

MINISTERUL EDUCATIEI AL REPUBLICII MOLDOVA

UNIVERSITATEA TEHNICA A MOLDOVEI

FACULTATEA CALCULATORARE INFORMATICA SI MICROELECTRONICA

RAPORT

Lucrarea de laborator nr.2

Disciplina: Medii Interactive de Dezvoltare a Produselor Soft

Tema: GUI Development

A efectuat
st.gr. TI-154

Popusoi Victor

A controlat
lect., asis.

Cojanu Irina

Chisinau 2017

1 Scopul lucrarii de laborator

- Cunoasterea noilor IDE si lucru cu acestea
- Repartizarea proiectului pe submodule in dependenta de functionalul prestat
- Insusirea detaliata a posibilitatilor limbajului ales si modulul lui matematic

2 Obiectivele lucrarii de laborator

- 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).

3 Cerintele lucrarii de laborator

Aplicatia trebuie sa fie divizata in doua module:

- Core module - contine functionalitatile de baza
- GUI module - include codul responsabil de crearea Interfetei Grafice si interactiunea ei cu elementele interfetei grafice si modulul de baza

* Incearca sa realizezi programul tau sa fie cross platform (Compatibil cu diferite platforme: Windows, Linux, Mac).

4 Efectuarea lucrarii de laborator

4.1 Sarcinile propuse pentru efectuare lucrarii de laborator

Basic Level (nota 5 - 6) :

- Realizeaza un simplu GUI calculator care suporta functiile de baza: $+$, $-$, $/$, $*$.

Normal Level (nota 7 - 8):

- Realizeaza un simplu GUI calculator care suporta urmatoare functii: $+$, $-$, $/$, $*$, putere, radical, InversareSemn($+/ -$).

Advanced Level (nota 9 - 10):

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

4.2 Realizarea lucrarii de laborator

Link-ul de la repository here

În această lucrare a fost propusă elaborarea unui GUI Calculator, un calculator cu interfață grafică. Pentru realizarea acestui GUI Calculator, a fost folosit mediul de dezvoltare Visual Studio Community 2015 și limbajul de programare C#. Modificările se monitorizează prin intermediul Git-ului. Dezvoltarea aplicației începe cu crearea și perfecționarea interfeței grafice. VS utilizează ”forma” drept spațiu de dezvoltare. Pentru a realiza un simplu calculator, forma (fereastra) a fost dezvoltată vizual astfel (Screen here 4.1)

Primul pas făcut în crearea calculatorului a fost UI-ul și definirea fiecărui buton de care am avut nevoie în acest proiect. După adăugarea butoanelor, am schimbat toate denumirile implicite `button10` în unul mai potrivit `divideButton`. Am creat metodele pentru fiecare acțiune generată din UI, de exemplu:

- Adăugarea unei noi cifre la ecran
- Stergerea unei cifre
- Adunarea, împartirea, înmulțirea, radical, ridicarea la puterea
- Etc.

Elementele utilizate:

- 1) 2 `TextBox`-uri, unul a fost utilizat pentru spațiul în care se introduc datele și se vizualizează rezultatul, iar cel de-al doilea a fost utilizat pentru afișarea operațiilor.
- 2) 24 butoane pentru interacțiunea utilizatorului cu programul.
- 3) Butonul ”About” care ne arată informația despre calculator și butonul ”File – Exit” cu ajutorul căruia putem ieși din program.

Butonul ”+”

```
private void plusButton_Click(object sender, EventArgs e)
{
    number01 = double.Parse(textBoxShow.Text);
    numberTextBox.Text += textBoxShow.Text + " + ";
    textBoxShow.Text = "";
    plusButtonCounter++;
}
```

Pentru adunarea numerelor, într-un cit valorile se introduc direct în câmpul casetei text, este necesară convertirea textului în valoare. Limbajul C# permite convertirea prin următoarea sintaxă `textBoxShow.Text`, convertind textul din câmpul `Text` al structurii `TextBox` în valoare. Adunarea se realizează ca și în majoritatea limbajelor de programare, operanții se adună, rezultatul se păstrează într-o variabilă. Respectiv, rezultatul adunării se afișează în câmpul structurii `textBoxShow.Text`. (Screen here 4.3)

Butoanele ”-”, ”*”, ”/”

Aceste butoane funcționează la fel ca și butonul ”+” doar că execută funcțiile de scădere, înmulțire și împartire. (Screen here 4.4 4.5 4.6 4.7). La operația de împartire a fost adăugat un `MessageBox` care ne spune că numitorul nu poate fi 0. (Screen here 4.7)

Butonul "C"

Sterge datele din ambele casete ale calculatorului (Screen here 4.8)

```
private void buttonC_Click(object sender, EventArgs e)
{
    textBoxShow.Text = "";
    numberTextBox.Text = "";
}
```

Butonul "Backspace"

Sterge doar cite o cifra din textBox (Backspace) (Screen here 4.9)

```
private void buttonDelete_Click(object sender, EventArgs e)
{
    textBoxShow.Text = textBoxShow.Text.Substring(0, textBoxShow.Text.Length - 1);
}
```

Butonul "sqrt"

Extrage radacina dintr-un numar. (Screen here 4.10)

```
private void buttonRadical_Click(object sender, EventArgs e)
{
    number01 = double.Parse(textBoxShow.Text);
    textBoxShow.Text = "" + Math.Sqrt(number01);
}
```

Pentru a realiza functionalul butonului, a fost necesara utilizarea functiei predefinite in C(Sharp) "Math.Sqrt()"

Butoanele "pow(x,2)" si "pow(x,y)"

Sunt folosite pentru ridicarea la putere a unui numar, doar ca "pow(x,2)" ridica un numar la puterea a 2, iar "pow(x,y)" ridica un numar la orice putere introdusa de utilizator. (Screen here 4.11 4.12)

```
private void buttonPutere_Click(object sender, EventArgs e)
{
    number01 = double.Parse(textBoxShow.Text);
    textBoxShow.Text = "" + (number01*number01);
}
```

```
private void buttonxy_Click(object sender, EventArgs e)
{
    number01 = double.Parse(textBoxShow.Text);
    textBoxShow.Text = "";
    xyButtonCounter++;
}
```

Butonul "+/-"

Este folosit pentru inversarea semnului. (Screen here 4.13)

```
private void buttonPlusMinus_Click(object sender, EventArgs e)
{
    if (textBoxShow.Text.StartsWith("-"))
    {
        textBoxShow.Text = textBoxShow.Text.Substring(1);
    }
    else if (!string.IsNullOrEmpty(textBoxShow.Text) && decimal.Parse(textBoxShow.Text) != 0)
    {
        textBoxShow.Text = "-" + textBoxShow.Text;
    }
}
```

```
textBoxShow.Text = "-" + textBoxShow.Text;
```

Butonul "."

Este folosit pentru lucrul cu numere zecimale. (Screen here 4.14)

```
private void buttonPunct_Click(object sender, EventArgs e)
{
    if (textBoxShow.Text.Contains("."))
        textBoxShow.Text = textBoxShow.Text;
    else
        textBoxShow.Text += ".";
}
```

4.3 Imagini

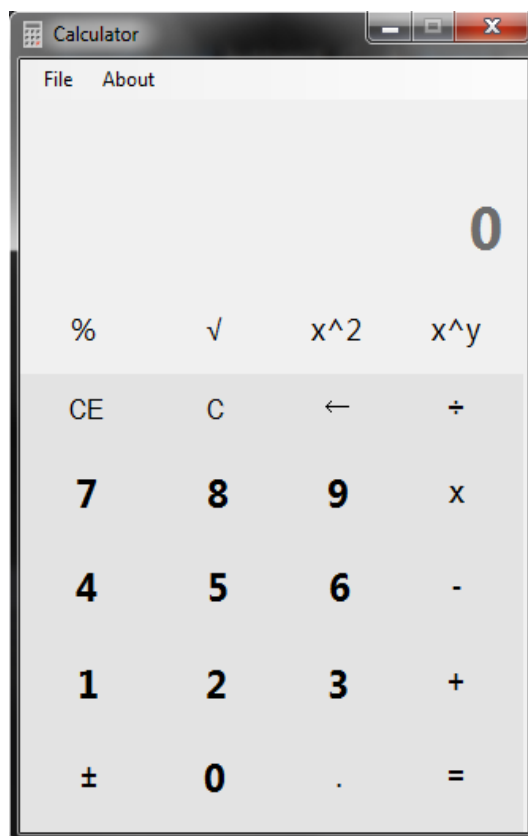


Figure 4.1 – Calculator(Windows 10 style)

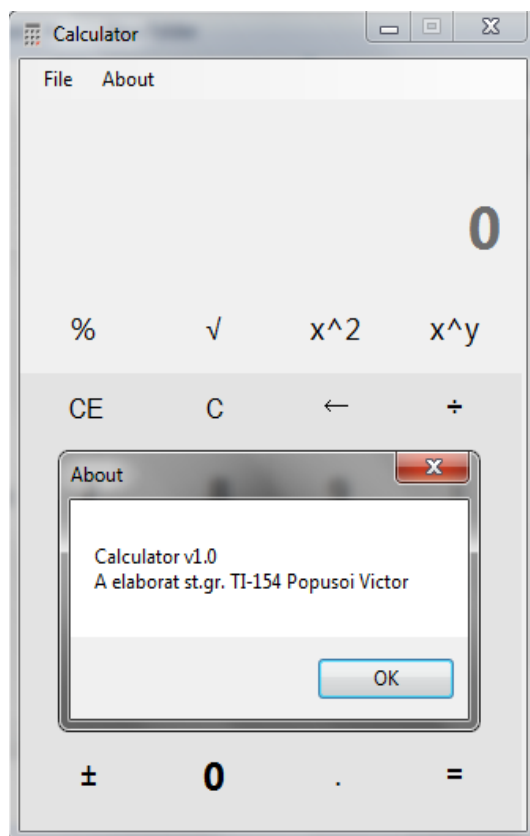


Figure 4.2– About calculuator

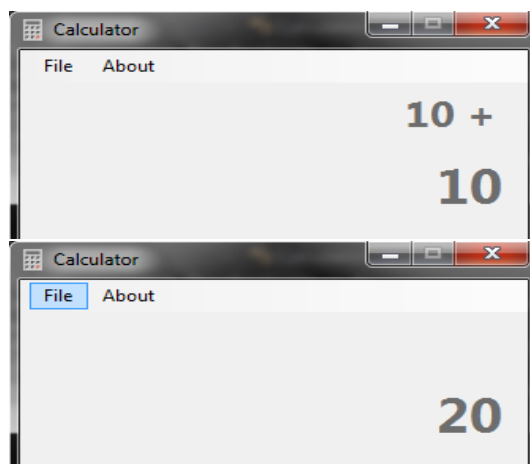


Figure 4.3– Adunarea

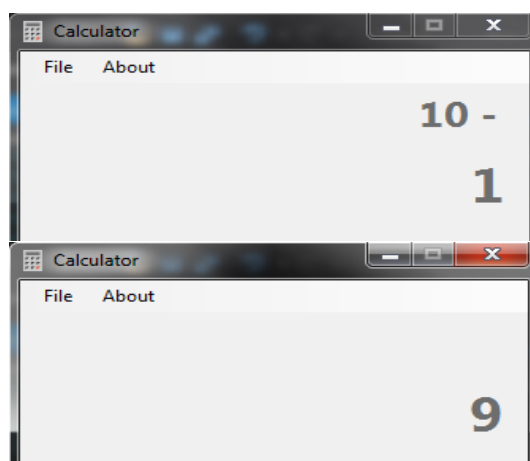


Figure 4.4– Scadere

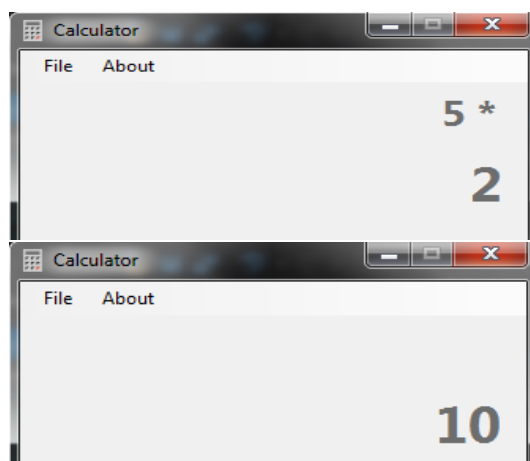


Figure 4.5– Inmultire

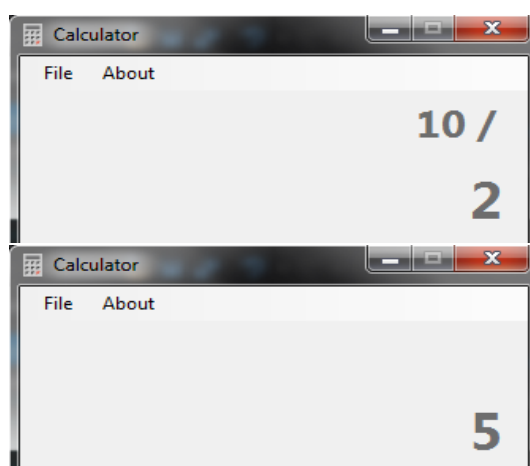


Figure 4.6– Impartire

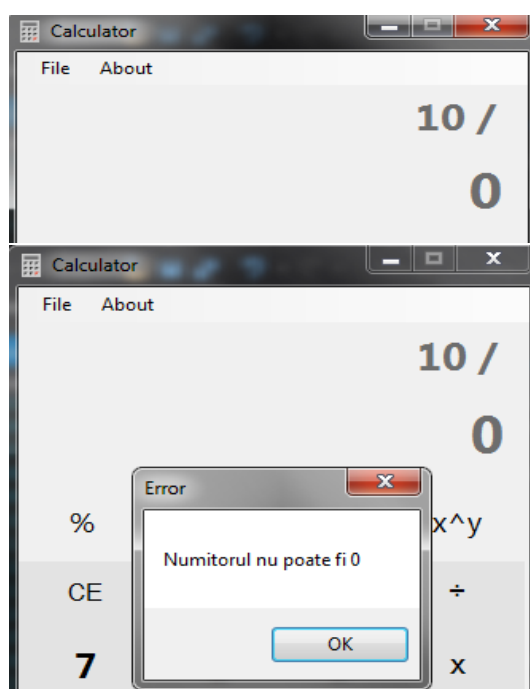


Figure 4.7– Message box

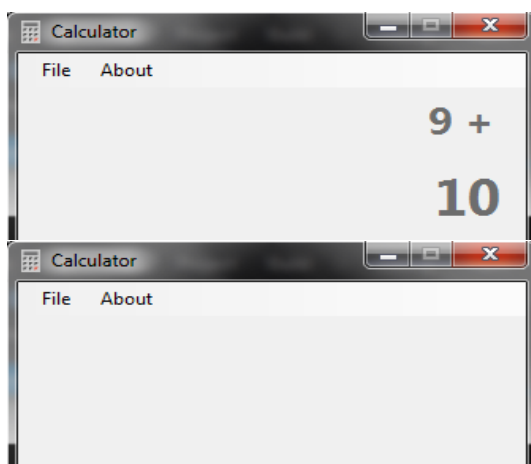


Figure 4.8– Clear text box

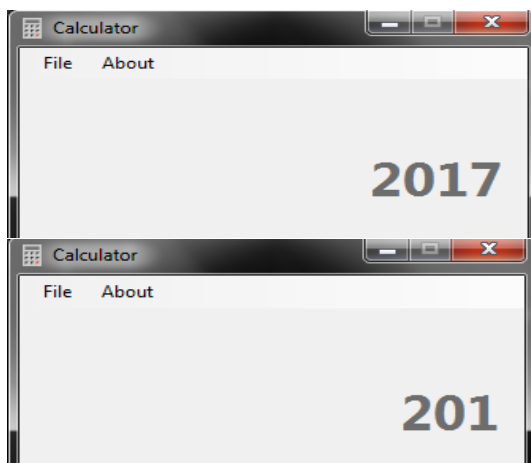


Figure 4.9– Backspace

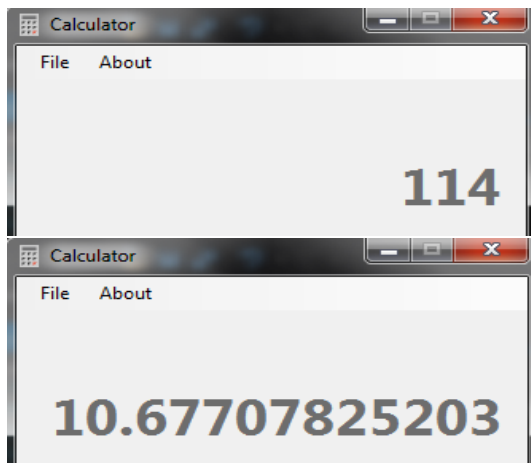


Figure 4.10– Extragerea radacinii

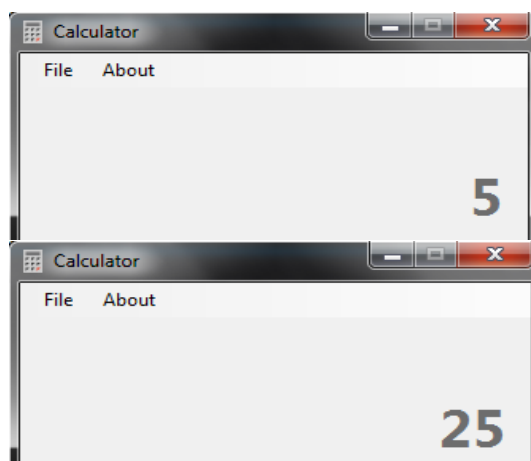


Figure 4.11 – Ridicarea la putere $\text{pow}(x,2)$



Figure 4.12 – Ridicarea la putere $\text{pow}(x,y)$

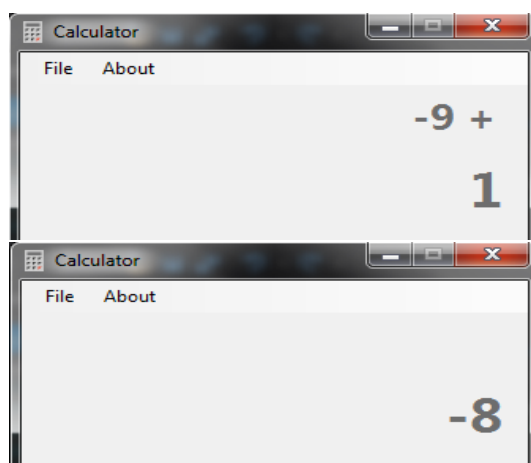


Figure 4.13 – Schimbarea semnului

