Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М. Машерова

Отчет

По модулю №8

« Графический интерфейс»

по учебной практике по программированию

Выполнил учащийся Войтеховская К.В.

группа 3ПОИС23 29.09.2025 г.

Проверил Алейников М.А.

30.09.2025 г.

Орша, 2025

**Задание.** Реализовать приложения используя концепцию ООП, разрешается реализация на мобильной платформе. Разработайте простой калькулятор, который выполняет основные математические операции, такие как сложение, вычитание, умножение и деление.

**Выполнение практического задания.** В ходе разработки калькулятора был создан графический интерфейс, который представлен на рисунке 1.

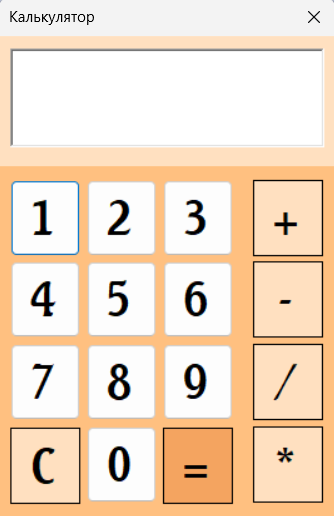


Рисунок 1 – Интерфейс калькулятора

Был разработан код программы представлен в листинге 1.

В нем класс Calculator управляет вводом, выполнением арифметических операций и обработкой ошибок. Для хранения состояния используются поля currentInput, firstOperand, currentOperation и isNewInput, которые отслеживают текущий ввод, первый операнд, выбранную операцию и признак нового ввода соответственно. Перечисление Operation определяет доступные математические операции: сложение, вычитание, умножение и деление.

Цифровые кнопки передают в калькулятор соответствующие символы через метод AppendNumber, а кнопки операций вызывают метод SetOperation с указанием типа операции. При нажатии кнопки вычисления активируется метод Calculate, который выполняет арифметическую операцию с учетом приоритетов и обрабатывает исключительные ситуации, такие как деление на ноль. Кнопка сброса вызывает метод Clear, восстанавливающий исходное состояние калькулятора.

Листинг 1. Код программы «Калькулятор»

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 WindowsFormsApp1

{

public partial class Form1 : Form

{

private Calculator calculator;

public Form1()

{

InitializeComponent();

calculator = new Calculator();

AttachEventHandlers();

}

private void AttachEventHandlers()

{

// Цифровые кнопки

yi.Click += (s, e) => AppendNumber("1");

er.Click += (s, e) => AppendNumber("2");

san.Click += (s, e) => AppendNumber("3");

si.Click += (s, e) => AppendNumber("4");

wu.Click += (s, e) => AppendNumber("5");

liu.Click += (s, e) => AppendNumber("6");

qi.Click += (s, e) => AppendNumber("7");

ba.Click += (s, e) => AppendNumber("8");

jiu.Click += (s, e) => AppendNumber("9");

ling.Click += (s, e) => AppendNumber("0");

// Операции

plus.Click += (s, e) => SetOperation(Operation.Add);

minus.Click += (s, e) => SetOperation(Operation.Subtract);

umnog.Click += (s, e) => SetOperation(Operation.Multiply);

delenie.Click += (s, e) => SetOperation(Operation.Divide);

// Специальные кнопки

clear.Click += (s, e) => ClearCalculator();

ravno.Click += (s, e) => CalculateResult();

}

private void AppendNumber(string number)

{

calculator.AppendNumber(number);

UpdateDisplay();

}

private void SetOperation(Operation operation)

{

calculator.SetOperation(operation);

UpdateDisplay();

}

private void CalculateResult()

{

calculator.Calculate();

UpdateDisplay();

}

private void ClearCalculator()

{

calculator.Clear();

UpdateDisplay();

}

private void UpdateDisplay()

{

richTextBox1.Text = calculator.DisplayValue;

}

}

// Перечисление для математических операций

public enum Operation

{

None,

Add,

Subtract,

Multiply,

Divide

}

// Основной класс калькулятора

public class Calculator

{

private string currentInput = "0";

private double? firstOperand = null;

private Operation currentOperation = Operation.None;

private bool isNewInput = true;

public string DisplayValue => currentInput;

public void AppendNumber(string number)

{

if (isNewInput)

{

currentInput = number;

isNewInput = false;

}

else

{

if (currentInput == "0" && number != "0")

{

currentInput = number;

}

else

{

currentInput += number;

}

}

}

public void SetOperation(Operation operation)

{

if (firstOperand == null)

{

firstOperand = double.Parse(currentInput);

}

else if (!isNewInput)

{

Calculate();

}

currentOperation = operation;

isNewInput = true;

}

public void Calculate()

{

if (firstOperand == null || currentOperation == Operation.None || isNewInput)

return;

double secondOperand = double.Parse(currentInput);

double result = 0;

try

{

switch (currentOperation)

{

case Operation.Add:

result = firstOperand.Value + secondOperand;

break;

case Operation.Subtract:

result = firstOperand.Value - secondOperand;

break;

case Operation.Multiply:

result = firstOperand.Value \* secondOperand;

break;

case Operation.Divide:

if (secondOperand == 0)

throw new DivideByZeroException("Деление на ноль невозможно");

result = firstOperand.Value / secondOperand;

break;

}

currentInput = FormatResult(result);

firstOperand = result;

currentOperation = Operation.None;

isNewInput = true;

}

catch (DivideByZeroException)

{

currentInput = "Ошибка";

Reset();

}

catch (Exception)

{

currentInput = "Ошибка";

Reset();

}

}

private string FormatResult(double result)

{

// Если число целое, отображение без десятичной части

if (result == Math.Truncate(result))

{

return result.ToString("0");

}

else

{

// Ограничение количества знаков после запятой

return result.ToString("0.##########");

}

}

public void Reset()

{

currentInput = "0";

firstOperand = null;

currentOperation = Operation.None;

isNewInput = true;

}

public void Clear()

{

Reset();

}

}

}

Для проверки корректности работы калькулятора было проведено тестирование. Для проверки правильности вычисления каждого действия были введены примеры: 484+5379, 782-862, 56:82, 972\*73. Результаты вычислений представлены на рисунке 2.

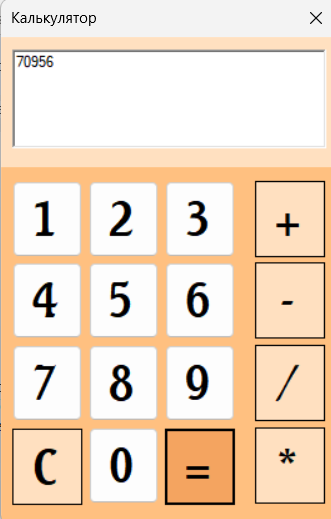
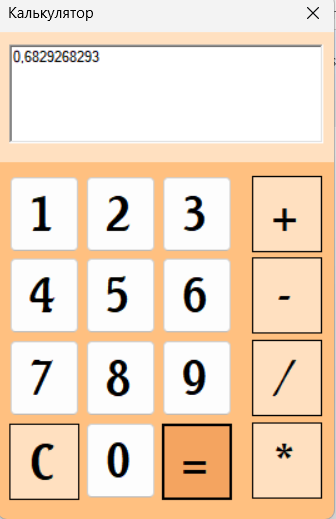
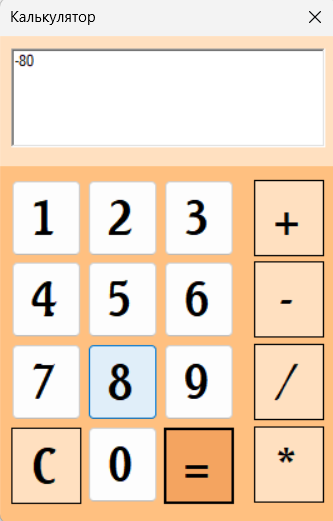
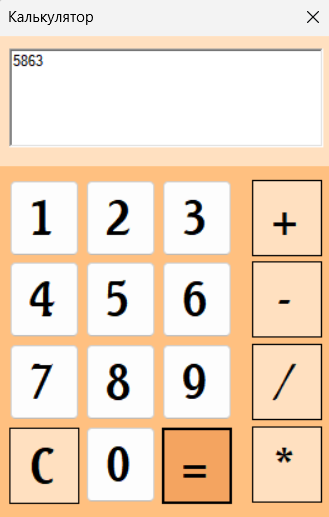


Рисунок 2 – Результаты вычисления калькулятора

Все полученные ответы соответствуют ожидаемым, что говорит о том, что калькулятор проводит правильные вычисления.