**Лабораторная работа №6**

**Создание графического интерфейса**

**Цель работы**

Создание графического интерфейса на языке программирования С#.

**Формулировка задачи**

Разработайте графический интерфейс для проекта из первой работы.

Определить класс Fraction, в котором производятся операции над обыкновенными дробями (сравнение, сложение, умножение, деление, сокращение).

**Листинг программы Program.cs на языке С#**

namespace lab6\_oop

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form2());

}

}

}

**Листинг программы Form1.cs на языке С#**

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 lab6\_oop

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Add\_Click(object sender, EventArgs e)

{

int num1, num2;

int dnum1=1, dnum2=1;

if (int.TryParse(Numerals1.Text,out num1) && int.TryParse(Numerals2.Text, out num2) &&

(int.TryParse(Denumerals1.Text, out dnum1) || Denumerals1.Text =="") &&

(int.TryParse(Denumerals2.Text, out dnum2) || Denumerals2.Text == ""))

{

if (Denumerals2.Text == "")

dnum2 = 1;

if (Denumerals1.Text == "")

dnum1 = 1;

Fraction Frac1 = new Fraction(num1, dnum1);

Fraction Frac2 = new Fraction(num2, dnum2);

Frac1 = Frac1.add(Frac2);

Numerals.Text = Frac1.getNumerals().ToString();

Denumerals.Text = Frac1.getDenominator().ToString();

isExist.Text = Frac1.getException().ToString() + ":Ex";

}

else isExist.Text = "error";

}

private void Exist\_Click(object sender, EventArgs e)

{

int num1, num2;

int dnum1 = 1, dnum2 = 1;

if (int.TryParse(Numerals1.Text, out num1) && int.TryParse(Numerals2.Text, out num2) &&

(int.TryParse(Denumerals1.Text, out dnum1) || Denumerals1.Text == "") &&

(int.TryParse(Denumerals2.Text, out dnum2) || Denumerals2.Text == ""))

{

if (Denumerals2.Text == "")

dnum2 = 1;

if (Denumerals1.Text == "")

dnum1 = 1;

Fraction Frac1 = new Fraction(num1, dnum1);

Fraction Frac2 = new Fraction(num2, dnum2);

isExist.Text = Frac1.getException().ToString() + ":Ex 1" + Frac2.getException().ToString() + ":Ex 2";

}

else if (int.TryParse(Numerals1.Text, out num1) &&

(int.TryParse(Denumerals1.Text, out dnum1) || Denumerals1.Text == ""))

{

Fraction Frac1 = new Fraction(num1, dnum1);

isExist.Text = Frac1.getException().ToString() + ":Ex 1";

}

else isExist.Text = "error";

}

private void Sub\_Click(object sender, EventArgs e)

{

int num1, num2;

int dnum1 = 1, dnum2 = 1;

if (int.TryParse(Numerals1.Text, out num1) && int.TryParse(Numerals2.Text, out num2) &&

(int.TryParse(Denumerals1.Text, out dnum1) || Denumerals1.Text == "") &&

(int.TryParse(Denumerals2.Text, out dnum2) || Denumerals2.Text == ""))

{

if (Denumerals2.Text == "")

dnum2 = 1;

if (Denumerals1.Text == "")

dnum1 = 1;

Fraction Frac1 = new Fraction(num1, dnum1);

Fraction Frac2 = new Fraction(num2, dnum2);

Frac1 = Frac1.sub(Frac2);

Numerals.Text = Frac1.getNumerals().ToString();

Denumerals.Text = Frac1.getDenominator().ToString();

isExist.Text = Frac1.getException().ToString() + ":Ex";

}

else isExist.Text = "error";

}

private void Mult\_Click(object sender, EventArgs e)

{

int num1, num2;

int dnum1 = 1, dnum2 = 1;

if (int.TryParse(Numerals1.Text, out num1) && int.TryParse(Numerals2.Text, out num2) &&

(int.TryParse(Denumerals1.Text, out dnum1) || Denumerals1.Text == "") &&

(int.TryParse(Denumerals2.Text, out dnum2) || Denumerals2.Text == ""))

{

if (Denumerals2.Text == "")

dnum2 = 1;

if (Denumerals1.Text == "")

dnum1 = 1;

