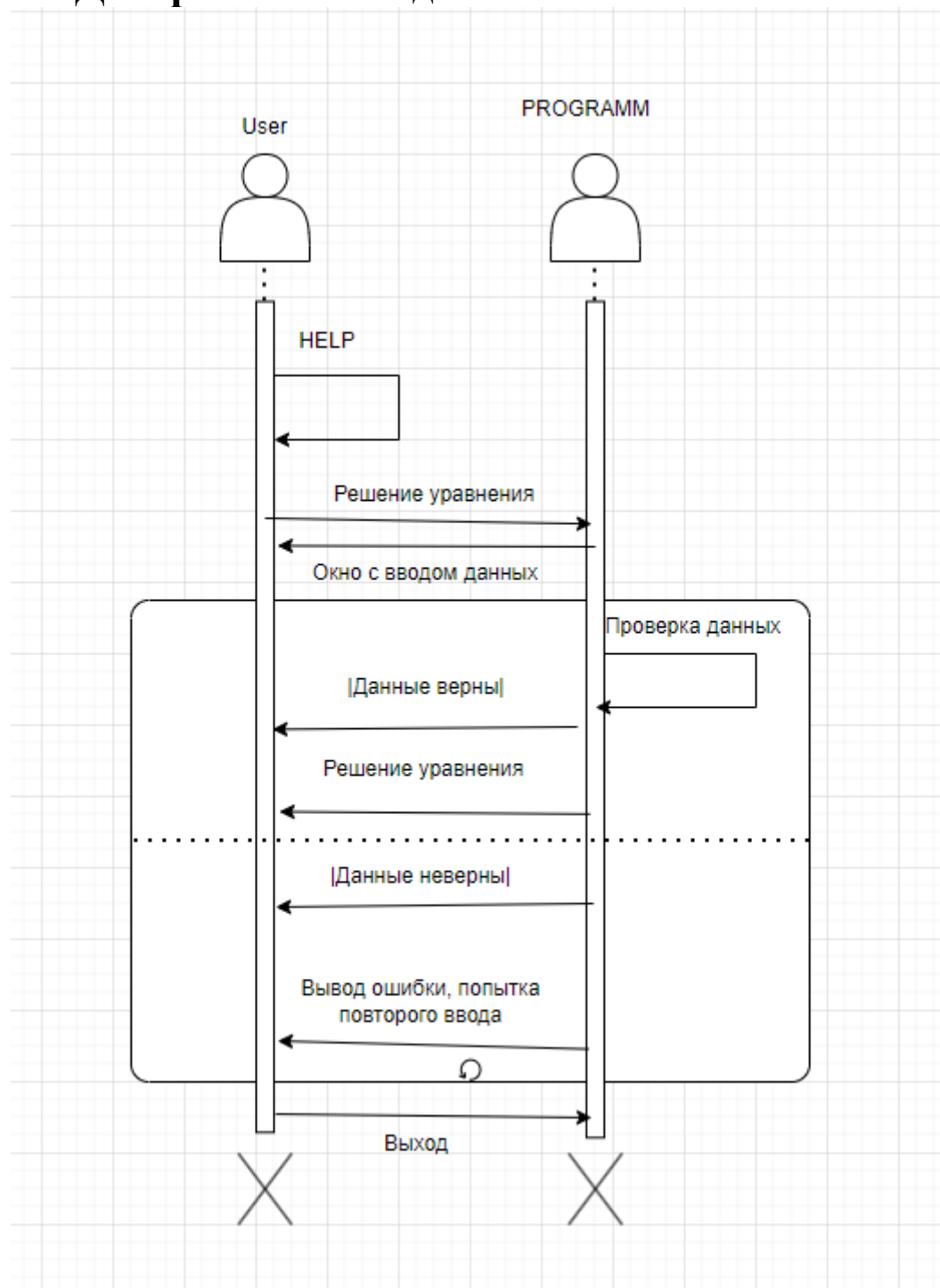


Рисунок 2.3 – Диаграмма последовательности системы

## 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

### 3.1 Реализация требований к системе

Требование №1: Программа должна быть написана на языке программирования C++.



```
моя курсовая.cpp X
E: > my_coursework > моя курсовая.cpp > ...
1  #include <iostream>
2  #include <cmath>
3  #include <fstream>
4  #include <string>
5  #include <Windows.h>
6  #undef max
7  using namespace std;
8
9  class quadratic_equation
10 {
11
12 private:
13     double a, b, c;
14
15     // функция выводящая ui
16     void output_data(string file_name)
17     {
18         string line;
19         ifstream file(file_name); // открываем файл
20
21         if (file.is_open())
22         {
23             while (getline(file, line))
24             {
25                 cout << line << endl;
26             }
27         }
28
29         file.close(); // закрываем файл
30     }
31 }
```

Требование №2: Приложение запускается и поддерживает свой цикл работы (для дальнейших примеров выполнения данного требования смотрите следующие страницы)

```
PS E:\my_coursework\output> & .\coursework.exe
```

Здравствуй, дорогой пользователь!  
Вас приветствует программа "Решатель".  
Она умеет решать квадратные уравнения.  
Далее будет представлено меню программы для более удобного пользования.

| МЕНЮ |

Напишите нужную цифру для выбора определенного варианта.

1. Подробное объяснение каждой функции меню.
2. Вывести таблицу видов квадратных уравнений.
3. Пример решения квадратного уравнения.
4. Решить квадратное уравнение с определенными коэффициентами.

\*\*\* или НАПИШИТЕ ЛЮБОЙ ДРУГОЙ СИМВОЛ для ВЫХОДА \*\*\*

Требование №3: При написании приложения, как минимум, были использованы классы в C++, объектный подход к проектированию системы, а также инкапсуляция.

```

> class quadratic_equation ...

class solver
{
private:
    bool process = true;
    int user_choice;

    tabnine: test | explain | document | ask
    void output_data(string file_name)
    {
        string line;
        ifstream file(file_name);

        if (file.is_open())
        {
            while (getline(file, line))
            {
                cout << line << endl;
            }

            file.close();
        }

        // функция которая чистит консоль пока сменяется диалоговое окно
        tabnine: test | explain | document | ask
        void clear_console()
        {
            COORD topLeft = { 0, 0 };
            HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
            CONSOLE_SCREEN_BUFFER_INFO screen;
            DWORD written;

            GetConsoleScreenBufferInfo(console, &screen);

```

Требование №4: Создан интерфейс пользователя, поддерживающий корректный пользовательский опыт и содержащий все необходимые пояснения к работе и эксплуатации.

На скриншоте ниже представлено меню по взаимодействию с программой, содержащее список доступных команд и инструкцию по их применению.

```
PS E:\my_coursework\output> & .\'coursework.exe'
```

Здравствуй, дорогой пользователь!  
Вас приветствует программа "Решатель".  
Она умеет решать квадратные уравнения.  
Далее будет представлено меню программы для более удобного пользования.

| МЕНЮ |

Напишите нужную цифру для выбора определенного варианта.

