

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

Выполнил:
студент группы ИУ5-35Б:
Тенишев А.А.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.В.
Подпись и дата:

Москва, 2024 г

Описание задания

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент A, B, C введен некорректно (не приводится к действительному числу), то необходимо проигнорировать некорректное значение и ввести коэффициент повторно.
4. Корни уравнения выводятся зеленым цветом. Если корней нет, то сообщение выводится красным цветом.
5. Коэффициенты A, B, C 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;

namespace lab2 {

    class Program {

        static void Main() {
            Console.BackgroundColor = ConsoleColor.Black;
            double A = 0, B = 0, C = 0;

            A = GetDoubleValue("Введите значение A: ");
            B = GetDoubleValue("Введите значение B: ");
            C = GetDoubleValue("Введите значение C: ");

            List<double> Roots = GetRoots(A, B, C);

            Console.WriteLine("Значения введены: A = {0}, B = {1}, C = {2}", A,
B, C);

            if (Roots.Count == 0){
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("Корней не существует");
            }
            else if (Roots.Count == 1){
                Console.Write("Существует один корень: ");
                Console.ForegroundColor = ConsoleColor.Green;
                Console.WriteLine(Roots[0]);
            }

        }

    }

}
```

```

        else if (Roots.Count == 2){
            Console.Write("Существует два корня \nПервый корень: ");
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine(Roots[0]);
            Console.ResetColor();
            Console.Write("Второй корень: ");
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine(Roots[1]);
        }
        else if (Roots.Count == 4){
            Console.Write("Существует четыре корня \nПервый корень: ");
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine(Roots[0]);
            Console.ResetColor();
            Console.Write("Второй корень: ");
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine(Roots[1]);
            Console.ResetColor();
            Console.Write("Третий корень: ");
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine(Roots[2]);
            Console.ResetColor();
            Console.Write("Четвертый корень: ");
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine(Roots[3]);
        }
        Console.ResetColor();
    }

    static double GetDoubleValue(string msg) {
        double value;
        while (true) {
            Console.Write(msg);
            var input = Console.ReadLine();

            if (double.TryParse(input, out value)) {
                return value;
            } else {
                Console.WriteLine("Некорректное значение. Попробуйте еще раз.");
            }
        }
    }

    static List<double> GetRoots(double A, double B, double C){
        List<double> Roots = new List<double>();
        double D = B*B - 4*A*C;
        if (D == 0.0){
            double Root = -B/(2.0*A);
            if (Root == 0){
                Roots.Add(Math.Abs(Root));
            }
        }
    }

```

```

    }
    else if (Root > 0){
        Roots.Add(Math.Sqrt(Root));
        Roots.Add(-Math.Sqrt(Root));
    }
}
else if (D > 0){
    double Sqd = Math.Sqrt(D);
    double Root1 = (-B + Sqd) / (2.0*A);
    double Root2 = (-B - Sqd) / (2.0*A);
    if (Root1 == 0){
        Roots.Add(Math.Abs(Root1));
    }
    else if (Root1 > 0){
        Roots.Add(Math.Sqrt(Root1));
        Roots.Add(-Math.Sqrt(Root1));
    }
    if (Root2 == 0){
        Roots.Add(Math.Abs(Root2));
    }
    else if (Root2 > 0){
        Roots.Add(Math.Sqrt(Root2));
        Roots.Add(-Math.Sqrt(Root2));
    }
}
return Roots;
}
}
}

```

Результаты выполнения.

```
PS C:\Users\ao17\Downloads\dlyagita\dlyagita\lab1> dotnet run
Введите значение A: 1
Введите значение B: -2
Введите значение C: 1
Значения введены: A = 1, B = -2, C = 1
Существует два корня
Первый корень: 1
Второй корень: -1
PS C:\Users\ao17\Downloads\dlyagita\dlyagita\lab1> dotnet run
Введите значение A: 5
Введите значение B: -6
Введите значение C: 3
Значения введены: A = 5, B = -6, C = 3
Корней не существует
PS C:\Users\ao17\Downloads\dlyagita\dlyagita\lab1> dotnet run
Введите значение A: 1
Введите значение B: -12
Введите значение C: 27
Значения введены: A = 1, B = -12, C = 27
Существует четыре корня
Первый корень: 3
Второй корень: -3
Третий корень: 1,7320508075688772
Четвертый корень: -1,7320508075688772
```