|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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)** |

**Отчёт**

***Лабораторная работа №1***

Выполнил студент: \_\_***Сафиуллин Антон Павлович\_\_\_\_\_\_\_\_\_\_***

*фамилия, имя, отчество*

Группа: \_\_\_\_***РТ5-31Б\_\_\_\_\_\_\_\_***

**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.

**Текст Программы**

using System;

using System.IO;

using static System.Math;

namespace LabRab\_1

{

class Program

{

static void cout (string per)

{

Console.WriteLine(per);

}

static double dcin ()

{

int error = 1;

double pev;

while (error == 1)//Вввод переменной

{

try

{

string per = Console.ReadLine();

pev = double.Parse(per);

return pev;

}

catch

{

cout("Введите корректную переменную: ");

}

finally { }

}

return 0;

}

static void Main(string[] args)

{

cout("Разработал: Сафиуллин Антон Павлович РТ5-31Б");

cout("Программа \n" +"Решение уравнения: Ax^4+Bx^2+C=Y: ");

cout("Введите переменную А: ");

double A = dcin();

cout("Введите переменную B: ");

double B = dcin();

cout("Введите переменную C: ");

double C = dcin();

double x1, x2, x3, x4;

double Diskr =(B\*B) - (4\*A\*C); // Вычисление дискриминанта

if (Diskr >= 0) // Если дискриминант больше 0, значит корни - числа R

{

double t1 = -B + Sqrt(Diskr);

double t2 = -B - Sqrt(Diskr);

if (t1>=0 && t2>=0) {

x1 = Sqrt(t1 / (2 \* A));

x2 = -Sqrt(t1 / (2 \* A));

x3 = Sqrt(t2 / (2 \* A));

x4 = -Sqrt(t2 / (2 \* A));

Console.ForegroundColor = ConsoleColor.Red;

cout($"Дискриминант : {Diskr}");

Console.ForegroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.Green;

cout($"Корень 1: {x1}\nКорень 2: {x2}\nКорень 3: {x3}\nКорень 4: {x4}\n ");

Console.ForegroundColor = ConsoleColor.White;

}

// Далее идёт разбор случаев с дискриминантом Если одно из значений или оба значения t меньше нуля

else if (t1 < 0 && t2 >= 0)

{

double kompt1 = -t1;

string x1k = $" {-B / (2 \* A)}+sqrt({kompt1:F4})i /{2 \* A}";

string x2k = $" {-B / (2 \* A)}-sqrt({kompt1:F4})i /{2 \* A}";

x3 = Sqrt(t2 / (2 \* A));

x4 = -Sqrt(t2 / (2 \* A));

Console.ForegroundColor = ConsoleColor.Red;

cout($"Дискриминант: {Diskr}");

Console.ForegroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.Green;

cout($"Корень 1: {x1k}\nКорень 2: {x2k}\nКорень 3: {x3}\nКорень 4: {x4}\n ");

Console.ForegroundColor = ConsoleColor.White;

}

else if (t1 >= 0 && t2 < 0)

{

double kompt1 = -t2;

string x1k = $" {-B / (2 \* A)}+sqrt({kompt1:F4})i /{2 \* A}";

string x2k = $" {-B / (2 \* A)}-sqrt({kompt1:F4})i /{2 \* A}";

x3 = Sqrt(t1 / (2 \* A));

x4 = -Sqrt(t1 / (2 \* A));

Console.ForegroundColor = ConsoleColor.Red;

cout($"Дискриминант: {Diskr}");

Console.ForegroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.Green;

cout($"Корень 1: {x1k}\nКорень 2: {x2k}\nКорень 3: {x3}\nКорень 4: {x4}\n ");

Console.ForegroundColor = ConsoleColor.White;

}

else if (t1<0 && t2<0 )

{

double kompt1 = -t1;

double kompt2 = -t2;

string x1k = $" {-B / (2 \* A)}+sqrt(-{B}+sqrt({kompt1:F4}) )i /{2 \* A}";

string x2k = $" {-B / (2 \* A)}-sqrt({B}+sqrt({kompt1:F4}) )i/{2 \* A}";

string x3k = $" {-B / (2 \* A)}+sqrt(-{B}-sqrt({kompt1:F4}) )i /{2 \* A}";

string x4k = $" {-B / (2 \* A)}-sqrt( {B}-sqrt({kompt1:F4}) )i/{2 \* A}";

Console.ForegroundColor = ConsoleColor.Red;

cout($"Дискриминант: {Diskr}");

Console.ForegroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.Green;

cout($"Корень 1: {x1k}\nКорень 2: {x2k}\nКорень 3: {x3k}\nКорень 4: {x4k}\n ");

Console.ForegroundColor = ConsoleColor.White;

}

}

else if (Diskr<0)// Если Дискриминант меньше 0, значит формируем комплексные корни

{

Console.ForegroundColor = ConsoleColor.Red;

double kompDiskr = - Diskr;

cout($"Дискриминант: {Diskr}");

Console.ForegroundColor = ConsoleColor.White;

string x1k, x2k, x3k, x4k;

x1k =$" {-B/(2\*A)}+sqrt({kompDiskr:F4})i /{2\*A}";

x2k = $" {-1\*-B / (2 \* A)}+sqrt({kompDiskr:F4})i /{2 \* A}";

x3k = $" {-B / (2 \* A)}-sqrt({kompDiskr:F4})i /{2 \* A}";

x4k = $" {-1\*-B / (2 \* A)}-sqrt({kompDiskr:F4})i /{2 \* A}";

Console.ForegroundColor = ConsoleColor.Green;

cout($"Корень 1: {x1k}\nКорень 2: {x2k}\nКорень 3: {x3k}\nКорень 4: {x4k}\n ");

Console.ForegroundColor = ConsoleColor.White;

}

Console.ResetColor();

}

}

}

**Экранные формы с примерами выполнения программы**