1. Подробное объяснение каждой функции меню.
2. Вывести таблицу видов квадратных уравнений.
3. Пример решения квадратного уравнения.
4. Решить квадратное уравнение с определенными коэффициентами.

\*\*\* или НАПИШИТЕ ЛЮБОЙ ДРУГОЙ СИМВОЛ для ВЫХОДА \*\*\*

Требование №5: Написана инструкция по эксплуатации, содержащая, в том числе, основные рекомендации по использованию и пояснения к возможным ошибкам в программе.

#### | ВСПОМОГАЮЩЕЕ МЕНЮ |

Здравствуй, дорогой пользователь, здесь я тебе расскажу, что делает каждый пункт меню.

1. Вызывает это диалоговое окно, если ты не понимаешь что делает эта программа.
2. Выводит таблицу со всеми 8 видами квадратных уравнений, подробно объясняя, как их определить, и как решить каждый из них.
3. Выводит примеры как пользоваться программой "Решатель" для решения квадратных уравнений.
4. Решает квадратное уравнение методом дискриминанта, подробно расписывая решение и определяя тип квадратного уравнения. Для использования данного алгоритма пользователь должен ввести три числа(поддерживаются числа с плавающей точкой) через пробел или через Enter. После успешного ввода коэффициентов, программа выведет решение на экран.

Примечание: находясь в меню, если ввести какой-либо символ, отличный от какого-либо пункта меню, программа закроется, так что будьте аккуратны, когда будете вводить команды.

\*\*\* ДЛЯ ВЫХОДА НАЖМИТЕ ENTER \*\*\*

Требование №6: Программа должна уметь решать все типы квадратных уравнений безошибочно.

```
Введите числовые коэффициенты для квадратного уравнения (через Enter/Пробел):
9 -5 -5

Принято! Готовлю решение для квадратного уравнения...
А вот и решение:
Уравнение принимает вид:
 $9x^2 - 5x - 5 = 0.$ 
Это уравнение 6 типа, поэтому решаем с помощью дискриминанта:
 $D = (-5)^2 - 4 * 9 * (-5) = 205.$ 
Так как  $D > 0$ ,  $\Rightarrow$  уравнение имеет два корня, которые вычисляются по формуле:


$$x_{1,2} = \frac{(-b \pm \sqrt{D})}{2 * a}$$


В итоге получаем корни:


$$x_1 = \frac{-(-5) + \sqrt{205}}{2 * 9} = 1.07321$$



$$x_2 = \frac{-(-5) - \sqrt{205}}{2 * 9} = -0.517657$$


Итоговый ответ:  $x = 1.07321; -0.517657$ 
Для перехода в главное меню напишите Enter.
```

Требование №7: Программа должна поддерживать различные типы данных, такие как целые, отрицательные, дробные числа для универсальности ее использования.

```

Введите числовые коэффициенты для квадратного уравнения (через Enter/Пробел):
3 -1.5 -1.5

Принято! Готовлю решение для квадратного уравнения...
А вот и решение:
Уравнение принимает вид:
3x^2 - 1.5x - 1.5 = 0.
Это уравнение 6 типа, поэтому решаем с помощью дискриминанта:
D = (-1.5)^2 - 4 * 3 * (-1.5) = 20.25.
Так как D > 0, => уравнение имеет два корня, которые вычисляются по формуле:

      (-b ± √D )
x1,2 = -----
      2 * a

В итоге получаем корни:

      (-(-1.5) + √20.25 )
x1 = ----- = 1
      2 * 3

      (-(-1.5) - √20.25 )
x2 = ----- = -0.5
      2 * 3

Итоговый ответ: x =1;-0.5
Для перехода в главное меню напишите Enter.

```

Данная программа решает уравнение для параметров  $a = 3$ ,  $b = -1.5$ ,  $c = -1.5$ .  
 Ответ предоставлен верно ( $x = 1; -0.5$ ).

Требование №8: Программа должна обрабатывать ошибки и исключительные случаи.

```

Введите числовые коэффициенты для квадратного уравнения (через Enter/Пробел):
6 нгне 6
Друг, ну давай по нормальному, введи уже числа и давай решим с тобой это квадратное уравнение:
2 4 2

Принято! Готовлю решение для квадратного уравнения...
А вот и решение:
Уравнение принимает вид:
2x^2 + 4x + 2 = 0.
Это уравнение 6 типа, поэтому решаем с помощью дискриминанта:
D = 4^2 - 4 * 2 * 2 = 0.
Так как дискриминант равен 0, уравнение имеет всего один корень:

      -4
x = ----- = -1.
      2 * 2

Итоговый ответ: x =-1
Для перехода в главное меню напишите Enter.

```

Требование №9: Программа должна быть способна работать на различных платформах, включая Windows, macOS, Linux

Введите числовые коэффициенты для квадратного уравнения (через Enter/Пробел):  
23 65 4

Принято! Готовлю решение для квадратного уравнения...

А вот и решение:

Уравнение принимает вид:

$$23x^2 + 65x + 4 = 0.$$

Это уравнение б типа, поэтому решаем с помощью дискриминанта:

$$D = 65^2 - 4 * 23 * 4 = 3857.$$

Так как  $D > 0$ ,  $\Rightarrow$  уравнение имеет два корня, которые вычисляются по формуле:

$$x_{1,2} = \frac{(-b \pm \sqrt{D})}{2 * a}$$

В итоге получаем корни:

$$x_1 = \frac{(-65 + \sqrt{3857})}{2 * 23} = -0.0629402$$

$$x_2 = \frac{(-65 - \sqrt{3857})}{2 * 23} = -2.76315$$

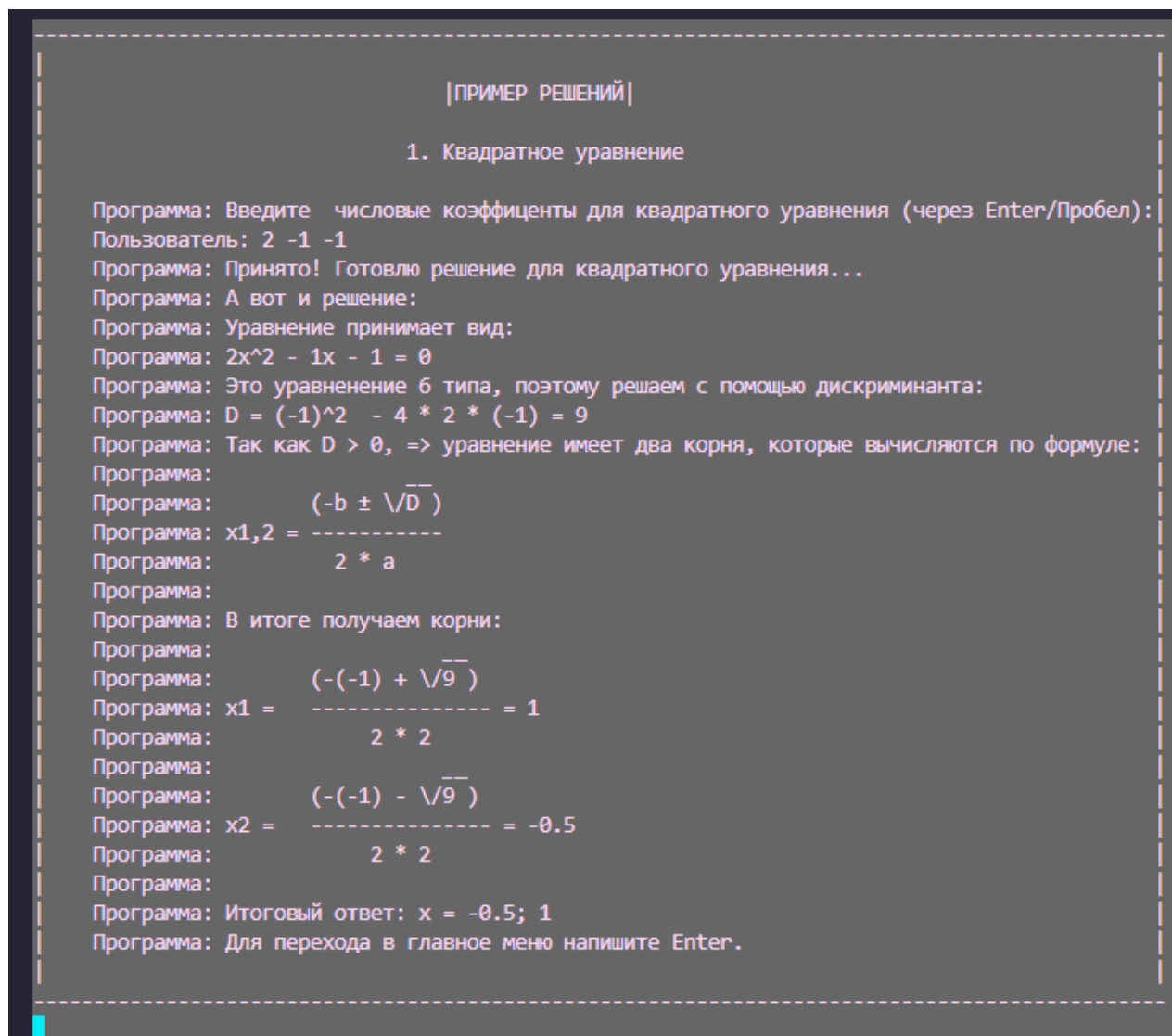
Итоговый ответ:  $x = -0.0629402; -2.76315$

Для перехода в главное меню напишите Enter.

Пример работы программы в операционной системе Windows.

Требование №10: Программа должна уметь выводить пример решения квадратного уравнения, чтобы пользователь мог понять, как программа будет решать квадратные уравнения.





## 3.2 Функциональное тестирование программного продукта

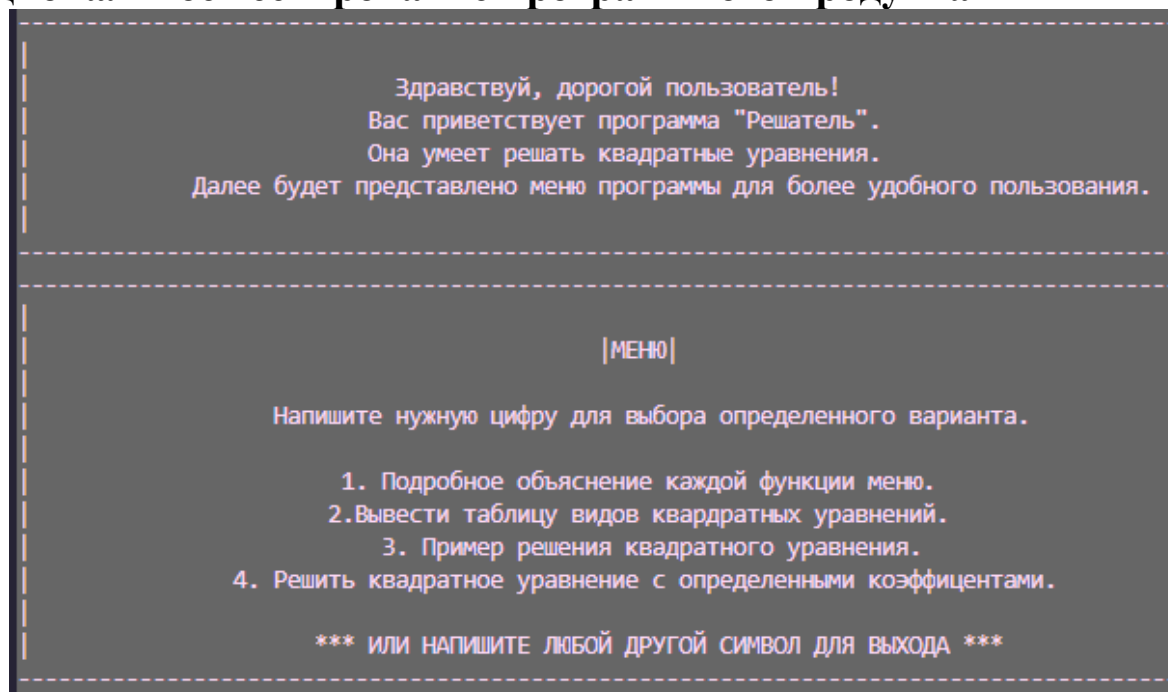


Рисунок 3.1 – Меню программы

На старте программы выводится контекстное меню, состоящее из 4 возможных команд на выбор. Выбор команды 1 выводит таблицу, содержащую описание всех команд в меню. При выборе команды 4 пользователь вводит 3 коэффициента, а программа в свою очередь решает уравнение с подробными шагами. Выбор неправильной команды завершает цикл работы программы и выводит сообщение о завершении работы. При выборе команды 2 пользователю выводится подробная таблица со всеми 8 типами уравнений, и наилучшими способами их решения. При выборе команды 3 пользователю выводится пример решения для квадратного уравнения, чтобы пользователь мог понять как работает программа.

Вот так выглядит таблица видов квадратных уравнений, с помощью которой можно определить тип квадратного уравнения и самый простой способ которым можно его решить

	B = 0		B != 0	
A = 0	$0 = 0$ (1)	$C = 0$ (3)	$Bx + C = 0$ (5)	$Bx = 0$ (7)
A != 0	$Ax^2 = 0$ (2)	$Ax^2 + C = 0$ (4)	$Ax^2 + Bx + C = 0$ (6)	$Ax^2 + Bx = 0$ (8)
	$C = 0$	$C != 0$	$C = 0$	

1. Уравнение верно при любом x.

2. Уравнение имеет единственный корень -  $x = 0$ .

3. Уравнение не имеет корней, и при любом x оно будет не верным.

4. Если  $-C / A >= 0$ , то уравнение имеет два корня по следующим формулам:

$x1 = \sqrt{-C/A}$	и	$x2 = -\sqrt{-C/A}$
--------------------	---	---------------------

Иначе уравнение корней не имеет.

