**Тема 3 «Использование подпрограмм для обработки  
простых типов данных»**

**Цель работы:** получить практические навыки разработки и использования подпрограмм для обработки простых типов данных.

***Задание 1.*** Создание дополнительного класса и разработка подпрограмм (методов) класса.

Подпрограммы расположить в отдельном файле (модуле).

Методы класса должны иметь уровень доступа **public**.

Подпрограммы должны получать исходные данные через параметры (для динамических методов – через экземпляр класса), вычислять результат и передавать его в место вызова (именно передавать, а не выводить на экран!).

Метод с результатом передает одно вычисленное значение (результат) через оператор **return**. Методы могут передавать и несколько результатов через параметры-переменные.

***Задание 2.*** Тестирование подпрограмм с помощью тест-кейсов.

* Для созданного модуля создать проект **UnitTestsProject**.
* Создать 2 функции тестирования.
* Запустить тесты и продемонстрировать успешное и неуспешное тестирование.

***Задание 3.*** Создание пользовательского интерфейса (модуля формы).

Необходимо настроить внешний вид и поведение компонентов формы и создать методы обработки событий работы пользователя с элементами пользовательского интерфейса. Модуль формы должен содержать код, демонстрирующий вызов подпрограмм из дополнительного модуля.

**Вариант 5.**

1. Разработать подпрограммы для нахождения наибольшего общего делителя двух натуральных чисел и наименьшего общего кратного.
2. Используя созданные подпрограммы, определить общий делитель или общее кратное (по выбору пользователя) для двух натуральных чисел.

**Выполнение задания**

1. **Разработка подпрограмм**
   1. **Разработка подпрограммы Class**
      1. **Математическая модель**

**Дано:** два натуральных числа - a и b

**ОДЗ:** дополнительных ограничений нет

* + 1. **Прототип**

/// <param name="a">Первое число</param>

/// <param name="b">Второе число</param>

/// <returns> Возвращает оба числа и хранит их для дальнейшего использования</returns>

* + 1. **Код подпрограммы** **Class**

namespace team\_3

{

public class Class

{

protected int a;

protected int b;

public Class()

{

}

public Class(int A, int B)

{

a = A;

b = B;

}

public Class(Class cl)

{

a = cl.a;

b = cl.b;

}

}

}

* 1. **Разработка подпрограммы Class\_nod (от класса Class)**
     1. **Математическая модель**

**Дано:** два натуральных числа - a и b

**Найти:** nod – натуральное число

**Связь:** см. Рис.

**ОДЗ:** дополнительных ограничений нет

* + 1. **Прототип**

/// <param name="a">Первое число</param>

/// <param name="b">Второе число</param>

/// <returns> Возвращает наибольший общий делитель</returns>

public int calculate\_nod()

* + 1. **Код подпрограммы** **Class\_nod**

namespace team\_3

{

public class Class\_nod : Class

{

public Class\_nod(int num1, int num2) : base(num1, num2)

{

}

public int calculate\_nod()

{

int x = a;

int y = b;

while (y != 0)

{

int temp = y;

y = x % y;

x = temp;

}

return x;

}

}

}

* + 1. **Алгоритм** **Class\_nod**
  1. **Разработка подпрограммы Class\_nok (от класса Class\_nod)**
     1. **Математическая модель**

**Дано:** два натуральных числа - a и b

**Найти:** nok – натуральное число

**Связь:** см. Рис.

**ОДЗ:** дополнительных ограничений нет

* + 1. **Прототип**

/// <param name="a">Первое число</param>

/// <param name="b">Второе число</param>

/// <returns> Возвращает наименьшее общее кратное </returns>

public int calculate\_nok()

* + 1. **Код подпрограммы** **Class\_nok**

namespace team\_3

{

public class Class\_nok : Class\_nod

{

public Class\_nok (int x, int y):base(x, y)

{

}

public int calculate\_nok()

{

int gcd = calculate\_nod();

return (a \* b) / gcd;