Fraction Frac1 = new Fraction(num1, dnum1);

Fraction Frac2 = new Fraction(num2, dnum2);

Frac1 = Frac1.mult(Frac2);

Numerals.Text = Frac1.getNumerals().ToString();

Denumerals.Text = Frac1.getDenominator().ToString();

isExist.Text = Frac1.getException().ToString() + ":Ex";

}

else isExist.Text = "error";

}

private void Div\_Click(object sender, EventArgs e)

{

int num1, num2;

int dnum1 = 1, dnum2 = 1;

if (int.TryParse(Numerals1.Text, out num1) && int.TryParse(Numerals2.Text, out num2) &&

(int.TryParse(Denumerals1.Text, out dnum1) || Denumerals1.Text == "") &&

(int.TryParse(Denumerals2.Text, out dnum2) || Denumerals2.Text == ""))

{

if (Denumerals2.Text == "")

dnum2 = 1;

if (Denumerals1.Text == "")

dnum1 = 1;

Fraction Frac1 = new Fraction(num1, dnum1);

Fraction Frac2 = new Fraction(num2, dnum2);

Frac1 = Frac1.div(Frac2);

Numerals.Text = Frac1.getNumerals().ToString();

Denumerals.Text = Frac1.getDenominator().ToString();

isExist.Text = Frac1.getException().ToString() + ":Ex";

}

else isExist.Text = "error";

}

private void Comp\_Click(object sender, EventArgs e)

{

int num1, num2;

int dnum1 = 1, dnum2 = 1;

if (int.TryParse(Numerals1.Text, out num1) && int.TryParse(Numerals2.Text, out num2) &&

(int.TryParse(Denumerals1.Text, out dnum1) || Denumerals1.Text == "") &&

(int.TryParse(Denumerals2.Text, out dnum2) || Denumerals2.Text == ""))

{

if (Denumerals2.Text == "")

dnum2 = 1;

if (Denumerals1.Text == "")

dnum1 = 1;

Fraction Frac1 = new Fraction(num1, dnum1);

Fraction Frac2 = new Fraction(num2, dnum2);

isExist.Text = Frac1.Comparison(Frac2);;

}

else isExist.Text = "error";

}

}

}

**Листинг программы Fraction.cs на языке С#**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab6\_oop

{

class Fraction

{

private int \_numerals = 0; //a-числитель

private int \_denominator = 1; //b-знаменатель

/\*public static void Main()

{

int \_numerals=0;

int \_denominator=1;

}\*/

public Fraction()

{

this.\_numerals = 0;

this.\_denominator = 1;

}

public Fraction(Fraction f)

{

this.\_numerals = (f.\_numerals);

this.\_denominator = (f.\_denominator);

}

public Fraction(int numerals, int denominator)

{

this.\_numerals=(numerals);

this.\_denominator=(denominator);

}

public Fraction add(Fraction buf)

{

Fraction Sum = new Fraction();

Sum.\_numerals = \_numerals \* buf.\_denominator + \_denominator \* buf.\_numerals;

Sum.\_denominator = \_denominator \* buf.\_denominator;

//Sum.Reduction();

return Sum;

}

public Fraction sub(Fraction buf)

{

Fraction Sum = new Fraction();

// Приводим обе дроби к общему знаменателю

Sum.\_numerals = \_numerals \* buf.\_denominator - \_denominator \* buf.\_numerals;

Sum.\_denominator = \_denominator \* buf.\_denominator;

Sum.Reduction();

return Sum;

}

public Fraction mult(Fraction buf)

{

Fraction Mult = new Fraction();

// Числитель конечной дроби равен произведению числителя первой и второй дроби

Mult.\_numerals = \_numerals \* buf.\_numerals;

Mult.\_denominator = \_denominator \* buf.\_denominator;

Mult.Reduction();

return Mult;

}

public Fraction div(Fraction buf)

{

Fraction Div = new Fraction();

// Числитель конечной дроби равен произведению числителя первой дроби и знаменателя второй

Div.\_numerals = \_numerals \* buf.\_denominator;

Div.\_denominator = \_denominator \* buf.\_numerals;

