Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator#1

Realizarea unui simplu GUI Calculator

Autor:
Daniela Cazac

lector asistent:
Irina Cojanu
lector superior:
Radu Melnic

Lucrarea de laborator #2

1 Scopul lucrarii de laborator :

Realizeaza un simplu GUI Calculator care suporta operatiile simple de +, -, *, /, putere, radical, InversareSemn(+/-), operatii cu numere zecimale.

2 Obiective si Conditii Necesare :

Familiarizarea cu un nou limbaj de programare, si folosirea unui nou IDE, pentru dezvoltarea cunostintelor noastre in limbaje si medii interactive de dezvoltare a programelor.

3 Mersul lucrarii :

In aceasta lucrare am elaborat un Calculator , care e friendly pentru utilizator.

Note si explicatii !

3.1 Partea Grafica

Cind am creat fereastra, implicit are o denumire pe care o vom modifica corespunzator in Calculator din menu-l **Properties**, modificam cimpul **Text**.

Pentru a nu schimba dimensiunea ferestrei , si a avea dimensiunea stabilita de noi vom dezactiva optiunea de **Maximize** si **Restore Down** din menu-l **Properties**, modificam valoarea cimpului MaximizeBox din True in False , si cimpul **FormBorderStyle** il modificam din **Sizable** in **FixedSingle**.

Pentru a adauga butoane pe forma , din **ToolBox** selectam **Button, TextBox** si **Label** ca componente vizuale caruia din **Properties** ii putem modifica orice proprietate,cum ar fi : marimea, font-ul, numele, etc. Analog adaugam un TextBox pentru afisarea rezultatului.

Pentru fiecare buton am scris codul, adica functionalitatea acestuia.Astfel,in continuare, voi explica momentele principale si importante de stiut.

3.2 Modulul de baza

Butonul CE (Clear Entry) / C (Clear all)
Functionalitatea acestuia consta in faptul ca cifra/cifrele introduse se sterg si "se transforma" in 0:
txtDisplay.Text = "0";
Adica la tastarea butonului **CE** , proprietatea Text a componentei vizuale TextBox, denumita de mine txtDisplay va lua valoarea 0, exact acelasi rezultat va fi si la tastarea butonului **C**.

```
1 reference
private void button4_Click(object sender, EventArgs e)
{
    txtDisplay.Text = "0";
    lblShowOp.Text = "";

}

1 reference
private void button3_Click(object sender, EventArgs e)
{
    txtDisplay.Text = "0";
    lblShowOp.Text = "";
}
```

Butonul Backspace

Functionalitatea acestuia consta in faptul ca la tastarea lui, ultima cifra introdusa va fi stearsa. Pentru asta am folosit 2 cicluri : primul if verifica lungimea stringului (pentru ca propietatea **Text** a TextBox-ului este de tip **String**) este mai mare ca 0 (adica putem ceva sterge), atunci din lungimea sirului de cifre, se sterge ultimul caracter :

txtDisplay.Text = txtDisplay.Text.Remove(txtDisplay.Text.Length - 1);

Si al doilea if, verifica daca proprietatea Text a TextBox-ului are un string null, atunci la tastarea butonului Backspace, valoarea va fi setata cu 0.

```
1 reference
private void btnSpace_Click_Click(object sender, EventArgs e)
{
    if(txtDisplay.Text .Length>0)
    {
        txtDisplay.Text = txtDisplay.Text.Remove(txtDisplay.Text.Length - 1, 1);
    }

    if (txtDisplay.Text == "")
    {
        txtDisplay.Text = "0";
    }
}
```

Operatorii aritmetici

Cind vom tasta pe unul din operatorii + , - , / , * atunci se va apela metoda ce urmeaza sa o descriu.Deci, am creat o variabila de tip Button in care se stocheaza butonul ce vine. Orice operator tastat el vine in aceasta functie, iar variabila **num** preia butonul cu toate caracteristicile lui,iar variabila operation preia semnul prin **num.Text;**

```
4 references
private void Arithmetic_Operator(object sender, EventArgs e)
{
    Button num = (Button)sender;
    operation = num.Text;
    results = Double.Parse(txtDisplay.Text);
    txtDisplay.Text = "";
    lblShowOp.Text = System.Convert.ToString(results) + " " + operation;
}
```

Butonul Egal

In aceasta metoda,am folosit o instructiune de selectie Switch, care in dependenta de operatorul ales, el va alege cazul corespunzator. Voi explica codul pentru un operator, pentru ca analogic este codul si pentru ceilalti operatori:

case "+":

txtDisplay.Text = (results + Double.Parse(txtDisplay.Text)) .ToString();

break;

Continutul proprietatii Text ai componentei vizuale txtDisplay (TextBox), va fi parsat in tipul de date Double, pentru a putea si adaugat cu valoarea lui results , care a fost initializata cu 0. Am obtinut un rezultat concret, un numar. Acest numar , cu ajutorul metodei ToString este convertita intr-un String, pentru ca proprietatea Text a Textbox-ului asteapta un String, daca primeste alt tip de date, atunci vom avea eroare. Respectiv se face break - care este iesirea conditionata din instructiune, celelalte case-uri fiind ignorate.

```
1 reference
private void btnEquals_Click_Click(object sender, EventArgs e)
{

    switch(operation)
    {
        case "+":
            txtDisplay.Text = (results + Double.Parse(txtDisplay.Text)).ToString();
            break;
        case "-":
            txtDisplay.Text = (results - Double.Parse(txtDisplay.Text)).ToString();
            break;
        case "*":
            txtDisplay.Text = (results * Double.Parse(txtDisplay.Text)).ToString();
            break;
        case "/":
            txtDisplay.Text = (results / Double.Parse(txtDisplay.Text)).ToString();
            break;

    }
}
```

Butonul Sqrt

Am declarat o variabila de tip Double, careia i-am atribuit valoarea preluata din txtDisplay si parsata in Double, respectiv acestei variabile locale, i-am atribuit rezultatul unei functii , si anume Math.Sqrt(), care primeste ca parametru insasi valoarea preluata din txtDisplay. Respectiv rezultatul final, va fi parsat cu ToString si atribuit proprietatii text:
txtDisplay.Text = System.Convert.ToString(sq);

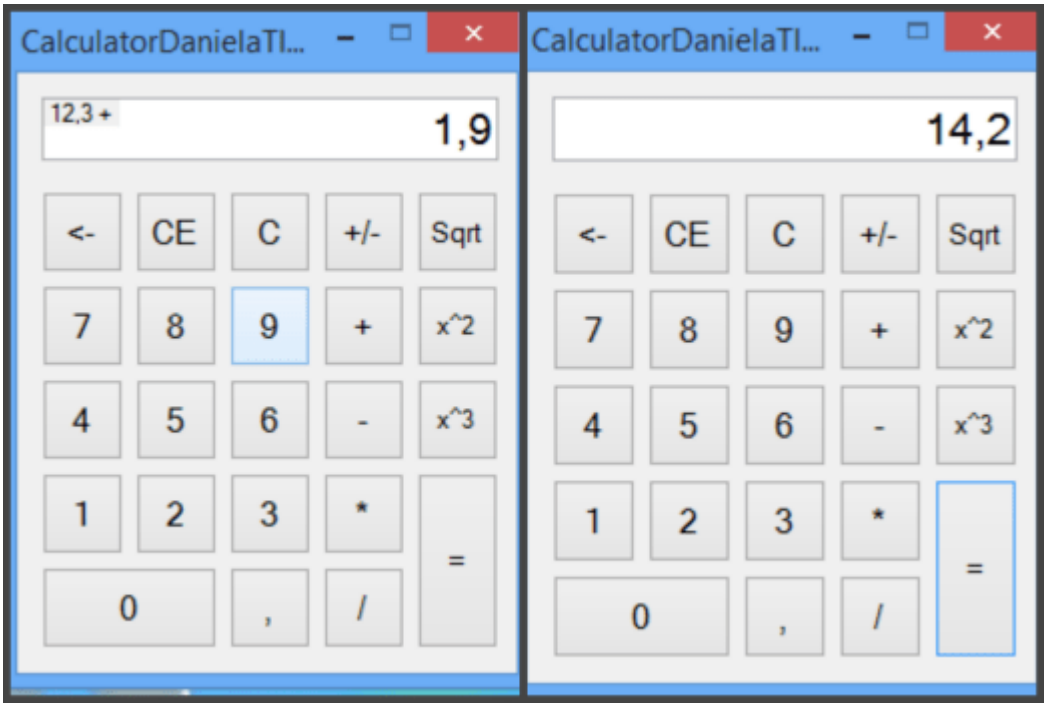
```
1 reference
private void btnSqrt_Click_Click(object sender, EventArgs e)
{
    double sq = Double.Parse(txtDisplay.Text);
    sq = Math.Sqrt(sq);
    txtDisplay.Text = System.Convert.ToString(sq);

}
```

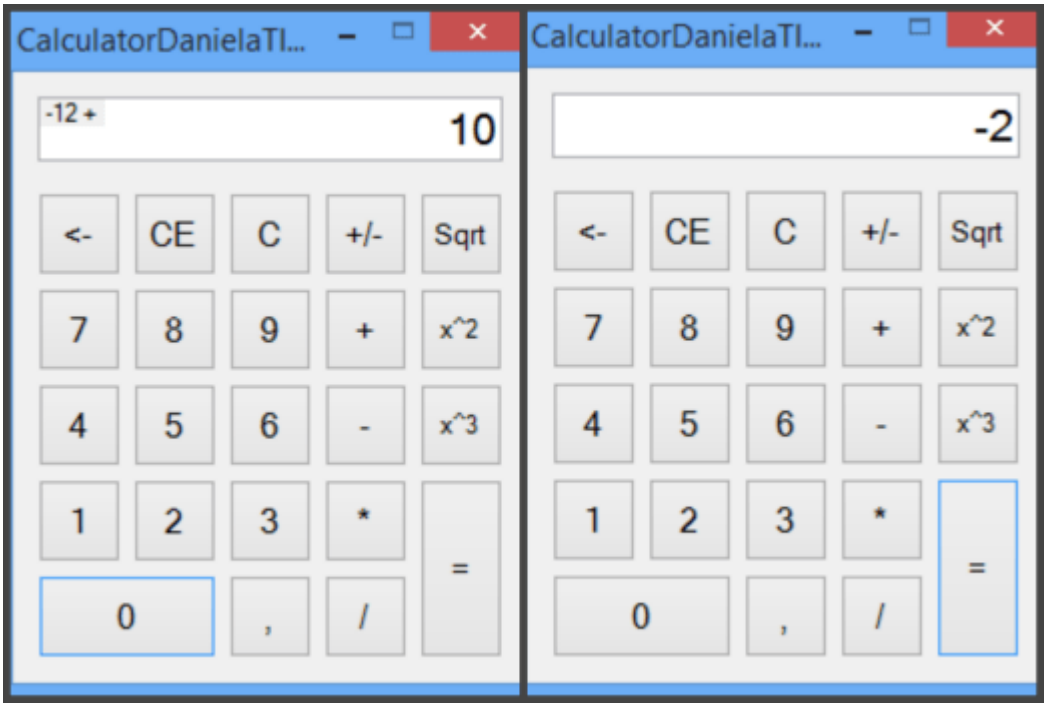
Butonul +/-

In aceasta metoda am declarat o variabila careia i-am atribuit valoarea din txtDisplay convertita in ToDouble. In continuare a = a*(-1); si valoarea obtinuta am convertit-o in ToString, si atribuit-o Text-ului:
txtDisplay.Text = System.Convert.ToString(a);

```
1 reference
private void PlusMinus_Click(object sender, EventArgs e)
{
    Double a;
    a = Convert.ToDouble(txtDisplay.Text) ;
    a = a*(-1);
    txtDisplay.Text = System.Convert.ToString(a);
}
```

4 Screenshoot-uri




CalculatorDanielaTI...
-12 +
10
<-
CE
C
+/-
Sqrt
7
8
9
+
x^2
4
5
6
-
x^3
1
2
3
*
=
0
,
/
CalculatorDanielaTI...
-2
<-
CE
C
+/-
Sqrt
7
8
9
+
x^2
4
5
6
-
x^3
1
2
3
*
=
0
,
/

5 Concluzii

In acest laborator, am facut cunostinta cu ceva total nou pentru mine, limbajul C-Sharp.Mi-a placut,e usor de utilizat,contine functii care m-au ajutat mult precum **Parse , ToString, ToDouble** si proprietati precum **.Text , .Remove , .Lenght , .Contains**.Pentru un limbaj de nivel inalt este foarte okey sa poti interactiona cu limbajul friendly. De asemenea si IDE-ul folosit (**Visual Studio**) este comod, este bine aranjat cu menu-uri,care ulterior pot fi rearanjate dupa comoditate.Totusi, bazindu-ma mai mult tehnic, calculatorul a avut nevoie de cunostinte bune si nunate de stiut, m-am documentat pe Internet, am folosit surse,tutoriale din **Youtube**, care m-au ajutat mult sa realizez aceasta lucrare.