Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică și Microelectronică

Catedra Tehnologii Informaționale

RAPORT

Lucrare de laborator Nr.3
Tema: GUI Development

A efectuat: St. gr. TI-142
Vîrlan Ion

A verificat: I. Cojan

Chişinău 2016

Scopul lucrării:

Analiza funcționalităților constructorilor GUI în mediile integrate de dezvoltare și înțelegerea lor prin realizarea unei aplicații

Obiective:

- 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)

Realizarea lucrării:

1. Inițial am proiectat design-ul calculatorului, aranjînd toate butoanele care vor realiza operețiile dorite și textbox-ul care va afișa calculele efectuate atribuindu-le fiecărui element în parte proprietățile cuvenite.
2. Conform sarcinii cerute în lucrarea de laborator am divizat proiectul în doua module - Interfata grafica(Modul GUI) si Modulul de baza(Core Module)
3. Am realizat pentru fiecare buton cifra evenimentul button_Click pentru a se introduce numerele care trebuiesc calculate.
4. Am realizat pentru fiecare buton cu operație evenimentul operator_Click pentru a se efectua operațiile necesare între numere.
5. Pentru butonul egal am realizaat evenimentul operator_Rezult care va monitoriza pentru calcularea operațiilor si de atransmite la elementul textbox rezultatul corect
6. Apoi în clasa butonului operator_Rezult în switch (operationPerfomed) am realizat funcționalitatea fiecărei din operațiile: +, -, /, *, *sqrt și* ^, prevăzînd posibilitatea ți a numerilor întregi.
7. Am realizat și butonnul , pentru a indentifica numerele întregi și de a le implica în calculele operațiilor.
8. Am implicat și butoanele *CE și C* care vor curăța textbox-ul sau vor aduce la starea inițială cu valoarea 0
9. Pentru a opera și cu numerele negative a fost implicat și butonul +/- prevăzînd posibilitatea de a schimba semnul oricărui număr în orce timp dorit, și avertizarea imposibilității de a extrage rădăcina pătrată din numere negative.
10. A fost prevăzută posibilitatea de a schimba semnul operației în orice timp dorit doar prin a apăsa un clic pe alt semn necesar fără a curăța mai întîi textbox-ul.
11. Au fost eliminate bag-urile de tipul mai multe virgule la rînd într-un număr sau cifra virgulă și repetări de acestea.
12. A fost implementat și un *labelCurrentOperation* pentru a afișa operația curentă care a fost aleasă pentru a se realiza calculele respective.
13. În raport este prezentat lisning-ul și cîteve screenhsot-uri ale aplicației calculatorului care a fost elaborat.
14. Toate materialele necesare și utile din cadrul acestei lucrări de laborator au fost încarcate pe github.com în repozitoriul personal împreună cu comentrariul proiectului realizat cu ajutorul lui *commit*

Lisning-ul în limbajul de programare C#:

```
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 Calcutator
{
    public partial class Form1 : Form
    {
        Double resultValue = 0;
        String operationPerfomed = "";
        bool isOperationPerfomed = false;
        public Form1()
        {
            InitializeComponent();
```

```
}

private void Form1_Load(object sender, EventArgs e)
{

}
private void button_click(object sender, EventArgs e)
{
    if ((textBox_Result.Text == "0") || (isOperationPerfomed))
        textBox_Result.Clear();

    isOperationPerfomed = false;
    Button button = (Button)sender;
    if (button.Text == ",")
    {
        if (!textBox_Result.Text.Contains(","))
            textBox_Result.Text = textBox_Result.Text + button.Text;

    }else
    textBox_Result.Text = textBox_Result.Text + button.Text;
}

private void operator_click(object sender, EventArgs e)
{
    Button button = (Button)sender;
        operationPerfomed = button.Text;
        resultValue = Double.Parse(textBox_Result.Text);
        labelCurrentOperation.Text = resultValue + " " + operationPerfomed;
        isOperationPerfomed = true;
}

private void button5_Click(object sender, EventArgs e)
{
    textBox_Result.Text = "0";
}

private void button10_Click(object sender, EventArgs e)
{
    textBox_Result.Text = "0";
    resultValue = 0;
}

private void button20_Click(object sender, EventArgs e)
{
    switch (operationPerfomed)
    {
        case "+":
            textBox_Result.Text = (resultValue + Double.Parse(textBox_Result.Text)).ToString();
            break;
        case "-":
            textBox_Result.Text = (resultValue - Double.Parse(textBox_Result.Text)).ToString();
            break;
        case "*":
            textBox_Result.Text = (resultValue * Double.Parse(textBox_Result.Text)).ToString();
            break;
        case "/":
            textBox_Result.Text = (resultValue / Double.Parse(textBox_Result.Text)).ToString();
            break;
        case "sqrt":
            textBox_Result.Text = (Math.Sqrt(Double.Parse(textBox_Result.Text))).ToString();
            break;
        case "^":
            textBox_Result.Text = Math.Pow(resultValue, Double.Parse(textBox_Result.Text)).ToString();
```

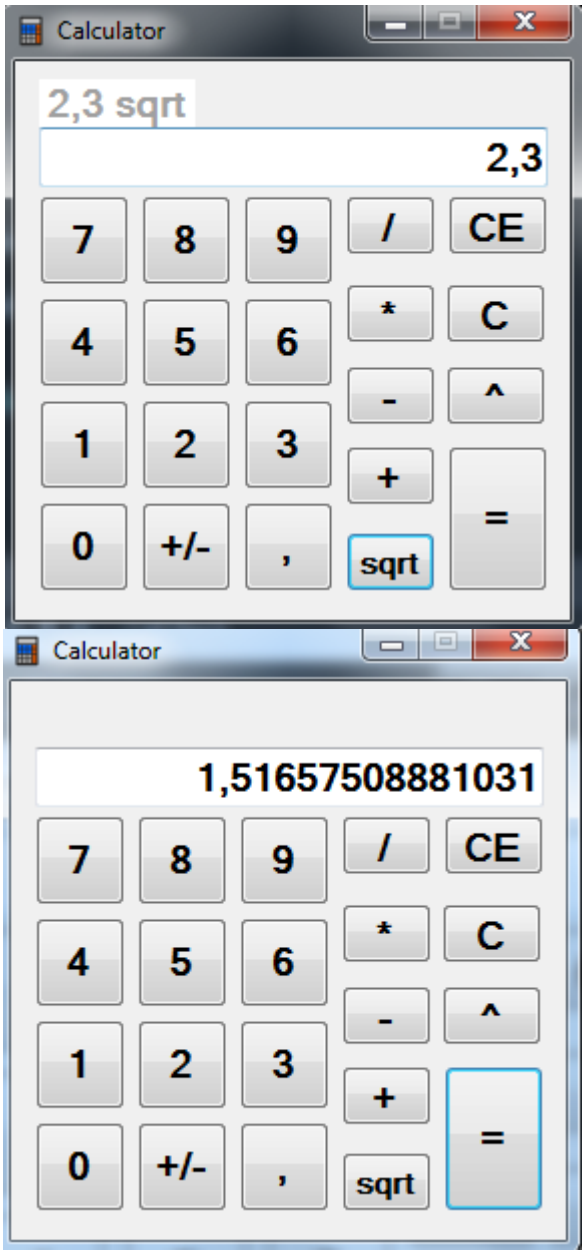
```
                break;
            default:
                break;
        }
        resultValue = Double.Parse(textBox_Result.Text);
        labelCurrentOperation.Text = "";
    }

    private void button21_Click(object sender, EventArgs e)
    {
        if (textBox_Result.Text.Contains("-"))
            textBox_Result.Text = textBox_Result.Text.Remove(0, 1);
        else
            textBox_Result.Text = "-" + textBox_Result.Text;
    }

    private void textBox_Result_TextChanged(object sender, EventArgs e)
    {

    }
  }
}
```

Rezultate screenshot:



Concluzie

În procesul efectuării lucrării de laborator am făcut cunoștință cu metodele de lucru cu mediul de dezvoltarea a produselor soft Visual Studio C# realizînd în limbajul de programare C# un Sample Calculator care efectuează operațiile de bază, aceasta fiind elaborat cu ajutorul instrumentelor din cadrul WindowsForm și implementarea codului în fișierul cu extensia .cs care a fost compilat executat și testat manual corectînd bag-urile apărute.

Pe parcursul lucrării am înțeles importanța, utilitatea, și eficiența acestor facilități pe gare ni le oferă mediul de dezvoltarea Visual Studio C# prin crearea si monitorizarea proiectului care este complex dar poate fi realizat cu ajutorul acestor facilități propuse de limpajul de programare C3 împreună cu pachetul de instrumente WindowsForm fiindcă este ușor, rapid pentru a elabora lucruri frumoase și utile.

Bibliografie:

Îndrumar de laborator Lucrarea nr.3
https://habrahabr.ru/hub/c/
https://habrahabr.ru/company/piter/blog/236985/
http://www.ozon.ru/catalog/1139483/
http://forcoder.ru/c-sharp/
http://bookwebmaster.narod.ru/csharp.html
http://www.proklondike.com/books/dotnet.html