}

}

}

* + 1. **Алгоритм** **Class\_no**
    2. **Схема работы подпрограммы Perimeter**

Return perimeter

Начало

Конец

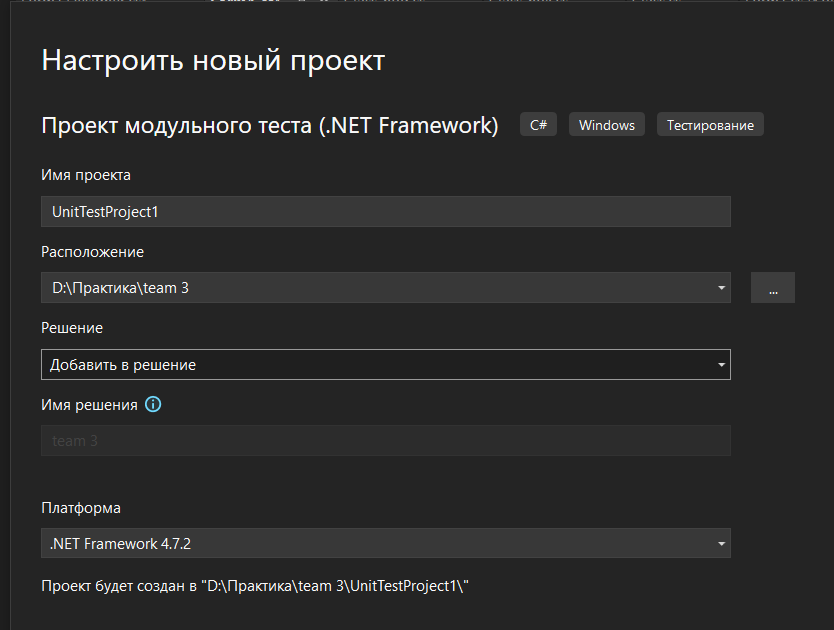
for(int i = 0; i < count; i++)

perimeter += Point.Distance(vertexes[i], vertexes[(i + 1) % count])

double perimeter = 0;

**Рис. 2**

1. **Тестирование подпрограмм с помощью тест-кейсов**

****

**Рис. 2.1 Создание проекта теста**

* 1. **Код модуля для тестов**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using team\_3;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod\_nok()

{

// исходные данные для теста

int first = 12;

int second = 18;

// ожидаемый результат

int experted = 36;

Class\_nok nk = new Class\_nok(first, second);

//вызов тестируемой функции

int actual = nk.calculate\_nok();

// сравнение результатов

Assert.AreEqual(experted, actual, 0.0, "Ожидаемый результат не получен!");

}

[TestMethod]

public void TestMethod\_nod()

{

// исходные данные для теста

int first = 12;

int second = 18;

// ожидаемый результат

int experted = 6;

Class\_nod nd = new Class\_nod(first, second);

//вызов тестируемой функции

int actual = nd.calculate\_nod();

// сравнение результатов

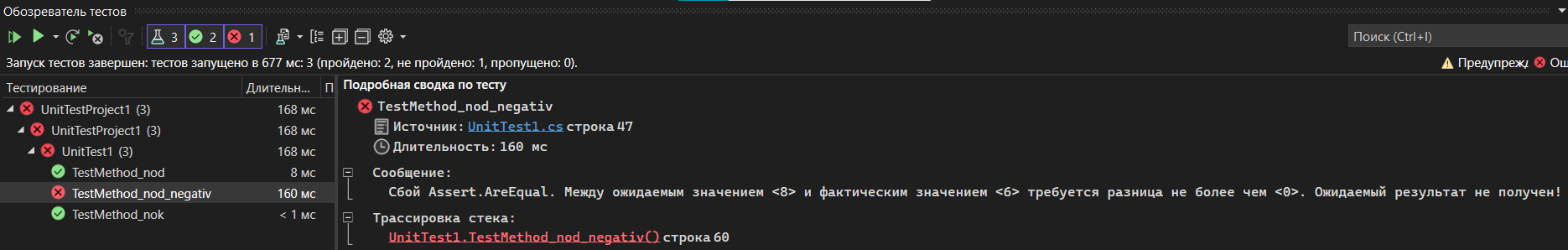
Assert.AreEqual(experted, actual, 0.0, "Ожидаемый результат не получен!");

