

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М.Є. Жуковського  
“Харківський авіаційний інститут”

Кафедра комп’ютерних систем та мереж

Лабораторна робота № 2

“Разработка и исследование программы решения квадратного уравнения.”

По дисциплине “Технологии программирования”

XAI.503.525B.123. 1705067.180

Виконав студент гр. 525B Пеналоза Г.  
(№ групи) (П.І.Б.)

18-02-19  
(підпис, дата)

Перевірів ст. викладач каф. 503  
(науковий ступінь, вчене звання, посада)

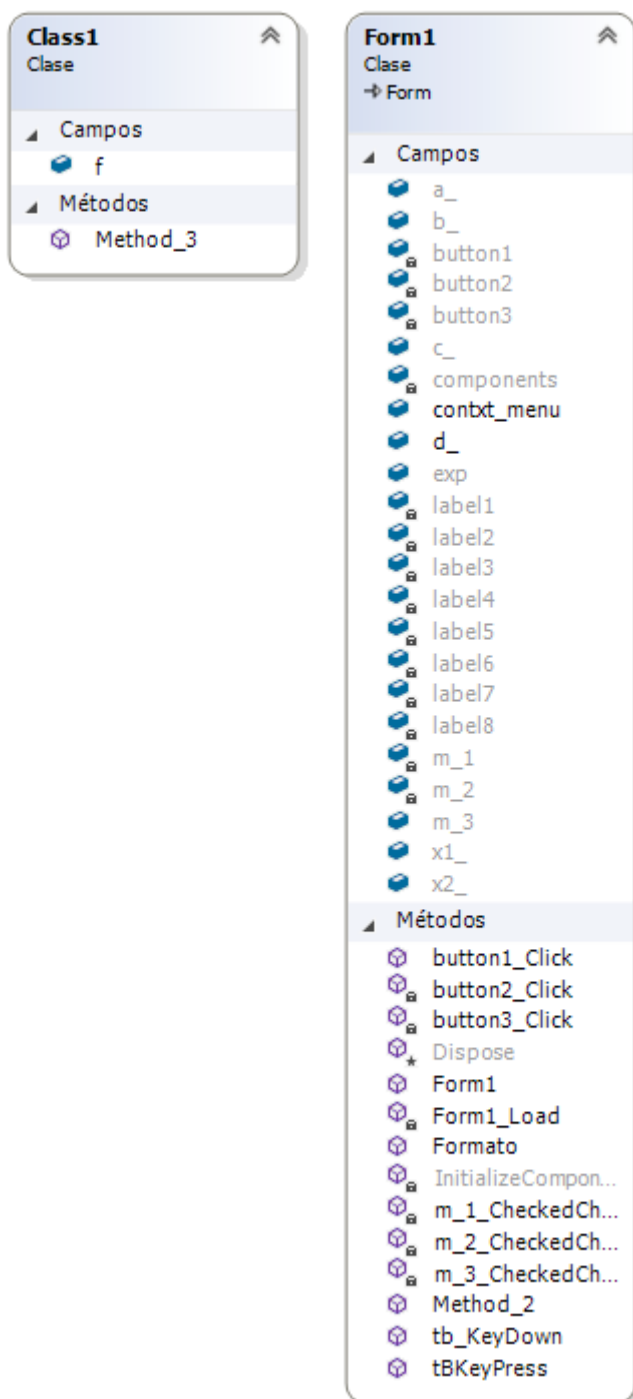
\_\_\_\_\_  
(підпис, дата) (П.І.Б.)

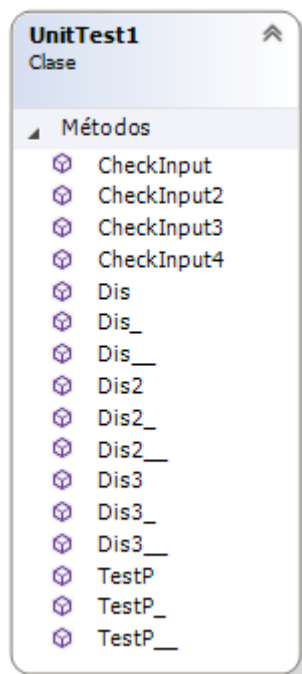
2018

**Тема работы:** “Разработка и исследование линейных программ” (Калькулятор)

**Задание:** разработать на Visual Studio C# проект, реализующий программу поиска действительных корней квадратного уравнения. Запретить ввод в окна редактирования: - символов отличных от (0 1 2 3 4 5 6 7 8 9 + - ,), - двух символов (+ -), - символов (+ и -) не в начале числа.

### Диаграмму классов





## Текст программы

### Form.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab2._2._2
{
    public partial class Form1 : Form
    {
        public ContextMenu ctxt_menu = new ContextMenu(); //Create an empty
context menu object
        public Form1()
    }
}
```

```

{
    InitializeComponent();
}

public string Formato(string n)//Check the formart
{
    Regex reg_dec_nul = new Regex("^(-)?[0-9]+([.][0-9]{1,7})?$");
    string input = n;
    return reg_dec_nul.IsMatch(n) ? "dec" : "nul";
}

public void button1_Click(object sender, EventArgs e)//Clean all the data to
do new operation
{
    a_.Text = "";
    b_.Text = "";
    c_.Text = "";
    x1_.Text = "";
    x2_.Text = "";
    m_1.Checked = false;
    m_2.Checked = false;
    m_3.Checked = false;
}

public void tBKeyPress(object sender, KeyPressEventArgs e)    //Detect
key entered according to the variant.
{
    Thread.CurrentThread.CurrentCulture = new CultureInfo("es-EC"); //Gets
or sets the current culture used by the Resource Manager to look up culture-
specific resources at run time.
    TextBox tb = (TextBox)sender;
    if (!char.IsDigit(e.KeyChar) && e.KeyChar != ',' && e.KeyChar != '-' &&
e.KeyChar != '\b')
    {
        e.KeyChar = char.MinValue;
    }
    else
    {
        if (e.KeyChar == ',')
        {
            if (tb.Text.IndexOf(',') != -1)
            {
                e.KeyChar = char.MinValue;
            }
        }
    }
}

```

```

else if (e.KeyChar == '-')
{
    if (tb.Text.Length == 0)
        return;
    e.KeyChar = char.MinValue;
}
else
{
    if (!char.IsDigit(e.KeyChar))
        return;
    else
    {
        string text = tb.Text;
        int num = text.IndexOf(',');
        if (num == -1 || text.Length - 1 - num <= 6)
            return;
        e.KeyChar = char.MinValue;
    }
}
}
}

```

```

public void tb_KeyDown(object sender, KeyEventArgs e)//Event when press
control + c or control + v to copy and paste
{
    TextBox tb = (TextBox)sender;
    if (e.Control && (e.KeyCode == Keys.V))
    {
        string n = Clipboard.GetText();
        string str = Formato(n);
        if (!(str == "dec"))
        {
            Help.ShowPopup(tb, "incorrect format!", new Point(tb.Right,
tb.Left));
            e.Handled = true;
        }
        else
        {
            if (n.Contains("."))
                n = n.Replace('.', ',');
            tb.Text = n;
            Help.ShowPopup(tb, "text pasted!", new Point(tb.Right, tb.Left));
        }
    }
    else

```

