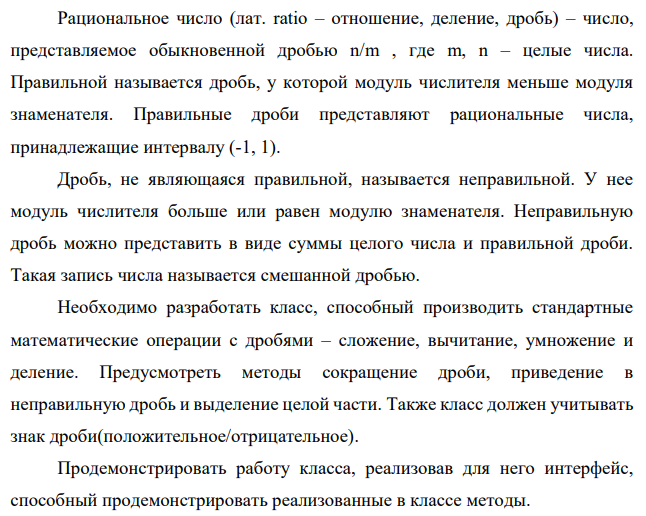
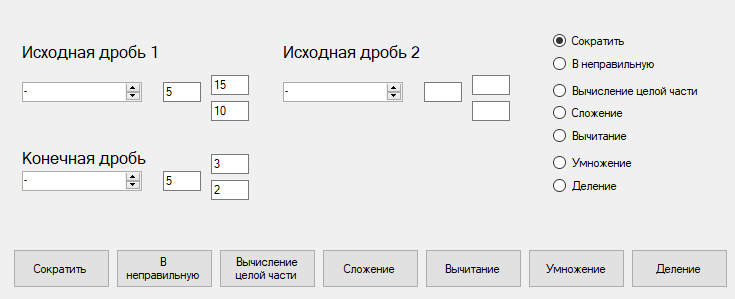
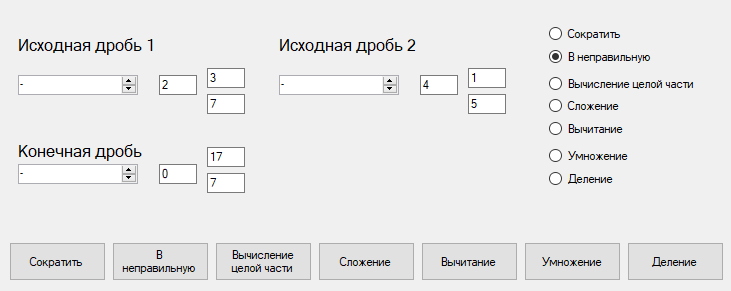
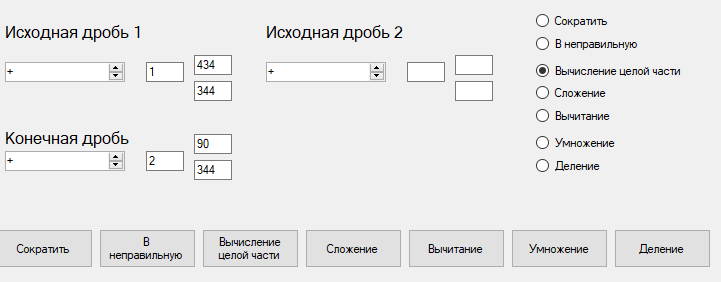
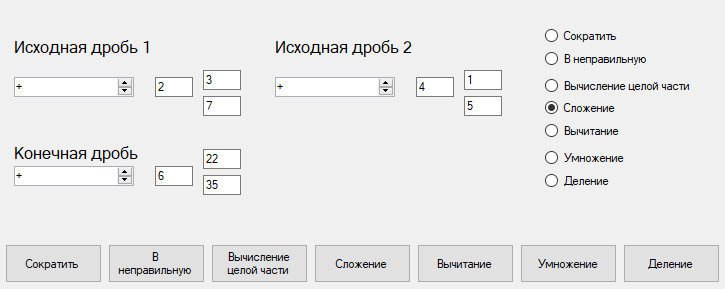
Практическая работа №8