}

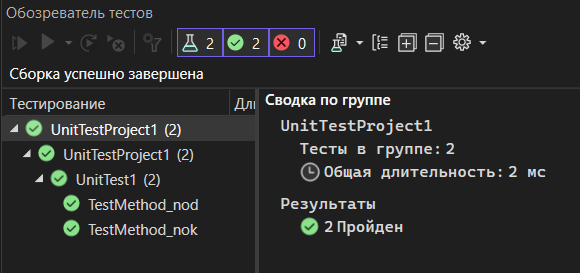
}

}

* 1. **Результаты тестов**



**Рис. 2.2.1 Работа тестов подпрограммы**

****

**Рис. 2.2.1Пройденные тесты подпрограммы**

1. **Создание пользовательского интерфейса**
   1. **Проектирование пользовательского интерфейса**

Внешний вид проекта представлен на рис. 3.1, значения измененных свойств компонентов представлены в таблице 3.1.

radioButton2

radioButton1

numericUpDown2

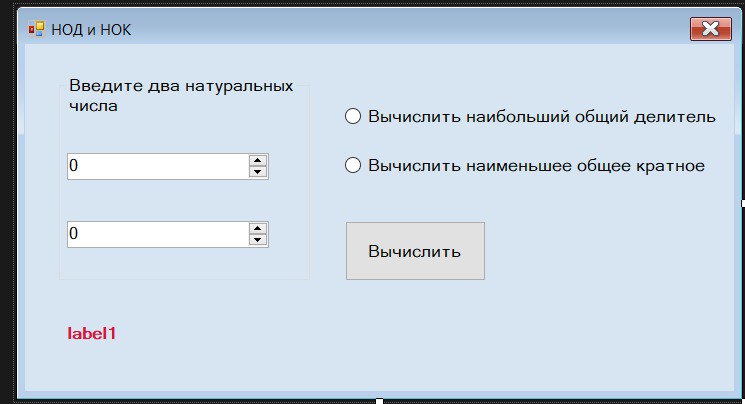
numericUpDown1

groupBox1

Form1

button1

label1



**Рис. 3.1**

* 1. **Свойства компонентов формы**

*Таблица 3.1.* Значения свойств, установленные на этапе конструирования интерфейса

|  |  |  |
| --- | --- | --- |
| **Название компонента** | **Свойства** | **Значения** |
| form1 | Text | «Вычисление по формулам» |
| BackColor | GradientInactiveCaption |
| Font | Гарнитура: Microsoft Sans Serif |
| Размер: 10,2 |
| Начертание: обычный |
| label1 | Text | «» |
| Font | Размер: 12 |
| FontColor | Crimson |
| button1 | Text | «Вычислить» |
| Enabled | False |

* 1. **Сценарий использования программы**
     1. Ввод двух натуральных чисел
     2. Выбор вычисления НОК или НОД
     3. Вычисления выводятся на экран.
     4. Возможность вернуться к пунктам 1.1.1, 1.1.2.
     5. Завершение работы программы.
  2. **Код модуля Form1.cs**

using System;

using System.Windows.Forms;

namespace team\_3

{

public partial class Form1 : Form

{

int res;

public Form1()

{

InitializeComponent();

radioButton1.Checked = false;

radioButton2.Checked = false;

label1.Text = null;

}

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

{

Class\_nod nod = new Class\_nod(((int)numericUpDown1.Value), ((int)numericUpDown2.Value));

res = nod.calculate\_nod();

label1.Text = res.ToString();

label1.Text = string.Format("НОД:{0}", res);

}

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

{

Class\_nok nok = new Class\_nok(((int)numericUpDown1.Value), ((int)numericUpDown2.Value));

res = nok.calculate\_nok();

label1.Text = res.ToString();

label1.Text = string.Format("НОК:{0}", res);

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

if (radioButton1.Checked)

{

Class\_nod nod = new Class\_nod(((int)numericUpDown1.Value), ((int)numericUpDown2.Value));

res = nod.calculate\_nod();

label1.Text = res.ToString();

label1.Text = string.Format("НОД:{0}", res);

}

else if (radioButton2.Checked) {

Class\_nok nok = new Class\_nok(((int)numericUpDown1.Value), ((int)numericUpDown2.Value));

res = nok.calculate\_nok();

label1.Text = res.ToString();

label1.Text = string.Format("НОК:{0}", res);

}

}

private void numericUpDown2\_ValueChanged(object sender, EventArgs e)

{

if (radioButton1.Checked)