5. Уравнение имеет единственный корень -  $x = -C/B$ .

6. Классическое квадратное уравнение.

$D = B^2 - 4 * A * C$

Уравнение имеет три случая:

I. При  $D > 0$ :

Уравнение имеет два корня:

$x1, x2 = (-B \pm \sqrt{D}) / 2 * a$

II. При  $D = 0$ :

$x = -B/2*A$

III. При  $D < 0$ :

Уравнение не имеет корней.

7. Уравнение имеет единственный корень  $x = 0$ .

8. Уравнение принимает вид  $x(Ax + B) = 0$  и имеет два корня:

$x = 0$   
 $x = -B/A$

НАЖМИТЕ ENTER, ЧТОБЫ ВЫЙТИ В ГЛАВНОЕ МЕНЮ.

Рисунок 3.2 – Таблица с решениями квадратных уравнений

Таблица содержит описание решения всех 8 типов квадратных уравнений, представленных в теоретической части. В начале выводится таблица, в которой

с помощью какие коэффициенты равняются 0 понять, какой это тип уравнения, после чего снизу посмотреть решение определенного типа уравнения.

Один из примеров решения уравнения представлен на рисунке 3.3. Пользователь вводит значение параметров a, b, c, а программа выдает подробное решение. На первом шаге решения выводится изначальный вид уравнения. После чего выводится тип уравнение, способ его решение, после чего выводятся корни уравнения, если они есть.

```
Введите числовые коэффициенты для квадратного уравнения (через Enter/Пробел):  
2 -1 -1  
  
Принято! Готовлю решение для квадратного уравнения...  
А вот и решение:  
Уравнение принимает вид:  
2x^2 - 1x - 1 = 0.  
  
Так как D > 0, => уравнение имеет два корня, которые вычисляются по формуле:  
  
      (-b ± √D )  
x1,2 = -----  
      2 * a  
  
В итоге получаем корни:  
  
      (-(-1) + √9 )  
x1 = ----- = 1  
      2 * 2  
  
      (-(-1) - √9 )  
x2 = ----- = -0.5  
      2 * 2  
  
Для перехода в главное меню напишите Enter.
```

Решение 3.4 – Решение квадратных уравнений

При выборе неправильной команды программа завершает свою работу и выводит сообщение о завершение работы. (Рисунок 3.5)

```
Прощай, дорогой пользователь! Надеюсь я сегодня был полезным, желаю тебе удачи, надеюсь скоро увидимся!
```

Рисунок 3.5 – Выход из программы

### 3.3 Инструкция по эксплуатации

Программа может использоваться в консольных приложениях либо используя преждевременную сборку либо сразу же запуская ее через .exe файл. На рисунке 3.6 представлен способ с запуском программы через .exe.

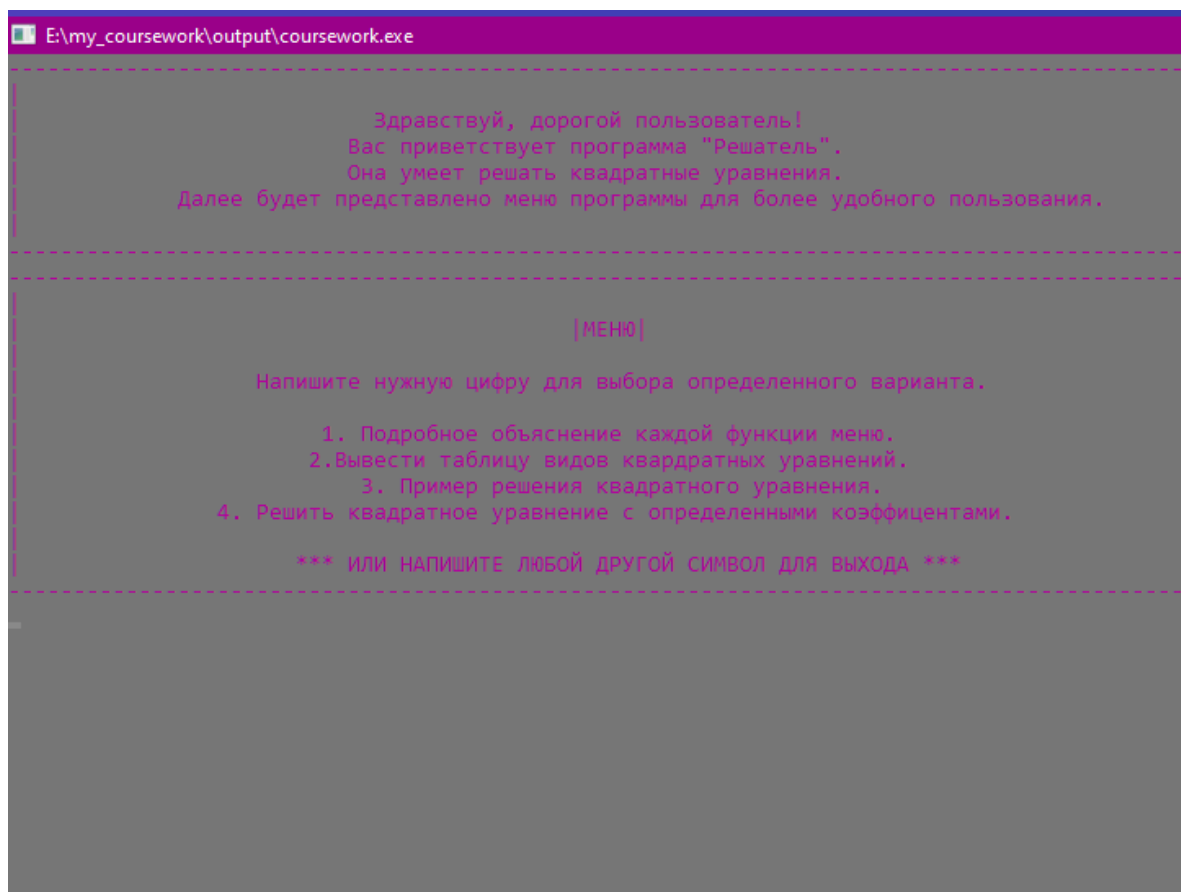


Рисунок 3.6 – запуск программы через .exe

Сразу же после запуска программы высвечивается меню для пользователя, в котором собраны список возможных операций и соответствующие им команды. (Рисунок 3.1)

Если вам нужно изучить 8 типов квадратных уравнений, то при выборе команды 2, отображается таблица с 8 типами квадратных уравнений, методами решения, и параметрами, которые при этом должны быть введены (Рисунок 3.2).

Для решения квадратного уравнения выберите команду 4, нажмите Enter, далее программа предоставит вам на ввод значение параметров  $a$ ,  $b$ ,  $c$ . Введите каждое из значений через Enter или через пробел. После ввода параметров программа начнет решать данное уравнение и выведет подробное решение с конечным ответом если он есть. В противном случае программа выведет сообщение о том, что решения не существует. (Рисунок 3.4).

После нахождения ответа вам снова станет доступен ввод команд из меню. Для получения списка команд и меню, выберите команду 1. В результате выполнения команды высветится отображение контекстного меню (Рисунок 3.7).

