**Libraries**

Modification of the application architecture for automated testing.

1 - Sample code not suitable for automated testing

2 - Modification of the program for automatic testing: moving the functionality not related to input and output to the screen into a separate method

3 - Creating a library. Connecting the library to the project. Analysis of the execution of the program using the library. Contents of the directory with the project when connecting libraries

**Option 1**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace umop7o3o3iLibraries

{

class Program

{

static void Main(string[] args)

{

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

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

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

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

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

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

Console.WriteLine(x1);

Console.WriteLine(x2);

}

}

}

**Option 2**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace umop7o3o3iLibraries

{

class Program

{

static double[] Solve(double a, double b, double c)

{

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

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

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

return new[] { x1, x2 };

}

static void Main(string[] args)

{

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

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

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

var result = Solve(a, b, c);

Console.WriteLine(result[0]);

Console.WriteLine(result[1]);

}

}

}

**Option 3**

**Contents of Program.cs (umop7.3.3iLibraries)**

using Solver;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace umop7o3o3iLibraries

{

class Program

{

static void Main(string[] args)

{

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

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

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

var result = QuadraticEquationsSolver.Solve(a, b, c);

Console.WriteLine(result[0]);

Console.WriteLine(result[1]);

}

}

}

**Contents of QuadraticEquationsSolver.cs (Solver)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Solver

{

public class QuadraticEquationsSolver

{

public static double[] Solve(double a, double b, double c)

{

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

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

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

return new[] { x1, x2 };

}

}

}