{

Class\_nod nod = new Class\_nod(((int)numericUpDown1.Value), ((int)numericUpDown2.Value));

res = nod.calculate\_nod();

label1.Text = res.ToString();

label1.Text = string.Format("НОД:{0}", res);

}

else if (radioButton2.Checked)

{

Class\_nok nok = new Class\_nok(((int)numericUpDown1.Value), ((int)numericUpDown2.Value));

res = nok.calculate\_nok();

label1.Text = res.ToString();

label1.Text = string.Format("НОК:{0}", res);

}

}

}

}

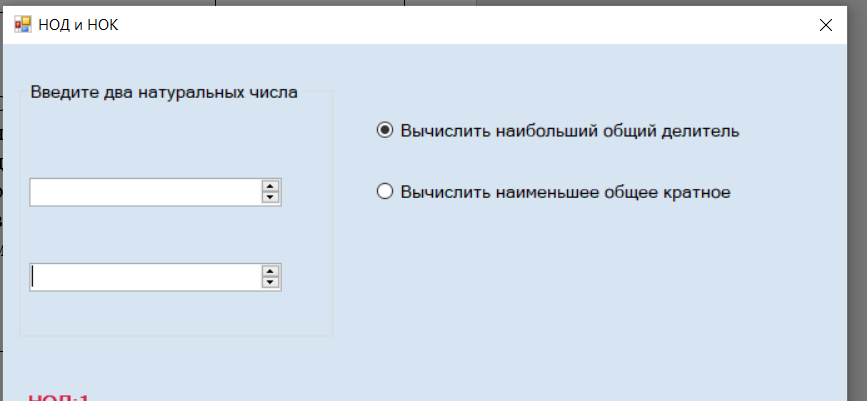
**Рис.3.2 Метод button2\_Click**

**3.3. Протокол испытаний:**

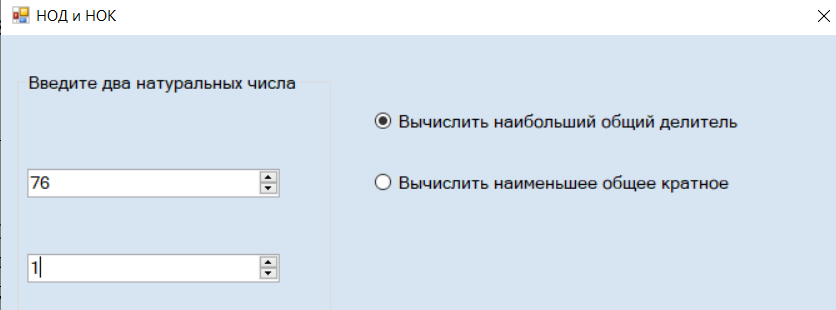
*Таблица 3.3.* Протокол испытаний

|  |  |  |  |
| --- | --- | --- | --- |
| **Проверяемые требования** | **Вариант входных данных** | **Ожидаемые результаты** | **Фактические результаты** |
| Способность контроля недопустимых значений в поля ввода | Входные данные 1  a b  Входные данные 2  76b 1  Входные данные 3  1,4 1 | Выходные данные 1  Null null  Выходные данные 2  76 1  Выходные данные 3  14 1 | Соответствуют ожидаемым результатам  (См. Рисунки 3.3.1-3.3.3) |
| Проверить способность программы вычислять наибольший общий делитель | Входные данные 1  25 5  Входные данные 2  455 56  Входные данные 3  5734 26 | Результат 1 5  Результат 2  7  Результат 3  2 | Соответствуют ожидаемым результатам  (См. Рисунки 3.3.4 – 3.3.6) |
| Проверить способность программы вычислять наименьшее общее кратное | Входные данные 1  45 26  Входные данные 2  85 4  Входные данные 3  78 3 | Результат 1 1170  Результат 2  340  Результат 3  3 | Соответствуют ожидаемым результатам  (См. Рисунки 3.3.7 – 3.3.9) |

