

FACULTATEA CALCULATOARE, INFORMATICA SI MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A PRODUSELOR SOFT

LUCRAREA DE LABORATOR#3

---

# GUI Development

---

*Autor:*

Victor Talpa

*lector asistent:*

Irina COJANU

*lector superior:*

Svetlana COJOCARU

## Laboratory work #3

### 1 Scopul lucrării de laborator

Studierea constructorilor GUI în medii integrate de dezvoltare.

### 2 Obiective

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

### 3 Laboratory work implementation

#### 3.1 Tasks and Points

1. Realizeaza un simplu GUI calculator care suporta urmatoare functii: +, -, /, \*, putere, radical, InversareSemn(+/-), operatii cu numere zecimale.
2. Divizare proiectului in doua module - Interfata grafica(Modul GUI) si Modulul de baza(Core Module).

#### 3.2 Analiza lucrarii de laborator

Link spre repozitoriu: <https://github.com/VictorTalpa/MIDPS/tree/master/MIDPS/Lab%233>

1. Initial am creat design-ul calculatorului, am aranjat butoanele, textbox-ul si le-am atribuit proprietatile dorite in fereastra de proprietati
2. Prima problema intilnita la dezvoltarea programului a fost divizarea programului in doua module
3. Am creat o clasa noua in care sa pastrez partea functionala a programului, dar am dat de problema ca clasa nu are acces catre control-uri, adica catre obiectele de pe forma
4. Dupa citeva incercari si multa cautare a solutiei pe internet, unind bucati din mai multe solutii am obtinut un rezultat.
5. Am facut obiectele de pe forma publice, dupa care in clasa ce va contine partea functionala am creat un pointer de tipul formei din partea GUI
6. La initializarea GUI am atribuit pointerului din clasa functionala adresa formei create, ceea ce a asigurat acces complet catre forma din clasa cu partea functionala
7. Apoi am creat functii pentru fiecare buton in parte si fiecare operatie care o va executa calculatorul
8. Prima variabila sa citeste la apasarea uneia dintre tastele cu operatii matematice (+, -, \*, /,  $x^y$ ) iar a 2-a variabila se citeste la apasarea butonului "=" unde se executa si operatia matematica
9. Operatiile matematice sunt alese intr-un switch care controleaza o variabila string
10. Am implementat butonul Backspace pentru a sterge o singura cifra, cit si butonul "C" pentru a sterge tot numarul introdus
11. Am limitat sirul de caractere afisat in textbox la 9 caractere, adica teoretic am un calculator 9-digit
12. Am testat calculatorul in mai multe situatii (ex.:impartirea la 0)
13. Am reparat mici bug-uri, erori cum ar fi punctul pentru numere zecimale putea fi apasat de mai multe ori (0...01 sau 0.0.1)
14. In final am obtinut un calculator destul de simplu, light, ce poate fi folosit pentru a efectua calcule simple, chiar poate fi inlocuire calculatorului standard Windows

### 3.3 Imagini

Divizînd programul în doua module, Codul aflat în clasa formei este scurt, avînd doar legături cu funcțiile necesare, aflate in altă clasă, în alt fișier, și anume clasa Functions.

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace Calculator
5  {
6      public partial class Calc : Form
7      {
8          public Calc()
9          {
10             InitializeComponent();
11             Functions.Calculator = this;
12
13             buttonBackspace.Click += new EventHandler(Functions.Backspace);
14             buttonMultiply.Click += new EventHandler(Functions.Multiply);
15             buttonDivide.Click += new EventHandler(Functions.Divide);
16             buttonEqual.Click += new EventHandler(Functions.Equal);
17             buttonMinus.Click += new EventHandler(Functions.Subtract);
18             buttonSqrt.Click += new EventHandler(Functions.SquareRoot);
19             buttonSign.Click += new EventHandler(Functions.ChangeSign);
20             buttonPlus.Click += new EventHandler(Functions.Add);
21             buttonPow.Click += new EventHandler(Functions.Exponent);
22             buttonDot.Click += new EventHandler(Functions.Dot);
23             buttonC.Click += new EventHandler(Functions.Clear);
24
25             num0.Click += new EventHandler(Functions.num0Click);
26             num1.Click += new EventHandler(Functions.num1Click);
27             num2.Click += new EventHandler(Functions.num2Click);
28             num3.Click += new EventHandler(Functions.num3Click);
29             num4.Click += new EventHandler(Functions.num4Click);
30             num5.Click += new EventHandler(Functions.num5Click);
31             num6.Click += new EventHandler(Functions.num6Click);
32             num7.Click += new EventHandler(Functions.num7Click);
33             num8.Click += new EventHandler(Functions.num8Click);
34             num9.Click += new EventHandler(Functions.num9Click);
35         }
36     }
37 }
```

Cîteva funcții din fișierul Functions.cs

