



**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)**

**Базовые компоненты интернет технологий
Отчет по лабораторной работе №1**

Студент: Мурзин В.В.
Группа: ИУ5Ц-51Б

Преподаватель: Гапанюк Ю. Е.

2019 г.

Лабораторная работа №1

Разработать программу для решения **биквадратного** уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа должна позволять осуществлять ввод с клавиатуры коэффициентов A , B , C , (уметь реализовывать ввод коэффициентов через параметры командной строки)
3. Программа должна вычислять дискриминант и корни уравнения (в зависимости от дискриминанта). В зависимости от введенных значений коэффициентов A , B , C , в программе должны быть реализованы варианты решений при $A=0$, $B=0$
4. Если хотя бы один из коэффициентов A , B , C введен некорректно, то программа должна сообщить об ошибке (цвет шрифта красный <https://www.videosharp.info/article/sharp/id=513>) и позволить и ввести коэффициент повторно.
5. отображать на экране информацию зеленым цветом (если корни имеются) и красным, если корней нет
6. Выводить в заголовке окна фамилию имя и номер группы

Приложение

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Лабораторная работа №1");

            Console.Title = "Мурзин Вячеслав, ИУ5Ц-51Б";

            //double x1, x2, x3, x4;
            double d, t1, t2, t;

            Console.WriteLine("Поиск корней биквадратного уравнения");
            Console.WriteLine("  $ax^4 + bx^2 + c = 0$ ");

            double a = ReadDouble("Введите коэффициент A: ");
            double b = ReadDouble("Введите коэффициент B: ");
            double c = ReadDouble("Введите коэффициент C: ");
            Console.ReadLine();

            Console.WriteLine("Вы ввели коэффициенты: ");
            Console.WriteLine("A = " + a);
            Console.WriteLine("B = " + b);
            Console.WriteLine("C = " + c);
            Console.WriteLine(a + " $x^4$  + " + b + " $x^2$  + " + c + " = 0");
            Console.ReadLine();

            if (a == 0 && b == 0 && c != 0)
            {
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("Это неравенство!");
                Console.ForegroundColor = ConsoleColor.Gray;
                Console.ReadKey();
            }
            else
            {
                Console.WriteLine("Заменим  $x^2$  на t:  $at^2 + bt + c = 0$  \r\n и найдем корни t1, t2");

                d = (b * b) - (4 * a * c);

                Console.WriteLine("Дискриминант = " + d);
                //Console.WriteLine("Корень из дискриминанта = " + Math.Sqrt(d));

                if (d > 0)
                {
                    Console.ForegroundColor = ConsoleColor.Green;
                    //t1 = Math.Sqrt((-b) - (Math.Sqrt(d))) / (2 * a));
                    t1 = ((-b) - (Math.Sqrt(d))) / (2 * a);
                    Console.WriteLine("t1 = " + t1);
                    //t2 = Math.Sqrt((-b) + (Math.Sqrt(d))) / (2 * a));
                    t2 = ((-b) + (Math.Sqrt(d))) / (2 * a);
                    Console.WriteLine("t2 = " + t2);
                }
            }
        }
    }
}
```

```

Console.ForegroundColor = ConsoleColor.Gray;
Console.ReadKey();

if (t1 >= 0 && t2 >= 0)
{
    Console.WriteLine("Теперь возведем переменные t1 и t2 в корни и
получим: ");

    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine("x1 = " + Math.Sqrt(t1));
    Console.WriteLine("x2 = " + -Math.Sqrt(t1));
    Console.WriteLine("x3 = " + Math.Sqrt(t2));
    Console.WriteLine("x4 = " + -Math.Sqrt(t2));
    Console.ForegroundColor = ConsoleColor.Gray;
    Console.ReadKey();
}
else if (t1 >= 0 && t2 < 0)
{
    Console.WriteLine("Теперь проведем обратную замену положительной
переменной t1: ");

    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine("x1 = " + Math.Sqrt(t1));
    Console.WriteLine("x2 = " + -Math.Sqrt(t1));
    Console.ForegroundColor = ConsoleColor.Red;
    Console.WriteLine("Нет корней от t2!");
    Console.ForegroundColor = ConsoleColor.Gray;
    Console.ReadKey();
}
else if (t1 < 0 && t2 >= 0)
{
    Console.WriteLine("Теперь проведем обратную замену положительной
переменной t2: ");

    Console.ForegroundColor = ConsoleColor.Red;
    Console.WriteLine("Нет корней от t1!");
    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine("x1 = " + Math.Sqrt(t2));
    Console.WriteLine("x2 = " + -Math.Sqrt(t2));
    Console.ForegroundColor = ConsoleColor.Gray;
    Console.ReadKey();
}
else if (t1 < 0 && t2 < 0)
{
    Console.ForegroundColor = ConsoleColor.Red;
    Console.WriteLine("Нет корней от t1 и t2!");
    Console.ForegroundColor = ConsoleColor.Gray;
    Console.ReadKey();
}
}
else if (d == 0)
{
    t = (-b) / (2 * a);
    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine("t = " + t);
    Console.ForegroundColor = ConsoleColor.Gray;
    Console.ReadKey();

    Console.WriteLine("Теперь возведем переменную t в корень и получим:
");

    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine("x1 = " + Math.Sqrt(t));
    Console.WriteLine("x2 = " + -Math.Sqrt(t));
    Console.ForegroundColor = ConsoleColor.Gray;
    Console.ReadKey();
}

```

```

    }
    else if (d < 0)
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("Нет корней!");
        Console.ForegroundColor = ConsoleColor.Gray;
        Console.ReadKey();
    }
}

}

static double ReadDouble(string message) // Ввод вещественного числа с проверкой
корректности ввода
{
    string resultString;
    double resultDouble;
    bool flag;
    do
    {
        Console.Write(message);
        resultString = Console.ReadLine();
        //Первый способ преобразования строки в число
        flag = double.TryParse(resultString, out resultDouble);

        if (!flag)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Необходимо ввести вещественное число");
            Console.ForegroundColor = ConsoleColor.Gray;
            Console.ReadKey();
        }
    }
    while (!flag);
    return resultDouble;
}

}
}

```

AssemblyInfo.cs:

```

using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// Общие сведения об этой сборке предоставляются следующим набором
// набора атрибутов. Измените значения этих атрибутов для изменения сведений,
// связанные с этой сборкой.
[assembly: AssemblyTitle("Lab1")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("HP Inc.")]
[assembly: AssemblyProduct("Lab1")]
[assembly: AssemblyCopyright("Copyright © HP Inc. 2019")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Установка значения False для параметра ComVisible делает типы в этой сборке невидимыми
// для компонентов COM. Если необходимо обратиться к типу в этой сборке через
// из модели COM задайте для атрибута ComVisible этого типа значение true.
[assembly: ComVisible(false)]

// Следующий GUID представляет идентификатор typelib, если этот проект доступен из модели
COM

```

```
[assembly: Guid("cf1a4fba-caeb-42fc-9b71-311c338dd327")]

// Сведения о версии сборки состоят из указанных ниже четырех значений:
//
//      Основной номер версии
//      Дополнительный номер версии
//      Номер сборки
//      Номер редакции
//
// Можно задать все значения или принять номера сборки и редакции по умолчанию
// используя "*", как показано ниже:
// [assembly: AssemblyVersion("1.0.*")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

Ссылка на репозиторий исходных кодов GitHub

<https://github.com/SalyVo/Lab1>