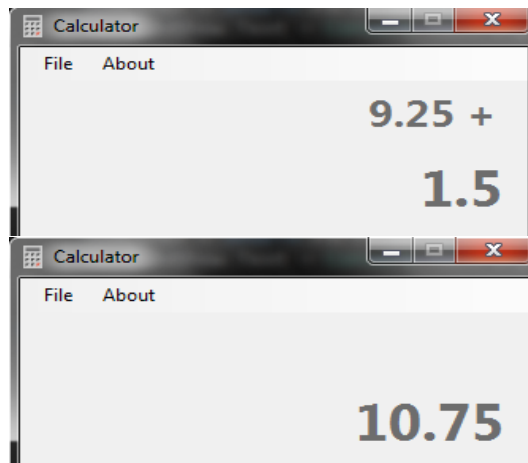


Figure 4.14– Numere zecimale

Concluzie

Pentru indeplinirea acestei lucrari de laborator a fost ales in calitate de IDE Visual Studio, iar limbajul de programare C# , deoarece limbajul are multe facilitati. Unele din ele sunt:

- Este un limbaj de programare simplu, modern, de utilitate generala, cu productivitate mare in programare.
- Este un limbaj orientat pe obiecte.
- Permite dezvoltarea de aplicatii.
- Oferă suport complet pentru dezvoltarea de componente software. C# se poate caracteriza ca fiind nu numai orientat obiect, ci si orientat spre componente.

A fost studiata o metoda de dezvoltare a unui calculator GUI, prin intermediul mediului Visual Studio si limbajul de programare C#. Acesta include operatii simple ca: +, -, *, /, putere, radical, inversare semn(+/-), operatii cu numere zecimale.

Efectuarea unui calculator, contribuie la dezvoltarea aplicatiilor. A fost cunoscut modul de programare a butoanelor si proprietatile acestora, cit si a Formelor utilizate in C#. Au fost obtinute noi cunostinte in domeniul Mediilor de dezvoltare a produselor soft si a fost obtinuta posibilitatea de a invata mai detaliat IDE-ul Visual Studio:

- Componentele grafice
- Visual C# - formele si componentele ei
- C# - modulul Math si supraincercarea unei componente vizuale

Bibliografie

1. Repozitoriul public GitHub:

<https://github.com/PopusoiVictor/MIDPS>

2. Video tutorial:

<https://www.youtube.com/watch?v=wwJ0j-NBfHc>

3. Visual Csharp, official page:

[https://msdn.microsoft.com/en-us/library/360kwx3z\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/360kwx3z(v=vs.90).aspx)