**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №1

«Создание программы для решения биквадратного уравнения»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31 |  | преподаватель каф. ИУ5 |
| Агличеев Михаил |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020г.

1. Описание задания

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.
6. 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.

2. UML-диаграмма классов

Диаграмма класса отсутствует, так как в данной лабораторной не предполагается создание собственных классов, лишь взаимодействие с классом Console.

3. Листинг программы

using System;

using System.Collections.Generic;

//using static System.Math;

namespace Лабораторная\_\_\_1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Aglicheev Mikhail Sergeevich, IU5-31B");

Console.WriteLine("Quadratic equation: A \* x ^ 4 + B \* x ^ 2 + C = 0");

string coefficient; // Temporary variable for storing input.

double[] coefficients = new double[3];

switch(args.Length)

{

case 0:

string[] coefficientsNames = new string[] { "A", "B", "C" };

for (int i = 0; i < 3; ++i)

{

while (true)

{

Console.Write($"Please input coefficient {coefficientsNames[i]}:");

coefficient = Console.ReadLine();

if (double.TryParse(coefficient, out coefficients[i]))

{

if (i != 0 || coefficients[i] != 0)

{

break;

}

else

{

Console.WriteLine("First coefficient ('A') cannot be 0. Please try again.");

}

}

else

{

Console.WriteLine("That's not a number. Please try again.");

}

}

}

break;

case 3:

for (int i = 0; i < 3; ++i)

{

if (double.TryParse(args[i], out coefficients[i]))

{

if (i == 0 && coefficients[i] == 0)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("First argument cannot be a 0");

Console.ForegroundColor = ConsoleColor.Black;

Console.ReadKey();

return;

}

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("One of arguments is not a number");

Console.ForegroundColor = ConsoleColor.Black;

Console.ReadKey();

return;

}

}

break;

default:

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine($"This function needs 3 arguments. You passed {args.Length}.");

Console.ForegroundColor = ConsoleColor.Black;

Console.ReadKey();

return;

}

var roots = new HashSet<double>();

double innerExpression = Math.Pow(coefficients[1], 2) - 4 \* coefficients[0] \* coefficients[2];

if (innerExpression >= 0)

{

double firstOuterExpression = (-coefficients[1] + Math.Sqrt(innerExpression)) / 2 / coefficients[0];

if (firstOuterExpression >= 0)

{

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

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

double secondOuterExpression = (-coefficients[1] - Math.Sqrt(innerExpression)) / 2 / coefficients[0];

if (secondOuterExpression >= 0)

{

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

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

}

}

}

if (roots.Count == 0)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("No roots have been found");

}

else

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"{roots.Count} Root{(roots.Count > 1 ? "s" : "")} found:");

foreach (double root in roots) {

Console.Write($"{root} ");

}

Console.WriteLine();

}

Console.ForegroundColor = ConsoleColor.Black;

Console.ReadKey();

}

}

}

4. Результаты

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |