

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Маргарит Шаралович

Должность: Ректор

Дата подписания: 13.10.2023 12:09:09

Уникальный программный ключ:

236bcc35c296f119d6aafdc22836b21db52dbc07971a86865a5825f91a4304cc

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ**

**ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**имени академика М.Д. Миллионщикова**

Кафедра «Информационные технологии»

**Д.А. Мачуева**

**Методические указания к выполнению лабораторных работ  
по дисциплине «Программирование»**

**Направление подготовки**

09.03.02 Информационные системы и технологии

**Направленность (профиль)**

«Информационные системы и технологии»

«Информационные технологии в образовании»

«Информационные технологии в дизайне»

**Квалификация**

бакалавр

Грозный 2021

## **Составители:**

Кандидат технических наук,  
доцент кафедры  
«Информационные технологии»

Мачуева Дина Алуевна

## **Рецензент:**

Э.Д. Алисултанова, доктор педагогических наук, кандидат физико-математических наук, профессор, директор Института прикладных информационных технологий, заведующая кафедрой «Информатика и вычислительная техника»

Методические указания предназначены для бакалавров по направлению подготовки 09.03.02 Информационные системы и технологии Института прикладных информационных технологий.

Методические рекомендации рассмотрены и утверждены на заседании кафедры «Информационные технологии»: Протокол № 7 от 19.02.2020 г.

Рекомендовано к изданию редакционно-издательским советом ГГНТУ

© Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Грозненский государственный нефтяной технический университет имени академика  
М.Д. Миллионщикова»

## Содержание

Введение .....	4
Лабораторная работа 1. Типичные операции при работе с массивами.....	5
Лабораторная работа 2. Задача с массивом .....	6
Лабораторная работа 3. Использование массивов для хранения входных данных и результатов вычислений .....	8
Лабораторная работа 4. Дополнительные задачи с массивами.....	9
Лабораторная работа 5. Обработка текстовой информации в C#. Функции обработки строк .....	9
Лабораторная работа 6. Работа со строками.....	11
Лабораторная работа 7. Автоматизация операций по обработке текста .....	12
Лабораторная работа 8. Работа с датами и временем .....	14
Лабораторная работа 9. Методы отладки приложения.....	15
Лабораторная работа 10. Модульное программирование. Разработка пользовательских функций и процедур.....	18
Лабораторная работа 11. Графика: рисование в программировании .....	20
Лабораторная работа 12. Создание графических примитивов .....	22
Лабораторная работа 13. Графические примитивы .....	23
Лабораторная работа 14. Разработка приложений для различных областей применения...	25
Лабораторная работа 15. Работа с файлами.....	26
Лабораторная работа 16. Связь приложений на C# с базами данных .....	29
Лабораторная работа 17. Взаимодействие приложения на языке C# с MS Excel .....	32
Список литературы.....	35

## Введение

Цель изучения дисциплины «Программирование» состоит в освоении студентами, получающими квалификацию бакалавра, подходов к составлению алгоритмов решения задач различной степени сложности, а также овладении навыками создания программ на примере среды разработки приложений Microsoft Visual Studio.

Задачами дисциплины являются: изучение основополагающих понятий и правил программирования; разработка алгоритмов обработки данных различной структуры, освоение правил создания и организации пользовательского интерфейса.

Данное методическое пособие содержит подробное описание лабораторных работ, каждая из которых направлена на освоение и закрепление определенной темы:

- Массивы. Типичные операции при работе с массивами.
- Использование массивов для хранения входных данных и результатов вычислений.
- Обработка текстовой информации в C#. Функции обработки строк.
- Автоматизация операций по обработке текста.
- Работа с датами и временем.
- Методы отладки приложения. Трассировка, точки останова.
- Обработка исключений, инструкция try...catch.
- Модульное программирование. Разработка пользовательских функций и процедур.
- Графика: рисование в программировании. Создание графических примитивов.
- Разработка приложений для различных областей применения.
- Работа с файлами. Использование файлов для обмена данными с приложением. Работа с каталогами как объектами файловой системы.
- Связь приложений на C# с базами данных. Взаимодействие приложений на C# с базами данных с помощью запросов на языке SQL.
- Взаимодействие приложения на языке C# с MS Excel.

## Лабораторная работа 1. Типичные операции при работе с массивами

**Цель работы:** рассмотреть ввод и обработку элементов одномерного массива.

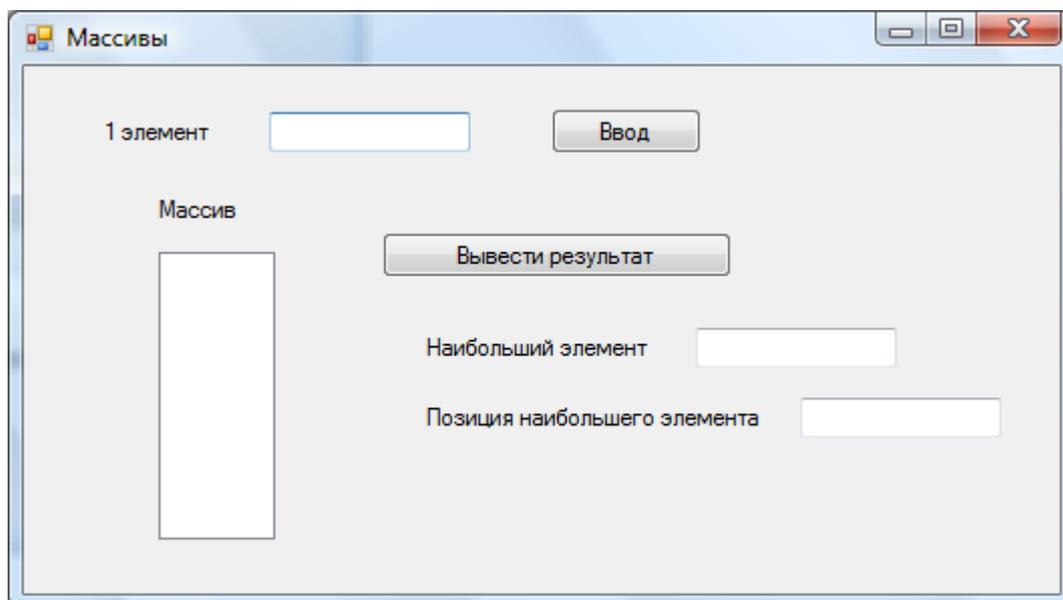
### Описание программы

Пользователь вводит массив из 5 целых чисел. Программа должна определить максимальный среди введенных элементов и указать его позицию в массиве.

Элементы массива поочередно вводятся в первое текстовое поле.

Введенные пользователем элементы массива отображаются в списке listBox.

### Интерфейс



The screenshot shows a Windows application window titled "Массивы". The interface includes the following elements:

- A label "1 элемент" followed by a single-line text input field.
- A button labeled "Ввод" to the right of the input field.
- A label "Массив" above a multi-line list box (listBox).
- A button labeled "Вывести результат" positioned between the list box and the output fields.
- A label "Наибольший элемент" followed by a single-line text input field.
- A label "Позиция наибольшего элемента" followed by a single-line text input field.

## Программный код

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int[] Massiv=new int[5];
        int n = 0;
        int max;           //Максимальный элемент
        int x;             //Позиция максимального элемента
        int i;