```

{
    if (!e.Control || e.KeyValue != 67)
        return;
    Clipboard.SetText(tb.Text);
    Help.ShowPopup(tb, "text copied!", new Point(tb.Right, tb.Left));
}
}

private void Form1_Load(object sender, EventArgs e)
{
    a_.ContextMenu = this.contxt_menu; //Set an empty context menu object
    b_.ContextMenu = this.contxt_menu; //Set an empty context menu object
    c_.ContextMenu = this.contxt_menu; //Set an empty context menu object
}

private void m_1_CheckedChanged(object sender, EventArgs e)//Checked the
first method and resolve the equation
{
    if (m_1.Checked == true && a_.Text != "" && b_.Text != "" && c_.Text !
= "")
    {
        m_2.Checked = false; m_3.Checked = false;
        x1_.Text = "";
        x2_.Text = "";
        exp.Text = "a . x^2 + b . x + c = 0";
        //Method 1 - посредством реализации алгоритма решения
квадратного уравнения непосредственно в обработчике;
        double a,b,c,x1,x2,d;
        a = Double.Parse(a_.Text);
        b = Double.Parse(b_.Text);
        c = Double.Parse(c_.Text);
        exp.Text = exp.Text.Replace("a", a_.Text);
        exp.Text = exp.Text.Replace("b", b_.Text);
        exp.Text = exp.Text.Replace("c", c_.Text);
        d = b * b - 4 * a * c;
        if (d<0)
        {
            MessageBox.Show("Дискриминант меньше нуля. Корней нет.");
            MessageBox.Show("Результат будет отображаться в комплексных
числах.");
            x1 = ((-b) / (2 * a));
            x1_.Text = x1.ToString("G4");
            x1 = ((-Math.Sqrt(Math.Abs(d))) / (2 * a));
            x1_.Text += x1.ToString("G4")+" * i";

```

```

        x2 = ((-b) / (2 * a));
        x2_.Text = x2.ToString("G4");
        x2 = ((Math.Sqrt(Math.Abs(d))) / (2 * a));
        x2_.Text += "+" + x2.ToString("G4") + " * i";
    }
    if (d == 0)
    {
        x1 = (-b / (2 * a));
        x1_.Text = x1.ToString("G7");
        //MessageBox.Show("Дискриминант равен нулю. Корень равен " +
x1 + ".");
    }
    if (d > 0)
    {
        x1 = ((-b - Math.Sqrt(d)) / (2 * a));
        x2 = ((-b + Math.Sqrt(d)) / (2 * a));
        x1_.Text = x1.ToString("G7");
        x2_.Text = x2.ToString("G7");
        //MessageBox.Show("Дискриминант равен " + d + ". Первый
корень равен " + x1 + ". Второй корень равен " + x2 + ".");
    }
}
else
{
    MessageBox.Show("Введите все переменные!");
    m_1.Checked = false;
}
}

```

```

private void m_2_CheckedChanged(object sender, EventArgs e) //Checked the
second method
{
    if (m_2.Checked == true && a_.Text != "" && b_.Text != "" && c_.Text !=
"")
    { m_1.Checked = false; m_3.Checked = false; x1_.Text = ""; x2_.Text =
""; exp.Text = "a . x^2 + b . x + c = 0"; }
    else
    {
        MessageBox.Show("Введите все переменные!");
        m_2.Checked = false;
    }
}

```

private void m\_3\_CheckedChanged(object sender, EventArgs e)//Checked the third method

```
{
    if (m_3.Checked == true && a_.Text != "" && b_.Text != "" && c_.Text != "")
    { m_1.Checked = false; m_2.Checked = false; x1_.Text = ""; x2_.Text = ""; exp.Text = "a . x^2 + b . x + c = 0"; }
    else
    {
        m_3.Checked = false;
        MessageBox.Show("Введите все переменные!");
    }
}
```

private void button3\_Click(object sender, EventArgs e)//Close the application

```
{
    this.Close();
}
```

public double d\_;

public void Method\_2()//Method 2 - посредством реализации алгоритма решения уравнения в методе класса формы;

```
{
    x1_.Text = "";
    x2_.Text = "";
    exp.Text = "a . x^2 + b . x + c = 0";
    //Method 1
    double a, b, c, x1, x2;
    a = Double.Parse(a_.Text);
    b = Double.Parse(b_.Text);
    c = Double.Parse(c_.Text);
    exp.Text = exp.Text.Replace("a", a_.Text);
    exp.Text = exp.Text.Replace("b", b_.Text);
    exp.Text = exp.Text.Replace("c", c_.Text);
    d_ = b * b - 4 * a * c;
    if (d_ < 0)
    {
        MessageBox.Show("Дискриминант меньше нуля. Корней нет.");
        MessageBox.Show("Результат будет отображаться в комплексных числах.");
        x1 = ((-b) / (2 * a));
        x1_.Text = x1.ToString("G4");
        x1 = ((-Math.Sqrt(Math.Abs(d_))) / (2 * a));
        x1_.Text += x1.ToString("G4") + " * i";
    }
}
```



```

        x2 = ((-b) / (2 * a));
        x2_.Text = x2.ToString("G4");
        x2 = ((Math.Sqrt(Math.Abs(d_)) / (2 * a));
        x2_.Text += "+" + x2.ToString("G4") + " * i";
    }
    if (d_ == 0)
    {
        x1 = (-b / (2 * a));
        x1_.Text = x1.ToString("G7");
        //MessageBox.Show("Дискриминант равен нулю. Корень равен " +
x1 + ".");
    }
    if (d_ > 0)
    {
        x1 = ((-b - Math.Sqrt(d_)) / (2 * a));
        x2 = ((-b + Math.Sqrt(d_)) / (2 * a));
        x1_.Text = x1.ToString("G7");
        x2_.Text = x2.ToString("G7");
        //MessageBox.Show("Дискриминант равен " + d + ". Первый корень
равен " + x1 + ". Второй корень равен " + x2 + ".");
    }
}

private void button2_Click(object sender, EventArgs e)//Identify methods
{
    if (m_2.Checked==true)
    {
        Method_2();
    }
    if (m_3.Checked==true)
    {
        Class1 class1 = new Class1
        {
            f = this
        };
        class1.Method_3();
    }
}
}
}

```

**Class1.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab2._2._2
{
    public class Class1
    {
        public Form1 f = new Form1();
        public void Method_3()//Method 3 - посредством реализации алгоритма
        решения уравнения в методе отдельного класса;
        {

            f.x1_.Text = "";
            f.x2_.Text = "";
            f.exp.Text = "a . x^2 + b . x + c = 0";
            //Method 1
            double a, b, c, x1, x2, d;
            a = Double.Parse(f.a_.Text);
            b = Double.Parse(f.b_.Text);
            c = Double.Parse(f.c_.Text);
            //Replace values in the expresion
            f.exp.Text = f.exp.Text.Replace("a", f.a_.Text);
            f.exp.Text = f.exp.Text.Replace("b", f.b_.Text);
            f.exp.Text = f.exp.Text.Replace("c", f.c_.Text);
            d = b * b - 4 * a * c; //Discriminate
            if (d < 0)//Complex roots
            {
                MessageBox.Show("Дискриминант меньше нуля. Корней нет.");
                MessageBox.Show("Результат будет отображаться в комплексных
числах.");
                x1 = ((-b) / (2 * a));
                f.x1_.Text = x1.ToString("G4");
                x1 = ((-Math.Sqrt(Math.Abs(d))) / (2 * a));
                f.x1_.Text += x1.ToString("G4") + " * i";

                x2 = ((-b) / (2 * a));
                f.x2_.Text = x2.ToString("G4");
                x2 = ((Math.Sqrt(Math.Abs(d))) / (2 * a));
            }
        }
    }
}

```

```

        f.x2_.Text += "+" + x2.ToString("G4") + " * i";
    }
    if (d == 0)//One root
    {
        x1 = (-b / (2 * a));
        f.x1_.Text = x1.ToString("G7");
        //MessageBox.Show("Дискриминант равен нулю. Корень равен " +
x1 + ".");
    }
    if (d > 0)//Two roots
    {
        x1 = ((-b - Math.Sqrt(d)) / (2 * a));
        x2 = ((-b + Math.Sqrt(d)) / (2 * a));
        f.x1_.Text = x1.ToString("G7");
        f.x2_.Text = x2.ToString("G7");
        //MessageBox.Show("Дискриминант равен " + d + ". Первый корень
равен " + x1 + ". Второй корень равен " + x2 + ".");
    }
}
}
}
}

```

### Unit-тестов.

```

using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Lab2._2._2;
using System.Windows.Forms;

namespace PruebasUnitarias
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void CheckInput()//In this test fail because it does not read the char has
sent to textbox
        {
            var dig = '-';
            var valespect = "-";
            Form1 main = new Form1();
            KeyPressEventArgs keyPress = new KeyPressEventArgs(dig);
            main.tBKeyPress(main.a_,keyPress);
            Assert.AreEqual(valespect,main.a_.Text);
        }
    }
}

```

```

}
//Verify different of numbers if they are correct.
[TestMethod]
public void CheckInput2()
{
    var dig = "34";
    var valespect = "dec";
    Form1 main = new Form1();
    Assert.AreEqual(valespect, main.Formato(dig));
}
[TestMethod]
public void CheckInput3()
{
    var dig = "-34,d";
    var valespect = "nul";
    Form1 main = new Form1();
    Assert.AreEqual(valespect, main.Formato(dig));
}
[TestMethod]
public void CheckInput4()
{
    var dig = "--45.3";
    var valespect = "nul";
    Form1 main = new Form1();
    Assert.AreEqual(valespect, main.Formato(dig));
}
//Verify discriminante
[TestMethod]
public void Dis()
{
    var a = 1;
    var b = 2;
    var c = 1;
    //double disc = 2 * (b) - 4 * a * c;
    double valespect = 0;
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis_()
{

```

```

var a = 2;
var b = 4;
var c = 2;
//double disc = 2 * (b) - 4 * a * c;
double valespect = 0;
Form1 main = new Form1();
main.a_.Text = a.ToString();
main.b_.Text = b.ToString();
main.c_.Text = c.ToString();
main.Method_2();
Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis__()
{
    var a = 4;
    var b = 8;
    var c = 4;
    //double disc = 2 * (b) - 4 * a * c;
    double valespect = 0;
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis2__()
{
    var a = 3;
    var b = 1;
    var c = 1;
    //double disc = 2 * (b) - 4 * a * c;
    double valespect = -11;
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis2__()
{

```

```

var a = 5;
var b = 1;
var c = 1;
//double disc = 2 * (b) - 4 * a * c;
double valespect = -19;
Form1 main = new Form1();
main.a_.Text = a.ToString();
main.b_.Text = b.ToString();
main.c_.Text = c.ToString();
main.Method_2();
Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis2()
{
    var a = 10;
    var b = 2;
    var c = 6;
    //double disc = 2 * (b) - 4 * a * c;
    double valespect = -236;
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis3()
{
    var a = 3;
    var b = 6;
    var c = 1;
    //double disc = 2 * (b) - 4 * a * c;
    double valespect = 24;
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis3_()
{

```

```

var a = 1;
var b = 4;
var c = 3;
//double disc = 2 * (b) - 4 * a * c;
double valespect = 4;
Form1 main = new Form1();
main.a_.Text = a.ToString();
main.b_.Text = b.ToString();
main.c_.Text = c.ToString();
main.Method_2();
Assert.AreEqual(valespect, main.d_);
}
[TestMethod]
public void Dis3__()
{
    var a = 2;
    var b = 7;
    var c = 3;
    //double disc = 2 * (b) - 4 * a * c;
    double valespect = 25;
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.d_);
}
//Verify test - results x1 and x2
[TestMethod]
public void TestP()
{
    var a = 1;
    var b = 2;
    var c = 1;
    //double disc = 2 * (b) - 4 * a * c;
    var valespect = "-1";
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.x1_.Text);
}
[TestMethod]
public void TestP_()

```

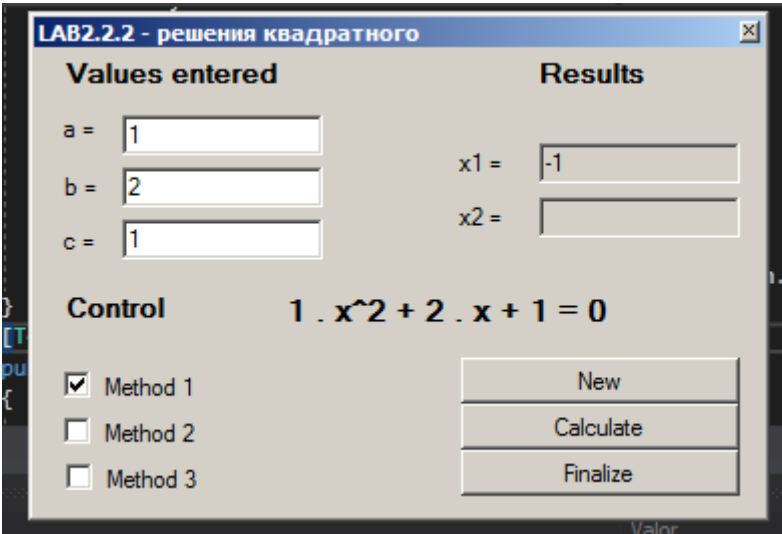
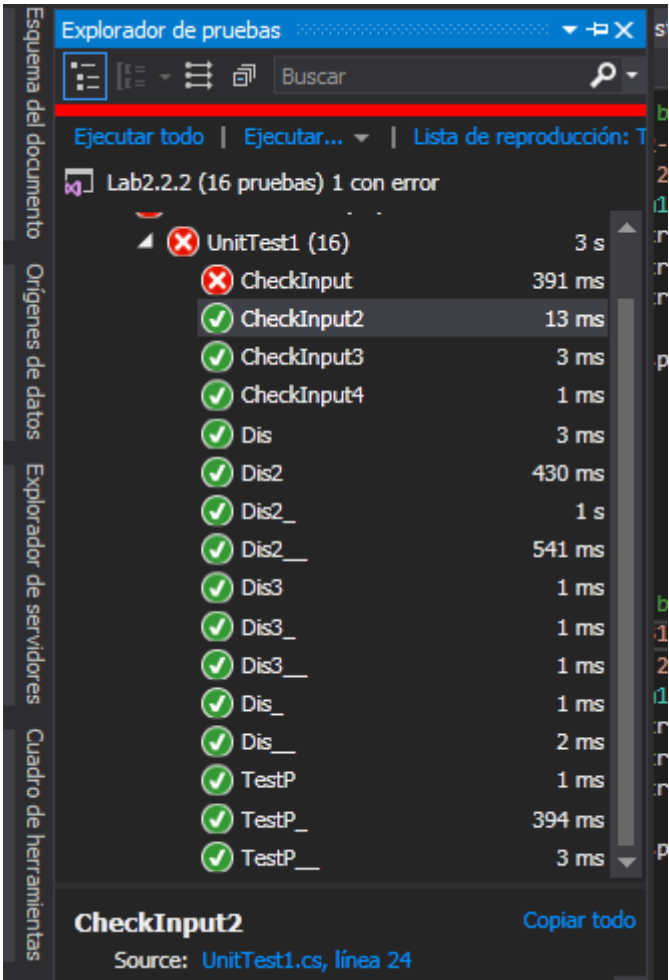
```

{
    var a = 5;
    var b = 2;
    var c = 1;
    //double disc = 2 * (b) - 4 * a * c;
    var valespect = "-0,2-0,4 * i";
    var valespect_ = "-0,2+0,4 * i";
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.x1_.Text,valespect_,main.x2_.Text);
}
[TestMethod]
public void TestP__()
{
    var a = 5;
    var b = 43;
    var c = 12;
    //double disc = 2 * (b) - 4 * a * c;
    var valespect = "-8,311234";
    var valespect_ = "-0,2887658";
    Form1 main = new Form1();
    main.a_.Text = a.ToString();
    main.b_.Text = b.ToString();
    main.c_.Text = c.ToString();
    main.Method_2();
    Assert.AreEqual(valespect, main.x1_.Text, valespect_, main.x2_.Text);
}
}
}

```



Скриншоты



**LAB2.2.2 - решения квадратного**

Values entered	Results
a = <input type="text" value="1"/>	x1 = <input type="text" value="-1-2,236 * i"/>
b = <input type="text" value="2"/>	x2 = <input type="text" value="-1+2,236 * i"/>
c = <input type="text" value="6"/>	

**Control**       $1 \cdot x^2 + 2 \cdot x + 6 = 0$

☐ Method 1  
☒ Method 2  
☐ Method 3

New  
 Calculate  
 Finalize

**LAB2.2.2 - решения квадратного**

Values entered	Results
a = <input type="text" value="1"/>	x1 = <input type="text" value="-42,86001"/>
b = <input type="text" value="43"/>	x2 = <input type="text" value="-0,1399906"/>
c = <input type="text" value="6"/>	

**Control**       $1 \cdot x^2 + 43 \cdot x + 6 = 0$

☐ Method 1  
☐ Method 2  
☒ Method 3

New  
 Calculate  
 Finalize

Valor

**Таблица 1 - Поля и методы класса А и их назначение (Пример)**

<b>№</b>	<b>Поле</b>	<b>Назначение</b>
1	a_	Variable 1
2	b_	Variable 2
3	b_2	--
4	button1	Clean fields
5	button2	Calcule the roots
6	button3	Close the app
7	c_	Variable 3
8	context_menu	Contex menu
9	d_	Discriminate
10	exp	Label with a expresion
11	label1	Titles
12	label2	Titles
13	label3	Titles
14	label4	Titles
15	label5	Titles
16	label6	Titles
17	label7	Titles
18	label8	Titles
19	m_1	Checkbox 1 (way)
20	m_2	Checkbox 2 (way)
21	m_3	Checkbox 3 (way)
22	x1_	Root 1
23	x2_	Root 2
24	f	Instance of form in the other file
25	<b>Метод</b>	<b>Назначение</b>
26	Method_2	Resolve the equation from a method of the form file
27	Formato	Indentify the correct format of the numbers entered.
28	Method_3	Resolve the equation from a

		method of other file (Class1.cs)
--	--	----------------------------------

**Таблица 2 - Обработчики событий проекта и их назначение  
(Пример)**

№	Обработчик события	Назначение
1	button1_Click	Clean
2	tBKeyPress	dentify the correct digits
3	tb_KeyDown	Copy and Paste with key combination
4	m_1_CheckedChanged	Identify the checkbox 1
5	m_2_CheckedChanged	Identify the checkbox 2
6	m_3_CheckedChanged	Identify the checkbox 3
7	button3_Click	Close
8	Form1_Load	Set null context menu to textbox
9	button2_Click	Calculate the roots

### **Выводы**

Однажды лабораторная практика была закончена. Знания и практика с регулярными выражениями и вводом параметров в текстовом поле были усилены. Кроме того, для проверки данных использовались юнит-тесты. На практике я не смог получить успешный тест, который подтвердил конкретную цифру. Поэтому я приступил к проверке введенных данных.

Улучшения для реализации:

- \* Уравнения высшего уровня.
- \* Получить график уравнения и его корней.

### **Использованные источники**

1. <https://docs.microsoft.com/es-es/dotnet/>