**Больше тестов!**

**Содержимое файла Program.cs (Equations):**

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]);

Console.ReadKey();

}

}

}

**Содержимое файла 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;

if (discriminant > 0)

{

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

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

return new[] { x1, x2 };

}

if (discriminant == 0)

{

return new[] { -b / (2 \* a) };

}

else return new double[0];

}

}

}

**Содержимое файла SolverTest.cs (Solver.Tests):**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace Solver.Tests

{

[TestClass] // <= так нужно пометить класс, чтобы система тестирования начала искать в нем тесты.

public class QuadraticEquatonSolverTest

{

void TestEquation(double a, double b, double c, params double[] expectedResult)

{

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

Assert.AreEqual(expectedResult.Length, result.Length);

for (int i = 0; i < result.Length; i++)

Assert.AreEqual(expectedResult[i], result[i]);

}

[TestMethod]

public void OrdinaryEquations()

{

TestEquation(1, -3, 2, 2, 1 );

}

[TestMethod]

public void NegativeDiscriminant()

{

TestEquation(1, 1, 1);

}

[TestMethod]

public void ZeroDiscriminant()

{

TestEquation(1, 2, 1, -1);

}

}

}