        private void button1_Click(object sender, EventArgs e) //кнопка Ввод
        {
            if (n <= 4 && textBox1.Text != "")
            {
                Massiv[n] = Convert.ToInt32(textBox1.Text);
                listBox1.Items.Add(Convert.ToString(Massiv[n]));
                textBox1.Text = "";
                n++;
                label1.Text = Convert.ToString(n + 1) + " элемент";
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            max = Massiv[0];
            x = 0;
            for (i = 1; i<=4; i++) //Поиск максимального элемента
                if (Massiv[i] > max)
                {
                    max = Massiv[i];
                    x = i;
                }
            textBox2.Text = Convert.ToString(max);
            textBox3.Text = Convert.ToString(x+1);
        }
    }
}
```

## Лабораторная работа 2. Задача с массивом

**Цель работы:** научиться применять одномерные массивы при решении задач.

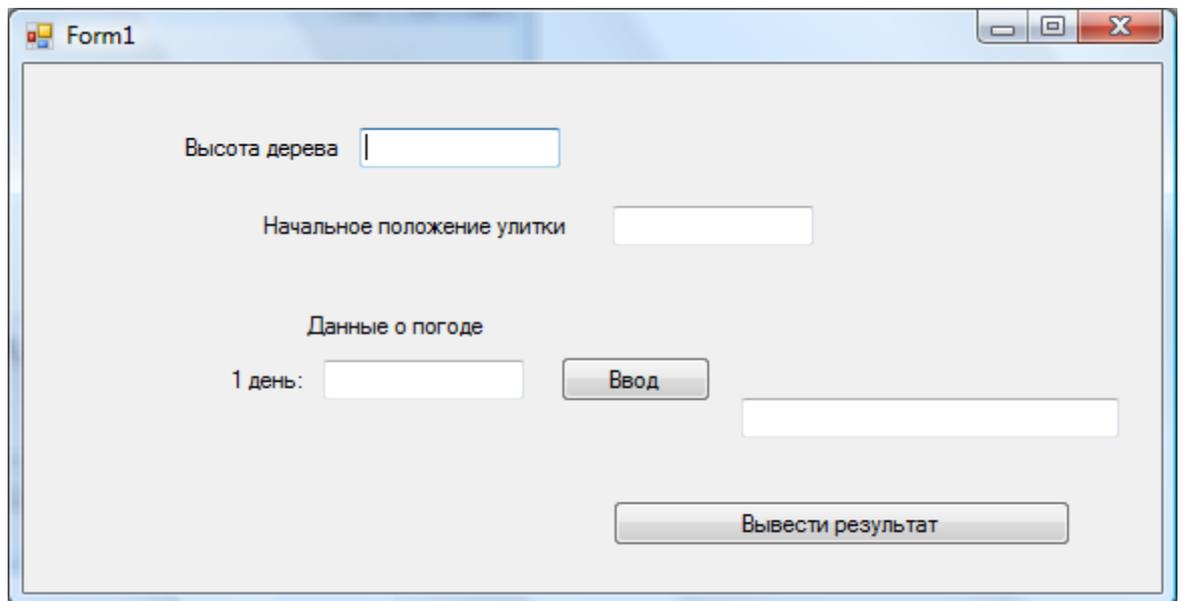
### Описание программы

Исходные данные: по дереву ползет улитка. Пользователь вводит высоту дерева и начальное положение улитки над уровнем земли (в сантиметрах). Затем вводятся данные о погоде в течение недели в следующем формате: 1 – если день был солнечный, 0 – если пасмурный. В солнечный день улитка поднимается на 10 см. вверх по дереву, в

пасмурный – опускается на 5 см. Программа должна определить конечное положение улитки по истечении недели.

Данные о погоде хранятся в одномерном массиве.

## Интерфейс



## Программный код

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int[] Pogoda = new int[7];
        int n = 0;
        int A, B;

        private void button1_Click(object sender, EventArgs e) //кнопка Ввод
        {
            if (n <= 6 && textBox3.Text != "")
            {
                Pogoda[n] = Convert.ToInt32(textBox3.Text);
                textBox4.Text = textBox4.Text + " " + Convert.ToString(Pogoda[n]);
                textBox3.Text = "";
                n++;
                label4.Text = Convert.ToString(n+1) + " день";
            }
        }
    }
}
```

```

private void button2_Click(object sender, EventArgs e) //кнопка Результат
{
    int i;
    A = Convert.ToInt32(textBox1.Text); //высота дерева
    B = Convert.ToInt32(textBox2.Text); //нач. положение улитки

    for (i = 0; i <= 6; i++)
    {
        if (Pogoda[i] == 0)
            B = B - 5;
        if (Pogoda[i] == 1)
            B = B + 10;
    }
    MessageBox.Show("Конечное положение улитки: "+Convert.ToString(B));
}
}
}

```

### Лабораторная работа 3. Использование массивов для хранения входных данных и результатов вычислений

**Цель работы:** рассмотреть понятие и применение двумерных массивов.

#### Описание программы

Двумерный массив вводится с помощью элемента dataGridView. Требуется найти сумму всех четных элементов в матрице 2 x 4.

Свойства объекта dataGridView:

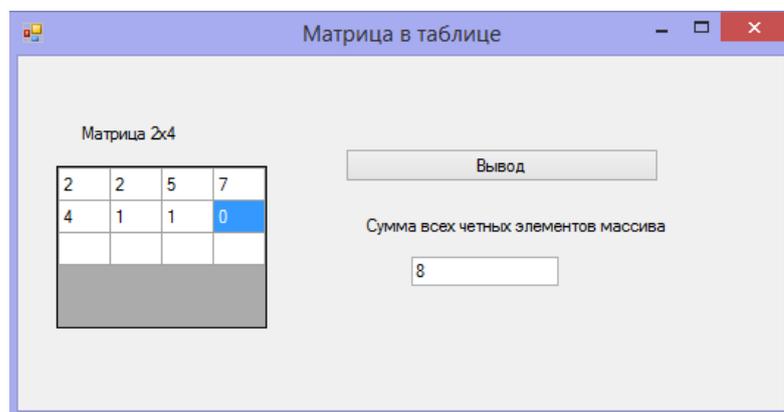
Columns – коллекция столбцов,

AutoSizeColumnsMode – присвоить значение Fill (столбцы по ширине занимают все пространство таблицы dataGridView),

RowHeadersVisible – присвоить значение False,

ColumnHeadersVisible – присвоить значение False.

#### Интерфейс



## Программный код

```
private void button1_Click(object sender, EventArgs e)
{
    int[,] massiv = new int[2, 4];
    int i, j, S;
    S = 0;

    // ввод массива из таблицы
    for (i = 0; i <= 1; i++) //номер строки
        for (j = 0; j <= 3; j++) // номер столбца
            massiv[i, j] = Convert.ToInt32(dataGridView1.Rows[i].Cells[j].Value);

    for (i = 0; i <= 1; i++)
        for (j = 0; j <= 3; j++)
            if (massiv[i, j]%2==0)
                S = S + massiv[i, j];

    textBox1.Text = Convert.ToString(S);
}
```

## Лабораторная работа 4. Дополнительные задачи с массивами

1. Вводится матрица 3 x 2. Определить, сколько среди ее элементов нулей.
2. Вводится матрица 4 x 2. Найти сумму элементов первого столбца.
3. Вводится матрица 3 x 3. Определить, является ли сумма элементов третьей строки четным числом.
4. Вводится матрица 3 x 3. Определить, является ли произведение элементов первого столбца четным числом.
5. Вводится матрица 2 x 3. Определить номер строки с наибольшей суммой элементов.
6. Вводится матрица 3 x 5. Найти произведение всех положительных элементов.

## Лабораторная работа 5. Обработка текстовой информации в C#. Функции обработки строк

**Цель работы:** изучить основные функции обработки текста.

### Описание программы

Вводится фамилия и имя студента через пробел. Программа разделяет фамилию и имя по разным текстовым полям.

## Интерфейс

Фамилия-имя студента

Введите фамилию и имя студента через пробел:

Разделить

Фамилия                      Имя

## Программный код

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string fn, fam, name;
            int L, n;
            fn = textBox1.Text;
            n = fn.IndexOf(" ");
            L = fn.Length;
            fam = fn.Substring(0, n);
            name = fn.Substring(n + 1, L - n - 1);
            textBox2.Text = fam;
            textBox3.Text = name;
        }
    }
}
```

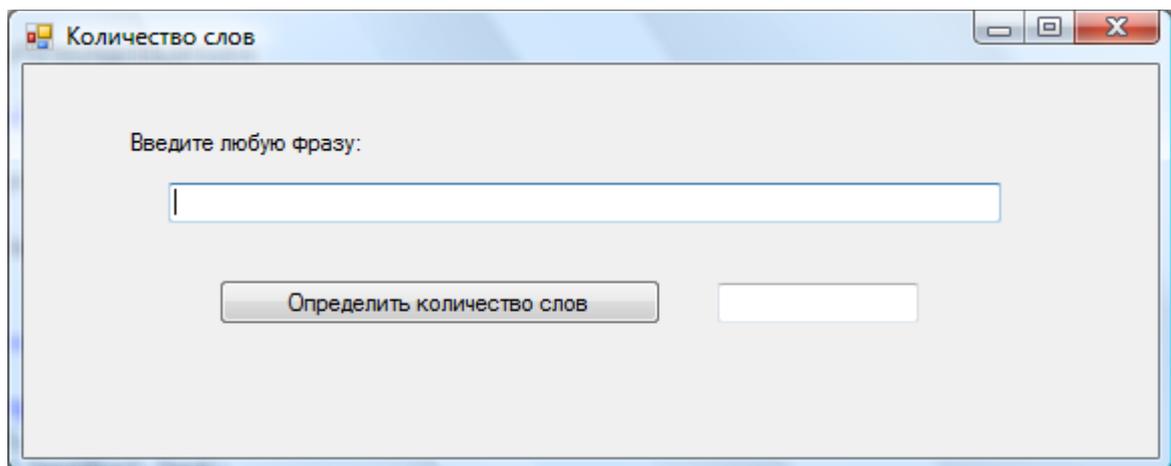
## Лабораторная работа 6. Работа со строками

**Цель работы:** научиться применять функции обработки строк при решении задач.

### Описание программы

Вводится фраза. Программа подсчитывает количество слов в ней.

### Интерфейс



### Программный код

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string str;
            int i,k;
            str = textBox1.Text;
            k = 0;
            for(i=0; i<=str.Length-1; i++)
                if (str.Substring(i,1) == " ")
                    k++;
            textBox2.Text = Convert.ToString(k + 1);
        }
    }
}
```

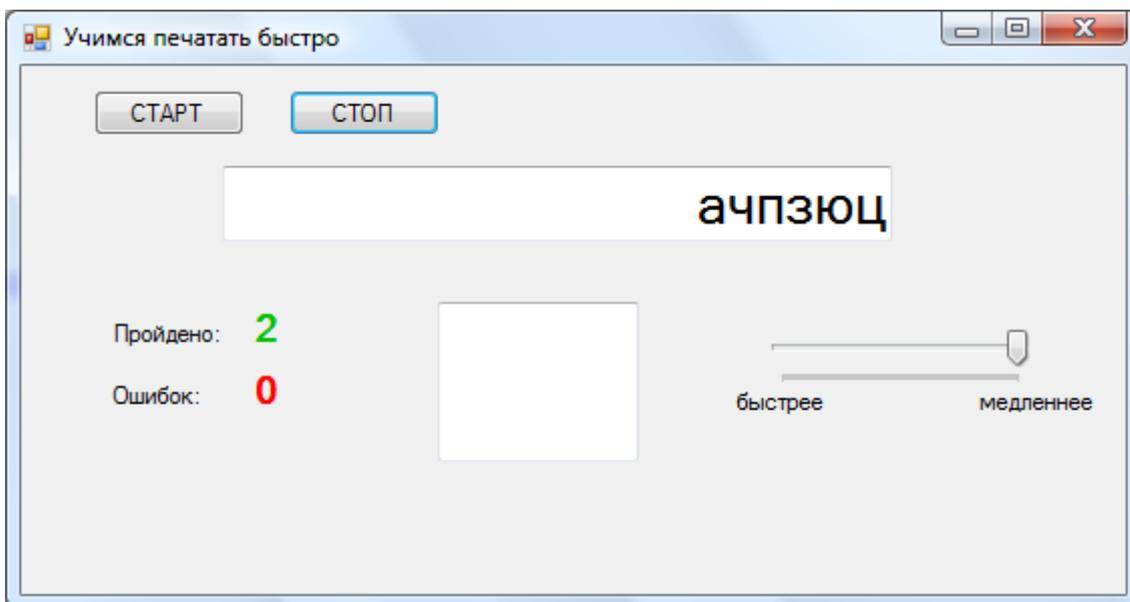
## Лабораторная работа 7. Автоматизация операций по обработке текста

**Цель работы:** рассмотреть применение функций обработки текста при решении практических задач.

### Описание программы

Игра «Учимся печатать быстро». Программа генерирует случайные десятичные числа и выводит соответствующие этим кодам символы по кодировке ASCII. Задача игрока – печатать эти символы в том же порядке. Верно набранный символ исчезает с экрана. Ведется отсчет общего количества набранных символов и ошибок. Игра заканчивается, когда на экране появляется 10 символов, которые игрок не успевает напечатать.

### Интерфейс



Дополнительные элементы:

1. timer с начальным интервалом = 1000
2. «ползунок» trackBar для изменения интервала таймера и скорости вывода

букв: свойство Minimum = 500, Maximum = 1000

Для текстового поля свойство TextAlign устанавливается Right

## Программный код

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int n, neprav;

        private void button1_Click(object sender, EventArgs e)
        {
            timer1.Enabled = true;
            n = 0;
            neprav = 0;
            label3.Text = "0";
            label4.Text = "0";
            textBox1.Text = "";
            textBox2.Focus();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            timer1.Enabled = false;
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            int code;
            Random rnd = new Random();
            code = rnd.Next(1072, 1103);
            textBox1.Text = textBox1.Text + (char)code;
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            string str;
            str = textBox1.Text;
            if (str.Length == 10)
            {
                timer1.Enabled = false;
                MessageBox.Show("Вы проиграли!", "Конец тренировки", MessageBoxButtons.OK, MessageBoxIcon.Stop);
            }
        }
    }
}
```

```

private void textBox2_TextChanged(object sender, EventArgs e)
{
    string буква, str;

    str = textBox1.Text;
    буква = textBox2.Text;
    if (буква != "")
    {
        n++;
        label3.Text = Convert.ToString(n);
        if (буква == str.Substring(0, 1))
            textBox1.Text=textBox1.Text.Remove(0, 1);
        else
        {
            neprav++;
            label4.Text = Convert.ToString(neprav);
        }
        textBox2.Text = "";
    }
}

private void trackBar1_Scroll(object sender, EventArgs e)
{
    timer1.Interval = trackBar1.Value;
}
}
}

```

## Лабораторная работа 8. Работа с датами и временем

**Цель работы:** изучить основные функции обработки дат и времени.

### Описание программы

Программа демонстрирует действие основных функций обработки даты и времени в языке C#. Для выбора произвольной даты используется календарь `dateTimePicker`.

### Интерфейс

## Программный код

```
private void button1_Click(object sender, EventArgs e)
{
    DateTime d;

    if (radioButton1.Checked == true)
        d = DateTime.Now;
    else
        d = dateTimePicker1.Value;

    textBox1.Text = d.ToString();
    textBox2.Text = Convert.ToString(d.Day);
    textBox3.Text = Convert.ToString(d.Month);
    textBox4.Text = Convert.ToString(d.Year);
    textBox5.Text = Convert.ToString(d.Hour);
    textBox6.Text = Convert.ToString(d.Minute);
    textBox7.Text = Convert.ToString(d.Second);
    textBox8.Text = Convert.ToString(d.DayOfYear);
    textBox9.Text = Convert.ToString(d.DayOfWeek);
    textBox10.Text = d.ToLongDateString();
    textBox11.Text = d.ToShortDateString();
    textBox12.Text = d.ToLongTimeString();
    textBox13.Text = d.ToShortTimeString();
    textBox14.Text = d.AddDays(10).ToString();
    textBox15.Text = d.AddMonths(10).ToString();
    textBox16.Text = d.AddYears(10).ToString();
}
```

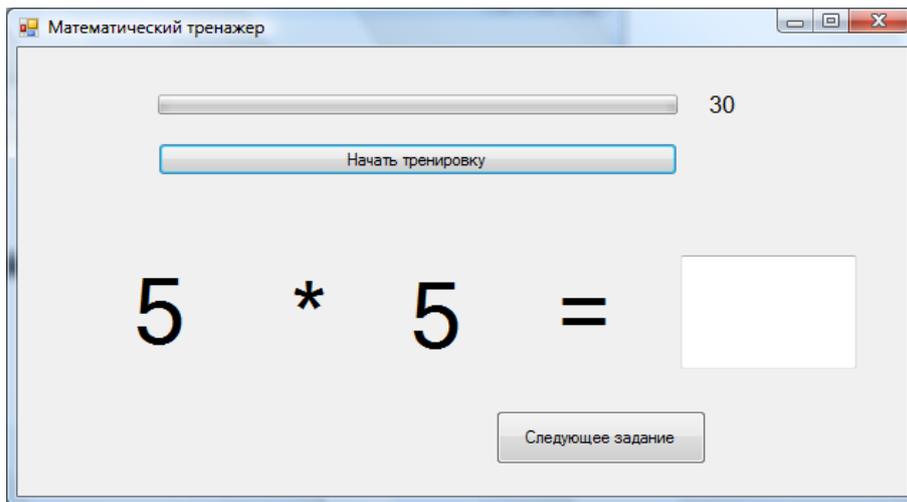
## Лабораторная работа 9. Методы отладки приложения

**Цель работы:** рассмотреть методы отладки на примере различных программ.

### Описание программы

Программа выдает в случайном порядке задания на сложение, вычитание, умножение и деление (случайным образом определяются два числа и знак операции между ними). Пользователю дается 30 секунд на тренировку, после чего выдается результат – количество пройденных заданий и количество правильных ответов.

### Интерфейс



## Программный код

```

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int s=30;           //секунды
        int prav, n;       //количество правильных ответов и пройденных заданий
        int x=5, y=5, res=0; //случайные числа и ответ пользователя
        int sign;         //знак вычислительного действия

        private void button1_Click(object sender, EventArgs e)
        {
            n = 0;
            prav = 0;
            x = 5;
            y = 5;
            sign = 3;
            label1.Text = "5";
            label2.Text = "*";
            label3.Text = "5";
            s = 30;
            label5.Text = "30";
            timer1.Enabled = true;
            button1.Enabled = false;
            button2.Enabled = true;
        }
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    Random rnd=new Random();
    n++;
    if (textBox1.Text != "")
        res = Convert.ToInt32(textBox1.Text);
    else
        res = 0;

    if (sign == 1 && res == x + y) prav++;
    if (sign == 2 && res == x - y) prav++;
    if (sign == 3 && res == x * y) prav++;
    if (sign == 4 && res == x / y) prav++;

    sign = rnd.Next(1, 5);
    if (sign==1)
    {
        x = rnd.Next(0, 50);
        y = rnd.Next(0, 50);
        label2.Text = "+";
    }

    if (sign==2)
    {
        x = rnd.Next(0, 100);
        y = rnd.Next(0, 100);
        if (x < y) x = x + y;
        label2.Text = "-";
    }
    if (sign==3)
    {
        x = rnd.Next(0, 25);
        y = rnd.Next(0, 10);
        label2.Text = "*";
    }
    if (sign==4)
    {
        y = rnd.Next(1, 15);
        res = rnd.Next(0, 10);
        x = res * y;
        label2.Text = "/";
    }

    label1.Text = Convert.ToString(x);
    label3.Text = Convert.ToString(y);
    textBox1.Text = "";
    textBox1.Focus();
}

```

```

private void timer1_Tick(object sender, EventArgs e)
{
    progressBar1.PerformStep();
    s--;
    label5.Text = Convert.ToString(s);
    if (s == 0)
    {
        timer1.Enabled = false;
        button2.Enabled = false;
        button1.Enabled = true;
        MessageBox.Show("Правильных ответов: " + Convert.ToString(prav) + " из " + Convert.ToString(n),
            "Конец тренировки", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
}

```

**Задание.** Попробуйте применить инструкцию try...catch для обработки возможных исключений в работе программы.

## Лабораторная работа 10. Модульное программирование. Разработка пользовательских функций и процедур

**Цель работы:** научиться создавать пользовательские процедуры и функции и осуществлять их вызов.

### Описание программы

Программа имитирует игру «Пятнашки» с восемью фишками, которые нужно разместить в порядке возрастания, передвигая в пределах поля 3 x 3.

В качестве фишек используются компоненты button.

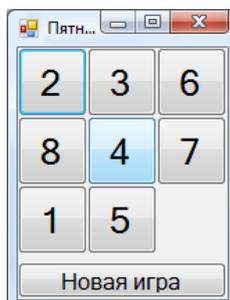
При нажатии кнопки «Новая игра» компьютер размещает на кнопках button цифры от 1 до 8 в случайном порядке, используя генератор случайных чисел.

При щелчке по любой из кнопок-фишек она перемещается на соседнее поле, если оно свободно.

Размеры кнопок с цифрами – свойство Size. Установить размеры 50 x 50.

Расположение кнопок – свойство Location.

### Интерфейс



### Программный код

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int a, b; //координаты пустой ячейки: номер строки и номер столбца
                //(нумеруются с нуля)

        private void Form1_Load(object sender, EventArgs e)
        {
            NewGame();
            a = 2;
            b = 2; //координаты пустой ячейки при запуске
        }
    }
}
```

```

private void NewGame()
{
    int[] Numbers=new int[8];
    bool X;
    int i,n,num;
    Random rnd=new Random();

    for(n=0;n<=7;n++)
        Numbers[n] = 0;

    //заполнение массива в случайном порядке:
    for (n = 0; n <= 7; n++)
        do
        {
            num = rnd.Next(9);
            X = false;          //X=True, если такая цифра уже использована
            for (i = 0; i <= n; i++)
                if (Numbers[i] == num)
                    X = true;

            if (X == false)
            {
                Numbers[n] = num;
                //расположение цифр на кнопках :
                this.Controls["button" + Convert.ToString(n + 1)].Text = Convert.ToString(Numbers[n]);
            }
        }
        while (Numbers[n] == 0);
}

```

```

private void button9_Click(object sender, EventArgs e)
{
    NewGame();          //Новая игра
}

private void BClick(string btname)
{
    int x,y;            //координаты кнопки: номер строки и номер столбца
    x = this.Controls[btname].Top / 50;
    y = this.Controls[btname].Left / 50;

    //если кнопка находится рядом с пустой ячейкой:
    if (((Math.Abs(x - a) == 1) && (y == b)) || ((Math.Abs(y - b) == 1) && (x == a)))
    {
        this.Controls[btname].Top = a * 50;
        this.Controls[btname].Left = b * 50;      //перемещение на пустое место
        a = x;
        b = y;
    }
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    BClick("button1");
}

private void button2_Click(object sender, EventArgs e)
{
    BClick("button2");
}

private void button3_Click(object sender, EventArgs e)
{
    BClick("button3");
}

private void button4_Click(object sender, EventArgs e)
{
    BClick("button4");
}

private void button5_Click(object sender, EventArgs e)
{
    BClick("button5");
}

private void button6_Click(object sender, EventArgs e)
{
    BClick("button6");
}

private void button7_Click(object sender, EventArgs e)
{
    BClick("button7");
}

private void button8_Click(object sender, EventArgs e)
{
    BClick("button8");
}
}
}

```

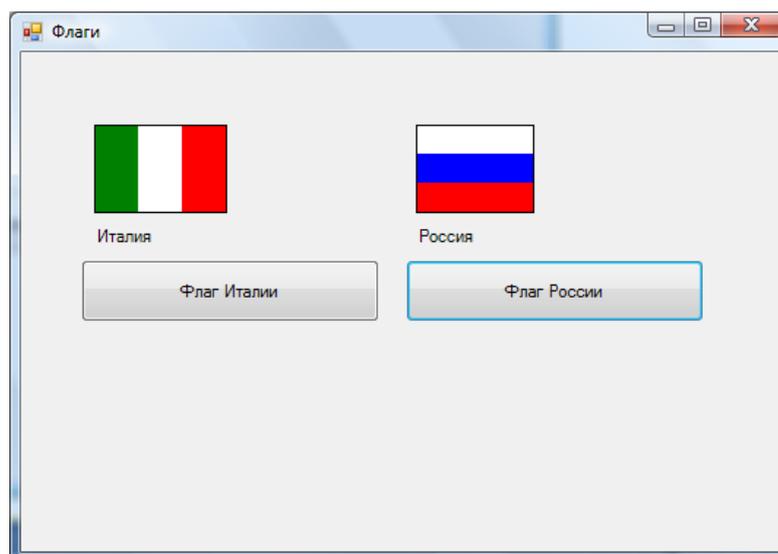
## Лабораторная работа 11. Графика: рисование в программировании

**Цель работы:** изучить основные методы создания графики в программировании.

### Описание программы

При нажатии на кнопку на форме прорисовывается флаг, состоящий из закрасенных прямоугольников и общего контура.

## Интерфейс



## Программный код

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Graphics g = this.CreateGraphics();
            // для каждой полосы задаются цвет кисти, координаты верхнего левого угла, ширина и высота
            // зеленая полоса
            g.FillRectangle(Brushes.Green, 50, 50, 30, 60);
            // белая полоса
            g.FillRectangle(Brushes.White, 80, 50, 30, 60);
            // красная полоса
            g.FillRectangle(Brushes.Red, 110, 50, 30, 60);
            // контур
            g.DrawRectangle(Pens.Black, 50, 50, 90, 60);
            // Подпись. Используется шрифт, заданный свойством Font формы.
            g.DrawString("Италия", this.Font, Brushes.Black, 50, 120);
        }
    }
}
```

## Задание

Флаг России прорисовать самостоятельно.

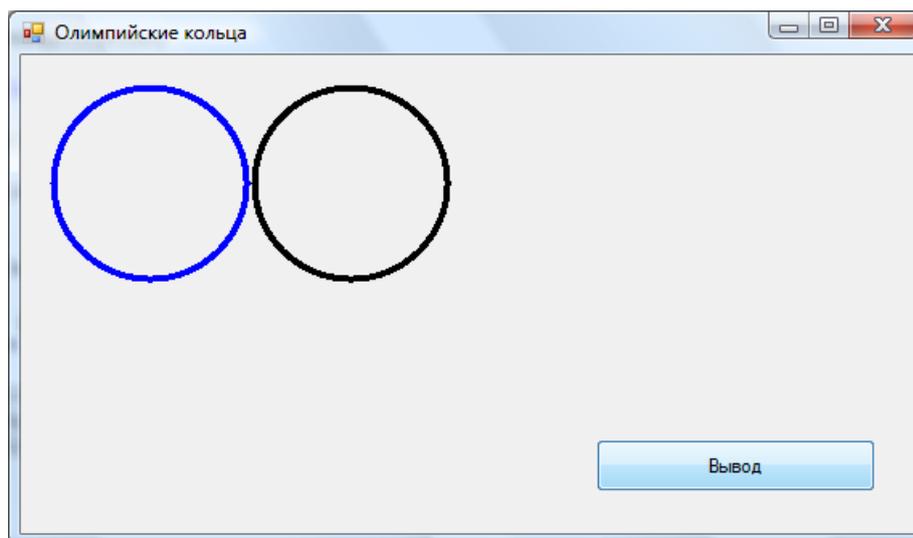
## Лабораторная работа 12. Создание графических примитивов

**Цель работы:** изучить основные методы создания графики в программировании.

### Описание программы

При нажатии на кнопку на форме прорисовывается символ олимпийских игр как рисунок, составленный из графических примитивов – окружностей.

### Интерфейс



### Программный код

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

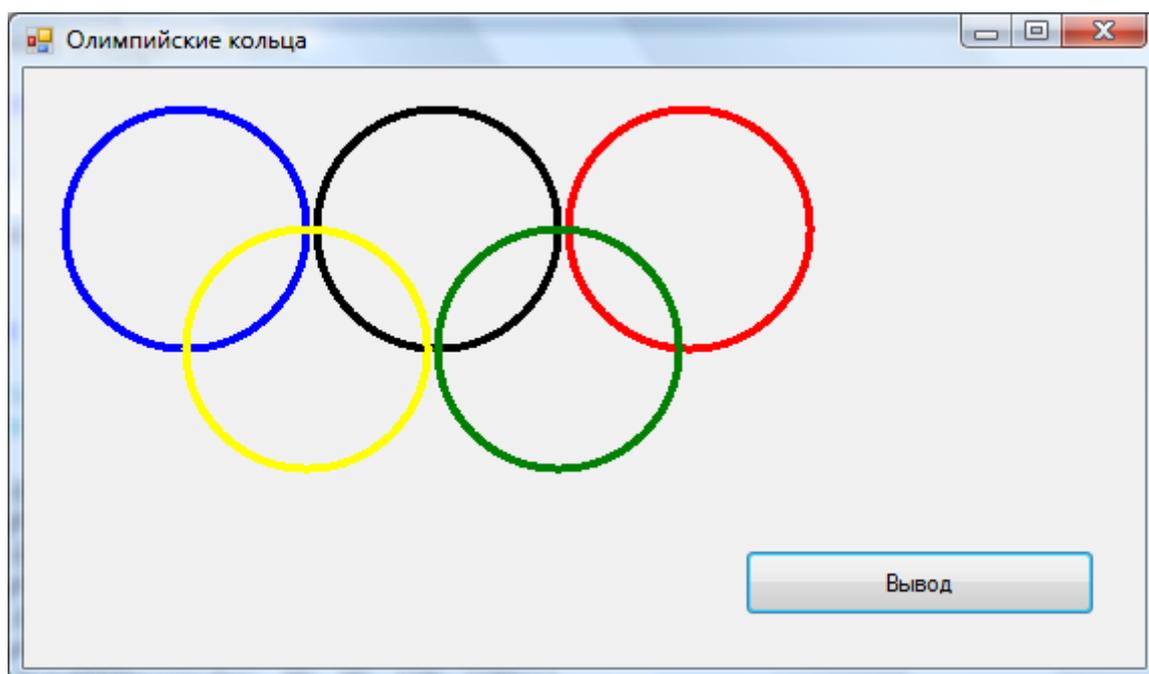
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Graphics g = this.CreateGraphics();
            Pen myPen = new Pen(Color.Blue, 4);

            g.DrawEllipse(myPen, 20, 20, 120, 120);
            myPen.Color = Color.Black;
            g.DrawEllipse(myPen, 145, 20, 120, 120);
        }
    }
}
```

## Задание

Закончить рисунок самостоятельно.



## Лабораторная работа 13. Графические примитивы

**Цель работы:** изучить основные методы создания графики в программировании.

### Описание программы

При нажатии на кнопку на форме прорисовывается пятиконечная звезда.

### Интерфейс



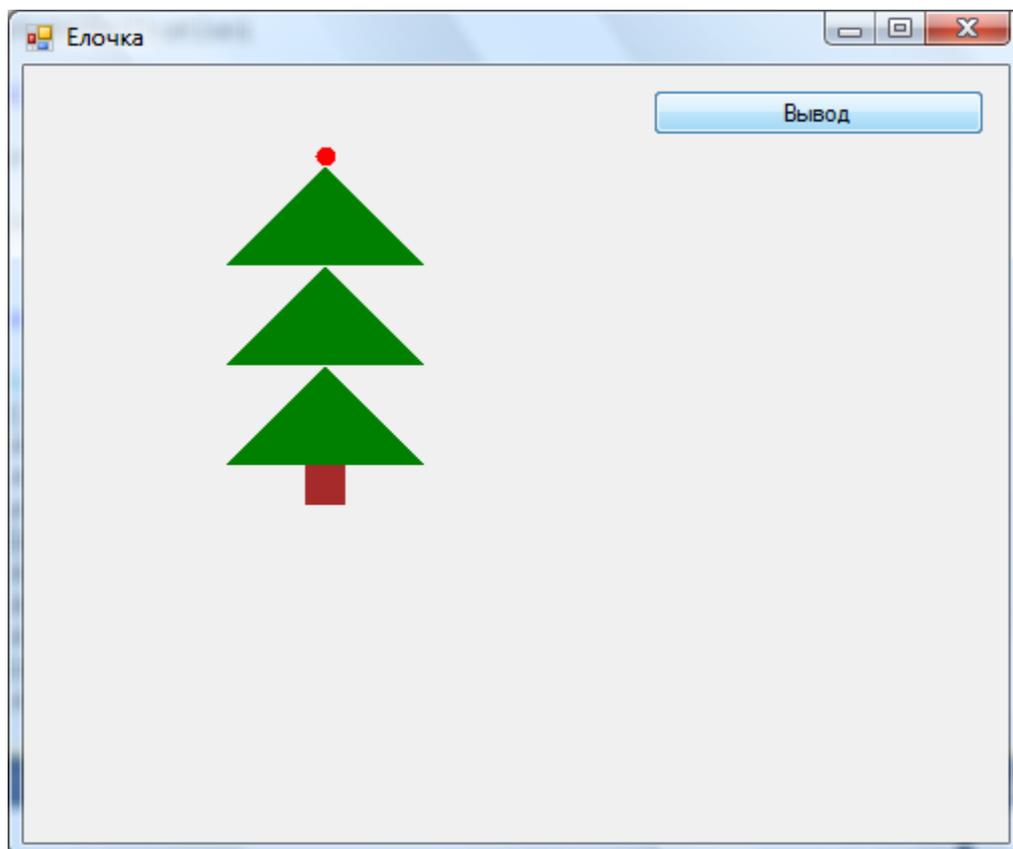
## Программный код

```
private void button1_Click(object sender, EventArgs e)
{
    Graphics g = this.CreateGraphics();
    Pen myPen = new Pen(Color.Coral, 4);

    Point[] p = new Point[5];
    p[0].X = 55; p[0].Y = 75;
    p[1].X = 190; p[1].Y = 115;
    p[2].X = 75; p[2].Y = 185;
    p[3].X = 140; p[3].Y = 50;
    p[4].X = 160; p[4].Y = 175;
    g.DrawPolygon(myPen, p);
}
```

## Задание

Нарисовать елочку, состоящую из трех ярусов – треугольников.



## Лабораторная работа 14. Разработка приложений для различных областей применения

**Цель работы:** научиться разрабатывать интерфейс и автоматизировать расчеты при решении различных прикладных задач.

### Интерфейс

Предоставляется скидка:  %

	Наименование материалов	Единица измерения	Количество	Стоимость, руб.	Общая стоимость закупки	Со скидкой
▶*		▼				

ИТОГО:

Рассчитать

### Описание программы

Информация о скидке, наименование материалов, количество и стоимость вводится пользователем.

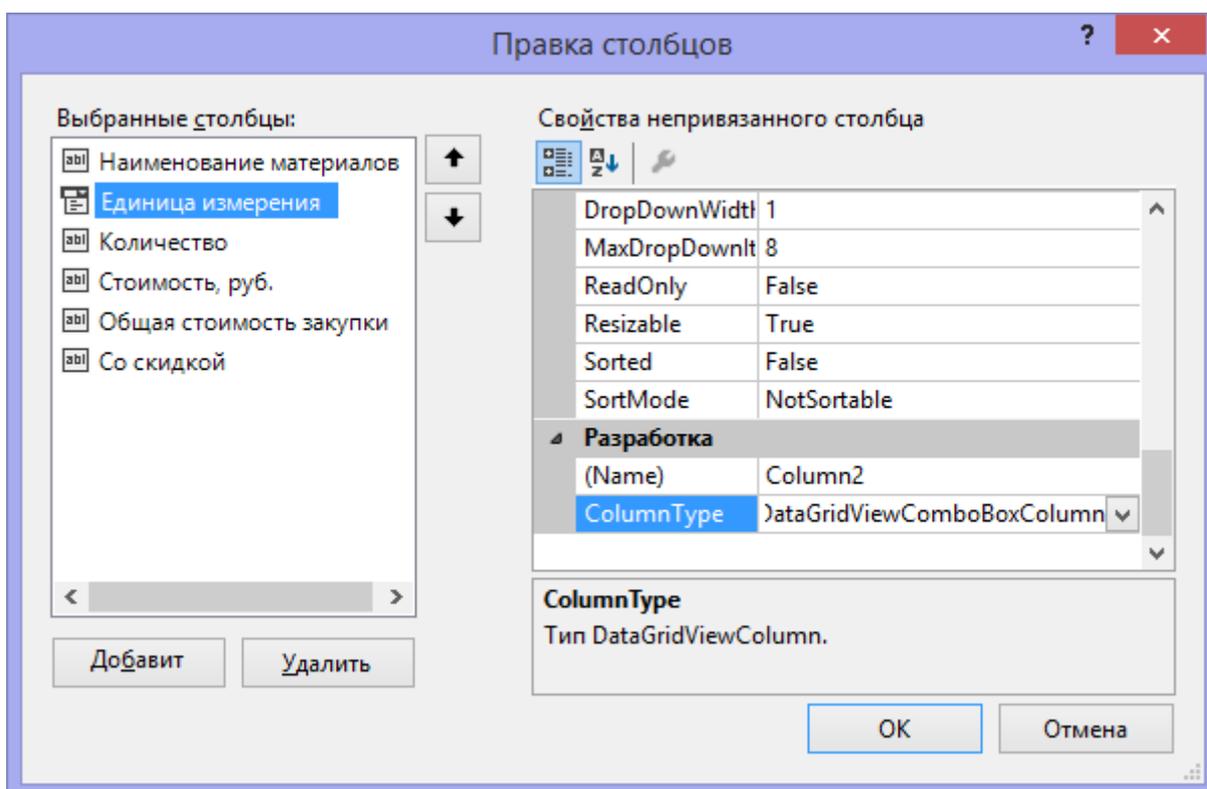
Общая стоимость по каждому пункту, стоимость с учетом скидки и суммарный итог рассчитывается программой.

Предоставляется скидка:  %

	Наименование материалов	Единица измерения	Количество	Стоимость, руб.	Общая стоимость закупки	Со скидкой
	цемент	тонн	10,5	400	4200	2100
	линолеум	кв. м.	230	120	27600	13800
▶	плитка	кв. м.	100	300	30000	15000
*		▼				

ИТОГО:

Рассчитать



Программный код разработать самостоятельно в соответствии с полученным описанием приложения.

## Лабораторная работа 15. Работа с файлами

**Цель работы:** рассмотреть методы работы с информацией о файлах.

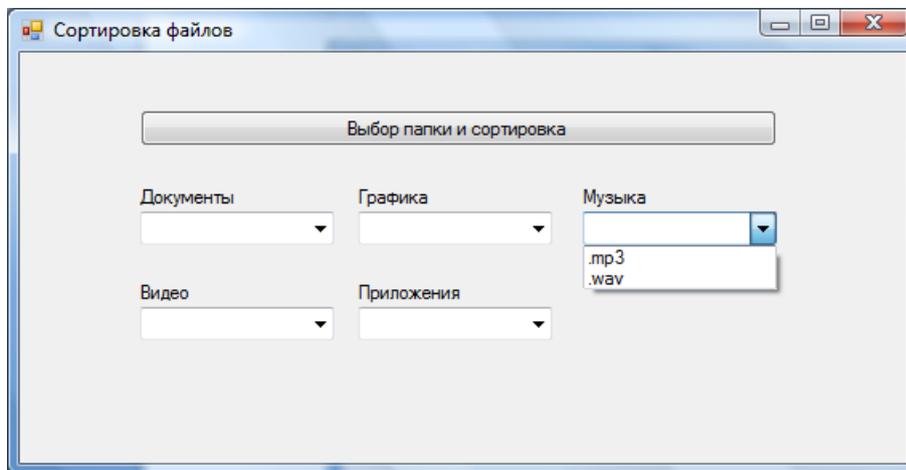
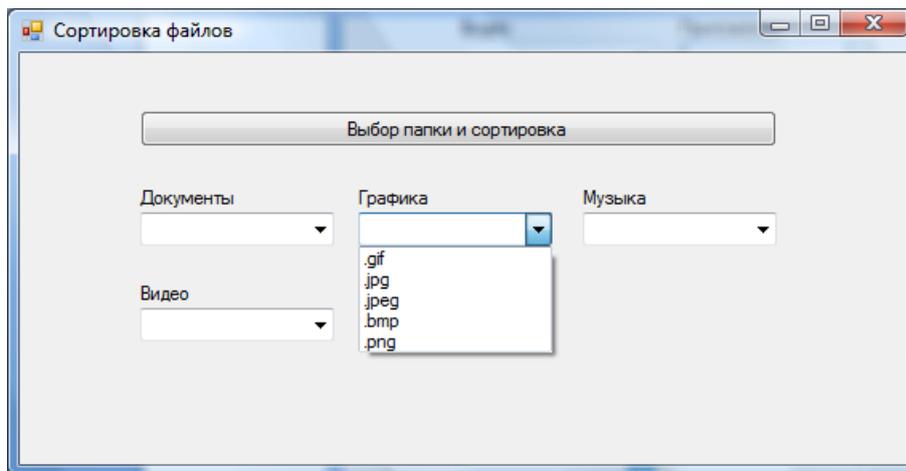
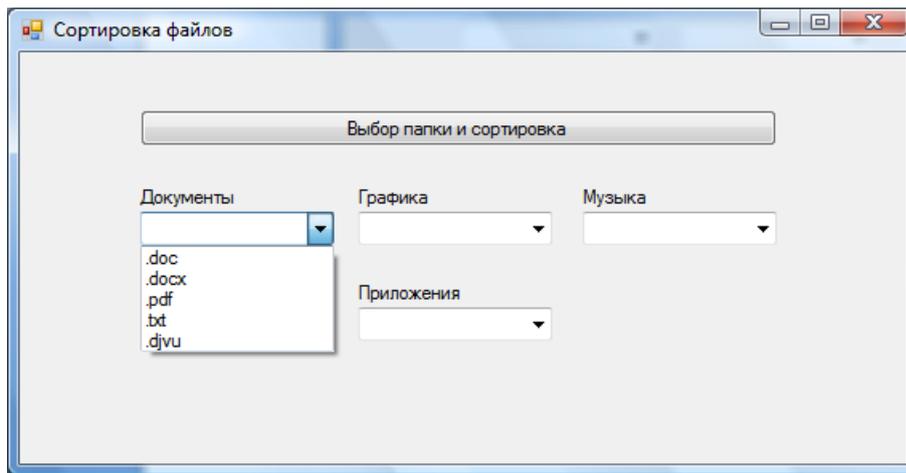
### Описание программы

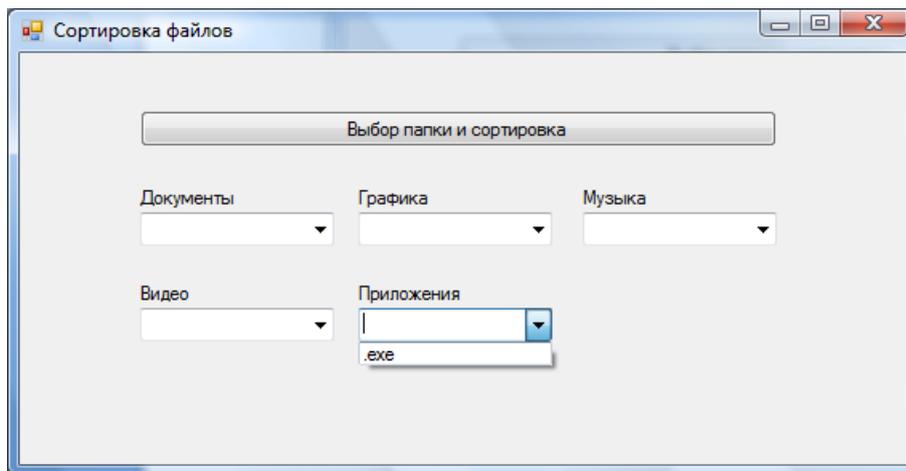
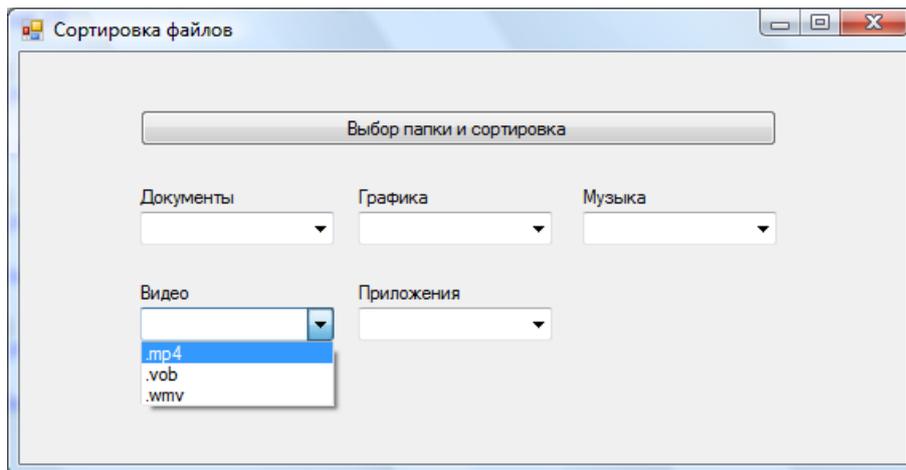
В программе предусмотрены 5 категорий файлов. Расширения файлов, относящихся к каждой категории, размещены в пяти списках comboBox.

Пользователь выбирает папку, в которой нужно отсортировать файлы. Программа создает в этой папке подкаталоги с именами категорий и перемещает туда файлы, расширения которых совпали с одним из вариантов, указанных в comboBox.

Для выбора папки используется компонент folderBrowserDialog.

### Интерфейс





## Программный код

```

using System.Windows.Forms;
using System.IO;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string catalog;
            string[] files;
            int i,j;

            folderBrowserDialog1.ShowDialog();
            catalog = folderBrowserDialog1.SelectedPath;
            if (Directory.Exists(catalog + "\\Документы") == false)
                Directory.CreateDirectory(catalog + "\\Документы");
            if (Directory.Exists(catalog + "\\Графика") == false)
                Directory.CreateDirectory(catalog + "\\Графика");
            if (Directory.Exists(catalog + "\\Музыка") == false)
                Directory.CreateDirectory(catalog + "\\Музыка");
            if (Directory.Exists(catalog + "\\Видео") == false)
                Directory.CreateDirectory(catalog + "\\Видео");
            if (Directory.Exists(catalog + "\\Приложения") == false)
                Directory.CreateDirectory(catalog + "\\Приложения");
        }
    }
}

```

```

files = Directory.GetFiles(catalog);
for (i = 0; i <= files.Length - 1; i++)
{
    for (j = 0; j <= comboBox1.Items.Count - 1; j++)
        if (Path.GetExtension(files[i]) == comboBox1.Items[j].ToString())
            File.Move(files[i], catalog + "\\Документы\\" + Path.GetFileName(files[i]));

    for (j = 0; j <= comboBox2.Items.Count - 1; j++)
        if (Path.GetExtension(files[i]) == comboBox2.Items[j].ToString())
            File.Move(files[i], catalog + "\\Графика\\" + Path.GetFileName(files[i]));
    for (j = 0; j <= comboBox3.Items.Count - 1; j++)
        if (Path.GetExtension(files[i]) == comboBox3.Items[j].ToString())
            File.Move(files[i], catalog + "\\Музыка\\" + Path.GetFileName(files[i]));
    for (j = 0; j <= comboBox4.Items.Count - 1; j++)
        if (Path.GetExtension(files[i]) == comboBox4.Items[j].ToString())
            File.Move(files[i], catalog + "\\Видео\\" + Path.GetFileName(files[i]));
    for (j = 0; j <= comboBox5.Items.Count - 1; j++)
        if (Path.GetExtension(files[i]) == comboBox5.Items[j].ToString())
            File.Move(files[i], catalog + "\\Приложения\\" + Path.GetFileName(files[i]));
}

MessageBox.Show("Сортировка прошла успешно!");
}
}