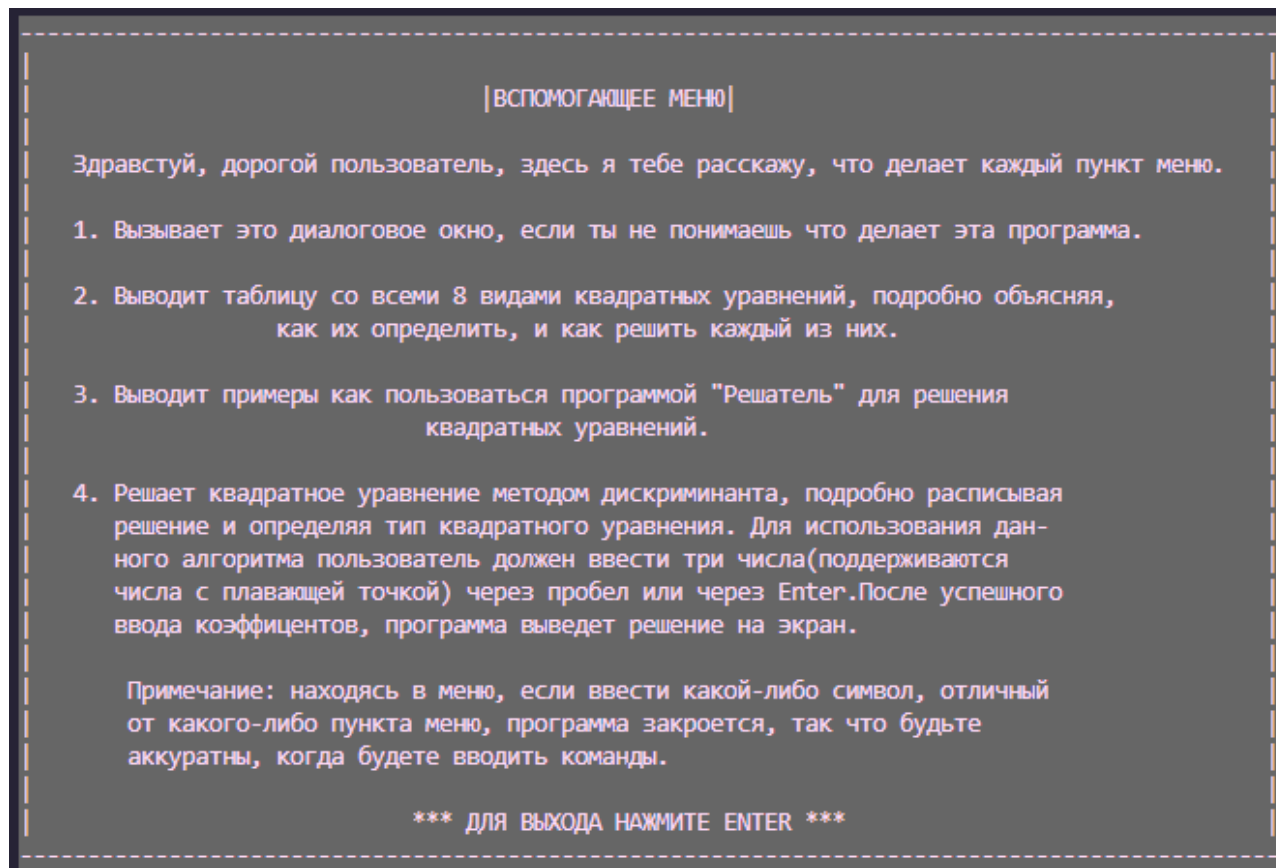


Рисунок 3.7 – вспомогательное меню

Для выхода из программы введите неправильную команду. Программа сразу же завершит свою работу и выведет сообщение о завершении своей работы (Рисунок 3.5).

## ЗАКЛЮЧЕНИЕ

Важно отметить, что успешное завершение Вашей курсовой работы представляет значительный вклад в область программирования систем. Вы провели основательный анализ текущего состояния технологии, четко определили системные требования, и разработали архитектуру системы, продемонстрировав её функциональность через диаграммы и последовательности. Разработанный программный продукт, который был должным образом проверен и сопровождается инструкцией, является полезным дополнением к данной технической области.

Дальнейшее развитие Вашего продукта с акцентом на добавление новых математических операций, интеграцию решения квадратных уравнений с комплексными корнями и совершенствование пользовательского интерфейса несомненно расширит его применение и улучшит пользовательский опыт. Это будет способствовать большей адаптивности и персонализации продукта с учетом потребностей и предпочтений потребителя, что сделает его более конкурентоспособным на рынке программных продуктов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Давыдова, Н.Е. Различные методы решения квадратных уравнений в 8 классе / Н.Е. Давыдова // Наука и образование: отечественный и зарубежный опыт : Сборник трудов конференции Двадцать первой Международной научно-практической конференции, Белгород, 17 июня 2019 года. — Белгород: ООО ГиК, 2019. — С. 153-155. — URL: <https://elibrary.ru/item.asp?id=39195788> (дата обращения: 09.03.2024).
2. Прямостанов, С.М. Метод коэффициентов при решении квадратных уравнений / С.М. Прямостанов, Л.В. Лысогорова // Юный учёный. — 2018. — № 1.1 (15.1). — С. 66-67. — URL: <https://moluch.ru/young/archive/15/1165/> (дата обращения: 09.03.2024).
3. Озерова Г.П. Информационные технологии: Mathcad: для студентов инженерных специальностей очной и заочных форм обучения: учебно-методическое пособие / Инженерная школа ДВФУ. — Владивосток: Дальневост. федерал. ун-т, 2020. — ISBN 978-5-7444-4776-2. — URL: <https://www.dvfu.ru/upload/medialibrary/aa4/Озерова%20Г.П.%20Информационные%20технологии.%20Mathcad.pdf> (дата обращения: 09.03.2024).
4. Тарасян К.В., Кондратьева Т.Н. Особенности и недостатки программы Mathcad // Материалы Международной студенческой научной конференции «Студенческий научный форум». — 2020. — № 2. — С. 72-74. — URL: <https://scienceforum.ru/2020/article/2018019680> (дата обращения: 09.03.2024).
5. Приложение Photomath — калькулятор онлайн в твоём телефоне // Apps4Life : [сайт]. — URL: <https://apps4.life/prilozhenie-photomath-kalkuljator-onlajn-v-tvoem-telefone/> (дата обращения: 09.03.2024).
6. Microsoft Math Solver : официальный сайт. — URL: <https://mathsolver.microsoft.com/ru> (дата обращения: 09.03.2024).