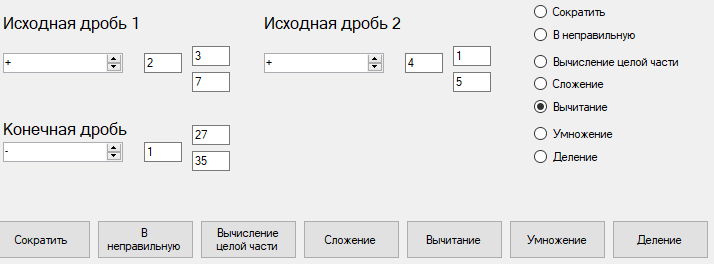


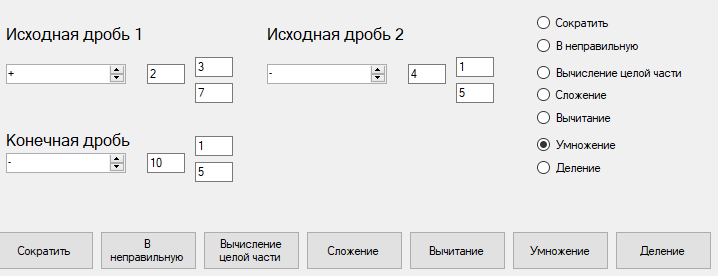


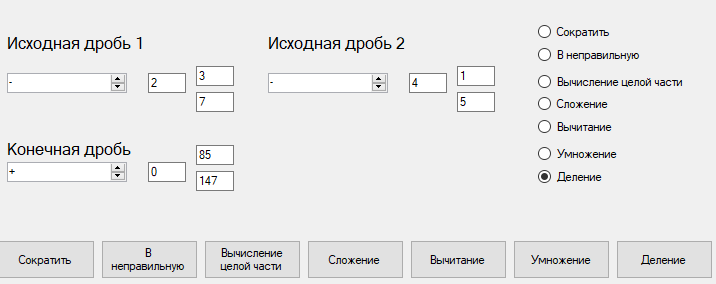












**Fraction.cs**

class Fraction

{

public int sign, integer, numeration, denominator;

public Fraction(int n\_sign, int n\_integer, int n\_numeration, int n\_denominator)//конструктор класса

{

sign = n\_sign;

integer = n\_integer;

numeration = n\_numeration;

denominator = n\_denominator;

}

private int NOD(int A, int B)

{

while (A != B)

{

if (A > B)

{

A = A - B;

}

else

{

B = B - A;

}

}

return A;

}

public void Reduction()// Сокращение дроби

{

int k = NOD(numeration, denominator);

if (k != 1)

{

numeration = numeration / k;

denominator = denominator / k;

}

}

public void Incorrect\_fraction()// приведение в неправильную дробь

{

numeration = integer \* denominator + numeration;

integer = 0;

}

public void Correct\_fraction()//Создание правильной дроби и выделение целой части

{

int k = integer;

integer = numeration / denominator;

numeration = numeration - integer \* denominator;

integer = integer + k;

}

public void Addition(Fraction d)//Сложение дробей

{

integer = sign \* integer + d.sign \* d.integer;

int k1 = sign \* numeration \* d.denominator + d.sign \* d.numeration \* denominator;

denominator = denominator \* d.denominator;

numeration = k1;

if (integer < 0)

{

integer = integer \* (-1);

sign = -1;

}

}

public void Addition2(Fraction d)

{

}

public void Subtrachion(Fraction d)//Вычитание дробей

{

Incorrect\_fraction();

d.Incorrect\_fraction();

int k1 = (sign \* numeration) \* d.denominator - (d.sign \* (d.numeration)) \* denominator;

numeration = k1;

denominator = d.denominator \* denominator;

if (numeration < 0)

{

numeration = numeration \* (-1);

sign = sign \* (-1);

}

Correct\_fraction();

Reduction();