```

## Лабораторная работа 16. Связь приложений на С# с базами данных

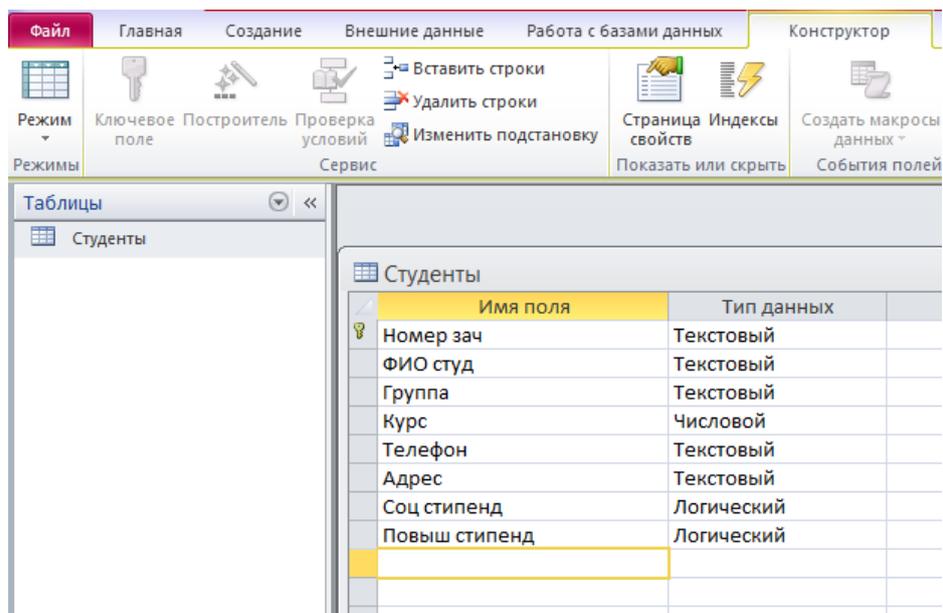
**Цель работы:** рассмотреть методы взаимодействия с базой данных, научиться составлять запросы на языке SQL.

### Описание программы

Программа предоставляет возможность:

1. вносить в базу данных сведения о студенте;
2. осуществлять поиск студентов в БД по названию группы.

Подготовленная база данных Access предоставляется студентам. БД состоит из одной таблицы Студенты.



Свойства объектов интерфейса:

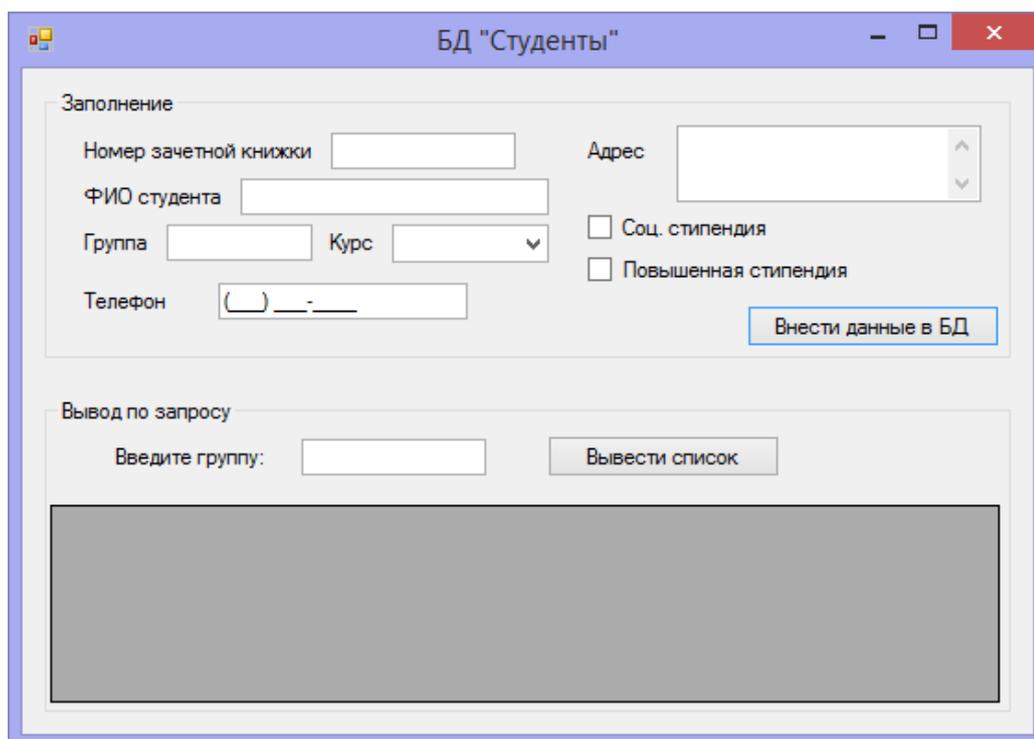
для указания курса (comboBox) – в свойстве Items значения 1,2,3,4,5;

для записи телефона (maskedTextBox) – в свойстве Mask выбрать Phone number;

для ввода адреса сделать текстовое поле многострочным – свойство Multiline = True;

для вывода результатов запроса – компонент dataGridView (настройки внешнего вида: свойства BackgroundColor,DefaultCellStyle).

## Интерфейс



## Программный код

```
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string s_stip, p_stip;
            if (checkBox1.Checked == true)
                s_stip = "TRUE";
            else
                s_stip = "FALSE";

            if (checkBox2.Checked == true)
                p_stip = "TRUE";
            else
                p_stip = "FALSE";

            OleDbConnection connect;
            connect = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
                Application.StartupPath.ToString() + "\\БД.mdb");
            connect.Open();

            OleDbCommand command = new OleDbCommand("INSERT INTO [Студенты] ([Номер зач],[ФИО студ],[Группа],"+
                "[Курс],[Телефон],[Адрес],[Соц стипенд],[Повыш стипенд]) VALUES ('" + textBox1.Text + "','" + textBox2.Text+
                "','" + textBox3.Text+ "','" + comboBox1.Text+ "','" + maskedTextBox1.Text+ "','" + textBox4.Text+
                "','" + s_stip+ "','" + p_stip+ "')", connect);
            command.ExecuteNonQuery();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            int col;
            string group;
            group = textBox5.Text;

            OleDbConnection connect;
            connect = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data Source=" +
                Application.StartupPath.ToString() + "\\БД.mdb");
            connect.Open();

            OleDbCommand command = new OleDbCommand("SELECT [Номер зач],[ФИО студ],[Телефон],[Соц стипенд],[Повыш стипенд] "+
                "FROM [Студенты] WHERE [Группа]='" + group + "'", connect);
            OleDbDataAdapter adapter = new OleDbDataAdapter(command);
            DataSet set = new DataSet();

            adapter.Fill(set, "Студенты");
            dataGridView1.DataSource = set;
            dataGridView1.DataMember = "Студенты";

            dataGridView1.Columns[0].HeaderText = "№ зачетной книжки";
            dataGridView1.Columns[1].HeaderText = "ФИО студента";
            dataGridView1.Columns[2].HeaderText = "Телефон";
            dataGridView1.Columns[3].HeaderText = "Соц стипендия";
            dataGridView1.Columns[4].HeaderText = "Повыш стипендия";

            col=dataGridView1.Rows.Count-1;
            MessageBox.Show("По запросу найдено записей: "+col, "Результат поиска");

            connect.Close();
        }
    }
}
```

Строка подключения: **Provider=Microsoft.ACE.OLEDB.12.0; Data Source=**  
или: **Provider=Microsoft.Jet.OLEDB.4.0; Data Source=**

## Задание

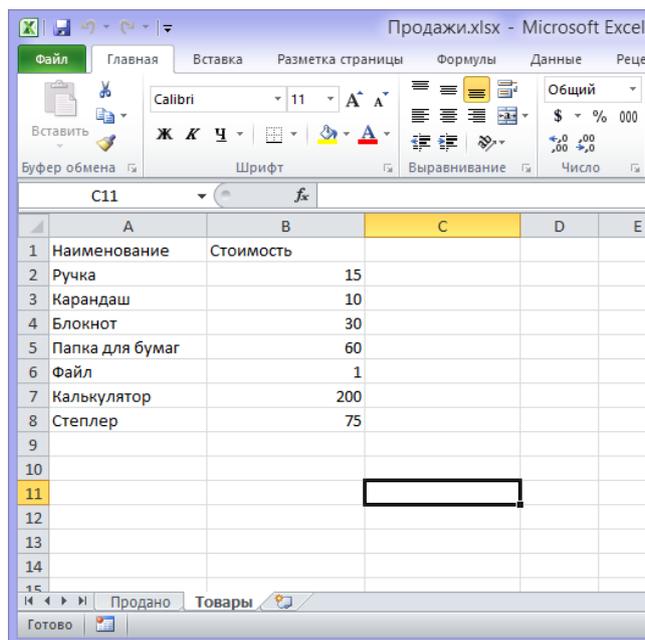
Разработать приложение, позволяющее вносить сведения о работниках офиса / продажах / выполненных работах и т.д. (тема на усмотрение студента) и получать информацию по запросу. Структуру базы данных продумать самостоятельно.