// Возвращаем дробь поделенную на другую

Div.Reduction();

return Div;

}

public Fraction Reduction()

{

int containers = \_numerals;

if (\_numerals > \_denominator)

containers = \_denominator;

// Ищем общий делитель для числителя и знаменателя, чтобы сократить дробь

for (int i = containers; i >1; i--)

{

if ((\_numerals % i) == 0 && (\_denominator % i) == 0)

{

\_numerals = \_numerals / i;

\_denominator = \_denominator / i;

containers = containers / i;

}

}

//Fraction Red(\_numerals, \_denominator);

return this;

}

public string Comparison(Fraction buf)

{

Fraction d1 = new Fraction(), d2 = buf;

// Приводим обе дроби к общему знаменателю

d1.\_numerals = \_numerals \* buf.\_denominator;

d2.\_numerals = buf.\_numerals \* \_denominator;

if (!(d1.getException() || d2.getException()))

return "exist";

if (d1.\_numerals > d2.\_numerals)

return "f1>f2";

if (d1.\_numerals < d2.\_numerals)

return "f1<f2";

if (d1.\_numerals == d2.\_numerals)

return "f1=f2";

return "exist";

}

public int getNumerals()

{

return \_numerals;

}

public int getDenominator()

{

return \_denominator;

}

public bool getException()

{

if (\_denominator == 0)

return false;

return true;

}

public void setNumerals(int numerals)

{

\_numerals = numerals;

}

public void setDenominator(int denominator)

{

\_denominator = denominator;

}

}

}

**Листинг программы Form1.cs на языке С#**

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 lab6\_oop

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Hide();

Form form = new Form1();

form.Show();

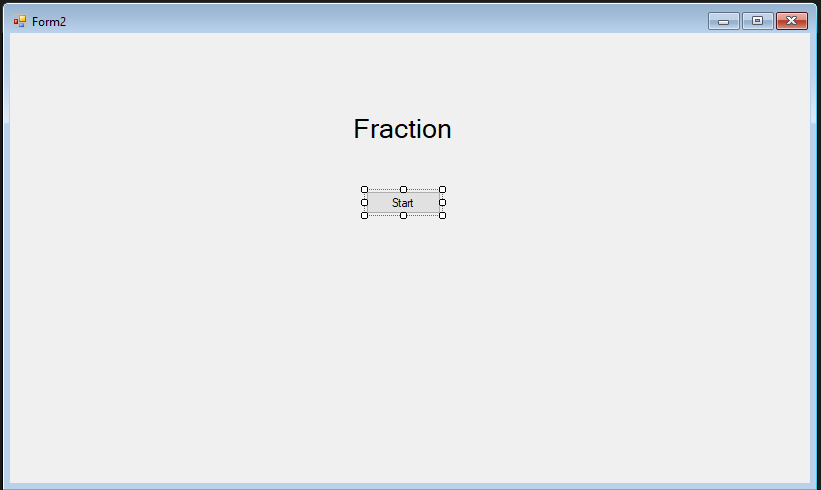
}

}

}

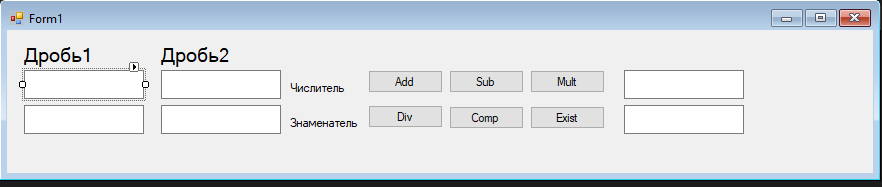
**Результат работы**

Первая часть программы приветствует пользователя и просит разрешение на продолжение калькулятора.



*Рисунок 1. Окно программы Form2*

Вторая часть программы производит математические и логические вычисления над введёнными дробями и выдают их в соответствующем окне.



*Рисунок 2. Окно программы Form1*

**Заключение**

В процессе выполнения лабораторной работы №6 мы ознакомились с инструментами создания графического интерфейса на языке С#.

Авторы: Преподаватель: Тарасов

Шутенко О.Р. Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_ Дата: \_\_.\_\_.2020