

Технический Университет Молдовы  
Кафедра Автоматики и Информационных Технологий

Лабораторная работа №2  
По MIDPS

Тема: **GUI Development**

---

---

Выполнила:

Проверил:

ст. гр. ТІ-155  
Архирий Ольга  
Гожин Виктор

Кишинев 2017

## Цель работы:

Ознакомиться с понятием графического интерфейса и реализовать простой GUI калькулятор.

## Задания:

- Создание простого GUI калькулятора, который реализует следующие функции: +, -, /, \*, степень, квадратный корень, смену знака(+/-), операции с десятичными числами.
- Реализация проекта в двух частях: графический интерфейс (Modul GUI) и базовый (Core Module).

## Выполнение заданий:

Для создания калькулятора мы воспользуемся языком программирования C# и средой разработки Visual Studio.

C# (произносится *си шарп*) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

**Microsoft Visual Studio** — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

При создании калькулятора мы работали с двумя файлами: *Form1.cs* (форма, графический интерфейс) и *Operators.cs* (класс, созданный нами для функций операторов). Сначала необходимо поработать с конструктором форм – добавить и расставить кнопки цифр и операторов, строку ввода-вывода текста. Важные математические операции, такие как +, -, /, \*, степень, квадратный корень мы реализуем в *Operators.cs* и вызываем в *Form1.cs*.

Наш калькулятор управляется клавиатурой и мышью, отображает предыдущую операцию, способен работать с десятичными числами. Также нет необходимости после каждого действия нажимать «=», можно работать с результатом. Для удобства я запретила изменять размеры окна калькулятора, чтобы он оставался привычным для пользователя.

### Содержание файла *Operators.cs*:

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```

using System.Text;
using System.Threading.Tasks;

namespace CalculatorCSharp
{
    class Operators
    {
        public static double Add(double n1, double n2)
        {
            return n1 + n2;
        }

        public static double Sub(double n1, double n2)
        {
            return n1 - n2;
        }

        public static double Div(double n1, double n2)
        {
            return n1 / n2;
        }

        public static double Mult(double n1, double n2)
        {
            return n1 * n2;
        }

        public static double sqrt(double n1)
        {
            return Math.Sqrt(n1);
        }
        public static double pw(double n1, double n2)
        {
            return Math.Pow(n1, n2);
        }
        public static double pl_min(double n1)
        {
            return (n1* (-1));
        }
    }
}

```

### Содержание файла *Form1.cs*:

```

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 CalculatorCSharp
{
    public partial class Form1 : Form
    {
        double value=0;
        String operation = "";
        bool operation_pressed = false;
        short flag = 0; //чтобы не приписывались цифры к результату

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        { }

        private void textBox1_TextChanged(object sender, EventArgs e)
        { }
        private void button_Click(object sender, EventArgs e)
        {

```

```

        Button b = (Button)sender;
        if ((result.Text == "0") || (operation_pressed) || (flag==1))
            result.Clear();
        flag = 0;
        operation_pressed = false;

        if (b.Text == ",")
        {
            if (result.Text=="") result.Text = result.Text.Insert(0, "0") + b.Text;
            if (!result.Text.Contains(",")) result.Text = result.Text + b.Text;
        }
        else
            result.Text = result.Text + b.Text;
    }

    private void button16_Click(object sender, EventArgs e)
    {
        result.Text = "0";
        equation.Text="";
        value = 0;
    }

    private void operator_click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        if (value != 0)
        {
            if (b.Name == "plus_minus")
                result.Text = (Double.Parse(result.Text) * (-1)).ToString();
            else if (b.Text == "√")
                result.Text = Operators.sqrt(Double.Parse(result.Text)).ToString();
            equal.PerformClick();
            operation_pressed = true;
            if ((b.Text!="√"))
                operation = b.Text;
            equation.Text = value + " " + operation;
        }
        else if (b.Text == "√") {
            result.Text = Operators.sqrt(Double.Parse(result.Text)).ToString();
            value = Double.Parse(result.Text); }
        else if (b.Name == "plus_minus")
        {
            result.Text = (Double.Parse(result.Text) * (-1)).ToString();
            value = Double.Parse(result.Text) * (-1);
        }
        else{
            operation = b.Text;
            value = Double.Parse(result.Text);
            operation_pressed = true;
            equation.Text = value + " " + operation;
        }
    }

