class Program

{

static void Main()

{

Console.WriteLine("Введите коэффициенты для квадратного уравнения (ax^2 + bx + c):");

Console.Write("a: ");

double a = Convert.ToDouble(Console.ReadLine());

Console.Write("b: ");

double b = Convert.ToDouble(Console.ReadLine());

Console.Write("c: ");

double c = Convert.ToDouble(Console.ReadLine());

if (a == 0)

{

Console.WriteLine("Это не квадратное уравнение.");

return;

}

double discriminant = b \* b - 4 \* a \* c;

if (discriminant > 0)

{

double root1 = (-b + Math.Sqrt(discriminant)) / (2 \* a);

double root2 = (-b - Math.Sqrt(discriminant)) / (2 \* a);

Console.WriteLine($"Уравнение имеет два решения: x1 = {root1}, x2 = {root2}");

}

else if (discriminant == 0)

{

double root = -b / (2 \* a);

Console.WriteLine($"Уравнение имеет одно решение: x = {root}");

}

else

{

Console.WriteLine("Уравнение не имеет вещественных решений.");

}

}

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название теста | Описание  сценария | Входные  данные | Выходные  данные | Удачное/  неудачное  тестирование | Предложения  по  исправлению  найденных  ошибок. |
| Правильность расчетов | Проверка расчета корней дискриминанта | Переменные:  a=2  b=8  c=4 | Уравнение имеет два решения: x1 = -0,5857864376269049,  x2 = -3,414213562373095 | удачное | - |
| **Два различных вещественных корня (D > 0):** | Проверка расчета ,когда у нас 2 корня | Переменные: a: 1  b: -3  c: 2 | Уравнение имеет два решения: x1 = 2, x2 = 1 | удачное | - |
| **Один вещественный корень** | Проверка расчета ,когда у нас 1 корня | Переменные: a: 1  b: -2  c: 1 | Уравнение имеет одно решение: x = 1 | удачное | - |
| **Два комплексных корня** | Проверка расчета при комплесных корнях | Переменные: a = 1, b = 2, c = 2 | Уравнение не имеет вещественных решений. | неудачное | Реализовать решение уровнений с комплесными корнями |
| **Деление на ноль** | **Деление на ноль** | Переменные: a = 0, b = 0, c = 0 | Это не квадратное уравнение. | неудачное | Реализовать решение уровнений при возможностях делить на 0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Неправильная обработка комплексных корней:** Программа не всегда корректно вычисляет и выводит комплексные корни, в некоторых случаях выдавая ошибку или неверный результат.  
• **Некорректная обработка случая a=0:**  В случае, когда a=0 (уравнение вырождается в линейное), программа может выдавать неверный результат или ошибку. Неправильная обработка случаев с бесконечным числом решений или отсутствием решений.  
• **Отсутствие обработки ошибок:** Программа не обрабатывает некорректный ввод данных (например, нечисловые значения коэффициентов). В таких случаях возникает исключение, приводящее к аварийному завершению программы.

2

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите коэффициенты для кубического уравнения (ax^3 + bx^2 + cx + d):");

Console.Write("a: ");

double a = Convert.ToDouble(Console.ReadLine());

Console.Write("b: ");

double b = Convert.ToDouble(Console.ReadLine());

Console.Write("c: ");

double c = Convert.ToDouble(Console.ReadLine());

Console.Write("d: ");

double d = Convert.ToDouble(Console.ReadLine());

if (a == 0)

{

Console.WriteLine("Это не кубическое уравнение.");

return;

}

Console.WriteLine($"Решаем уравнение: {a}x^3 + {b}x^2 + {c}x + {d} = 0");

FindRoots(a, b, c, d);

}

static void FindRoots(double a, double b, double c, double d)

{

double f = ((3 \* b / a) - ((b \* b) / (a \* a))) / 3;

double g = ((2 \* b \* b \* b) / (a \* a \* a) - (9 \* b \* c) / (a \* a) + (27 \* d / a)) / 27;

double h = (g \* g / 4) + (f \* f \* f / 27);

if (h > 0)

{

double R = -(g / 2) + Math.Sqrt(h);

double S = Math.Cbrt(R);

double T = -(g / 2) - Math.Sqrt(h);

double U = Math.Cbrt(T);

double root1 = (S + U) - (b / (3 \* a));

Console.WriteLine($"Корень: {root1}");

}

else if (h == 0)

{

double R = -(g / 2);

double S = Math.Cbrt(R);

double root1 = 2 \* S - (b / (3 \* a));

double root2 = -S - (b / (3 \* a));

Console.WriteLine($"Корень 1: {root1}");

Console.WriteLine($"Корень 2: {root2}");

}

else

{

double i = Math.Sqrt((g \* g / 4) - h);

double j = Math.Cbrt(i);

double k = Math.Acos(-(g / (2 \* i)));

double L = j \* -1;

double M = Math.Cos(k / 3);

double N = Math.Sqrt(3) \* Math.Sin(k / 3);

double P = -(b / (3 \* a));

double root1 = 2 \* j \* Math.Cos(k / 3) - P;

double root2 = L \* (M + N) + P;

double root3 = L \* (M - N) + P;

Console.WriteLine($"Корни: {root1}, {root2}, {root3}");

}

}

}

Задание 3. Подготовить набор тестовых вариантов для обнаружения ошибок в программах заданий 1 и 2, для каждого из заданий должна быть своя таблица.

Результат оформить в следующем виде таблицы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название теста | Описание  сценария | Входные  данные | Выходные  данные | Удачное/  неудачное  тестирование | Предложения  по  исправлению  найденных  ошибок. |
| Правильность расчетов | Проверка расчета корней дискриминанта | Переменные:  a=1  b=-6  c=11  d =-6 | Корни: 2,2426406871192848, -2,2426406871192848, 1,9999999999999996 | неудачное | Переделать алгоритм решения уровнения |
| **Три различных вещественных корня** | Проверка | a: 1  b: -6  c: 11  d: -6 | **Корни: 2,2426406871192848, -2,2426406871192848, 1,9999999999999996** | удача | - |
| **Случаи с** a=0 | Проверка частных случаев с a=0 | a = 0, b = 1, c = -2, d = 1 | Это не кубическое уравнение. | неудачное | Реализовать все случаи ,когда a= 0 ,когда у нас корен-это квадратное уровнение,линейное и тд |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

• **Неправильное определение типа корней:** Программа не всегда корректно определяет, какие типы корней (вещественные, комплексные) содержит уравнение. В некоторых случаях она выдаёт неверный тип или не обрабатывает комплексные корни.  
• **Неполное решение:** В некоторых случаях программа не находит все три корня, особенно когда есть комплексные корни.  
• **Отсутствие обработки ошибок:** Программа не обрабатывает некорректный ввод данных (например, нечисловые значения коэффициентов). В таких случаях возникает исключение, что приводит к аварийному завершению программы.