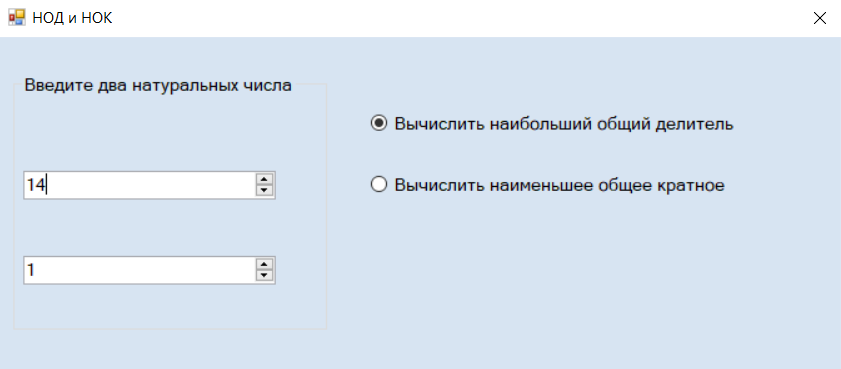
* 1. **Результаты тестирования проекта:**

****

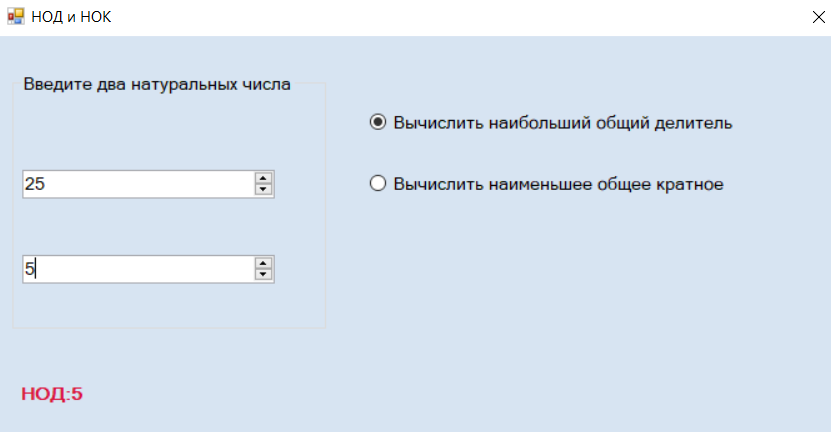
**Рис 3.3.1**

****

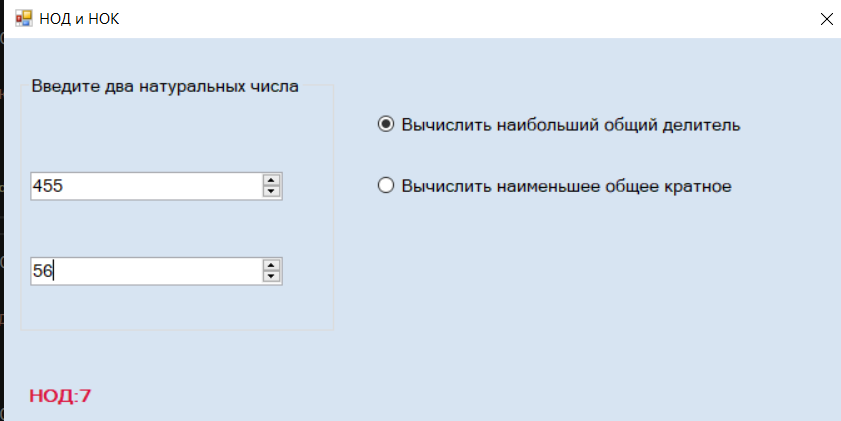
**Рис 3.3.2**

****

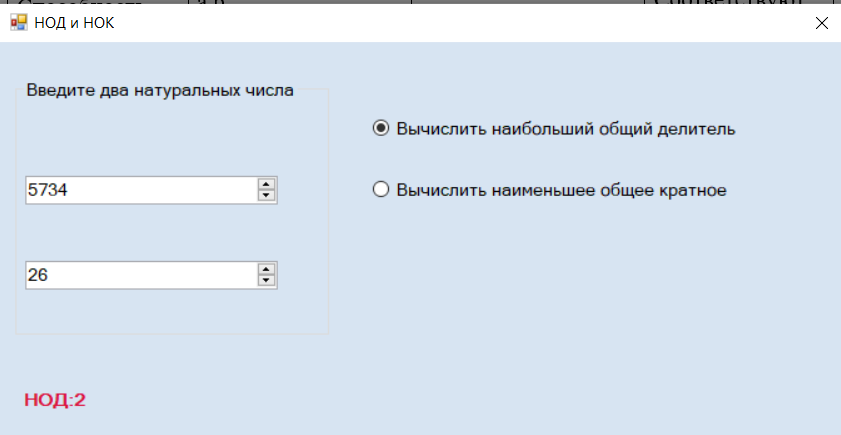