}

public void Multiplication(Fraction d)//Умножение дробей

{

Incorrect\_fraction();

d.Incorrect\_fraction();

int t = sign \* numeration \* d.sign \* d.numeration;

int t2 = denominator \* d.denominator;

numeration = t;

denominator = t2;

if (numeration < 0)

{

numeration = numeration \* (-1);

sign = -1;

}

else

{

sign = 1;

}

Correct\_fraction();

Reduction();

}

public void Division(Fraction d)//Деление дробей

{

Incorrect\_fraction();

d.Incorrect\_fraction();

int t = sign \* numeration \* d.sign \* d.denominator;

int t2 = d.numeration \* denominator;

numeration = t;

denominator = t2;

if (numeration < 0)

{

numeration = numeration \* (-1);

sign = -1;

}

else

{

sign = 1;

}

Correct\_fraction();

Reduction();

}

}

**Form1.cs**

public partial class Form1 : Form

{

Fraction f,f2;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

domainUpDown1.SelectedIndex = 0;

domainUpDown2.SelectedIndex = 0;

domainUpDown3.SelectedIndex = 0;

}

public void Print()

{

if (f.sign > 0)

{

domainUpDown2.SelectedIndex = 0;

}

else

{

domainUpDown2.SelectedIndex = 1;

}

textBox4.Text = f.integer.ToString();

textBox5.Text = f.numeration.ToString();

textBox6.Text = f.denominator.ToString();

}

public void ReceiveF()

{

int sign = 1;

if (domainUpDown1.SelectedIndex == 1)

{

sign = -1;

}

else if (domainUpDown1.SelectedIndex == 0)

{

sign = 1;

}

int integer = Convert.ToInt32(textBox1.Text);

int numerator = Convert.ToInt32(textBox2.Text);

int denominator = Convert.ToInt32(textBox3.Text);

f = new Fraction(sign, integer, numerator, denominator);

}

public void ReceiveF2()

{

int sign2 = 1;

if (domainUpDown3.SelectedIndex == 1)

{

sign2 = -1;

}

else if (domainUpDown3.SelectedIndex == 0)

{

sign2 = 1;

}

int integer2 = Convert.ToInt32(textBox9.Text);

int numerator2 = Convert.ToInt32(textBox8.Text);

int denominator2 = Convert.ToInt32(textBox7.Text);

f2 = new Fraction(sign2, integer2, numerator2, denominator2);

}

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

{

ReceiveF();

f.Reduction();

Print();

}

private void button2\_Click(object sender, EventArgs e)

{

ReceiveF();

f.Incorrect\_fraction();

Print();

}

private void button3\_Click(object sender, EventArgs e)

{

ReceiveF();

f.Correct\_fraction();

Print();

}

private void button4\_Click(object sender, EventArgs e)

{

ReceiveF();

ReceiveF2();

f.Addition(f2);

Print();

}

private void button5\_Click(object sender, EventArgs e)

{

ReceiveF();

ReceiveF2();

f.Subtrachion(f2);

Print();

}

private void button6\_Click(object sender, EventArgs e)

{

ReceiveF();

ReceiveF2();

f.Multiplication(f2);

Print();

}

private void button7\_Click(object sender, EventArgs e)

{

ReceiveF();

ReceiveF2();

f.Division(f2);

Print();

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton1.Checked)

{

ReceiveF();

f.Reduction();

Print();

}

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton2.Checked)

{

ReceiveF();

f.Incorrect\_fraction();

Print();

}

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton3.Checked)

{

ReceiveF();

f.Correct\_fraction();

Print();

}

}

private void radioButton4\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton4.Checked)

{

ReceiveF();

ReceiveF2();

f.Addition(f2);

Print();

}

}

private void radioButton5\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton5.Checked)

{

ReceiveF();

ReceiveF2();

f.Subtrachion(f2);

Print();

}

}

private void radioButton6\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton6.Checked)

{

ReceiveF();

ReceiveF2();

f.Multiplication(f2);

Print();

}

}

private void radioButton7\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton7.Checked)

{

ReceiveF();

ReceiveF2();

f.Division(f2);

Print();

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

Ссылка на гитхаб:

<https://github.com/Alexandrov911/Practical.8-2022.git>