**Министерство науки и высшего образования Российской** **Федерации** **Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Радиотехнический»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Лабораторная работа №1

по дисциплине «Базовые компоненты интернет-технологий» на тему:

«Решение биквадратного уравнения»

Выполнил:

студент группы РТ5-31Б

В. И. Плешаков

Проверил:

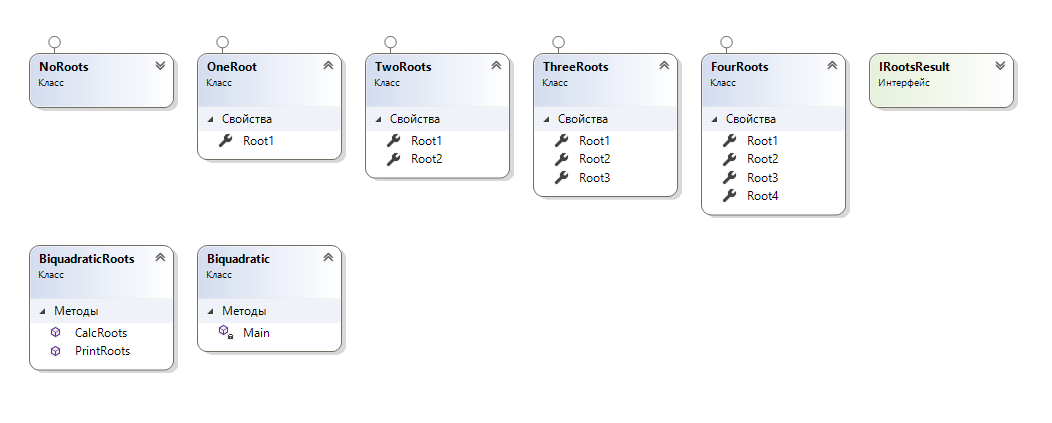
Ю. Е. Гапанюк

2020 г.

Задание:

Разработать программу для решения биквадратного уравнения.

Диаграмма классов:



Текст программы:

using System;

namespace Biquadratic

{

interface IRootsResult { }

class NoRoots : IRootsResult { }

class OneRoot : IRootsResult

{

public double Root1 { get; set; }

}

class TwoRoots: IRootsResult

{

public double Root1 { get; set; }

public double Root2 { get; set; }

}

class ThreeRoots : IRootsResult

{

public double Root1 { get; set; }

public double Root2 { get; set; }

public double Root3 { get; set; }

}

class FourRoots : IRootsResult

{

public double Root1 { get; set; }

public double Root2 { get; set; }

public double Root3 { get; set; }

public double Root4 { get; set; }

}

class BiquadraticRoots

{

/// <summary>

/// Вычисление корней биквадратного уравнения

/// </summary>

public IRootsResult CalcRoots(double a, double b, double c)

{

//Вычисление дискриминанта

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

//В случае, если дискриминант равен нулю

if (D == 0)

{

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

if (sqRoot == 0)

{

return new OneRoot()

{

Root1 = 0

};

}

else if (sqRoot > 0)

{

return new TwoRoots()

{

Root1 = Math.Sqrt(sqRoot),

Root2 = -Math.Sqrt(sqRoot)

};

}

else return new NoRoots();

}

//Если дискриминант больше нуля

else if (D > 0)

{

double sqrtD = Math.Sqrt(D);

double rt1 = (-b + sqrtD) / (2 \* a);

double rt2 = (-b - sqrtD) / (2 \* a);

//Если rt1 < rt2, то меняем их местами

if (rt1 < rt2)

{

double buff = rt1;

rt1 = rt2;

rt2 = buff;

}

//4 корня

if (rt1 > 0 && rt2 > 0)

{

return new FourRoots()

{

Root1 = Math.Sqrt(rt1),

Root2 = -Math.Sqrt(rt1),

Root3 = Math.Sqrt(rt2),

Root4 = -Math.Sqrt(rt2),

};

}

//3 корня

else if (rt1 > 0 && rt2 == 0)

{

return new ThreeRoots()

{

Root1 = Math.Sqrt(rt1),

Root2 = -Math.Sqrt(rt1),

Root3 = 0

};

}

//2 корня

else if (rt1 > 0 && rt2 < 0)

{

return new TwoRoots()

{

Root1 = Math.Sqrt(rt1),

Root2 = -Math.Sqrt(rt1)

};

}

//1 корень

else if (rt1 == 0 && rt2 < 0)

{

return new OneRoot()

{

Root1 = 0

};

}

else return new NoRoots();

}

//Если дискриминант меньше нуля

else

{

return new NoRoots();

}

}

/// <summary>

/// Вывод корней

/// </summary>

public void PrintRoots(double a, double b, double c)

{

IRootsResult roots = this.CalcRoots(a, b, c);

string resultType = roots.GetType().Name;

if (resultType == "NoRoots")

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Корней нет.");

Console.ResetColor();

}

else if (resultType == "OneRoot")

{

OneRoot rt = (OneRoot)roots;

Console.WriteLine("Один корень {0:F4}", rt.Root1);

}

else if (resultType == "TwoRoots")

{

TwoRoots rt = (TwoRoots)roots;

Console.WriteLine("Два корня: {0:F4} и {1:F4}", rt.Root1, rt.Root2);

}

else if (resultType == "ThreeRoots")

{

ThreeRoots rt = (ThreeRoots)roots;

Console.WriteLine("Три корня: {0:F4}, {1:F4} и {2:F4}", rt.Root1, rt.Root2, rt.Root3);

}

else if (resultType == "FourRoots")

{

FourRoots rt = (FourRoots)roots;

Console.WriteLine("Четыре корня: {0:F4}, {1:F4}, {2:F4} и {3:F4}", rt.Root1, rt.Root2, rt.Root3, rt.Root4);

}

}

}

class Biquadratic

{

static void Main(string[] args)

{

double a, b, c;

bool ConvertResult;

string str;

do

{

Console.WriteLine("Введите коэффицент a");

str = Console.ReadLine();

ConvertResult = double.TryParse(str, out a);

} while (!ConvertResult || a == 0);

do

{

Console.WriteLine("Введите коэффицент b");

str = Console.ReadLine();

ConvertResult = double.TryParse(str, out b);

} while (!ConvertResult);

do

{

Console.WriteLine("Введите коэффицент c");

str = Console.ReadLine();

ConvertResult = double.TryParse(str, out c);

} while (!ConvertResult);

BiquadraticRoots eq = new BiquadraticRoots();

Console.ForegroundColor = ConsoleColor.Green;

eq.PrintRoots(a, b, c);

Console.ResetColor();

}

}

}

Пример выполнения программы:

