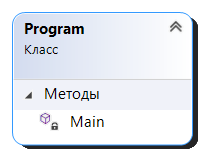
**Отчёт по лабораторной работе № 1**

**Волгина А. Д., ИУ5-33Б**

Разработать программу для решения биквадратного уравнения – <https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B5%D1%82%D0%B2%D1%91%D1%80%D1%82%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8#%D0%91%D0%B8%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5>

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. Первой строкой программа выводит ФИО разработчика и номер группы.
5. Корни уравнения выводятся зеленым цветом. Если корней нет, то сообщение выводится красным цветом.

ДОПОЛНИТЕЛЬНОЕ ТРЕБОВАНИЕ. Коэффициенты А, В, С задаются в виде параметров командной строки. Если они не указаны, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.



using System.Text;

using System.Threading.Tasks;

namespace Lab1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Волгина Анна Денисовна, ИУ5-33Б");

double A, B, C;

HashSet<double> roots = new HashSet<double>();

try

{

if(args.Length != 3)

{

A = Convert.ToDouble(Console.ReadLine());

B = Convert.ToDouble(Console.ReadLine());

C = Convert.ToDouble(Console.ReadLine());

}

else

{

A = Convert.ToDouble(args[0]);

B = Convert.ToDouble(args[1]);

C = Convert.ToDouble(args[2]);

}

double D = B \* B - 4 \* A \* C;

if (D < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("No roots");

Console.ResetColor();

return;

}

double x1 = (-B + Math.Sqrt(D)) / (2 \* A), x2 = (-B - Math.Sqrt(D)) / (2 \* A);

if (x1 >= 0)

{

roots.Add(Math.Sqrt(x1));

roots.Add(-Math.Sqrt(x1));

}

if (x2 >= 0)

{

roots.Add(Math.Sqrt(x2));

roots.Add(-Math.Sqrt(x2));

}

switch (roots.Count)

{

case 0:

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("No roots");

Console.ResetColor();

break;

case 1:

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("One root: {0}", roots.ElementAt(0));

Console.ResetColor();

break;

case 2:

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Two roots: {0} {1}", roots.ElementAt(0), roots.ElementAt(1));

Console.ResetColor();

break;

case 3:

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Three roots: {0} {1} {2}", roots.ElementAt(0), roots.ElementAt(1), roots.ElementAt(2));

Console.ResetColor();

break;

case 4:

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Four roots: {0} {1} {2} {3}", roots.ElementAt(0), roots.ElementAt(1), roots.ElementAt(2), roots.ElementAt(3));

Console.ResetColor();

break;

}

return;

}

catch (FormatException e)

{

Console.WriteLine("Некорректный ввод");

Console.WriteLine(e.Message);

}

catch(Exception e)

{

Console.WriteLine("Собственное исключение");

Console.WriteLine(e.Message);

}

}

}

}





