

Практическая работа №8

Рациональное число (лат. ratio – отношение, деление, дробь) – число, представляемое обыкновенной дробью $\frac{n}{m}$, где m, n – целые числа. Правильной называется дробь, у которой модуль числителя меньше модуля знаменателя. Правильные дроби представляют рациональные числа, принадлежащие интервалу $(-1, 1)$.

Дробь, не являющаяся правильной, называется неправильной. У нее модуль числителя больше или равен модулю знаменателя. Неправильную дробь можно представить в виде суммы целого числа и правильной дроби. Такая запись числа называется смешанной дробью.

Необходимо разработать класс, способный производить стандартные математические операции с дробями – сложение, вычитание, умножение и деление. Предусмотреть методы сокращения дроби, приведение в неправильную дробь и выделение целой части. Также класс должен учитывать знак дроби (положительное/отрицательное).

Продемонстрировать работу класса, реализовав для него интерфейс, способный продемонстрировать реализованные в классе методы.

Исходная дробь 1		Исходная дробь 2		<input checked="" type="radio"/> Сократить	
-	5	15	-		<input type="radio"/> В неправильную
		10			<input type="radio"/> Вычисление целой части
Конечная дробь		3	<input type="radio"/> Сложение		
-	5	2	<input type="radio"/> Вычитание		
			<input type="radio"/> Умножение		
			<input type="radio"/> Деление		
Сократить		В неправильную		Вычисление целой части	
Сложение		Вычитание		Умножение	
Деление					

Исходная дробь 1

-

2

3

7

Исходная дробь 2

-

4

1

5

☐ Сократить
☒ В неправильную
☐ Вычисление целой части
☐ Сложение
☐ Вычитание
☐ Умножение
☐ Деление

Конечная дробь

-

0

17

7

Сократить

В неправильную

Вычисление целой части

Сложение

Вычитание

Умножение

Деление

Исходная дробь 1

+

1

434

344

Исходная дробь 2

+

☐ Сократить
☐ В неправильную
☒ Вычисление целой части
☐ Сложение
☐ Вычитание
☐ Умножение
☐ Деление

Конечная дробь

+

2

90

344

Сократить

В неправильную

Вычисление целой части

Сложение

Вычитание

Умножение

Деление

Исходная дробь 1

+

2

3

7

Исходная дробь 2

+

4

1

5

☐ Сократить
☐ В неправильную
☐ Вычисление целой части
☒ Сложение
☐ Вычитание
☐ Умножение
☐ Деление

Конечная дробь

+

6

22

35

Сократить

В неправильную

Вычисление целой части

Сложение

Вычитание

Умножение

Деление

Исходная дробь 1	Исходная дробь 2	<input type="radio"/> Сократить <input type="radio"/> В неправильную <input type="radio"/> Вычисление целой части <input type="radio"/> Сложение <input checked="" type="radio"/> Вычитание <input type="radio"/> Умножение <input type="radio"/> Деление
<div>+</div> <div>2</div> <div>3</div> <div>7</div>	<div>+</div> <div>4</div> <div>1</div> <div>5</div>	
Конечная дробь		
<div>-</div> <div>1</div> <div>27</div> <div>35</div>		
Сократить	В неправильную	Вычисление целой части
Сложение	Вычитание	Умножение
Деление		

Исходная дробь 1	Исходная дробь 2	<input type="radio"/> Сократить <input type="radio"/> В неправильную <input type="radio"/> Вычисление целой части <input type="radio"/> Сложение <input type="radio"/> Вычитание <input checked="" type="radio"/> Умножение <input type="radio"/> Деление
<div>+</div> <div>2</div> <div>3</div> <div>7</div>	<div>-</div> <div>4</div> <div>1</div> <div>5</div>	
Конечная дробь		
<div>-</div> <div>10</div> <div>1</div> <div>5</div>		
Сократить	В неправильную	Вычисление целой части
Сложение	Вычитание	Умножение
Деление		

Исходная дробь 1	Исходная дробь 2	<input type="radio"/> Сократить <input type="radio"/> В неправильную <input type="radio"/> Вычисление целой части <input type="radio"/> Сложение <input type="radio"/> Вычитание <input type="radio"/> Умножение <input checked="" type="radio"/> Деление
<div>-</div> <div>2</div> <div>3</div> <div>7</div>	<div>-</div> <div>4</div> <div>1</div> <div>5</div>	
Конечная дробь		
<div>+</div> <div>0</div> <div>85</div> <div>147</div>		
Сократить	В неправильную	Вычисление целой части
Сложение	Вычитание	Умножение
Деление		

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>