## Лабораторная работа 17. Взаимодействие приложения на языке C# с MS Excel

**Цель работы:** научиться вносить и считывать информацию из внешнего файла на примере Microsoft Excel.

### Описание программы

В файле Excel на листе Товары содержится список товаров и цен на них. При запуске программы этот список считывается и представляется пользователю в компоненте comboBox.



The screenshot shows a Microsoft Excel window titled "Продажи.xlsx - Microsoft Excel". The ribbon includes "Файл", "Главная", "Вставка", "Разметка страницы", "Формулы", "Данные", and "Рецензирование". The active cell is C11. The spreadsheet contains the following data:

	A	B	C	D	E
1	Наименование	Стоимость			
2	Ручка		15		
3	Карандаш		10		
4	Блокнот		30		
5	Папка для бумаг		60		
6	Файл		1		
7	Калькулятор		200		
8	Степлер		75		
9					
10					
11					
12					
13					
14					
15					

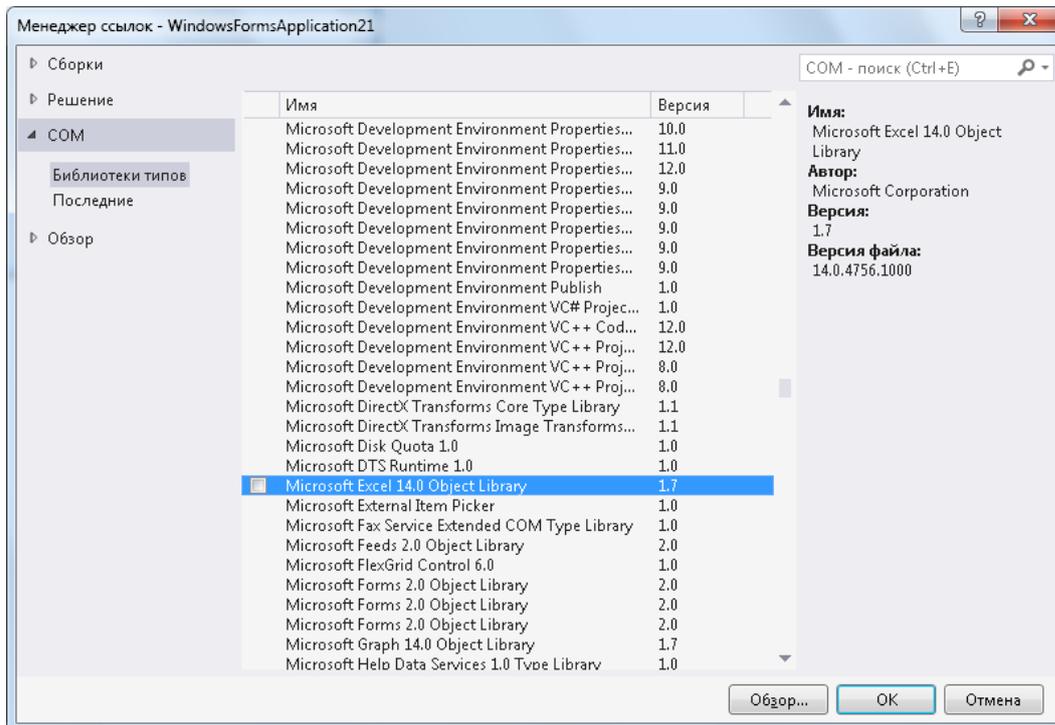
At the bottom of the window, there are two tabs: "Продано" and "Товары". The "Товары" tab is active. The status bar at the bottom left shows "Готово".

Информация о выбранном товаре при нажатии кнопки Продано фиксируется в том же файле Excel. Подсчитывается и отображается в компоненте label на форме общая стоимость купленных товаров.

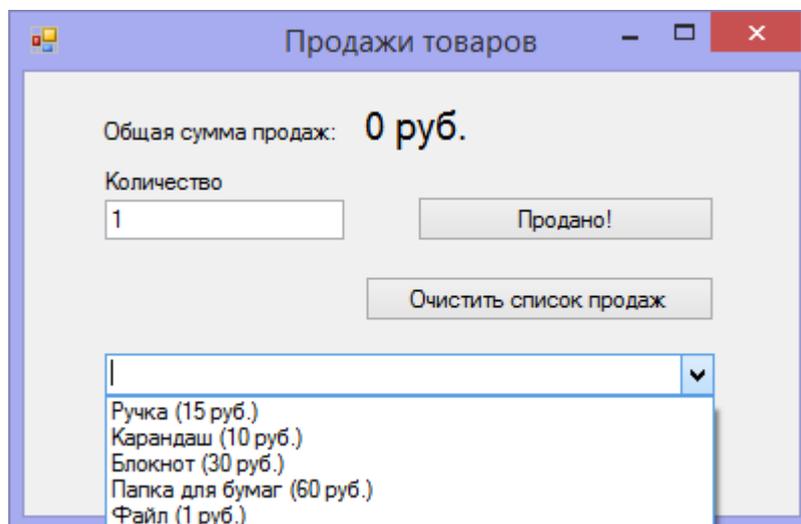
Предусмотрена возможность очистить список продаж.

Подготовленный файл Excel предоставляется студентам.

Для взаимодействия с приложением Excel необходимо: с помощью менеджера ссылок добавить в проект ссылку на библиотеку Microsoft Excel Object Library (меню Проект – Добавить ссылку).



## Интерфейс



## Программный код

```
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            Microsoft.Office.Interop.Excel.Application app = new Microsoft.Office.Interop.Excel.Application();
            Workbook wb;
            int summa=0;

            private void Form1_Load(object sender, EventArgs e)
            {
                wb = app.Workbooks.Open(System.Windows.Forms.Application.StartupPath + "\\Продажи.xlsx");

                int tovar_kol;
                tovar_kol = wb.Worksheets["Товары"].UsedRange.Rows.Count;
                for(int i=2;i<=tovar_kol;i++)
                    comboBox1.Items.Add(wb.Worksheets["Товары"].Cells[i,1].Value+" ("+
                        wb.Worksheets["Товары"].Cells[i, 2].Value+" руб.)");
            }

            private void button1_Click(object sender, EventArgs e)
            {
                int n; //номер пустой строки в списке с продажами
                int kol,cena;
                n = wb.Worksheets["Продано"].UsedRange.Rows.Count+1;

                wb.Worksheets["Продано"].Cells[n, 1].Value = comboBox1.Text;
                kol = Convert.ToInt32(textBox1.Text);
                wb.Worksheets["Продано"].Cells[n, 2].Value = kol;
                cena = Convert.ToInt32(wb.Worksheets["Товары"].Cells[comboBox1.SelectedIndex + 2, 2].Value);
                wb.Worksheets["Продано"].Cells[n, 3].Value = cena * kol;

                summa = summa + cena * kol;
                label2.Text = Convert.ToString(summa) + " руб.";
            }

            private void Form1_FormClosed(object sender, FormClosedEventArgs e)
            {
                wb.Save();
                app.Quit();
            }

            private void button2_Click(object sender, EventArgs e)
            {
                int n;
                n = wb.Worksheets["Продано"].UsedRange.Rows.Count;
                for (int i = 2; i <= n; i++)
                {
                    wb.Worksheets["Продано"].Cells[i, 1].Value="";
                    wb.Worksheets["Продано"].Cells[i, 2].Value = "";
                    wb.Worksheets["Продано"].Cells[i, 3].Value = "";
                }
                MessageBox.Show("Данные о продажах удалены", "Внимание", MessageBoxButtons.OK,
                    MessageBoxIcon.Exclamation);
            }
    }
}
```

## Список литературы

1. Агапов, В.П. Основы программирования на языке C# [Электронный ресурс]: учебное пособие / В.П. Агапов. – Электрон. текстовые данные. – М.: Московский государственный строительный университет, ЭБС АСВ, 2012. – 128 с. – Режим доступа: <http://www.iprbookshop.ru/16366.html> (ЭБС «IPRbooks»).

2. Поляков, А.Ю. Программирование [Электронный ресурс]: практикум / А.Ю. Поляков, А.Ю. Полякова, Е.Н. Перышкова. – Электрон. текстовые данные. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2015. – 55 с. – Режим доступа: <http://www.iprbookshop.ru/55494.html> (ЭБС «IPRbooks»).

3. Котов, О.М. Язык C#. Краткое описание и введение в технологии программирования [Электронный ресурс]: учебное пособие / О.М. Котов. — Электрон. текстовые данные. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2014. – 208 с. – Режим доступа: <http://www.iprbookshop.ru/68524.html> (ЭБС «IPRbooks»).

4. Биллиг, В.А. Основы программирования на C# [Электронный ресурс] / В. А. Биллиг. – 2-е изд. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 574 с. – Режим доступа: <http://www.iprbookshop.ru/73695.html> (ЭБС «IPRbooks»).