7. Разница между Microsoft Math Solver и Photomath // AskAnyDifference.com : [сайт]. — URL: <https://askanydifference.com/ru/difference-between-microsoft-math-solver-and-photomath/> (дата обращения: 09.03.2024)



# ПРИЛОЖЕНИЯ

## Приложение А – Исходный код программы

*Листинг А.1 — Файл coursework.cpp*

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <string>
#include <Windows.h>
#undef max
using namespace std;

class quadratic_equation
{
private:
    double a, b, c;

    // функция выводящая ui
    void output_data(string file_name)
    {
        string line;
        ifstream file(file_name); // открываем файл

        if (file.is_open())
        {
            while (getline(file, line))
            {
                cout << line << endl;
            }
        }

        file.close(); // закрываем файл
```

```

    }

public:
    explicit quadratic_equation(double set_a = 0, double set_b = 0, double
set_c = 0)
    {
        a = set_a;
        b = set_b;
        c = set_c;
    }

    void write_qe_answer()
    {
        output_data("ui/qe/3.start_solving.txt");

        if (a == 0 && b == 0 && c == 0)
        {
            output_data("ui/qe/4.first_type.txt");
        }

        else if (a != 0 && b == 0 && c == 0)
        {
            output_data("ui/qe/universal_start.txt");
            cout << a << "x^2 = 0." << endl;
            output_data("ui/qe/5.second_type_end.txt");
        }

        else if (a == 0 && b == 0 && c != 0)
        {
            output_data("ui/qe/universal_start.txt");
            cout << c << " = 0." << endl;
            output_data("ui/qe/6.third_type_end.txt");
        }

        else if (a != 0 && b == 0 && c != 0)
        {

```

```

        output_data("ui/qe/universal_start.txt");
        if (c > 0)
        {
            cout << a << "x^2 + " << c << " = 0." << endl;
        }
        else
        {
            cout << a << "x^2 - " << fabs(c) << " = 0." << endl;
        }

        output_data("ui/qe/7.fourth_type_end.txt");

        if (-c / a >= 0) {
            double x = sqrt(-c / a);
            cout << "Ура, у нас все получилось, теперь найдем корни
уравнения!" << endl;
            cout << "          _____ " << endl;
            cout << "x1 = \\\ / (-(" << c << ") / " << a << ") = " << x <<
endl;
            cout << "          _____ " << endl;
            cout << "x2 = -\\\ / (-(" << c << ") / " << a << ") = " << -x <<
endl;

            cout << "" << endl;
            cout << "Итоговый ответ: x = " << x << "; " << -x << "." << endl;
        }
        else
        {
            cout << "-c/a = " << (-c / a) << endl;
            cout << "К сожалению получилось так, что -c/a < 0, а значит
корней нет." << endl;
            cout << "Итоговый ответ: Корней нет." << endl;
        }
    }

    else if (a == 0 && b != 0 && c != 0)
    {
        double x = -c / b;

```

```

        output_data("ui/qe/universal_start.txt");
        if (c > 0)
        {
            cout << b << "x + " << c << " = 0." << endl;
        }
        else
        {
            cout << b << "x - " << fabs(c) << " = 0." << endl;
        }

        output_data("ui/qe/8.fifth_type_end.txt");
        cout << "x = -c / b = " << x << endl;
        cout << "Итоговый ответ: x = " << x << '.' << endl;
    }

    else if ( a != 0 && b != 0 && c != 0)
    {
        double d;
        d = pow(b, 2) - 4 * a * c;

        output_data("ui/qe/universal_start.txt");
        if (a > 0 && b > 0 && c > 0)
        {
            cout << a << "x^2 + " << b << "x + " << c << " = 0." << endl;
            cout << "Это уравнение 6 типа, поэтому решаем с помощью  
дискриминанта:" << endl;
            cout << "D = " << b << "^2 - 4 * " << a << " * " << c << " = "
<< d << "." << endl;
        }

        else if (a > 0 && b > 0 && c < 0)
        {
            cout << a << "x^2 + " << b << "x - " << fabs(c) << " = 0." <<
endl;
            cout << "Это уравнение 6 типа, поэтому решаем с помощью  
дискриминанта:" << endl;

```

```

        cout << "D = " << b << "^2 - 4 * " << a << " * (" << c << ") = "
<< d << "." << endl;
    }

    else if (a > 0 && b < 0 && c > 0)
    {
        cout << a << "x^2 - " << fabs(b) << "x + " << c << " = 0." <<
endl;

        cout << "Это уравнение 6 типа, поэтому решаем с помощью
дискриминанта:" << endl;

        cout << "D = (" << b << ")^2 - 4 * " << a << " * " << c << " = "
<< d << "." << endl;
    }

    else if (a < 0 && b > 0 && c > 0)
    {
        cout << a << "x^2 + " << b << "x + " << c << " = 0." << endl;

        cout << "Это уравнение 6 типа, поэтому решаем с помощью
дискриминанта:" << endl;

        cout << "D = " << b << "^2 - 4 * (" << a << ") * " << c << " = "
<< d << "." << endl;
    }

    else if (a > 0 && b < 0 && c < 0)
    {
        cout << a << "x^2 - " << fabs(b) << "x - " << fabs(c) << " = 0."
<< endl;

        cout << "Это уравнение 6 типа, поэтому решаем с помощью
дискриминанта:" << endl;

        cout << "D = (" << b << ")^2 - 4 * " << a << " * (" << c << ") = "
" << d << "." << endl;
    }

    else if (a < 0 && b > 0 && c < 0)
    {
        cout << a << "x^2 + " << b << "x - " << fabs(c) << " = 0." <<
endl;

        cout << "Это уравнение 6 типа, поэтому решаем с помощью
дискриминанта:" << endl;

        cout << "D = " << b << "^2 - 4 * (" << a << ") * (" << c << ") = "
" << d << "." << endl;
    }

```

```

    }

    else if (a < 0 && b < 0 && c > 0)
    {
        cout << a << "x^2 - " << fabs(b) << "x + " << c << " = 0." <<
endl;

        cout << "Это уравнение 6 типа, поэтому решаем с помощью
дискриминанта:" << endl;

        cout << "D = (" << b <<")^2 - 4 * (" << a << " ) * " << c << " =
" << d << "." << endl;
    }

    else if (a < 0 && b < 0 && c < 0)
    {
        cout << a << "x^2 - " << fabs(b) << "x - " << fabs(c) << " = 0."
<< endl;

        cout << "Это уравнение 6 типа, поэтому решаем с помощью
дискриминанта:" << endl;

        cout << "D = (" << b <<")^2 - 4 * (" << a << " ) * (" << c << ")
= " << d << "." << endl;
    }

    if (d > 0)
    {
        double x1, x2;

        x1 = (-b + sqrt(d))/(2 * a);
        x2 = (-b - sqrt(d))/(2 * a);

        output_data("ui/qe/9.1.sixth_type_d_higher_0.txt");

        if (b > 0)
        {
            cout << "          ____" << endl;
            cout << "          (-" << b << " + \\/" << d <<")" << endl;
        }
        else
        {
            cout << "          ____" << endl;

```

