Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei

RAPORT

La lucrarea de laborator nr. 3
la disciplina «MIDPS»

«GUI Calculator»

A efectuat: studentul gr. T-145 Ialticenco A.
A verificat: lector univ. Cojocaru S.

Chisinau 2016

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).

Sarcina lucrarii

- Realizează 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).

1. Modul GUI

Modul GUI va fi encapsulat in clasa Form1.

```
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 PW9
{
    public partial class Form1 : Form
    {
        public bool error;
        public Form1()
        {
            InitializeComponent();
        }

        private void checkBox1_Click(object sender, EventArgs e)
        {
            if (check_round.Checked)
                select_round.Enabled = true;
            else
                select_round.Enabled = false;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            field_a.Clear();
            field_b.Clear();
            field_c.Clear();
            field_x1.Clear();
            field_x2.Clear();
            full.Clear();
        }
```

```
        private void checkEnter(Object sender, KeyPressEventArgs e)
        {
            if (((e.KeyChar < 48 || e.KeyChar > 57) && e.KeyChar != 8 &&
e.KeyChar != 46))
            {
                e.Handled = true;
                return;
            }

            if (e.KeyChar == 46)
            {
                if ((sender as TextBox).Text.IndexOf(e.KeyChar) != -1)
                    e.Handled = true;
            }
        }

        private void field_x1_KeyPress(object sender, KeyPressEventArgs
e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            error = false;
            if (check_round.Checked)
            switch (select_round.SelectedIndex)
            {
                case 1: Resolver.setRound(2); break;
                case 2: Resolver.setRound(3); break;
                default: Resolver.setRound(1); break;
            }
            else
                Resolver.setRound(2);
            if (Resolver.resolve(field_a.Text, field_b.Text,
field_c.Text))
            {
                    full.Clear();
                    for (pb.Value = 0; pb.Value < 100; pb.Value += 1)
                    {
                        addText(pb.Value);
                        Application.DoEvents();
                        System.Threading.Thread.Sleep(100);
                    if (error) break;
                    }
                pb.Value = 0;
                if (error == false)
                {
                    MessageBox.Show("-ешение получено!", ""спех",
                 MessageBoxButtons.OK, MessageBoxIcon.Information);

                    field_x1.Text = Resolver.x1 + "";
                    field_x2.Text = Resolver.x2 + "";
                }
```

```
            }
            else
                MessageBox.Show("sожалуйста, заполните все об€зательные
пол€!", "ќшибка!",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

        public void addText(int i)
        {
            String s = "";
            switch (i)
            {
                case 10: s += "ƒано:\n"; colorize(s, Color.Black);
break;
                case 15: s += "\na = " + Resolver.getA() + "\nb = " +
Resolver.getA() + "\nc = " + Resolver.getA() + "\n"; colorize(s,
Color.Red); break;
                case 20: s += "\n¬ычисл€ем дискриминант"; colorize(s,
Color.Blue);  break;
                case 25: s += "."; colorize(s, Color.Blue);  break;
                case 30: s += "."; colorize(s, Color.Blue);  break;
                case 35: s += "."; colorize(s, Color.Blue);  break;
                case 40: s += "\n\nD = " + Resolver.getPreD();
colorize(s, Color.Black);  break;
                case 50: if (only_real.Checked && Resolver.wasComplex) {
s += "\n\nƒействительных корней нет"; MessageBox.Show("" заданного вами
уравнени€ нет действительных корней!", " орней нет!",
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
pb.Value = 100; error = true;
                            colorize(s, Color.Red);
                    }  break;
                case 55: s += "\n\nSQRT(D) = " + Resolver.getD();
colorize(s, Color.Black); break;
                case 60: s += "\n\n¬ычисл€ем корни уравнени€";
colorize(s, Color.Blue); break;
                case 63: s += "."; colorize(s, Color.Blue); break;
                case 66: s += "."; colorize(s, Color.Blue); break;
                case 69: s += "."; colorize(s, Color.Blue); break;
                case 70: s += "\n\nx1 = "+Resolver.x1; colorize(s,
Color.Red);  break;
                case 80: s += "\n\nx2 = " + Resolver.x2; colorize(s,
Color.Red);  break;
                case 85: if (Resolver.wasComplex) break;  s +=
"\n\nќпредел€ем вершину параболы"; colorize(s, Color.Blue); break;
                case 89: if (Resolver.wasComplex) break; s += ".";
colorize(s, Color.Blue); break;
                case 92: if (Resolver.wasComplex) break; s += ".";
colorize(s, Color.Blue); break;
                case 95: if (Resolver.wasComplex) break; s += ".";
colorize(s, Color.Blue); break;
                case 99: if (Resolver.wasComplex) break; s += "\n\n" +
Resolver.getVertix(); colorize(s, Color.Red); break;

            }
        }
```

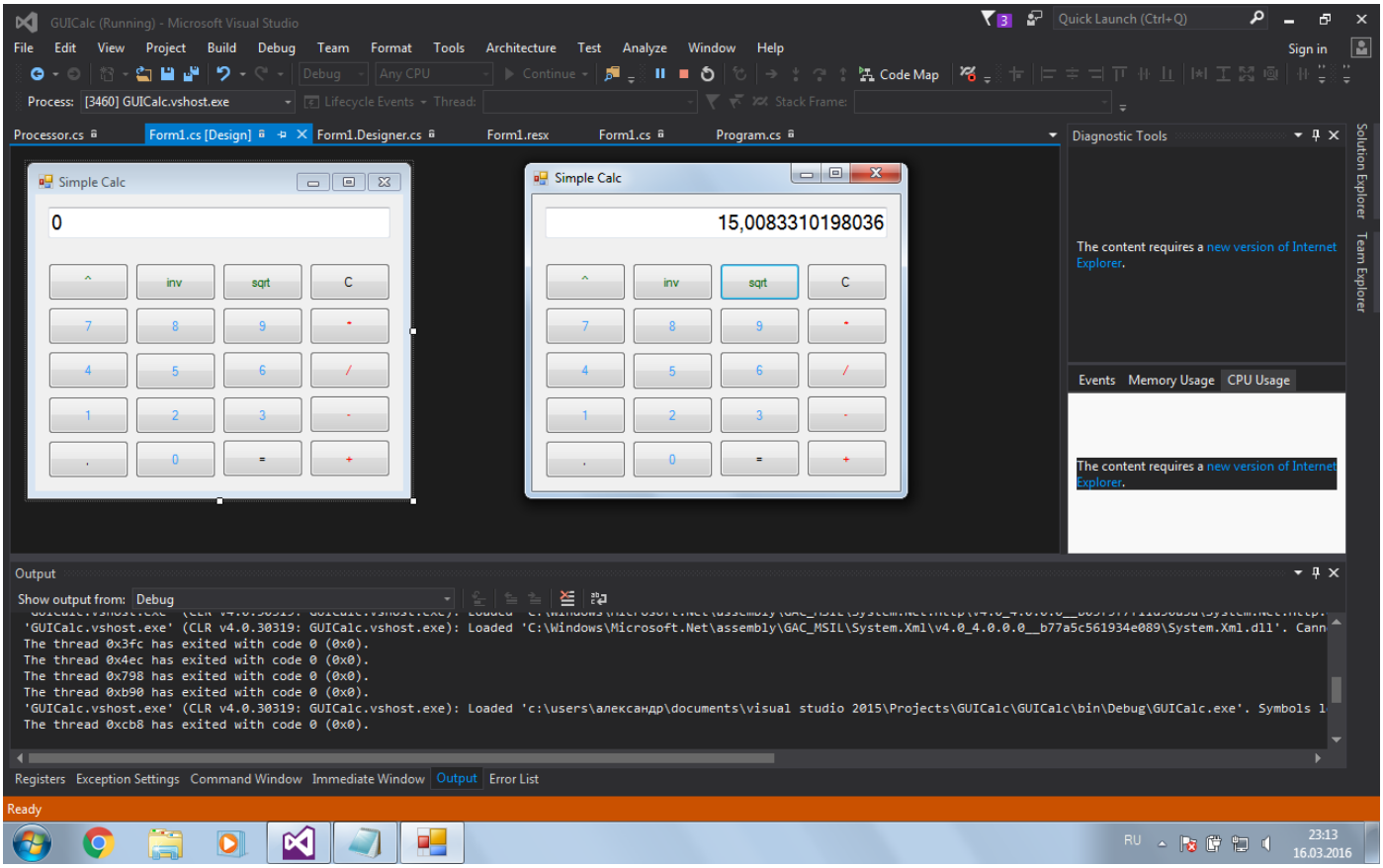
```
        public void colorize(String s, Color c)
        {
            int lineNum = full.Lines.Count();
            full.AppendText(s);
            full.Select(full.TextLength - s.Length, s.Length);
full.SelectionColor = c;
            full.SelectionStart = full.Text.Length;
            if (full.Lines.Count() > lineNum)
                full.ScrollToCaret();
        }

    }


}
```

2. Modul de baza (Core Module) va fi encapsulat in clasa Processor:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace GUICalc.CalcCore
{
    public abstract class Processor
    {
        public static double mem1 = 0;
        public static double mem2 = 0;
        public static bool v1 = false;
        public static bool v2 = false;
        public static operations curOp;
        public static int opCount;
        public enum operations {plus, minus, mult, div, sqrt, pow, inv
};

        public static double getResult(operations op)
        {
            switch (op)
            {
                case operations.plus: return mem1 + mem2;
                case operations.minus: return mem1 - mem2;
                case operations.mult: return mem1*mem2;
                case operations.div: return mem1/mem2;
                case operations.sqrt: return Math.Sqrt(mem1);
                case operations.pow: return Math.Pow(mem1,mem2);
                case operations.inv: return -mem1;
            }
            return 0;
        }

        public static bool setVal(double val)
        {
            if (v1 == false) {
            mem1 = val;
```

```
                v1 = true;
                Console.WriteLine("1");
                return false;
            } else
            {
                mem2 = val;
                //v1 = false;
                Console.WriteLine("2");
                mem1 = getResult(curOp);
                mem2 = 0;
                return true;
            }
        }
    }
}
```

Rezultatele lucrării



Concluziile

În cadrul acestei lucrări de laborator am creat simplu calculator GUI prin intermediul IDE Visual Studio 2015 în limbajul C#. Calculatorul suportă operațiile simple: +,-,*,/,putere,radical,InversareSemn(+/-), operatii cu numere zecimale. Produsul soft realizat poate fi executat nu doar sub Windows, dar si sub alte platforme (Linux, Mac) în cazul în care este instalat Mono Framework, deși aplicația este de tip cross-platform. Cunoștințele obținute pe parcursul desfășurării lucrării de laborator vor fi utile pentru realizarea proiectelor ce urmeaza.

Bibliografie

1. https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx - C# Programming