    private void button19_Click(object sender, EventArgs e)
    {
        equation.Text = "";
        switch (operation)
        {
            case "^":
                result.Text = Operators.pw(value, Double.Parse(result.Text)).ToString();
                break;
            case "+":
                result.Text = Operators.Add(value, Double.Parse(result.Text)).ToString();
                break;
            case "-":
                result.Text = Operators.Sub(value, Double.Parse(result.Text)).ToString();
                break;
            case "/":
                result.Text = Operators.Div(value, Double.Parse(result.Text)).ToString();
                break;
            case "*":
                result.Text = Operators.Mult(value, Double.Parse(result.Text)).ToString();

```

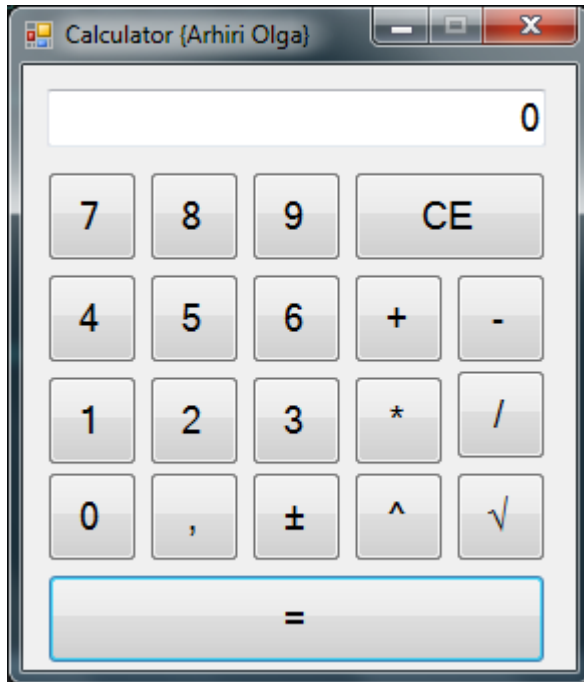
```

        break;
    default: break;
}
value = Double.Parse(result.Text);
operation = "";
flag = 1;
}
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    switch (e.KeyChar.ToString()) {
        case "1": one.PerformClick();
            break;
        case "2":
            two.PerformClick();
            break;
        case "3":
            three.PerformClick();
            break;
        case "4":
            four.PerformClick();
            break;
        case "5":
            five.PerformClick();
            break;
        case "6":
            six.PerformClick();
            break;
        case "7":
            seven.PerformClick();
            break;
        case "8":
            eight.PerformClick();
            break;
        case "9":
            nine.PerformClick();
            break;
        case "0":
            zero.PerformClick();
            break;
        case "*":
            times.PerformClick();
            break;
        case "/":
            div.PerformClick();
            break;
        case "+":
            add.PerformClick();
            break;
        case "-":
            sub.PerformClick();
            break;
        case ".":
            dec.PerformClick();
            break;
        case "=":
            equal.PerformClick();
            break;

        default: break;
    }
}
}
}

```

### Скриншот калькулятора:



### ***Выводы:***

В ходе данной лабораторной работы я ознакомилась с понятием графического интерфейса пользователя (GUI) и реализовала калькулятор на языке C#. Я убедилась, что графика даёт лучшее восприятие действительного положения вещей на экране, визуально богатую среду для передачи информации. Это необходимо для того, чтобы облегчить пользователю работу и изучение возможностей программы. Среда разработки Visual Studio представляет собой очень полезный инструмент для программиста, так как в ней можно с лёгкостью как писать код, так и разрабатывать графический интерфейс программы. С помощью среды разработки мне удалось расположить на экране такие элементы как button, text box, label и управлять ими в ходе работы программы. Существует большое количество параметров, которые могут видоизменить окно и его элементы в зависимости от идеи разработчика. На мой взгляд, грамотно составленное графическое приложение привлечет внимание пользователя, поэтому очень важно понять основные принципы работы с формами и объектами формы.