**Рис 3.3.3**

****

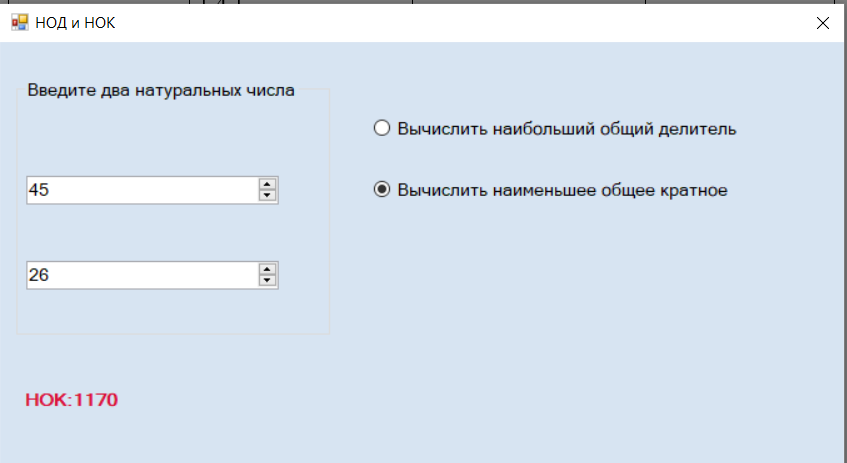
**Рис 3.3.4**

****

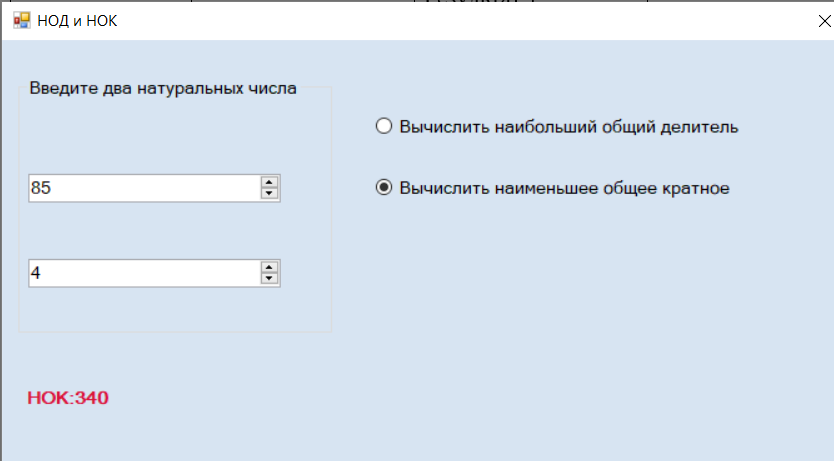
**Рис 3.3.5**

****

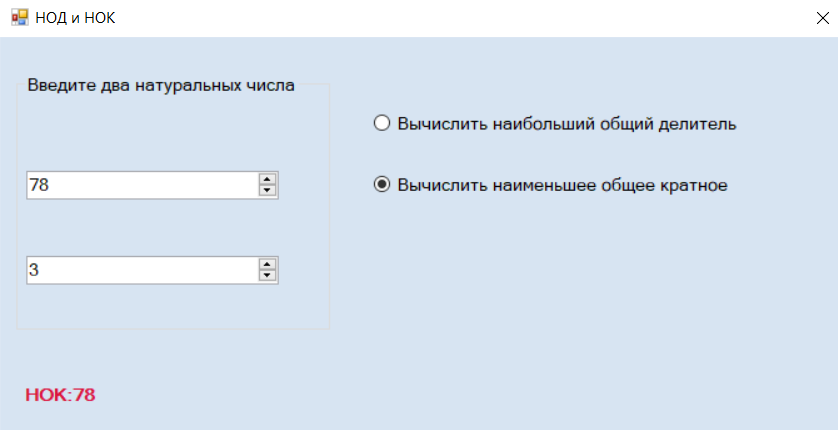
**Рис 3.3.6**

****

**Рис 3.3.7**

****

**Рис 3.3.8**

****

**Рис 3.3.9**