```

        cout << "          (-(" << b << ") + \\/" << d <<" )" << endl;
    }
    cout << "x1 =  ----- = " << x1 << endl;
    cout << "          2 * " << a << " " << endl;

    if (b > 0)
    {
        cout << "          ____" << endl;
        cout << "          (-" << b << " - \\/" << d <<" )" << endl;
    }
    else
    {
        cout << "          ____" << endl;
        cout << "          (-(" << b << ") - \\/" << d <<" )" << endl;
    }
    cout << "x2 =  ----- = " << x2 << endl;
    cout << "          2 * " << a << " " << endl;
    cout << "" << endl;
    cout << "ИТОГОВЫЙ ответ: x =" << x1 << ";" << x2 << endl;
}

else if (d == 0)
{
    double x = -b/(2 * a);

    cout << "Так как дискриминант равен 0, уравнение имеет всего
один корень:" << endl;

    if (b > 0)
    {
        cout << "          -" << b << " " << endl;
    }
    else
    {
        cout << "          -(" << b << ") " << endl;
    }
    cout << "x = ----- = " << x << '.' << endl;
}

```

```

        cout << "          2 * " << a << " " << endl;

        cout << "Итоговый ответ: x =" << x << endl;

    }

    else {

        cout << "Так как дискриминант меньше 0, следовательно, уравнение
корней не имеет" << endl;

        cout << "Итоговый ответ: Корней нет." << endl;

    }

}

else if (a == 0 && b != 0 && c == 0)
{
    output_data("ui/qe/universal_start.txt");
    cout << b << "x = 0." << endl;
    output_data("ui/qe/10.seventh_type_end.txt");
}

else if (a != 0 && b != 0 && c == 0)
{
    double x = -b / a;
    output_data("ui/qe/universal_start.txt");

    if (b > 0)
    {
        cout << a << "x^2 + " << b << "x = 0." << endl;
        cout << "Данное уравнение эквивалентно следующему:" << endl;
        cout << "x(" << a << "x + " << b << ") = 0." << endl;
    }

    else
    {
        cout << a << "x^2 - " << fabs(b) << "x = 0.\n" << endl;
        cout << "Данное уравнение эквивалентно следующему:\n" << endl;
        cout << "x(" << a << "x - " << fabs(b) << ") = 0." << endl;
    }
}

```



```

    }

    output_data("ui/qe/11.eigth_type_end.txt");
    if ( b > 0 )
    {
        cout << "x2 = -" << b << " / " << a << " = " << x << '.' <<
endl;
    }
    else
    {
        cout << "x2 = -(" << b << ") / " << a << " = " << x << '.' <<
endl;
    }
    cout << "Итоговый ответ: x = 0; " << x << '.' << endl;
}
}
};

class solver
{
private:
    bool process = true;
    int user_choice;

    void output_data(string file_name)
    {
        string line;
        ifstream file(file_name);

        if (file.is_open())
        {
            while (getline(file, line))
            {
                cout << line << endl;
            }
        }
    }
};

```

```

        }

    }

    file.close();
}

// функция которая чистит консоль пока сменяется диалоговое окно
void clear_console()
{
    COORD topLeft = { 0, 0 };
    HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_SCREEN_BUFFER_INFO screen;
    DWORD written;

    GetConsoleScreenBufferInfo(console, &screen);

    FillConsoleOutputCharacterA
    (
        console, ' ', screen.dwSize.X * screen.dwSize.Y, topLeft, &written
    );

    FillConsoleOutputAttribute
    (
        console, FOREGROUND_GREEN | FOREGROUND_RED | FOREGROUND_BLUE,
        screen.dwSize.X * screen.dwSize.Y, topLeft, &written
    );

    SetConsoleCursorPosition(console, topLeft);
    system("color 8D");
}

public:
    void get_qe_solve()
    {
        double a, b, c;

        output_data("ui/qe/1.entering_coef_in_qe.txt");
    }
}

```

```

    cin >> a >> b >> c;

    while(cin.fail())
    {

        cin.clear();

        cin.ignore(numeric_limits<streamsize>::max(), '\n');

        output_data("ui/qe/2.bad_entering.txt");
        cin >> a >> b >> c;
    }

    quadratic_equation equation(a, b, c);
    equation.write_qe_answer();
    output_data("ui/qe/12.end.txt");
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cin.get();
}

void show_help_menu()
{
    output_data("ui/3.help_menu.txt");
    cin.get();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
}

void show_qe_table()
{
    output_data("ui/4.table_equation.txt");
    cin.get();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
}

void show_qe_qi_example()
{
    output_data("ui/5.example_solution.txt");

```

```

        cin.get();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    void main_menu()
    {
        while (process)
        {
            output_data("ui/2.first_instruction.txt");
            cin >> user_choice;
            clear_console();
            switch (user_choice)
            {
            case 1:
                show_help_menu();
                clear_console();
                break;
            case 2:
                show_qe_table();
                clear_console();
                break;
            case 3:
                show_qe_qi_example();
                clear_console();
                break;
            case 4:
                get_qe_solve();
                clear_console();
                break;
            case 5:
                clear_console();
                break;
            default:
                output_data("ui/6.goodbye.txt");
                process = false;
            }
        }
    }

```

```

        cout << endl;

    }

}

};

void output_data(string file_name)
{
    string line;
    ifstream file(file_name);

    if (file.is_open())
    {
        while (getline(file, line))
        {
            cout << line << endl;
        }
    }

    file.close();
}

int main() {
    system("color 8D");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    output_data("ui/1.hello_to_user.txt");
    solver menu;
    menu.main_menu();
    return 0;
}

```

*Листинг A.2 — Файл ui/1.hello\_to\_user.txt*

```

-----
|                                     |
|                                     |
|               Здравствуй, дорогой пользователь!               |
|               Вас приветствует программа "Решатель".           |
|                                     |

```

	Она умеет решать квадратные уравнения.	
	Далее будет представлено меню программы для более удобного пользования.	
-----		

```
-----
|
|                                     |
|                               |МЕНЮ|                               |
|                                     |
|
|      Напишите нужную цифру для выбора определенного варианта.      |
|                                     |
|      1. Подробное объяснение каждой функции меню.                    |
|      2. Вывести таблицу видов квадратных уравнений.                  |
|      3. Пример решения квадратного уравнения.                        |
|      4. Решить квадратное уравнение с определенными коэффициентами. |
|                                     |
|      *** ИЛИ НАПИШИТЕ ЛЮБОЙ ДРУГОЙ СИМВОЛ ДЛЯ ВЫХОДА ***            |
|
|
|-----
```

