Отчёт по лабораторной работе №1 по курсу БКИТ

«Решение квадратного уравнения»

Подготовил

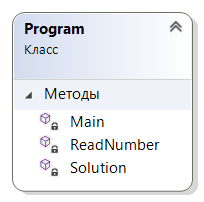
Ионов С.А.

ИУ5-31Б

1. Описание задания

Разработать программу для решения квадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент А, В, С введен некорректно, то необходимо проигнорировать некорректное значение и ввести коэффициент повторно.
4. Диаграмма классов



1. Текст программы

using System;

namespace Lab1

{

class Program

{

static void Main(string[] args)

{

Solution(ReadNumber('a'), ReadNumber('b'), ReadNumber('c'));

Console.ReadLine();

}

/// <summary>

/// Ввод и проверка

/// </summary>

/// <param name="t">Отображение названия коэффициента</param>

/// <returns>Введенный коэффициент</returns>

static double ReadNumber(char t)

{

double r = 0;

bool k = true;

do

{

Console.Write("Write " + t + ": ");

k = Double.TryParse(Console.ReadLine(), out r);

} while (!k);

return r;

}

/// <summary>

/// решение уравнения

/// </summary>

static void Solution(double a, double b, double c)

{

if (a == 0) //поиск решений

{

if (b != 0)

{

Console.WriteLine("\nAnswer: " + -c / b);

}

else

{

if (c != 0)

Console.WriteLine("\nNo solution");

else

Console.WriteLine("\nInfinite numbers of solutions");

}

}

else

{

if (b != 0)

{

if (c != 0)

{

double d = Math.Sqrt(b \* b - 4 \* a \* c);

if (d >= 0)

{

Console.WriteLine("\nx1 = " + (-b - d) / (2 \* a));

Console.WriteLine("x2 = " + (-b + d) / (2 \* a));

}

else

{

Console.WriteLine("\nNo real solution");

}

}

else

{

Console.WriteLine("\nx1 = " + 0);

Console.WriteLine("x2 = " + -b / a);

}

}

else

{

double r = -c / a;

if (r >= 0)

{

Console.WriteLine("\nx1 = " + -Math.Sqrt(r));

Console.WriteLine("x2 = " + Math.Sqrt(r));

}

else

Console.WriteLine("\nNo real solution");

}

}

}

}

}

1. Экранные формы с примерами выполнения программы
2. Проверка функции ввода информации (ReadNumber)

В данном примере производится попытка неправильного ввода первого коэффициента (вместо числа пользователь ошибочно вводит то букву, то слово). В итоге при таком вводе логический метод TryParse класса Double возвращает значение false, и вводить пользователь начинает заново, пока не введёт то значение, которое можно преобразовать в тип Double. В итоге далее вводятся правильно все 3 коэффициента, но уравнение с введёнными коэффициентами не имеет решений на действительном поле чисел, о чём и свидетельствует вывод. Так как в главной функции Main в конце был прописан метод Console.ReadLine(); , то после вывода информации на экран, не выводится надпись «о необходимости нажать кнопку», и мы спокойно можем анализировать выходную информацию.



1. Проверка функции ввода информации (ReadNumber)

В данном примере я демонстрирую, как программа ведёт себя при правильном вводе 1-го коэффициента, но неправильном вводе 2-го, а в дальнейшем 3-го. Если пользователь ввёл верно первый коэффициент, но ошибся во втором, то ему не нужно заново вводить всё сначала, программа запрашивает неправильно введённый коэффициент. Также программа предназначена для решения не только квадратных уравнений. В частном случае, когда старший коэффициент равен 0, уравнение превращается в линейное и имеет один корень.



1. Проверка функции решения уравнения (Solution)

Ниже приведён случай, когда все коэффициенты квадратного уравнения равны 0. В таком случае мы имеем бесконечное количество решений.



1. Проверка функции решения уравнения (Solution)

Далее проверено поведение при вводе коэффициентов 0 при переменной x и ненулевого коэффициента c. Логичный результат – решений нет.



1. Проверка функции решения уравнения (Solution)

Ниже отмечена особенность вывода при совпадающих корнях, в случаях, когда уравнение представляет собой полный квадрат, а также, если первый коэффициент не 0, а остальные 0. В таких случаях корень печатается два раза, потому что, так как коэффициент при переменной в квадрате ненулевой, то уравнение имеет 2 корня, другое дело, что они могут совпадать.





1. Проверка функции решения уравнения (Solution)

Наконец проверена способность правильно находить корни квадратного уравнения при числовом значении дискриминанта.



1. Проверка функции решения уравнения (Solution)

Также проверен случай, когда дискриминант уравнения меньше 0, то есть в действительном поле чисел решений квадратное уравнение не имеет.



1. Проверка функции решения уравнения (Solution)

Правильно программа работает и при решении неполного квадратного уравнения (свободный член равен 0).