```
static public void SquareRoot(object Sender, EventArgs e)
{
    x = Convert.ToDouble(Calculator.textBox.Text);
    Calculator.textBox.Text = Math.Sqrt(x).ToString();
    if (Calculator.textBox.Text.Length > 9)
        Calculator.textBox.Text = Calculator.textBox.Text.Substring(0, 9);
    operation = "none";
}

static public void Exponent(object Sender, EventArgs e)
{
    x = Convert.ToDouble(Calculator.textBox.Text);
    Calculator.textBox.Text = "0";
    operation = "power";
}

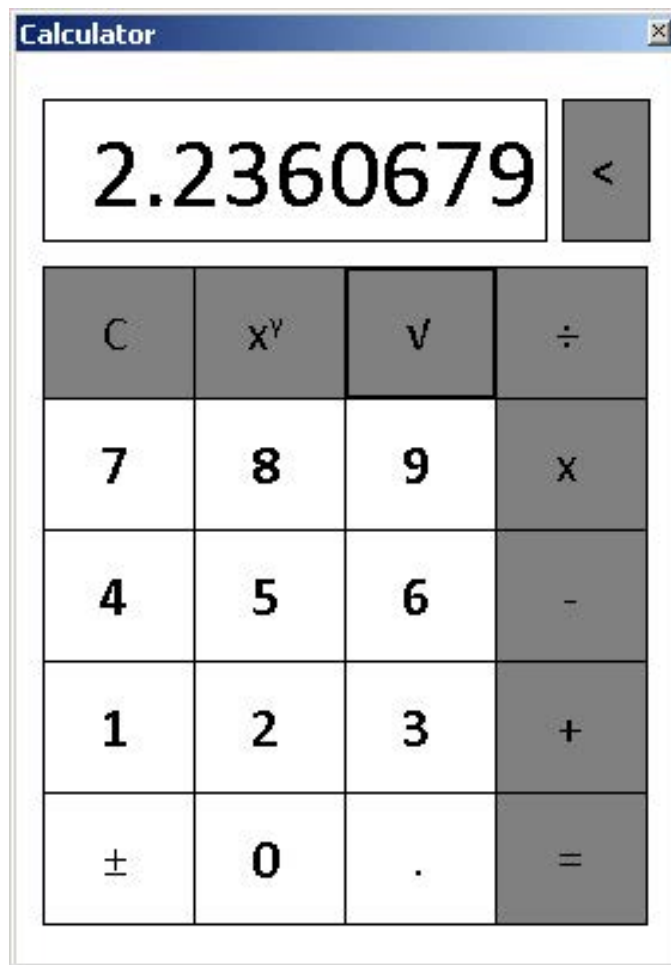
static public void Backspace(object Sender, EventArgs e)
{
    string s = Calculator.textBox.Text;
    if (s.Length > 1)
        s = s.Substring(0, s.Length - 1);
    else
        s = "0";
    Calculator.textBox.Text = s;
}

static public void Dot(object Sender, EventArgs e)
{
    string str = Calculator.textBox.Text;
    int lenght = str.Length;
    bool dot = false;

    foreach(char c in str)
    {
        if (c == '.') dot = true;
    }
    if(!dot)
        Calculator.textBox.AppendText(".");
}
```

Interfața calculatorului și produsul final.

Calculatorul prevede lucru cu numere zecimale, a fost testat și toate butoanele operează corect.



## Concluzie

În urma efectuării acestei lucrări de laborator, am studiat mai avansat mediul integrat de programare Visual Studio, cât și limbajul de programare C#. Acesta este un limbaj de programare strict orientat pe obiecte, spre deosebire de C++ care combină OOP cu programarea procedurală. Visual Studio este un IDE extrem de puternic. Este instrumentul principal al companiei Microsoft și a developerilor pentru efectuarea programelor cu GUI și anume ferestre în SO Windows. Calculatorul este de fapt unul dintre scopuri pentru care s-a creat tehnica de calcul (computerele) și anume pentru a efectua calcule complicate. Astfel efectuând această lucrare, noi studiem și logica programării unui calculator ceea ce creează un fundament solid în viitoarele sarcini de programare. Am învățat să găsim soluții la probleme de programare și mai complicate care se întâlnesc în cursul dezvoltării unui proiect și anume partea cu divizarea programului în două module, care este de fapt o tehnică de programare destul de bună, dar și am creat un instrument util pentru viața de zi cu zi. Începând cu un calculator simplu, putem continua dezvoltarea acestuia la nivele mai mari, adăugând operații matematice mai complicate cum ar fi trigonometrie, derivate sau integrale. Un scop ar fi proiectul WolframAlpha, cam așa ar trebui să lucreze un calculator performant.

## References

1 Microsoft Developer Network (MSDN), C# Tutorials

[https://msdn.microsoft.com/en-us/library/aa288436\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/aa288436(v=vs.71).aspx)

2 Stack Overflow Community <http://stackoverflow.com/>

3 Google and Youtube