Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatică şi Microelectronică

Raport

la disciplina:
MIDPS

Lucrarea de laborator Nr.3

Tema: GUI Development

A efectuat: st.gr.Ti-141 Bulgac Ion

A verificat lect. univ. Cojanu Irina

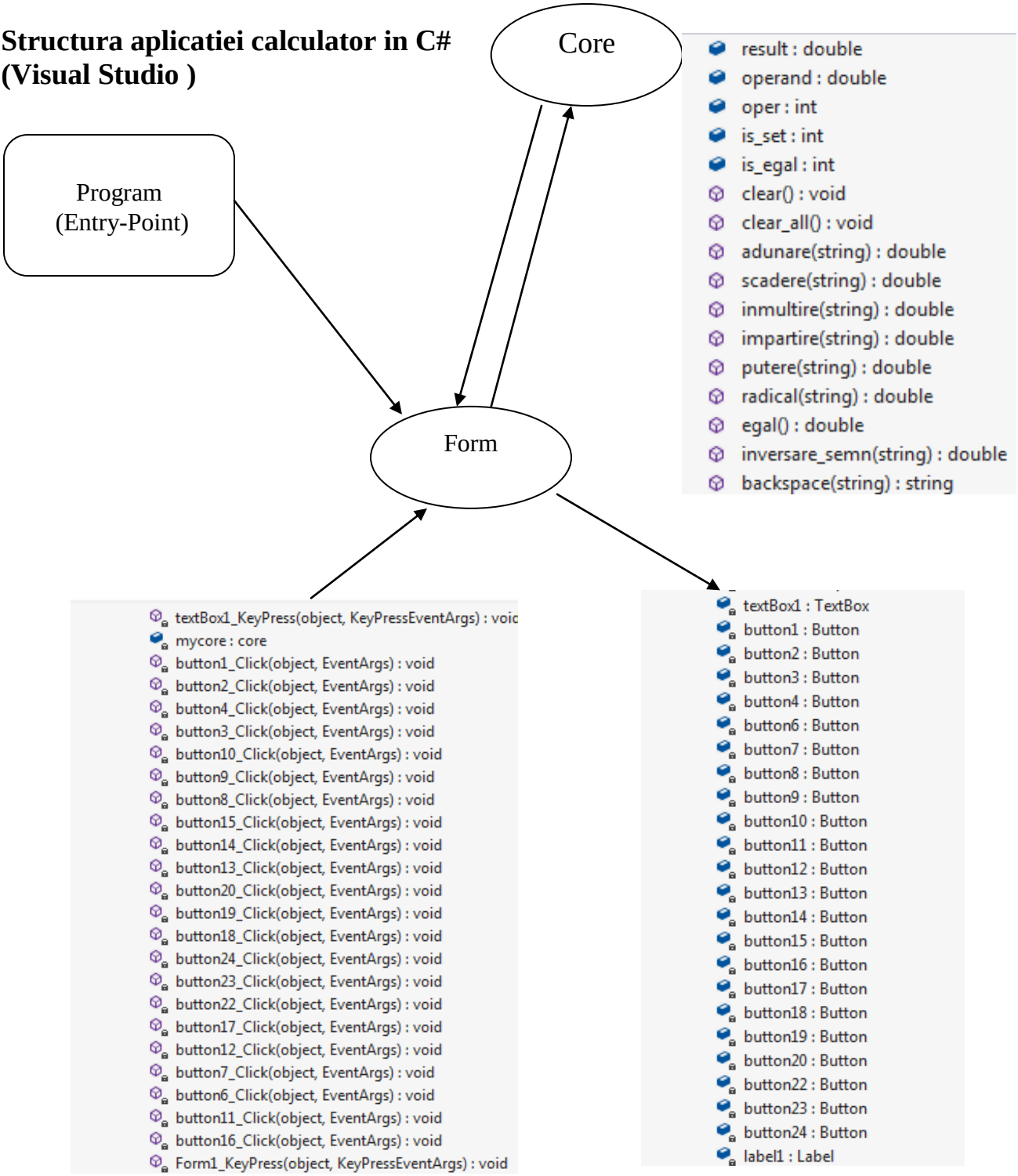
Chişinău 2016

Scopul lucrării: Crearea aplicatiilor cu interfata utilizator

Obiectivele lucrării:

- Realizeaza un simplu GUI Calculator
- Operatiile simple: +,-,*,/,putere,radical,InversareSemn(+/-),operatii cu numere zecimale.
- Divizare proiectului in doua module - Interfata grafica(Modul GUI) si Modulul de baza(Core Module).

Structura aplicatiei calculator in C# (Visual Studio)



Codul sursa

Program.cs (contine clasa Program(entry-pointul) si clasa Core(logica programului))

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

namespace WindowsFormsApplication1
{

    public class core
    {


        public double result = 0;
        public double operand = 0;

        public int oper = 0;
        public int is_set = 0;
        public int is_egal = 0;

        public void clear() {
            operand = 0;
        }
        public void clear_all() {
            result = 0;
            operand = 0;
            oper = 0;

        }

        public double adunare(string val) {
            is_set = 0;
            double number = double.Parse(val);


            if (result == 0 || is_egal == 1){
                result = number;
            }
            else {
                operand = number;
                egal();
            }

            oper = 1;
            is_egal = 0;
            return result;
        }

        public double scadere(string val){
            is_set = 0;
            double number = double.Parse(val);


            if (result == 0 || is_egal == 1)
            {
                result = number;
            }
            else
            {
                operand = number;
```

```
            egal();
        }

        oper = 2;
        is_egal = 0;
        return result;
    }


    public double inmultire(string val){
        is_set = 0;
        double number = double.Parse(val);


        if (result == 0 || is_egal == 1)
        {
            result = number;
        }
        else
        {
            operand = number;
            egal();
        }

        oper = 3;
        is_egal = 0;
        return result;
    }

    public double impartire(string val){
        is_set = 0;
        double number = double.Parse(val);


        if (result == 0 || is_egal == 1)
        {
            result = number;
        }
        else
        {
            operand = number;
            egal();
        }

        oper = 4;
        is_egal = 0;
        return result;
    }


    public double putere(string val){
        is_set = 0;
        double number = double.Parse(val);


        if (result == 0 || is_egal == 1)
        {
            result = number;
        }
        else
        {
            operand = number;
            egal();
        }

        oper = 5;
```

```
    is_egal = 0;
    return result;
}

public double radical(string val){
    double number = double.Parse(val);

    oper = 6;
    result = Math.Sqrt(number);

    return result;
}




public double egal()
{

    switch (oper)
    {
        case 0:
            break;
        case 1:
            result = result + operand;
            break;
        case 2:
            result = result - operand;
            break;
        case 3:
            result = result * operand;
            break;
        case 4:
            result = result / operand;
            break;
        case 5:
            result = Math.Pow(result,operand);
            break;
        case 6:
            result = Math.Sqrt(result);
            break;

    }

    is_egal = 1;
    return result;
}


public double inversare_semn(string  val)
{

    double number = double.Parse(val);
    number = number * -1;

    return number;
}

public string backspace(string val) {


    string res;
    int number = val.Count();


        if (number != 0)
```

```
                {
                    res = val.Substring(0, number - 1);
                }
                else
                {
                    res = " ";
                }

            return res;
        }


    }


    public class Program
    {

        [STAThread]
        public static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());

        }

    }

}
```

Form1.cs (Definirea elementelor grafice si prelucrarea evenimentelor)

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

namespace WindowsFormsApplication1
{

    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&
               (e.KeyChar != '.'))
            {
                e.Handled = true;
            }

            // only allow one decimal point
            if ((e.KeyChar == '.') && ((sender as TextBox).Text.IndexOf('.') > -1))
```

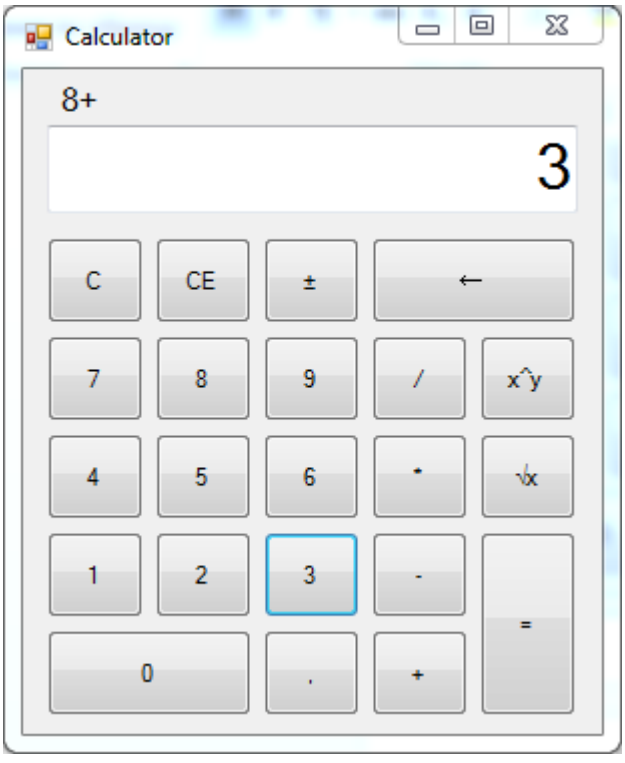
```
    {
        e.Handled = true;
    }
}

core mycore = new core();

// C
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    label1.Text = "";

    mycore.clear();

}

// CE
private void button2_Click(object sender, EventArgs e)
{
    textBox1.Text = "";

    mycore.clear_all();
}

// +-
private void button4_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        double val = mycore.inversare_semn(textBox1.Text);
        textBox1.Text = " " + val;
    }
}

// backspace
private void button3_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        string val = mycore.backspace(textBox1.Text);
        textBox1.Text = val;
    }

}

// 7
private void button10_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 7;
}

//8
private void button9_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 8;
}

//9
private void button8_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 9;
}

//4
private void button15_Click(object sender, EventArgs e)
```

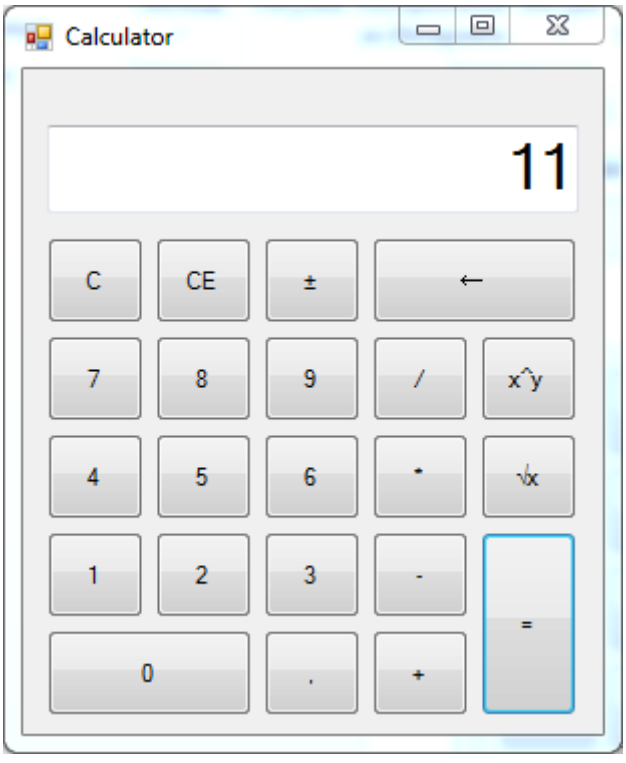
```
{
    textBox1.Text = textBox1.Text + 4;
}

//5
private void button14_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 5;
}

//6
private void button13_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 6;
}

//1
private void button20_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 1;
}

//2
private void button19_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 2;
}

//3
private void button18_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 3;
}

//0
private void button24_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + 0;
}

// ,
private void button23_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + ",";
}


// +
private void button22_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        double val = mycore.adunare(textBox1.Text);
        label1.Text = "" + val + "+";
        //label1.Text = label1.Text + textBox1.Text + "+";
        textBox1.Text =" ";
    }
}

//-
private void button17_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        double val = mycore.scadere(textBox1.Text);
        label1.Text = "" + val + "-";
```

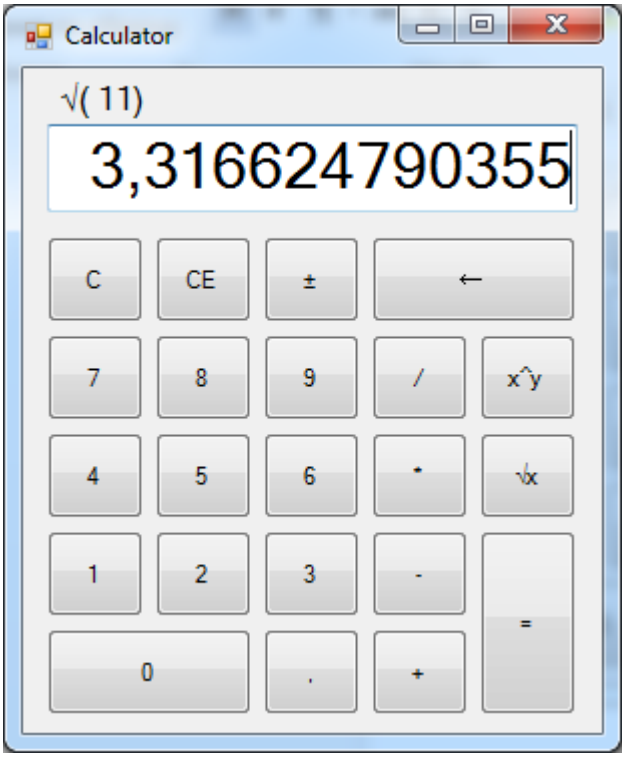
```
        textBox1.Text = " ";
    }
}

//*
private void button12_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        double val = mycore.inmultire(textBox1.Text);
        label1.Text = "" + val + "*";
        textBox1.Text = " ";
    }
}

// /
private void button7_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        double val = mycore.impartire(textBox1.Text);
        label1.Text = "" + val + "/";
        textBox1.Text = " ";
    }
}

// pow
private void button6_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        double val = mycore.putere(textBox1.Text);
        label1.Text = "" + val + "^";
        textBox1.Text = " ";
    }
}

// sqrt
private void button11_Click(object sender, EventArgs e)
{
    if (textBox1.Text != " ")
    {
        double val = mycore.radical(textBox1.Text);
        label1.Text = "√(" + textBox1.Text + ")";
        textBox1.Text = " " + val;
    }
}

// egal
private void button16_Click(object sender, EventArgs e)
{
    double val;

    if (textBox1.Text != " ")
    {
        if (mycore.is_set==0) {
            mycore.operand = double.Parse(textBox1.Text);
            mycore.is_set = 1;
        }

        val = mycore.egal();

        label1.Text = " ";
        textBox1.Text = " " + val;
    }
```

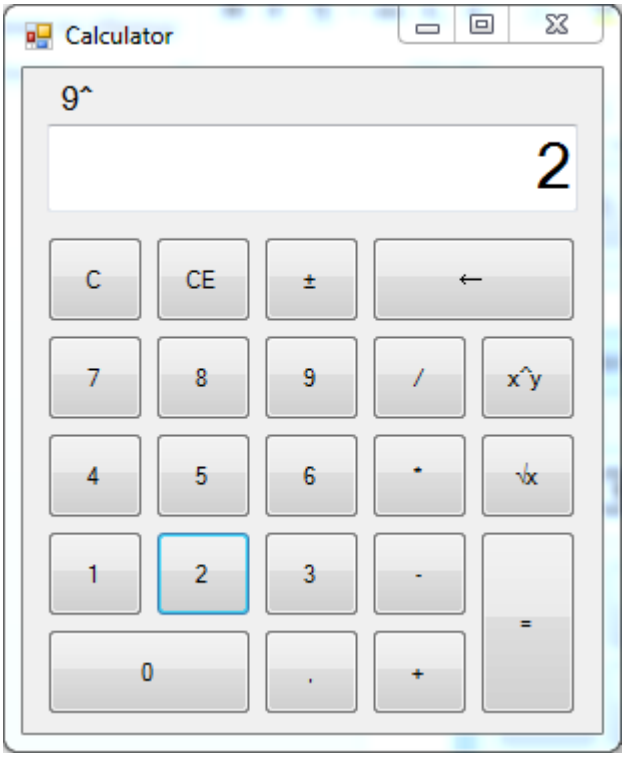
```
        }

        // ------------------------------------------------------------------------------------
        // keyboard events
        // ------------------------------------------------------------------------------------

        private void Form1_KeyPress(object sender, KeyPressEventArgs e)
        {

            switch ((int)e.KeyChar) {
                case 108:
                    button16.PerformClick();
                    break;

                case 107:
                    button22.PerformClick();
                    break;

                case 105:
                    button17.PerformClick();
                    break;

                case 100:
                    button12.PerformClick();
                    break;

                case 95:
                    button7.PerformClick();
                    break;
            }

        }

    }

}
```

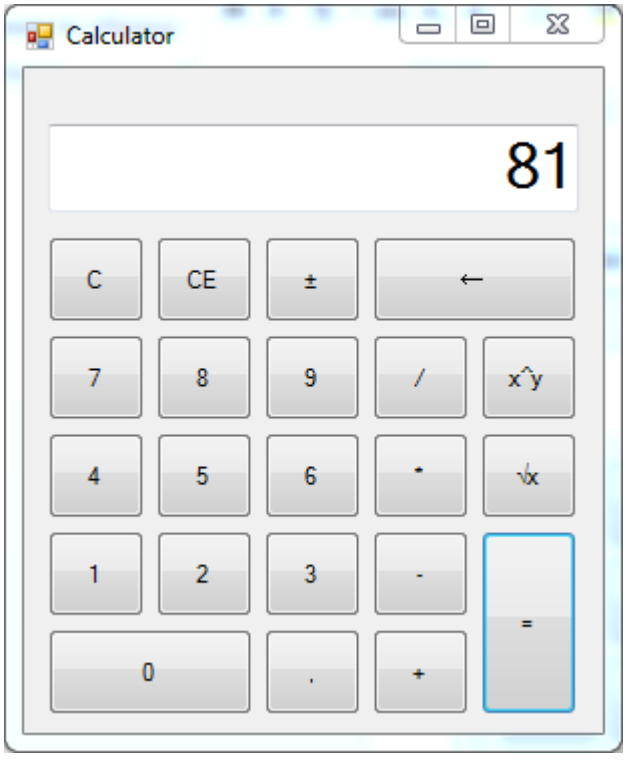
Executia programului






Calculator
√(11)
3,316624790355
C
CE
±
←
7
8
9
/
x^y
4
5
6
*
√x
1
2
3
-
=
0
.
+


Calculator
9^
2
C
CE
±
←
7
8
9
/
x^y
4
5
6
*
√x
1
2
3
-
=
0
.
+


Calculator
81
C
CE
±
←
7
8
9
/
x^y
4
5
6
*
√x
1
2
3
-
=
0
.
+

Concluzii

In urma efectuarii lucrarii de laborator Nr.3 la MIDPS am obtinut capacitati practice de creare a aplicatiilor grafice in mediul de dezvoltare Visual Studio . De asemenea am utilizat conceptele si proprietatile un nou limbaj de programare , C# . In urma acestei lucrari de laborator am obtinut cunostinte practice asupra crearii unei aplicatii grafice in Visual Studio pentru sistemul Windows , si am deprins lucrul cu elemente grafice si implementarea evenimentelor.