```
-----
|
|                               |ВСПОМОГАЮЩЕЕ МЕНЮ|                               |
|                               |
|
|      Здравстуй, дорогой пользователь, здесь я тебе расскажу, что делает каждый пункт
|      меню. |
|
|
|      1. Вызывает это диалоговое окно, если ты не понимаешь что делает эта программа.
|
|
|
|      2. Выводит таблицу со всеми 8 видами квадратных уравнений, подробно объясняя,
|      как их определить, и как решить каждый из них.
|
|
|      3. Выводит примеры как пользоваться программой "Решатель" для решения
|      квадратных уравнений.
|
|
|      4. Решает квадратное уравнение методом дискриминанта, подробно расписывая
|      решение и определяя тип квадратного уравнения. Для использования дан-
|      ного алгоритма пользователь должен ввести три числа(поддерживаются
|      числа с плавающей точкой) через пробел или через Enter. После успешного
|      ввода коэффициентов, программа выведет решение на экран.
|
|-----
```

Примечание: находясь в меню, если ввести какой-либо символ, отличный от какого-либо пункта меню, программа закроется, так что будьте аккуратны, когда будете вводить команды.

\*\*\* ДЛЯ ВЫХОДА НАЖМИТЕ ENTER \*\*\*

Листинг A.5 — Файл ui/4.table\_equation.txt

Вот так выглядит таблица видов квадратных уравнений, с помощью которой можно определить тип квадратного уравнения

и самый простой способ которым можно его решить

	$B = 0$	$B \neq 0$	
$A = 0$	$0 = 0$ (1)	$C = 0$ (3)	$Bx + C = 0$ (5)
			$Bx = 0$ (7)
$A \neq 0$	$Ax^2 = 0$ (2)	$Ax^2 + C = 0$ (4)	$Ax^2 + Bx + C = 0$ (6)
			$Ax^2 + Bx = 0$ (8)
	$C = 0$	$C \neq 0$	$C = 0$

1. Уравнение верно при любом  $x$ .



2. Уравнение имеет единственный корень -  $x = 0$ .

3. Уравнение не имеет корней, и при любом  $x$  оно будет не верным.

4. Если  $-C / A \geq 0$ , то уравнение имеет два корня по следующим формулам:

$$x_1 = \sqrt{-C/A} \quad \text{и} \quad x_2 = -\sqrt{-C/A}$$

Иначе уравнение корней не имеет.

5. Уравнение имеет единственный корень -  $x = -C/B$ .

6. Классическое квадратное уравнение.

$$D = B^2 - 4 * A * C$$

Уравнение имеет три случая:

I. При  $D > 0$ :

Уравнение имеет два корня:

$$x_{1,2} = \frac{-B \pm \sqrt{D}}{2 * a}$$

II. При  $D = 0$ :

$$x = -B/2*A$$

III. При  $D < 0$ :

Уравнение не имеет корней.

7. Уравнение имеет единственный корень  $x = 0$ .

8. Уравнение принимает вид  $x(Ax + B) = 0$  и имеет два корня:

$$x = 0$$

$$x = -B/A$$

НАЖМИТЕ ENTER, ЧТОБЫ ВЫЙТИ В ГЛАВНОЕ МЕНЮ.

*Листинг А.6 — Файл ui/5.example\_solution.txt*

```

-----
|
|                                     |
|               |ПРИМЕР РЕШЕНИЙ|               |
|
|                                     |
|               1. Квадратное уравнение               |
|
|                                     |
|  Программа: Введите числовые коэффициенты для квадратного уравнения
(через Enter/Пробел):|
|  Пользователь: 2 -1 -1               |
|  Программа: Принято! Готовлю решение для квадратного уравнения...
|
|  Программа: А вот и решение:               |
|  Программа: Уравнение принимает вид:               |
|  Программа: 2x^2 - 1x - 1 = 0               |

```

| Программа: Это уравнение 6 типа, поэтому решаем с помощью дискриминанта: |

| Программа:  $D = (-1)^2 - 4 * 2 * (-1) = 9$  |

| Программа: Так как  $D > 0$ ,  $\Rightarrow$  уравнение имеет два корня, которые вычисляются по формуле: |

| Программа:  $_____$  |

| Программа:  $(-b \pm \sqrt{D})$  |

| Программа:  $x_{1,2} = \frac{_____}{_____}$  |

| Программа:  $2 * a$  |

| Программа: |

| Программа: В итоге получаем корни: |

| Программа:  $_____$  |

| Программа:  $(-(-1) + \sqrt{9})$  |

| Программа:  $x_1 = \frac{_____}{2 * 2} = 1$  |

| Программа:  $2 * 2$  |

| Программа:  $_____$  |

| Программа:  $(-(-1) - \sqrt{9})$  |

| Программа:  $x_2 = \frac{_____}{2 * 2} = -0.5$  |

| Программа:  $2 * 2$  |

| Программа: |

| Программа: Итоговый ответ:  $x = -0.5; 1$  |

| Программа: Для перехода в главное меню напишите Enter. |

|

|

-----  
Листинг А.7 — Файл ui/6.goodbye.txt

Прощай, дорогой пользователь! Надеюсь я сегодня был полезным, желаю тебе удачи, надеюсь скоро увидимся!

Листинг А.8 — Файл ui/qe/1.entering\_coef\_in\_qe

Введите числовые коэффициенты для квадратного уравнения (через Enter/Пробел):

Листинг A.9 — Файл ui/qe/2.bad\_entering.txt

Друг, ну давай по нормальному, введи уже числа и давай решим с тобой это квадратное уравнение:

Листинг A.10 — Файл ui/qe/3.start\_solving.txt

Принято! Готовлю решение для квадратного уравнения...

А вот и решение:

Листинг A.11 — Файл ui/qe/4.first\_type.txt

$$0x^2 + 0x + 0 = 0$$

Это уравнение 1 типа, поэтому мы сразу получаем:

$$0 = 0$$

Следовательно, при любом  $x$  это уравнение будет верно.

Итоговый ответ:  $x = \text{любое число}$

Листинг A.12 — Файл ui/qe/5.second\_type\_end.txt

Это уравнение 2 типа, поэтому у уравнение всего единственное решение:

$$x = 0$$

Итоговый ответ:  $x = 0$

Листинг A.13 — Файл ui/qe/6.third\_type\_end.txt

Это уравнение 3 типа, поэтому у этого уравнения нет решений.

Так как при любом  $x$  уравнение не будет равняться 0.

Итоговый ответ: Корней нет

Листинг A.14 — Файл ui/qe/7.fourth\_type\_end.txt

Это уравнение 4 типа, и если  $-C/A \geq 0$ , тогда это уравнение будет иметь два корня, иначе - 0.

Сейчас мы это и проверим:

Листинг A.15 — Файл ui/qe/8.fifth\_type\_end.txt

Это уравнение 5 вида, поэтому оно имеет единственное решение:

Листинг A.16 — Файл ui/qe/9.1.sixth\_type\_d\_higher\_0.txt

Так как  $D > 0$ ,  $\Rightarrow$  уравнение имеет два корня, которые вычисляются по формуле:

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2 * a}$$

В итоге получаем корни:

Листинг A.17 — Файл ui/qe/10.seventh\_type\_end.txt

Это уравнение 7 типа, поэтому у уравнение всего единственное решение:

$$x = 0$$

Итоговый ответ:  $x = 0$

Листинг A.18 — Файл ui/qe/11.eigth\_type\_end.txt

Это уравнение 8 типа, поэтому оно имеет 2 корня:

$$x = 0$$

Листинг A.19 — Файл ui/qe/12.end.txt

Для перехода в главное меню напишите Enter.

Листинг A.20 — Файл ui/qe/universal\_start.txt

Уравнение принимает вид:

