Министерство просвещения Приднестровской Молдавской Республики

Государственное образовательное учреждение среднего профессионального образования «Тираспольский техникум информатики и права»

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

«Разработка консольного приложения для решения квадратного уравнения»

по учебной дисциплине «Информатика»

по специальности 12.02.24 «Информационные системы и программирование»

Выполнил Белько. В.И

обучающийся I курса

специальность 12.02.24 «Информационные системы и программирование»

Руководитель Шандригоз Наталья Николаевна

Преподаватель информатики высшей квалификационной категории

Допущен к защите

«\_\_\_»\_\_\_\_\_\_\_202\_\_ г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тирасполь 20\_\_

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 3 |
| 1. ВЫЧИСЛЕНИЕ КВАДРАТНОГО КОРНЯ | 4 |
| 1.1 Интерфейс пользователя |  |
| 1.2 Решение квадратного уравнения  1.3 Вывод результата  1.4 Обработка ошибок |
| 2. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ | 5 |
| 2.1 Язык программирования  2.2 Обработка ввода  2.3 Алгоритмы |
| 3. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ | 6 |
| 3.1 Работоспособное приложение  3.2 Реализация обработки ошибок  3.3 Тестирование приложения |
| ТЕСТИРОВАНИЕ ПРОГРАММЫ | 7 |
| ЗАКЛЮЧЕНИЕ | 8 |
| СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ | 9 |
| ПРИЛОЖЕНИЯ | 10 |

**ВВЕДЕНИЕ**

Современный мир информационных технологий предоставляет уникальные возможности для автоматизации решения различных задач. В контексте программирования, создание консольных приложений остается актуальным и эффективным подходом для разработки легковесных, удобных в использовании инструментов.

Данный проект направлен на разработку консольного приложения, способного решать квадратные уравнения. Решение квадратных уравнений является широко используемой задачей в математике и физике. Создание удобного инструмента для данного расчета позволит пользователям быстро и эффективно находить корни уравнения, углубляясь в основы алгебры.

Проект предлагает использование языка программирования C# для реализации консольного интерфейса, обеспечивающего ввод коэффициентов от пользователя и вывод результата. Решение квадратного уравнения, основанное на формуле дискриминанта, станет основой для программы, которая не только предоставляет ответ, но и обрабатывает различные случаи, такие как наличие двух корней, одного корня или отсутствие действительных корней.

Разработка данного консольного приложения не только позволит применить базовые концепции программирования, но и даст пользователям инструмент для быстрого и простого решения квадратных уравнений без необходимости использования сложных математических методов.

1. **ВЫЧИСЛЕНИЕ КВАДРАТНОГО КОРНЯ**

**1.1 Интерфейс пользователя**

* Создание интерфейса для ввода коэффициентов:
  + Разработка текстового интерфейса, явно указывающего пользователю, какие значения нужно ввести (a, b, c).
* Обработка пользовательского ввода:
  + Реализация механизма обработки введенных пользователем данных, включая проверку наличия всех необходимых коэффициентов и их корректность.
* Проверка валидности введенных данных:
  + Внедрение механизмов проверки на валидность введенных данных, таких как обнаружение ошибок в формате и предотвращение возможных исключений.

**1.2 Решение квадратного уравнения**

* Разработка алгоритма для решения уравнения:
  + Создание эффективного алгоритма, основанного, например, на формуле дискриминанта, для вычисления корней квадратного уравнения.
* Обработка различных случаев:
  + Реализация кода, способного корректно обрабатывать ситуации, такие как уравнение без корней, с одним корнем или с двумя корнями.

**1.3 Вывод результата**

* Организация вывода корней уравнения на экран:
  + Разработка механизма для красивого и понятного вывода корней уравнения на консоль.
* Предоставление дополнительной информации:
  + Добавление информативных сообщений, например, в случае отсутствия корней, наличия одного корня или двух различных корней.

**1.4 Обработка ошибок**

* Реализация механизма обработки ошибок:
  + Создание системы, способной обнаруживать и корректно реагировать на ошибки, связанные с вводом данных или решением уравнения.

1. **ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ**

**4.1 Язык программирования**

* Выбор C# в качестве языка программирования:
  + Обоснование выбора C# для реализации консольного приложения.
  + Рассмотрение преимуществ C# в контексте данной задачи, таких как простота синтаксиса и поддержка объектно-ориентированного программирования.

**4.2 Обработка ввода**

* Использование стандартных библиотек C#:
  + Имплементация механизма обработки ввода данных от пользователя с использованием стандартных библиотек C#.

**4.3 Алгоритмы**

* Реализация алгоритмов для решения квадратного уравнения на C#:
  + Написание алгоритма, основанного на формуле дискриминанта, для эффективного решения квадратного уравнения на языке C#.

**3. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ**

**5.1 Работоспособное приложение**

* Разработка работоспособного консольного приложения на C#:
  + Написание кода, который корректно принимает ввод от пользователя, решает квадратное уравнение и выводит результаты.

**5.2 Интуитивный интерфейс**

* Создание интуитивного интерфейса на C#:
  + Разработка пользовательского интерфейса, который является понятным и удобным в использовании.

**5.3 Устойчивость к ошибкам**

* Реализация механизмов обработки ошибок на C#:
  + Создание системы, способной обнаруживать и обрабатывать ошибки, связанные с вводом данных или вычислениями в контексте языка C#.

**5.4 Тестирование**

* Тестирование на C#:
  + Проведение тестирования приложения на различных наборах входных данных для подтверждения корректности работы и выявления возможных проблем.

**ТЕСТИРОВАНИЕ ПРОГРАММЫ**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp929100

{

internal class Program

{

static void Main(string[] args)

{

try

{

double x1 = 0;

double x2 = 0;

//Введение коэфицентов

Console.Write("Введите коэфицент a = ");

double a = double.Parse(Console.ReadLine());

Console.Write("Введите коэфицент b = ");

double b = double.Parse(Console.ReadLine());

Console.Write("Введите коэфицент c = ");

double c = double.Parse(Console.ReadLine());

//Вычисление дискриминанта

double d = Math.Pow(b, 2) - 4 \* a \* c;

Console.WriteLine($"D = {b}^2 - 4 \* {a} \* {c}");

if (d > 0)

{

//Вычисление корней

Console.WriteLine("Дискриминант имеет 2 корня");

x1 = (-b + Math.Sqrt(d)) / 2 \* a;

x2 = (-b - Math.Sqrt(d)) / 2 \* a;

Console.WriteLine($"x1 = {x1:f1}");

Console.WriteLine($"x2 = {x2:f1}");

}

else if (d < 0)

{

Console.WriteLine("Корней нет");

}

else

{

Console.WriteLine("Дискриминант имеет 1 корень");

x1 = -b / 2 \* a;

Console.WriteLine($"x1 = {x1}");

}

}

//Вывод ошибки, при вводе неправильных данных

catch

{

Console.Write("Произошла ошибка!");

}

Console.Read();

}

}

}

**ЗАКЛЮЧЕНИЕ**

В ходе данной проектной работы было предложено и разработано консольное приложение на языке программирования C#, предназначенное для решения квадратного уравнения. Проект включал в себя различные этапы, начиная от создания интерфейса пользователя и обработки ввода, заканчивая реализацией алгоритмов для решения уравнения.

**Результаты проекта:**

**1. Интерфейс пользователя:**

* Создан понятный текстовый интерфейс для ввода коэффициентов уравнения.
* Реализован механизм обработки пользовательского ввода с проверкой валидности данных.

**2. Решение квадратного уравнения:**

* Написан эффективный алгоритм, основанный на формуле дискриминанта, для нахождения корней квадратного уравнения.
* Реализован код, способный обрабатывать различные случаи, такие как отсутствие корней, наличие одного корня или двух корней.

**3. Вывод результата:**

* Создан механизм красивого вывода корней уравнения на экран.
* Добавлена дополнительная информация о результатах решения, учитывая различные сценарии.

**4. Обработка ошибок:**

* Реализован механизм обработки ошибок, предотвращающий возможные проблемы при вводе данных или решении уравнения.

**5. Тестирование:**

* Проведено тестирование приложения на различных входных данных для проверки корректности работы.
* Выявлены и устранены потенциальные ошибки, обеспечив стабильность работы программы.

В итоге, разработанное консольное приложение представляет собой полезный инструмент для решения квадратных уравнений с простым и понятным интерфейсом, что делает его доступным для широкого круга пользователей.

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

https://www.google.ru

<https://metanit.com>

**ПРИЛОЖЕНИЯ**

Microsoft Visual Studio

Word

Google Chrome