

ЛАБОРАТОРНАЯ РАБОТА №8

«Прототип информационной системы в среде VISUAL STUDIO»

Интерфейс пользователя информационной системы

В системах построенных по технологии клиент-сервер существует два вида интерфейса:

Интерфейс, реализуемый при помощи клиентского приложения;

- Web-интерфейс.
- Интерфейс, реализуемый при помощи клиентского приложения – это компьютерная программа,

устанавливаемая на клиентские компьютеры, предназначенная для работы с файлами данных через сеть. Основными элементами клиентских приложений являются формы (окно программы) и отчёты.

Элементы управления на форме называется объектами. Каждый объект обладает своим набором свойств, событий и методов.

- Свойства объекта – это его характеристики (высота, ширина и т.д.).
- События объекта – это события операционных систем или события инициируемые пользователем, на которые может реагировать объект (нажатие кнопки).
- Методы объекта – действия, которые можно производить с объектом в ходе выполнения программ.

В БД все объекты форм делятся на два класса:

- Объекты управления – объекты, осуществляющие управление БД (Например, Кнопка или Выпадающий список).
- Объекты для отображения информации – элементы, отображающие содержимое таблиц, запросов или фильтров, позволяющие добавлять изменять и удалять информацию, и проводить ее анализ.

Все формы в клиентском приложении делятся на три группы:

1 Формы для работы с данными – формы, содержащие как объекты управления, так и объекты просмотра данных. Такие формы предназначены для отображения, изменения, удаления и анализа данных.

2 Кнопочные формы – формы, содержащие только объекты управления, предназначены для открытия всех других форм. Кнопочная

форма, которая появляется первой после запуска программы, называется, главной кнопочной формой.

3 Информационные и служебные формы – формы, содержащие только элементы управления, предназначены для отображения служебной информации (справки), несвязанной с таблицами, запросами и фильтрами, либо для выполнения служебных операций, не связанных с данными (Например: форма с калькулятором).

Существует два вида дизайна форм:

- 1 Ленточные формы - формы, выводящие информацию по одной записи.
- 2 Табличные формы – формы, выводящие информацию в виде таблицы.

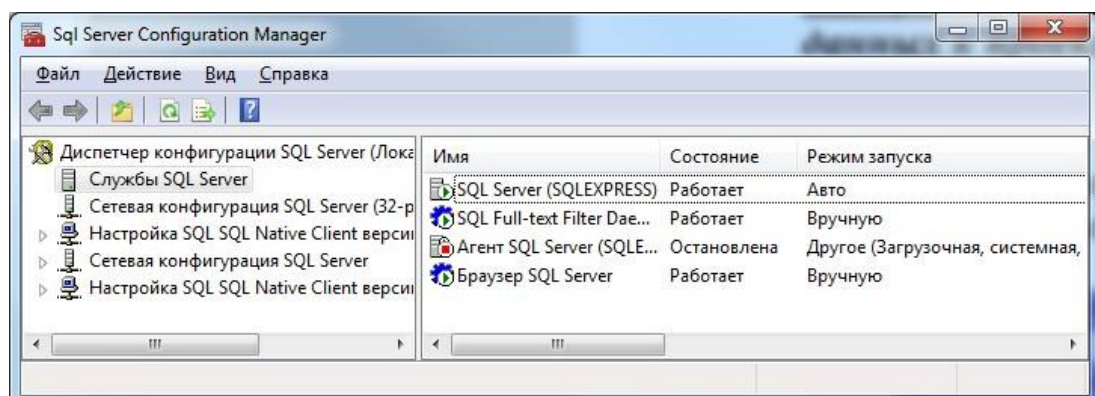
Объекты связи используются только в клиентском интерфейсе. В web-интерфейса функции объекта связи выполняет сервер.

Основой web-интерфейса являются страницы (файл с расширенным htm или html). Работа со страницами осуществляется с помощью программы – браузера. Изначально страницы находятся на сервере, пользователь сначала загружает их на свой компьютер с сервера, а затем с помощью страниц пользователь работает с файлом данных.

В web-интерфейсе отсутствуют отчёты, их роль выполняют сами страницы.

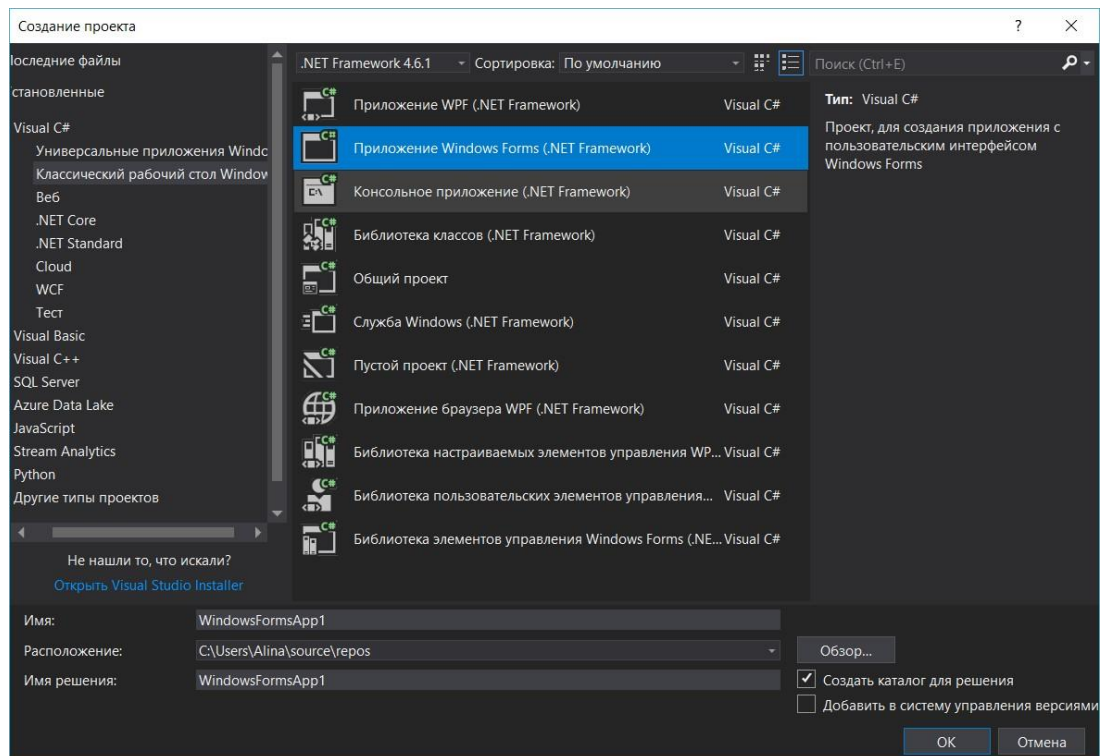
Подключения файла данных к проекту

Теперь рассмотрим создание пользовательского интерфейса БД в Microsoft Visual Studio. Прежде чем создавать новый проект в Microsoft Visual Studio, необходимо запустить службу Браузер SQL Server (SQL Server Browser) в утилите Диспетчер конфигурации SQL Server. Данная служба обеспечивает доступ сторонним приложениям к Microsoft SQLServer.

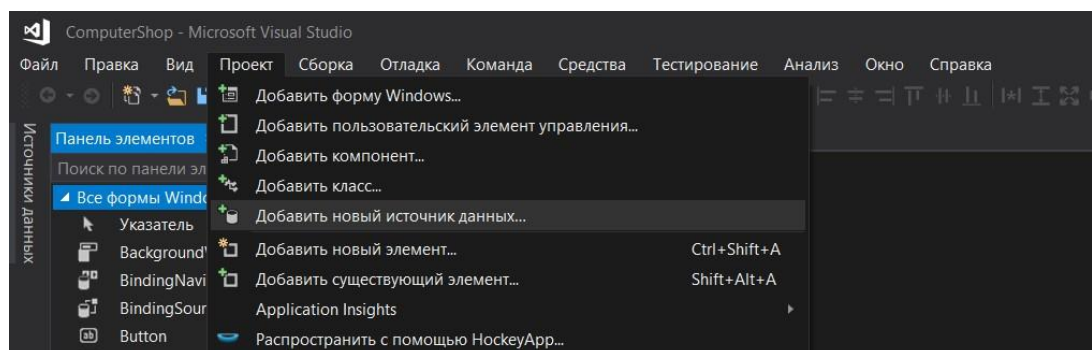


Теперь перейдём непосредственно к созданию пользовательского интерфейса БД

«Computer Shop» в Microsoft Visual Studio. Для начала необходимо создать новый проект. Запустите Microsoft Visual Studio и выберите Файл > Создать > Проект. В окне Выбор проекта выберите Приложение Windows Forms:



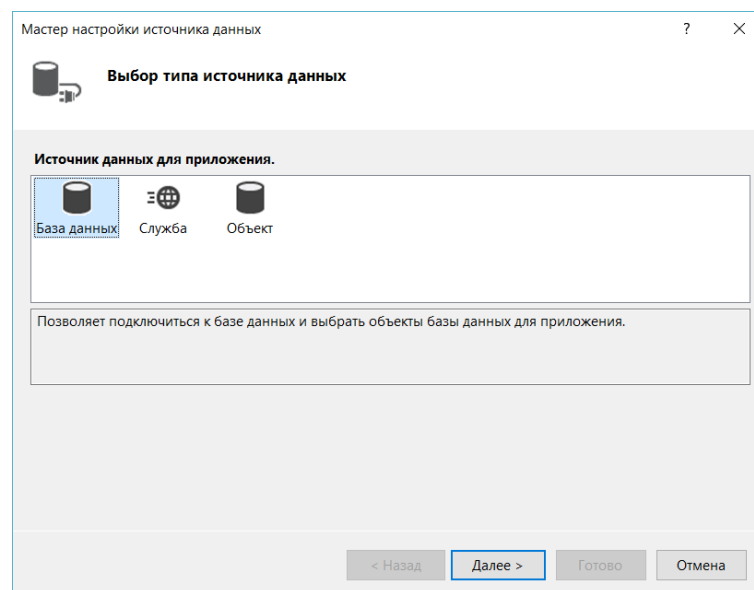
После создания нового проекта необходимо подключить к проекту созданную ранее в Microsoft SQL Server БД «Computer Shop». Для подключения БД к проекту в оконном меню среды разработки выберите Проект > Добавить новый источник данных:



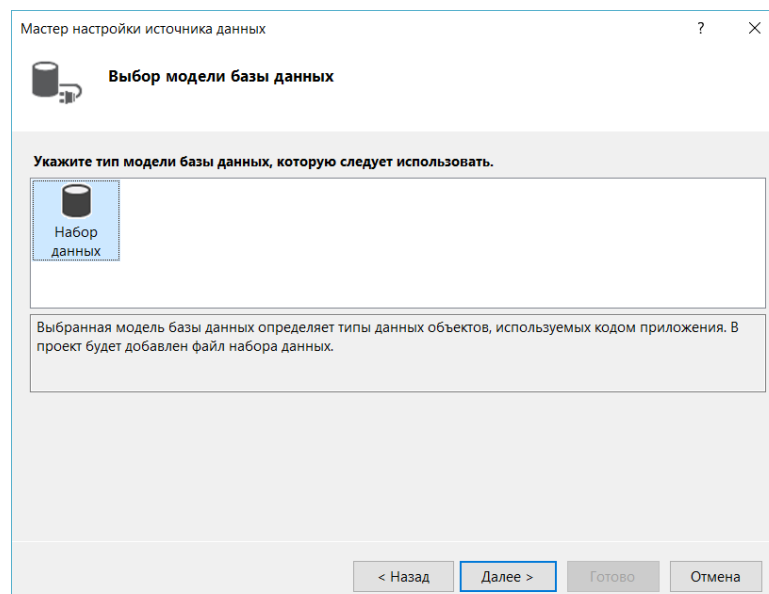
В окне Выбор типа источника данных (Choose a Data Source Type) можно выбрать один из трёх видов источников данных:

- База данных (Database);
- Служба (Service);
- Объект (Object).

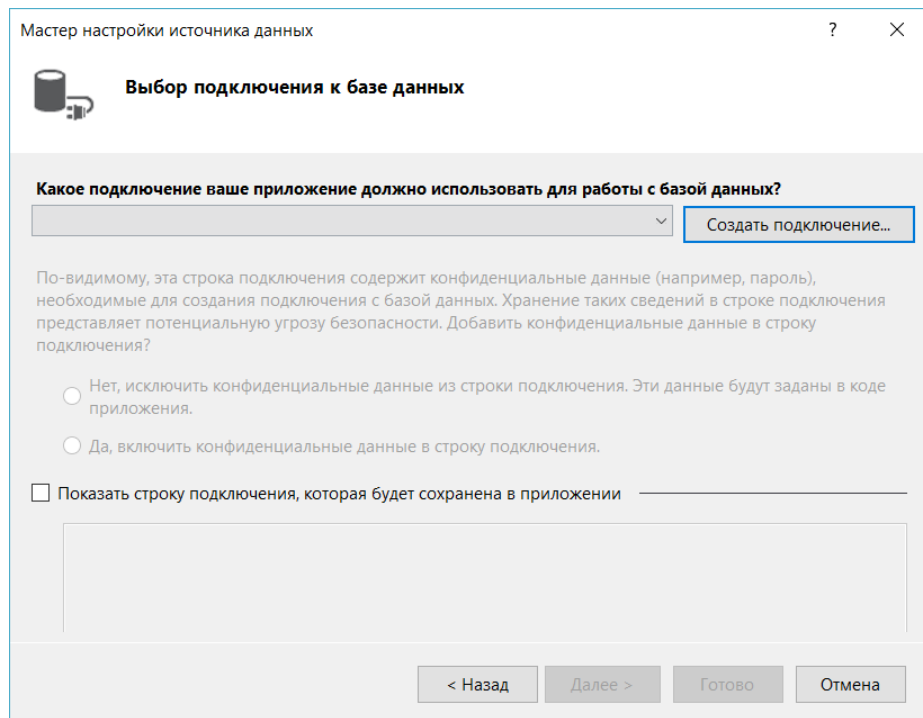
Так как мы подключаем наш проект к БД «Computer Shop», то выбираем вариант База данных (Database) и нажимаем кнопку Далее. Появится окно Выбора подключения БД (Choose Your Data Connection):



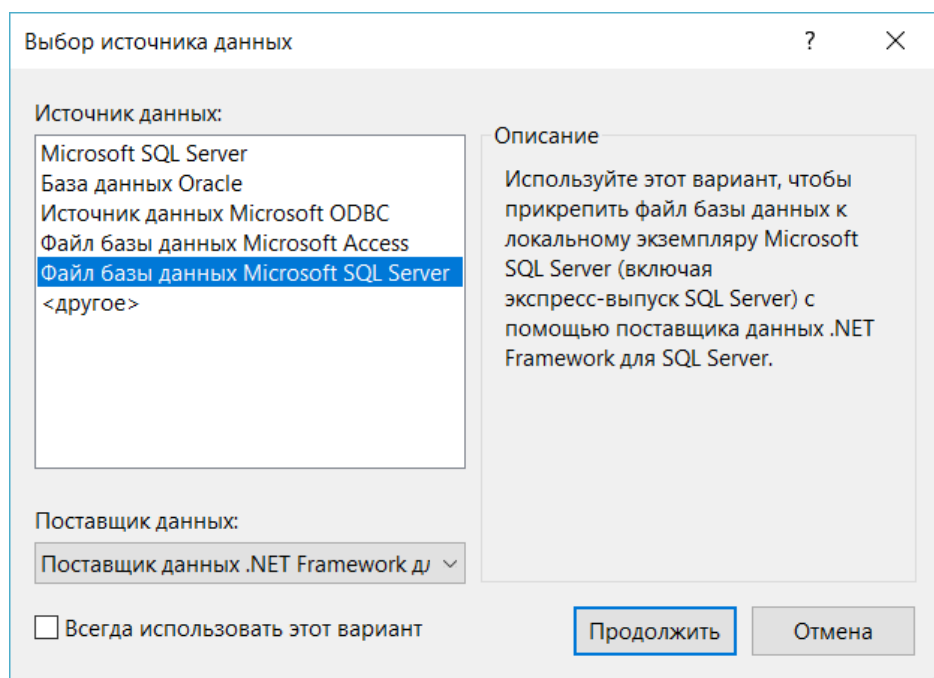
Дальше необходимо указать тип модели БД, которую следует использовать. Выберите «Набор данных».



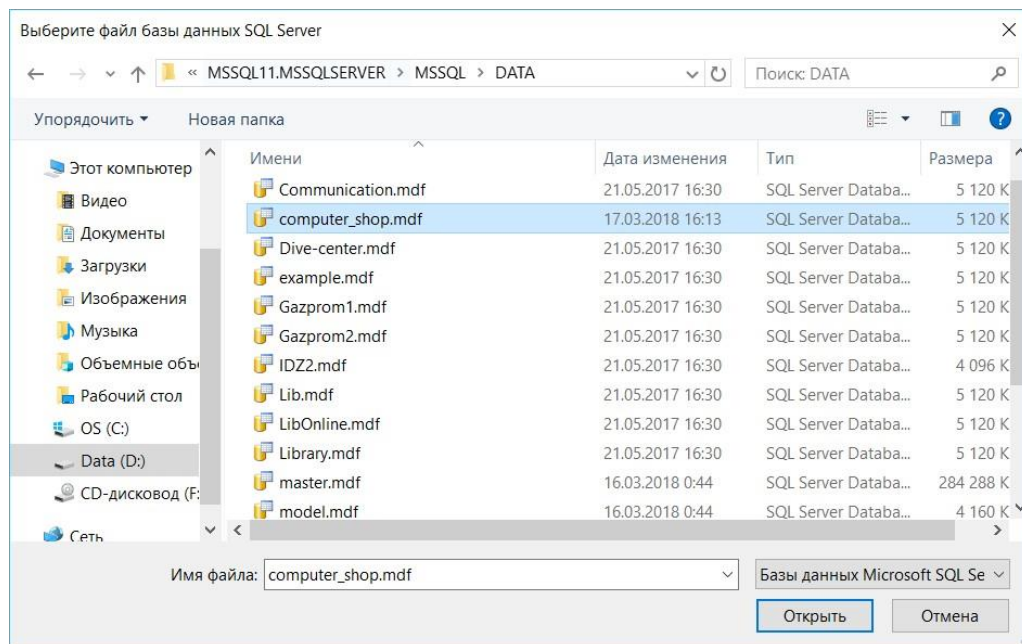
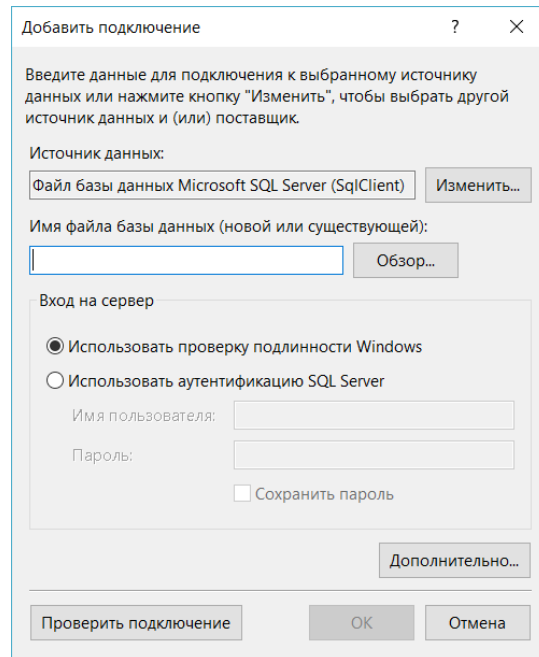
В окне выбора подключения к БД, для создания нового подключения нажмите кнопку Создать подключение (New Connection). Появится окно добавления нового источника данных:



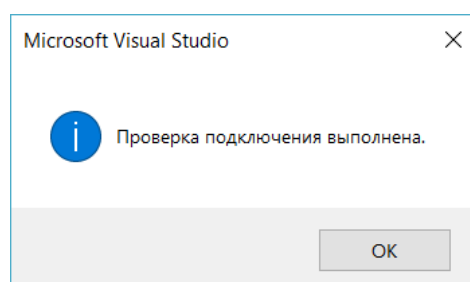
Выберите Файл базы данных Microsoft SQL Server и нажмите «Продолжить»:



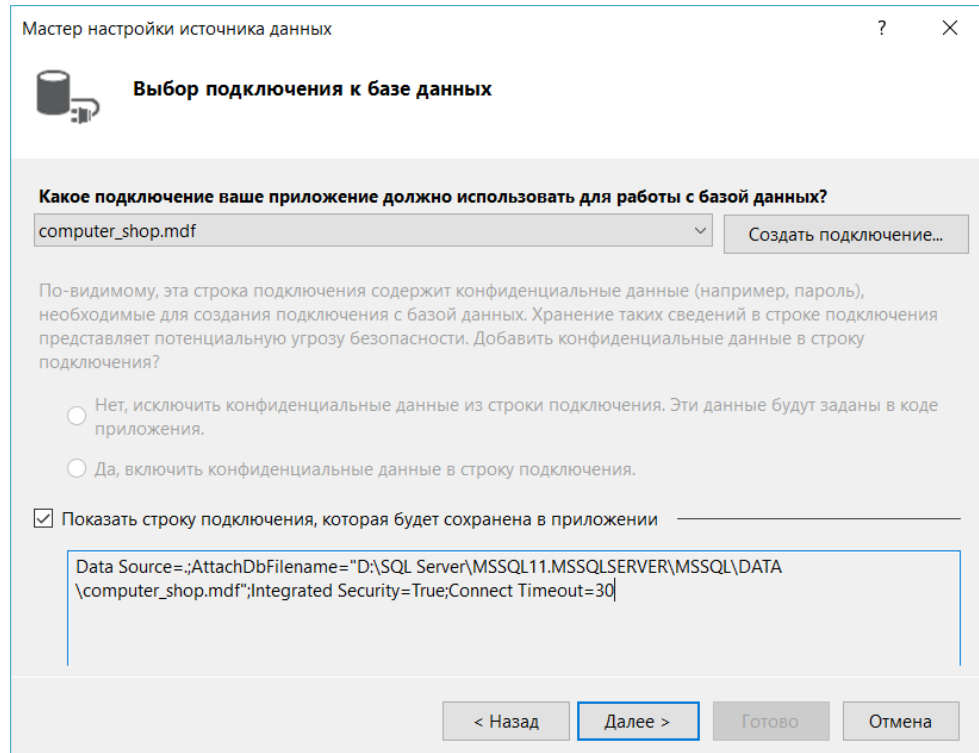
Нажмите Обзор и выберите файл базы данных:



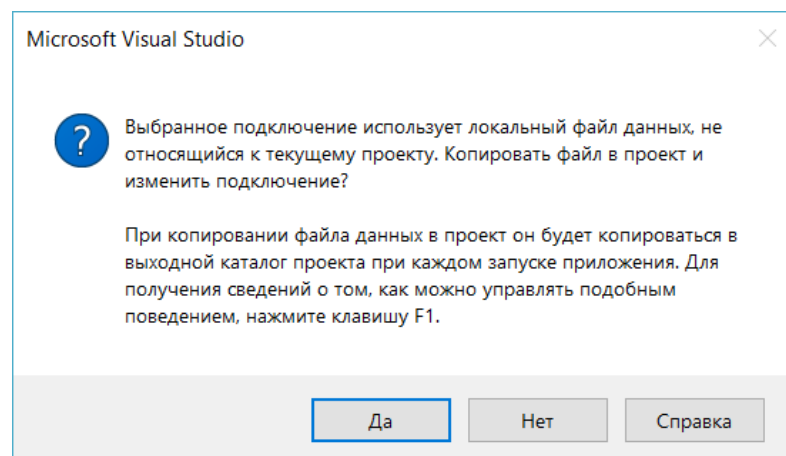
Для проверки работоспособности создаваемого соединения нажмите кнопку Проверить подключение. Появится сообщение Проверка подключения выполнена, говорящее о том, что соединение работоспособно.



Закройте окно, , а затем в окне добавления нового соединения нажмите кнопку ОК. Произойдёт возвращение к окну выбора подключения к БД. Просмотрите созданную строку подключения, щёлкнув по знаку «+» в нижней части окна:



В следующем диалоговом окне (рисунок 16) выберите Нет.



Появится окно с запросом о сохранении строки подключения в файле конфигурации приложения. Для сохранения строки подключения включите опцию Да, сохранить подключение как и нажмите кнопку Далее. Появится окно выбора объектов подключаемой БД. Выберите все объекты к и нажмите кнопку Готово. Подключение завершено.

Мастер настройки источника данных

Сохранение подключения в файле конфигурации приложения

Хранение строк с параметрами подключений в файле конфигурации приложения облегчает сопровождение и развертывание. Чтобы сохранить строку подключения в файле конфигурации приложения, введите имя в это поле, а затем нажмите кнопку "Далее".

Сохранить строку подключения в файле конфигурации приложения?

☒ Да, сохранить подключение как

< Назад **Далее >** Готово Отмена

Мастер настройки источника данных

Выбор объектов базы данных

Объекты базы данных для набора данных

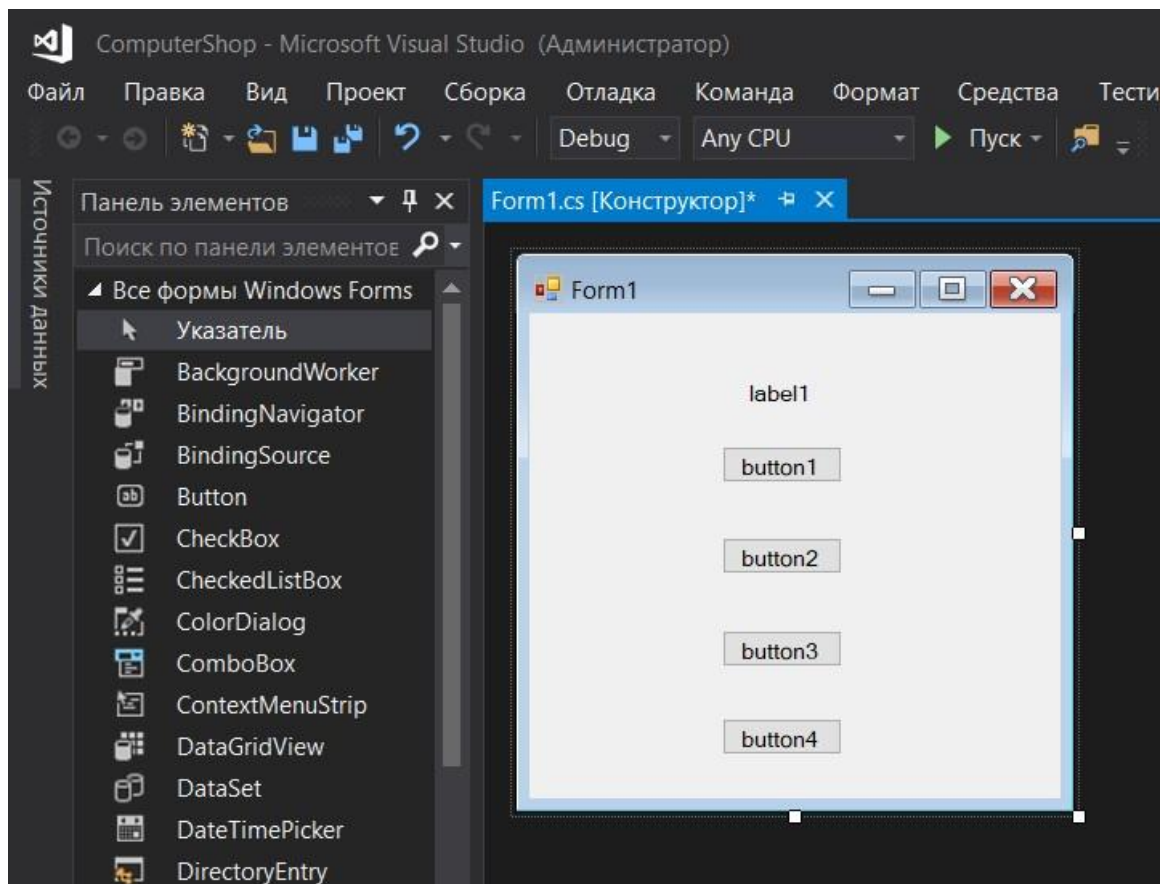
- ☒ Таблицы
- ☒ Представления
- ☒ Хранимые процедуры
- ☒ Функции

Имя набора данных (DataSet):

< Назад Далее > **Готово** Отмена

Главная кнопочная форма. Создание простых ленточных форм для работы с данными

Поместите на форму надпись (Label) и четыре кнопки (Button) как показано на рисунке:



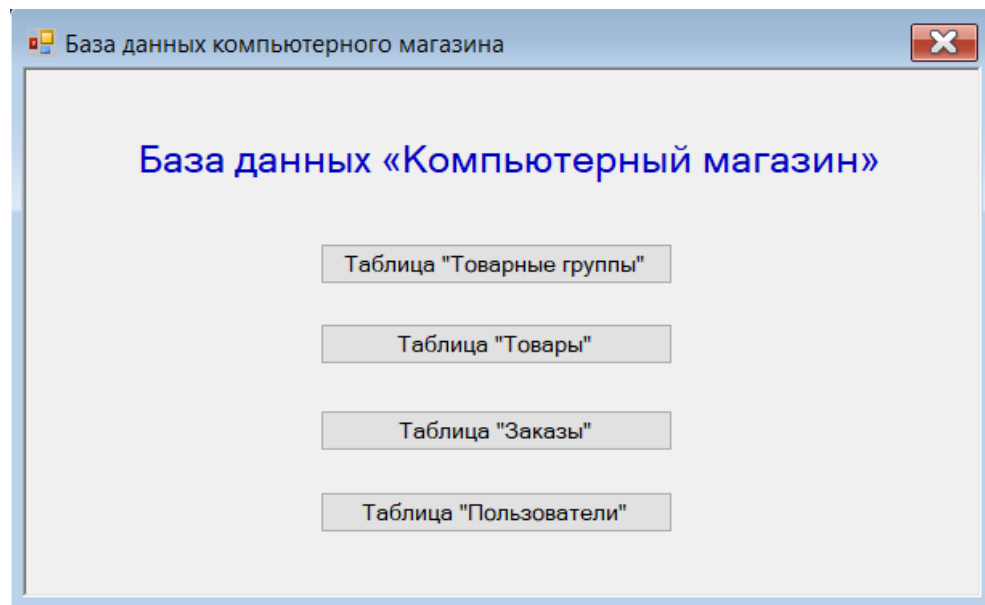
После создания элементов перейдём к настройке их свойств. Начнём с настройки свойств формы. Выделите форму, щёлкнув ЛКМ в пустом месте формы. На панели свойств задайте свойства формы как представлено ниже:

- `FormBorderStyle` (Стиль границы формы): `Fixed3D`;
- `MaximizeBox` (Кнопка развёртывания формы во весь экран): `False`;
- `MinimizeBox` (Кнопка свёртывания формы на панель задач): `False`;
- `Text` (Текст надписи в заголовке формы): База данных компьютерного магазина.

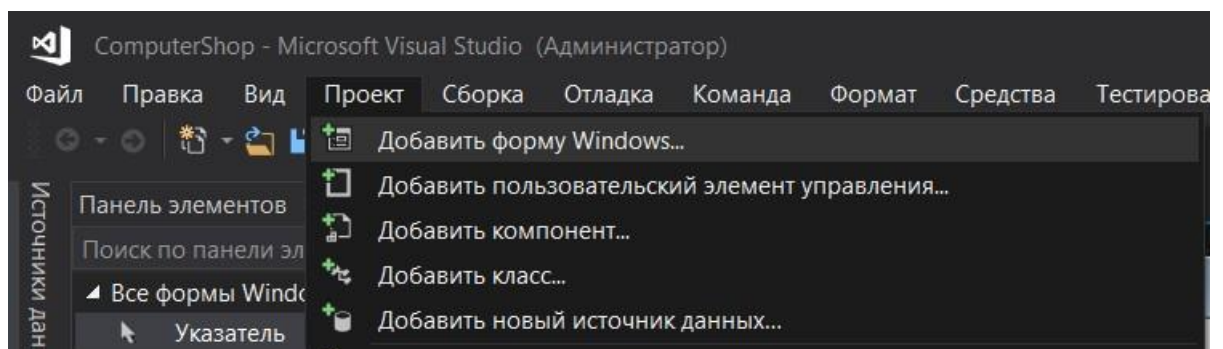
На форме выделите надпись `label1`, щёлкнув по ней ЛКМ и на панели свойств, задайте свойства надписи следующим образом:

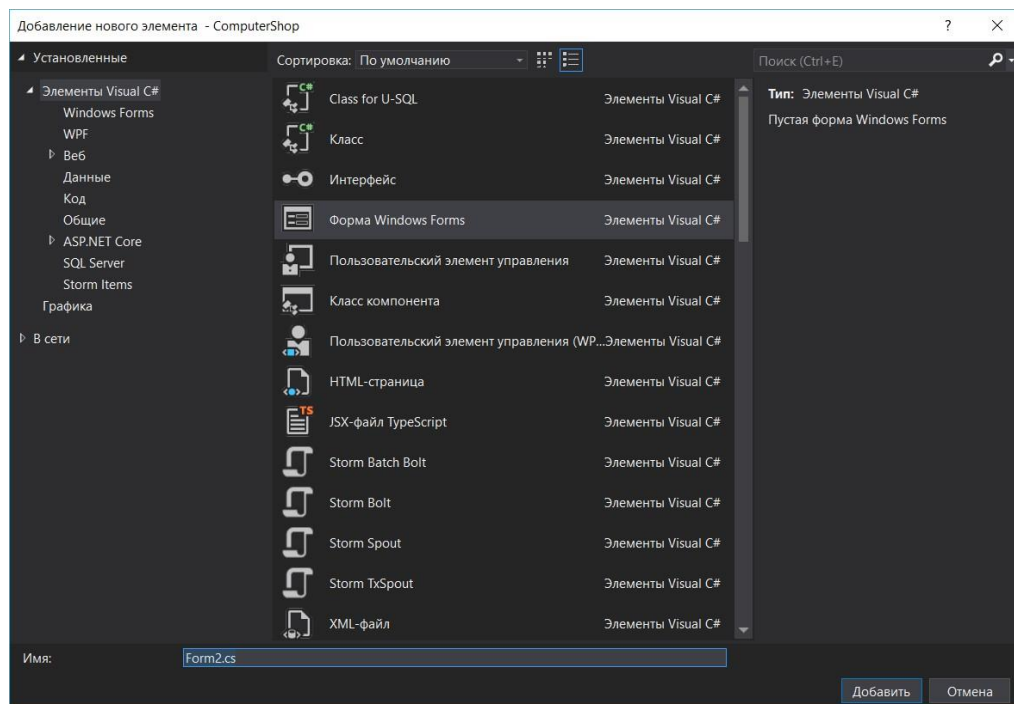
- `AutoSize` (Авторазмер): `False`;
- `Font` (Шрифт): `Microsoft Sans Serif`, размер 14;
- `ForeColor` (Цвет текста): Тёмно синий (0; 0; 192);
- `Text` (Текст надписи): База данных «Компьютерный магазин»;
- `TextAlign` (Выравнивание текста): `MiddleCenter`.

У кнопок задайте надписи (свойство «`Text`»), как показано на рисунке 20.



Теперь перейдём к созданию простых ленточных форм для работы с данными. Для начала создадим ленточную форму, отображающую таблицу «Товарные группы». Добавим в проект новую пустую форму. Для этого в оконном меню выберите пункт «Project/Add Windows Form». Появится окно «Add New Item - StudentsDB» (Добавить новый компонент):





После создания элементов перейдём к настройке их свойств. Начнём с настройки свойств формы. Выделите форму, щёлкнув ЛКМ в пустом месте формы. На панели свойств задайте свойства формы как представлено ниже:

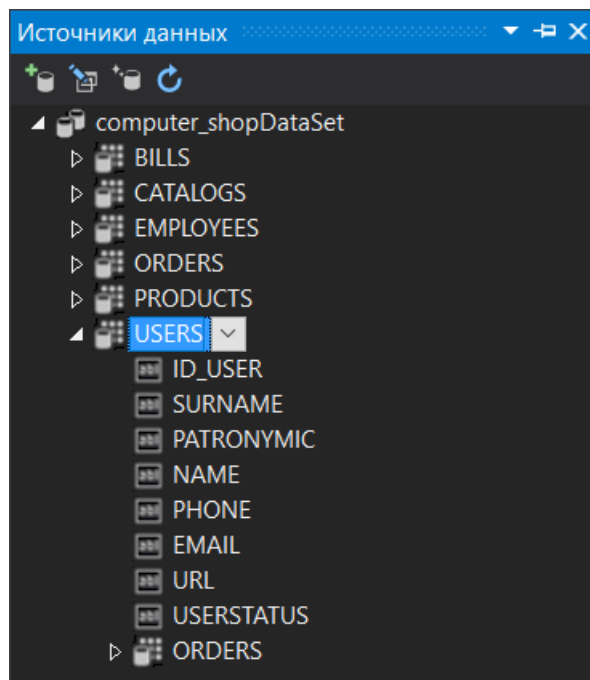
- **FormBorderStyle** (Стиль границы формы): **Fixed3D**;
- **MaximizeBox** (Кнопка развёртывания формы во весь экран): **False**;
- **MinimizeBox** (Кнопка свёртывания формы на панель задач): **False**;
- **Text** (Текст надписи в заголовке формы): **Пользователи**;
- **(Name)** (Наименование формы): **Users**.

На форме выделите надпись **label1**, щёлкнув по ней ЛКМ и на панели свойств, задайте свойства надписи следующим образом:

- **AutoSize** (Авторазмер): **False**;
- **Font** (Шрифт): **Microsoft Sans Serif**, размер **14**;
- **ForeColor** (Цвет текста): **Тёмно синий (0; 0; 192)**;
- **Text** (Текст надписи): **Пользователи**;
- **TextAlign** (Выравнивание текста): **MiddleCenter**.

Теперь поместим на форму поля таблицы «Пользователи». Сначала откройте панель

«Источники данных» (**Data Sources**). На панели «Источники данных» отобразите поля таблицы «Пользователи», щёлкнув по значку «+», расположенному слева от имени таблицы.



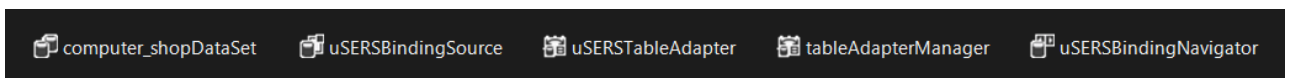
Под полями таблицы «Пользователи» в виде подтаблицы располагается таблица «Заказы». Подтаблица показывает, что таблица «Заказы» является вторичной по отношению к таблице «Пользователи».

Для того чтобы поместить на новую форму поля таблицы их необходимо перетащить из панели «Источники данных» на форму. Из таблицы «Пользователи» перетащите мышью на форму поля «Фамилия», «Отчество», «Имя», «Телефон», «E-mail», «URL» и «Статус». Форма примет вид, представленный на рисунке:

Мы не помещаем поле «Код пользователя» на нашу форму, так как данное поле является первичным полем связи и заполняется автоматически. Конечный пользователь не должен видеть такие поля.

Обратите внимание, что после перетаскивания полей с панели «Источники данных» на форму в верхней части формы появилась навигационная панель, а в нижней части рабочей области среды разработки появились пять невидимых объектов. Эти объекты предназначены для связи нашей формы с таблицей «Пользователи», расположенной на сервере. Рассмотрим функции этих объектов:

- COMPUTER_SHOPDataSet (Набор данных COMPUTER_SHOP) – обеспечивает подключение формок конкретной БД на сервере (в нашем случае это БД COMPUTER_SHOP);
- USERSBindingSource (Источник связи для таблицы «Пользователи») – обеспечивает подключение к конкретной таблице (в нашем случае к таблице специальности), а также позволяет управлять таблицей;
- USERSTableAdapter (Адаптер таблиц для таблицы «Пользователи») – обеспечивает передачу данных с формы в таблицу и наоборот.
- TableAdapterManager (Менеджер адаптера таблиц) – управляет работой объекта USERSTableAdapter;
- USERSBindingNavigator (Панель управления таблицей «Пользователи») – голубая панель с кнопками управления таблицей, расположенная в верхней части формы.



Теперь нам необходимо проверить работоспособность новой формы. Для отображения формы «Пользователи» её необходимо подключить к главной кнопочной форме, а затем запустить проект и открыть форму «Специальности» при помощи кнопки наглавной кнопочной форме.


Отобразите главную кнопочную форму в рабочей области среды разработки, щёлкнув по вкладке «Form1.cs [Конструктор]» в верхней части рабочей области. Для подключения новой формы «Пользователи» к главной кнопочной форме дважды щёлкните ЛКМ по кнопке «Таблица

«Пользователи»», расположенной на главной кнопочной форме. В появившемся окне кода формы в процедуре «button4_Click» наберите код, предназначенный для открытия формы «Таблица «Пользователи»» (Users), как это показано в листинге 1.

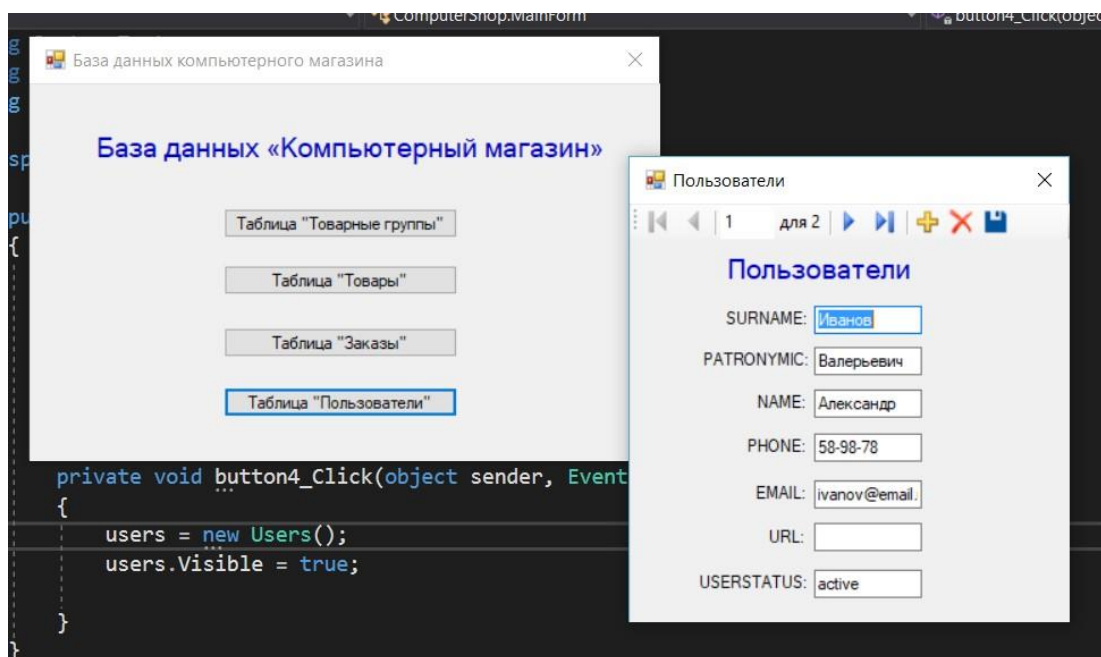
Листинг 1 – Код, предназначенный для открытия формы Users

```
public partial class MainForm : Form
{
    private Users users;

    public MainForm()
    {
        InitializeComponent();
    }
    private void button4_Click(object sender, EventArgs e)
    {
        users = new Users();
        users.Visible = true;
    }
}
```

Теперь запустим проект, нажав на панели инструментов кнопку . На экране появится главная кнопочная форма. Для открытия формы, отображающей таблицу

«Пользователи» на главной кнопочной форме нажмите кнопку «Таблица «Пользователи »». Появится форма с соответствующей таблицей:



Проверьте работу панели навигации, расположенной в верхней части формы, понажимав на ней различные кнопки. Вернитесь в среду разработки,

просто закрыв форму с таблицей «Пользователи» и главную кнопочную форму.

Подобным образом создайте формы для остальных таблиц БД и подключите их к главной кнопочной форме (листинг 2).

Листинг 2 – Код для открытия форм

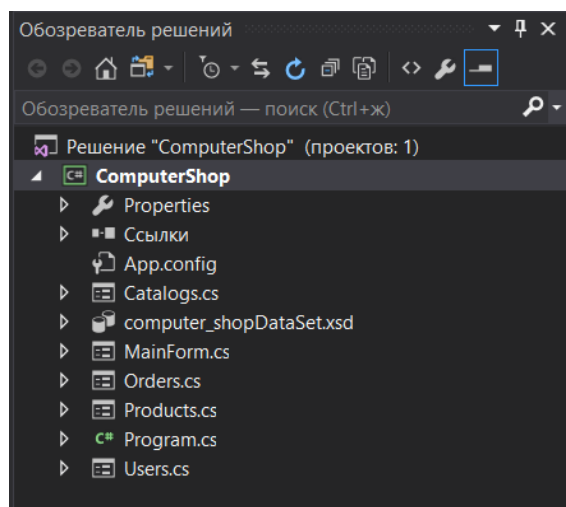
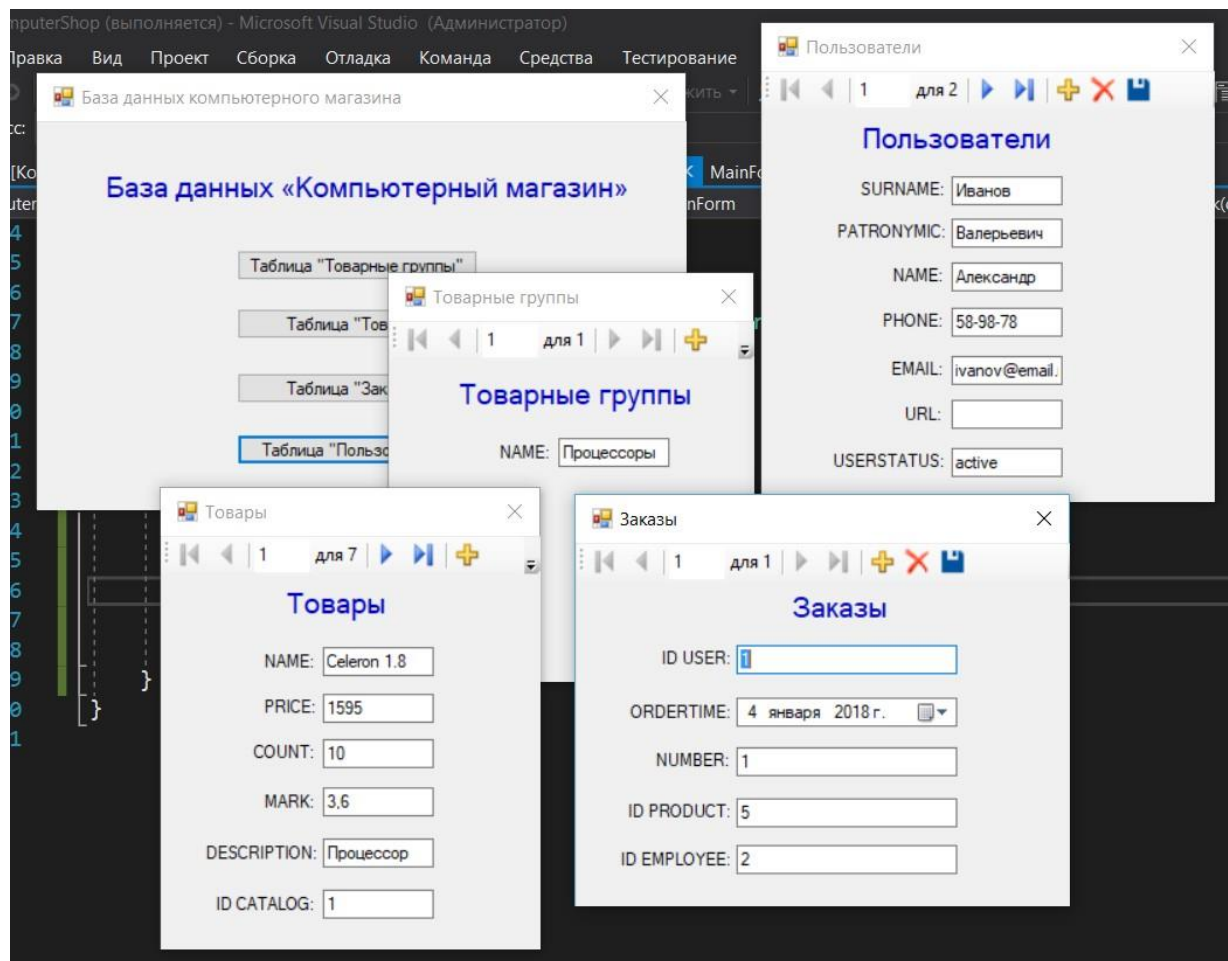
```
namespace WindowsFormsApplication1
{
    public partial class MainForm : Form
    {
        private Users users;
private Catalogs catalogs;
private Products products;
private Orders orders;

        public MainForm()
        {
            InitializeComponent();
        }
        private void button4_Click(object sender, EventArgs e)
        {
            users = new Users();
users.Visible = true;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            catalogs = new
Catalogs(); catalogs.Visible =
true;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            products = new
Products(); products.Visible =
true;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            orders = new Orders();
orders.Visible = true;
        }
    }
}
```

Создание сложных ленточных форм для работы с данными

Модернизируем форму для таблицы «Товары». Сначала программно продублируем кнопки панели навигации, расположенной в верхней части формы.

В качестве надписей на созданных кнопках задайте: «Первая», «Предыдущая», «Добавить», «Последняя», «Следующая», «Удалить» и «Сохранить»:

Дважды щёлкните ЛКМ по кнопке «Первая» и в появившемся окне кода формы

«Products» в процедуре «Button1_Click» наберите команду для перехода к первой записи

«pPRODUCTSBindingSource.MoveFirst()» (листинг 3). Подобным образом добавьте команды для остальных кнопок (листинг 4).

Листинг 3 – Команда для перехода к первой записи

```
private void button1_Click(object sender, EventArgs e)
{
    pPRODUCTSBindingSource.MoveFirst();
}
```

Листинг 4 – Команды для остальных кнопок

```
private void button2_Click(object sender, EventArgs e)
{
    pPRODUCTSBindingSource.MovePrevious();
}

private void button3_Click(object sender, EventArgs e)
{
    pPRODUCTSBindingSource.MoveNext();
}

private void button4_Click(object sender, EventArgs e)
{
    pPRODUCTSBindingSource.MoveLast();
}

private void button5_Click(object sender, EventArgs e)
{
    pPRODUCTSBindingSource.AddNew();
}

private void button6_Click(object sender, EventArgs e)
{
    pPRODUCTSBindingSource.RemoveCurrent();
}

private void button7_Click(object sender, EventArgs e)
{
    //проверяет введённые в поля данные на соответствие типам
данных полейthis.Validate();
    //закрывает подключение с сервером
this.pPRODUCTSBindingSource.EndEdit();
    //обновляет данные на сервере
this.tableAdapterManager.UpdateAll(this.computer_shopDataSet);
}
```

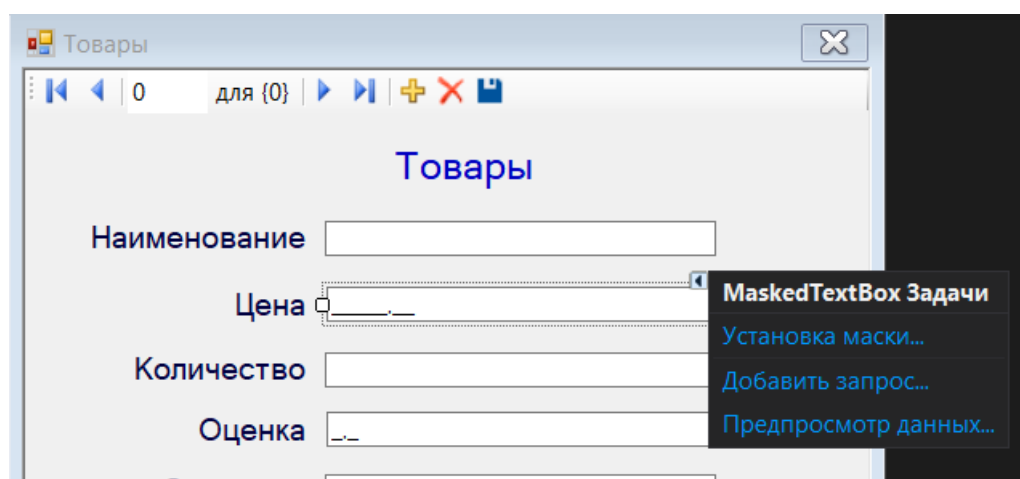
Для проверки работы созданных кнопок запустите проект откройте форму «Таблица «Товары»» и нажмите каждую из кнопок.

Теперь изменим объекты, отображающие поля для более удобного ввода информации. Для начала удалите текстовые поля ввода (TextBox), отображающие следующие поля таблицы «Products»: «Цена», «Оценка» и «Товарные группы».

Для отображения полей «Цена» и «Оценка» будем использовать текстовые поля ввода по маске (MaskedTextBox). Объект текстовое поле ввода по маске отсутствует в выпадающем списке объектов для отображения полей в окне «Источники данных», поэтому будем создавать данные объекты при помощи панели объектов (Toolbox), а затем подключать их к соответствующим полям вручную. Для создания текстовых полей ввода по маске на панели объектов используется кнопка

Создайте текстовые поля ввода по маске справа от надписей «Цена» и «Оценка».

Теперь у созданных объектов настроим маски ввода. Начнём с объекта, отображающего цену. На форме выделите соответствующее полю «Цена» текстовое поле ввода по маске. Для задания маски в меню действий с объектом выберите пункт «Установка маски...»



После этого на экране появится окно задания маски «Маска ввода»

Символы для создания масок ввода приведены в таблице 1.

Маска ввода

?

×

Выберите готовое описание маски из списка ниже или задайте специальную маску, выбрав команду "Специальный".

Описание маски	Формат данных	Проверка типа
Numeric (5-digits)	12345	Int32
Phone number	(574) 555-0123	(нет)
Phone number no area code	555-0123	(нет)
Short date	12/11/2003	DateTime
Short date and time (US)	12/11/2003 11:20	DateTime
Social security number	000-00-1234	(нет)
Time (European/Military)	23:20	DateTime
Time (US)	11:20	DateTime
Zip Code	98052-6399	(нет)
<Специальный>		(нет)

Маска:

☒
Использовать ValidatingType

Предварительный просмотр:

OK

Отмена

Таблица 1 – Символы для создания масок ввода

Знак	Описание
0	Пользователь должен ввести цифру (от 0 до 9).
9	Пользователь может ввести цифру (от 0 до 9).
#	Пользователь может ввести цифру, пробел, знак "плюс" или "минус". Если ничего не ввести, будет вставлен пробел.
L	Пользователь должен ввести букву.
?	Пользователь может ввести букву.
A	Пользователь должен ввести букву или цифру.
a	Пользователь может ввести букву или цифру.
&	Пользователь должен ввести какой-либо знак или пробел.
C	Пользователь может ввести знаки или пробелы.
.,;-	Разделитель целой и дробной части, групп разрядов, значений дат и времени.
/	Выбираемый знак зависит от региональных параметров Windows.
>	Все последующие знаки будут переведены в верхний регистр.
<	Все последующие знаки будут переведены в нижний регистр.
!	Маска ввода заполняется слева направо, а не справа налево.
\	Знаки, следующие непосредственно за обратной косой чертой, отображаются без изменений.


""	Знаки, заключенные в двойные кавычки, отображаются без изменений.
----	---

В окне «Маска ввода» выберите маску «Специальный» и введите маску «99990.00» нажмите кнопку «ОК». Для поля оценка введите маску «0.0».

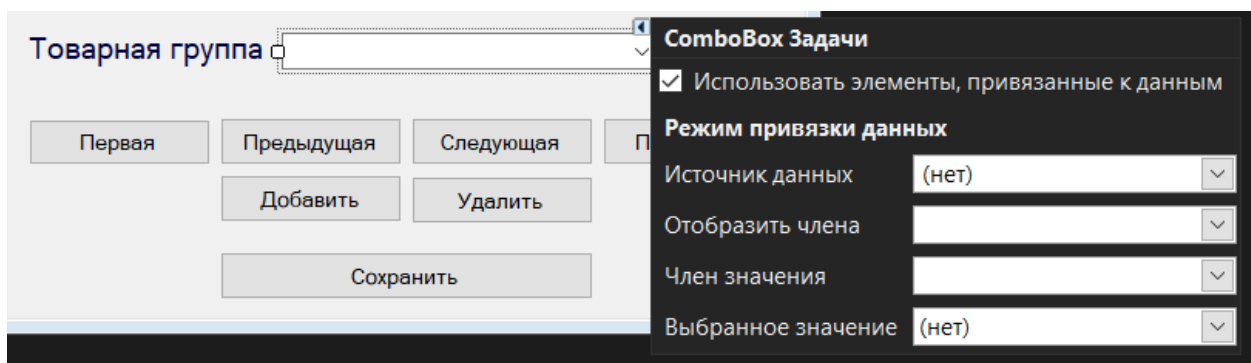
Теперь нам необходимо подключить созданные текстовые поля ввода по маске к соответствующим полям. Для этого с панели «Источники данных» (DataSources) перетащите поле «PRICE» на текстовое поле ввода по маске, расположенное справа от надписи «Цена». Прodelайте такую же операцию с полем «Оценка».

На данном этапе вместо Товарной группы на форме отображается только ее код. Исправим это с помощью добавления поля типа «Выпадающий список». При этом сам выпадающий список будет заполнен товарными группами из таблицы «Товарные группы» и при выборе товара её код будет автоматически подставляться в поле «Товарные группы».

Поместите справа от надписи «Товарная группа», неподключенный ни к каким

полям выпадающий список  **ComboBox**. После создание выпадающего списка подключим его к полю «Код товарный группы» из таблицы «Товары» и настроим заполнение списка значениями поля «Наименование товарной группы» из таблицы

«Товарные группы». Для этого выделите вновь созданный выпадающий список, отобразите меню действий и в меню действий включите опцию «Использовать элементы, привязанные к данным»:

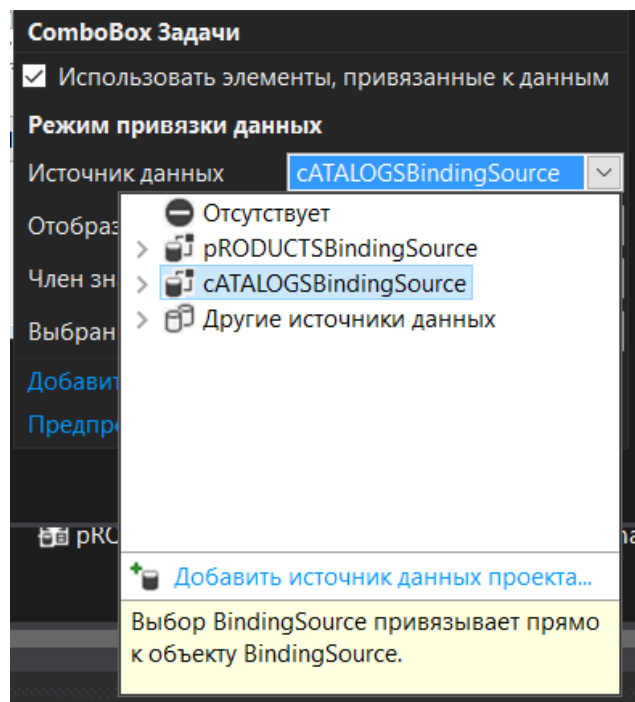


В панели действий под опцией «Использовать элементы, привязанные к данным» расположены следующие параметры:

- Data Source (Источник данных) – определяет таблицу или запрос из которого заполняется список;
- Display Member (Отобразить члена) – определяет поле значениями которого заполняется список;
- Value Member (Член значения) – определяет значения какого поля подставляются в связанное с выпадающим списком поле;
- Selected Value (Выбранное значение) – определяет связанное с выпадающим списком поле.

Для изменения параметров необходимо нажать кнопку в нутрии поля параметра.

Появится древовидная структура выбора источника данных:



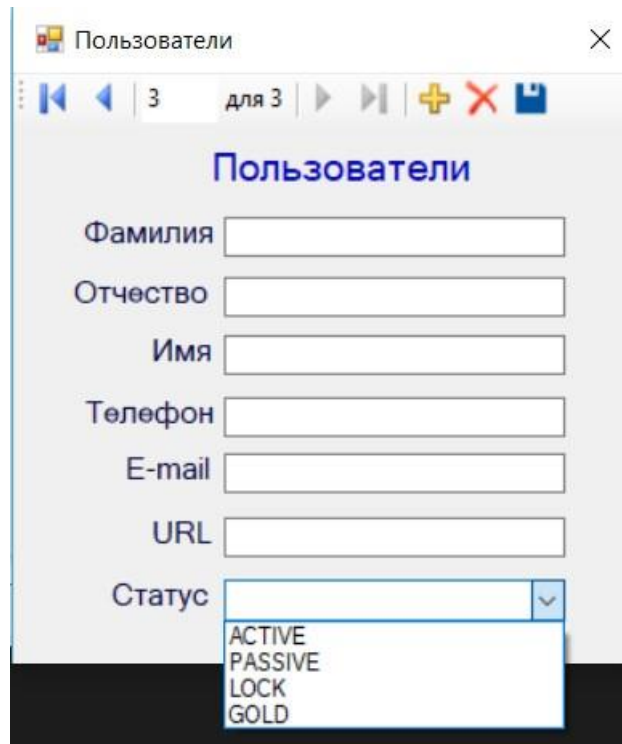
Заполните параметры согласно:

The image shows a software window titled 'Товары' (Goods). It contains several input fields: 'Наименование' (Name), 'Цена' (Price), 'Количество' (Quantity), 'Оценка' (Rating), and 'Описание' (Description). Below these is a 'Товарная группа' (Goods group) dropdown menu. At the bottom are buttons for 'Первая' (First), 'Предыдущая' (Previous), 'Следующая' (Next), 'Последняя' (Last), 'Добавить' (Add), 'Удалить' (Delete), and 'Сохранить' (Save). A context menu is open over the 'Товарная группа' dropdown, titled 'ComboBox Задачи' (Task ComboBox). It includes a checked option 'Использовать элементы, привязанные к данным' (Use elements bound to data), a 'Режим привязки данных' (Data binding mode) section with settings for 'Источник данных' (Data source: cATALOGSBindingSource), 'Отобразить члена' (Show member: NAME), 'Член значения' (Member value: ID_CATALOG), and 'Выбранное значение' (Selected value: pPRODUCTSBindingSource -). It also has links for 'Добавить запрос...' (Add query...) and 'Предпросмотр данных...' (Preview data...).

Рассмотрим на примере поля «Статус» в таблице «Пользователи» создание выпадающего списка с predetermined значениями. Для этого удалите поле ввода справа от строки «Статус» на форме «Пользователи».

The image shows a dialog box titled 'Редактор коллекции строк' (String collection editor). It has a text area with the instruction 'Введите строки в коллекцию (одну на строку):' (Enter lines in the collection (one per line):). The text area contains the following text: 'ACTIVE', 'PASSIVE', 'LOCK', and 'GOLD'. At the bottom are 'OK' and 'Отмена' (Cancel) buttons.

Добавьте поле типа выпадающий список и свяжите с полем БД, как Вы это делали с полями «Цена» и «Оценка». В свойстве поля Items укажите значения, которые будут содержаться в выпадающем списке. На рисунке показан результат добавления поля типа выпадающий список.



Изменим форму «Заказы». На данный момент на форме «Заказы» просто вынесены все поля таблицы «Orders». Однако для пользователя БД полей таблицы «Orders» недостаточно. Создадим в БД «COMPUTER_SHOP» представление (листинг 5), которое будет содержать всю необходимую информацию.

Листинг 5 – Представление ViewORDER


```
USE
COMPUTER_SHOPGO
CREATE VIEW
ViewORDERAS
SELECT PRODUCTS.NAME AS PRODUCTSNAME, PRODUCTS.DESRIPTION,
PRODUCTS.PRICE, ORDERS.NUMBER, ORDERS.ORDERTIME, USERS.SURNAME AS
USERSURNAME, USERS.NAME AS USERNAME, USERS.PATRONYMIC, USERS.PHONE,
EMPLOYEES.SURNAME AS EMPLOYEESURNAME, EMPLOYEES.POST
FROM ORDERS INNER JOIN PRODUCTS ON ORDERS.ID_PRODUCT =
PRODUCTS.ID_PRODUCT INNER JOIN EMPLOYEES ON ORDERS.ID_EMPLOYEE =
```

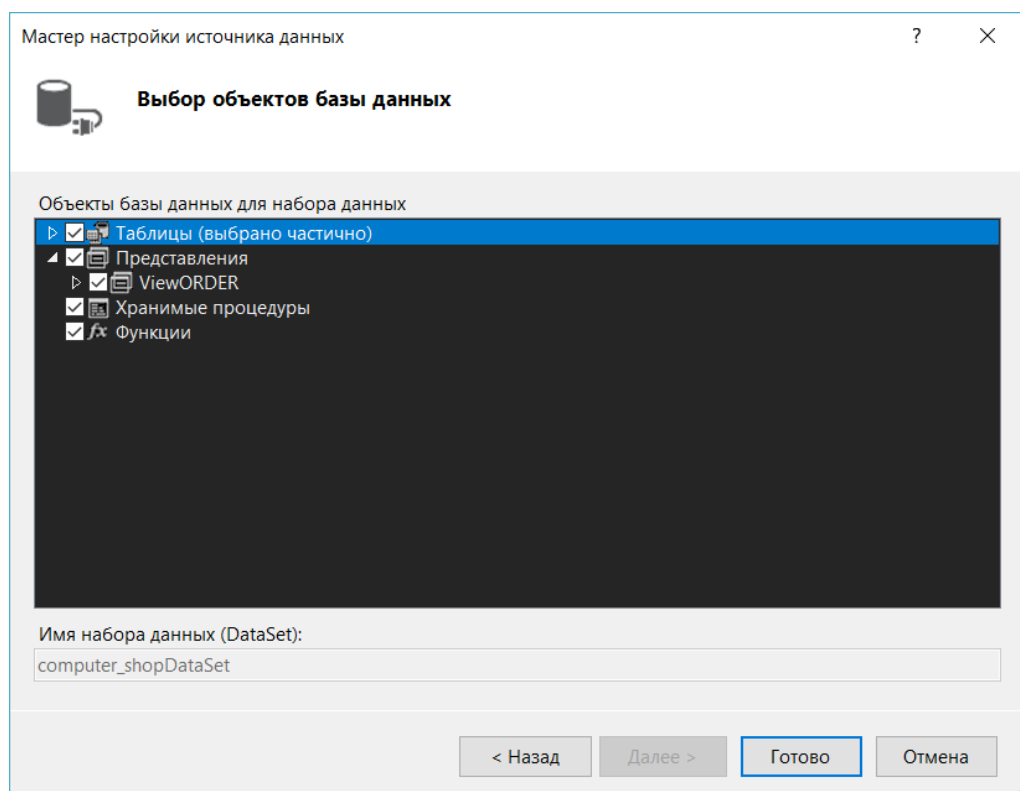

EMPLOYEES.ID_EMPLOYEE INNER JOIN USERS ON ORDERS.ID_USER =
dbo.USERS.ID_USER

GO

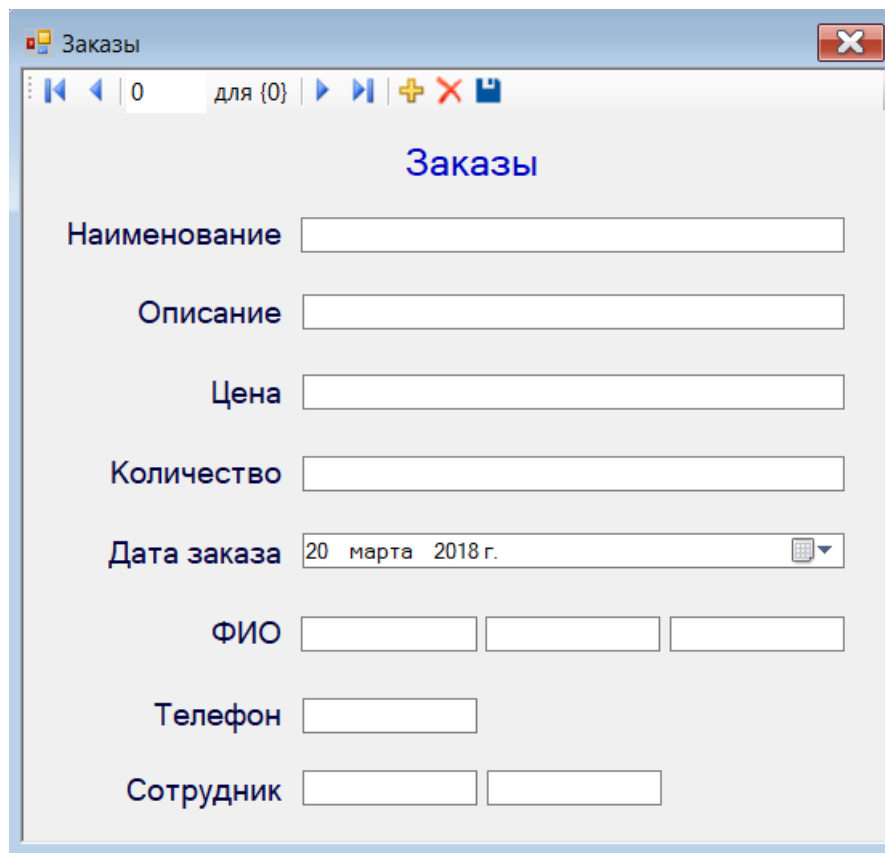
Содержимое представления ViewORDER показано на рисунке:

PRODUCTSNAME	DESCRIPTION	PRICE	NUMBER	ORDERTIME	USERSURNAME	USERNAME	PATRONYMIC	PHONE	EMPLOYEEESU...	POST
Intel Pentium 4 ...	Процессор Intel...	6147,00	1	2005-01-04 10:39:38...	Семенов ...	Игорь ...	Вячеславович ...	9056666100	Михайлова ...	Оператор...
Gigabyte GA-8I8...	Материнская п...	1896,00	2	2005-02-10 09:40:29...	Корнеев ...	Александр ...	Александрович ...	89-78-36	Михайлова ...	Оператор...
Maxtor 6Y120P0...	Винчестер 120 ...	2456,00	4	2005-02-18 13:41:05...	Иванов ...	Александр ...	Валерьевич ...	58-98-78	Михайлова ...	Оператор...
Maxtor 6B200P0...	Винчестер 200 ...	3589,00	1	2005-03-10 18:20:00...	Прокопчук ...	Сергей ...	Иванович ...	9057777777	Вязова ...	Оператор...
ASUSTEK A9600...	Видеоадаптер ...	5156,00	1	2005-03-12 19:15:36...	Семенов ...	Игорь ...	Вячеславович ...	9056666100	Вязова ...	Оператор...
Asustek P4P800...	Материнская п...	2518,00	2	2005-03-15 19:15:36...	Иванов ...	Александр ...	Валерьевич ...	58-98-78	Вязова ...	Оператор...
Epox EP-4PDA3I...	Материнская п...	2289,00	1	2005-03-18 19:15:36...	Семенов ...	Игорь ...	Вячеславович ...	9056666100	Михайлова ...	Оператор...
Intel Pentium 4 ...	Процессор Intel...	6147,00	3	2005-03-20 19:15:36...	Корнеев ...	Александр ...	Александрович ...	89-78-36	Михайлова ...	Оператор...
Celeron D 315 2...	Процессор Cele...	1880,00	1	2005-03-22 19:15:36...	Семенов ...	Игорь ...	Вячеславович ...	9056666100	Михайлова ...	Оператор...
Gigabyte GA-8I...	Материнская п...	2420,00	1	2005-03-25 19:15:36...	Лосев ...	Анатолий ...	Юрьевич ...	NULL	Вязова ...	Оператор...
DDR-400 512MB...	Оперативная п...	1932,00	4	2005-03-26 19:15:36...	Семенов ...	Игорь ...	Вячеславович ...	9056666100	Вязова ...	Оператор...
Seagate ST3120...	Винчестер 120 ...	2468,00	1	2005-04-01 19:15:36...	Петров ...	Макоим ...	Валерьевич ...	NULL	Вязова ...	Оператор...
Samsung SP081...	Винчестер 80 G...	2093,00	1	2005-04-10 19:15:36...	Лосев ...	Анатолий ...	Юрьевич ...	NULL	Михайлова ...	Оператор...
Maxtor 6B200P0...	Винчестер 200 ...	3589,00	2	2005-04-11 19:15:36...	Иванов ...	Александр ...	Валерьевич ...	58-98-78	Вязова ...	Оператор...
Samsung SP081...	Винчестер 80 G...	2093,00	1	2005-04-12 19:15:36...	Прокопчук ...	Сергей ...	Иванович ...	9057777777	Михайлова ...	Оператор...
Gigabyte GA-8I8...	Материнская п...	1896,00	1	2005-04-15 19:15:36...	Семенов ...	Игорь ...	Вячеславович ...	9056666100	Вязова ...	Оператор...
Celeron 2.0GHz ...	Процессор Cele...	1969,00	2	2005-04-16 19:15:36...	Семенов ...	Игорь ...	Вячеславович ...	9056666100	Михайлова ...	Оператор...
Celeron 1.8 ...	Процессор Cele...	1595,00	1	2005-04-17 19:15:36...	Петров ...	Макоим ...	Валерьевич ...	NULL	Вязова ...	Оператор...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Visual Studio. Нажмите  Настроить источники данных с помощью мастера на панели инструментов Источники данных. В мастере выберите созданное представление:



Изменим форму заказы. Обратите внимание, что кнопки на навигационной панели связаны с таблицей «Orders». Для того чтобы связать навигационную панель с представлением «ViewOrder» в меню задачи навигационной панели выберите пункт Правка элементов. В Свойстве BindingSource выберите viewORDERBindingSource.



Добавим поле, в котором будет высчитываться сумма заказа. При этом сумма заказа рассчитывается по нажатию кнопки «Рассчитать»:



Кликните 2 раза на кнопку «Рассчитать» и наберите команду для события «button1_Click» как показано в листинге 6.

Листинг 6 – Событие «button1_Click»

```
private void button1_Click(object sender, EventArgs e)
{
    //класс System.Convert позволяет преобразовывать несовместимые
    в C# типы данных textBox1.Text = Convert.ToString(
```

```
Convert.ToDouble(pRICETextBox.Text) *
Convert.ToDouble(NUMBERTextBox.Text));
}
```

Недостатком такой реализации является то, что пользователю каждый раз придется нажимать на кнопку «Рассчитать».

Еще одним способом реализации поля «Сумма заказа» является добавление данного поля в представление «ViewOrder».

Добавим в представление «ViewOrder» столбец, в который будет записываться сумма заказа. Для этого закройте Visual Studio, перезапустите службы SQL Server и браузер SQL Server в Диспетчере конфигурации SQL Server, запустите SQL Management Studio и добавьте столбец «SUM» в представление «ViewOrder» (либо с помощью запроса, либо с помощью мастера).

Листинг 7 – Создание представления «ViewORDER»

```
USE
COMPUTER_SHOPGO

ALTER VIEW
ViewORDERAS
SELECT PRODUCTS.NAME AS PRODUCTSNAME, PRODUCTS.DESCRPTION,
PRODUCTS.PRICE, ORDERS.NUMBER, ORDERS.ORDERTIME, USERS.SURNAME AS
USERSURNAME, USERS.NAME AS USERNAME, USERS.PATRONYMIC, USERS.PHONE,
EMPLOYEES.SURNAME AS EMPLOYEESURNAME, EMPLOYEES.POST, ORDERSUM =
(PRODUCTS.PRICE * ORDERS.NUMBER) FROM ORDERS INNER JOIN PRODUCTS ON
ORDERS.ID_PRODUCT = PRODUCTS.ID_PRODUCT
INNER JOIN EMPLOYEES ON ORDERS.ID_EMPLOYEE =
EMPLOYEES.ID_EMPLOYEE INNER JOIN USERS ON ORDERS.ID_USER =
dbo.USERS.ID_USER
GO
```

После редактирования таблицы закройте SQL Management Studio, перезапустите службы SQL Server и браузер SQL Server в Диспетчере конфигурации SQL Server.

В Мастере настройки источника данных выберите добавленный столбец «ORDERSUM». Перетащите столбец «ORDERSUM» с панели

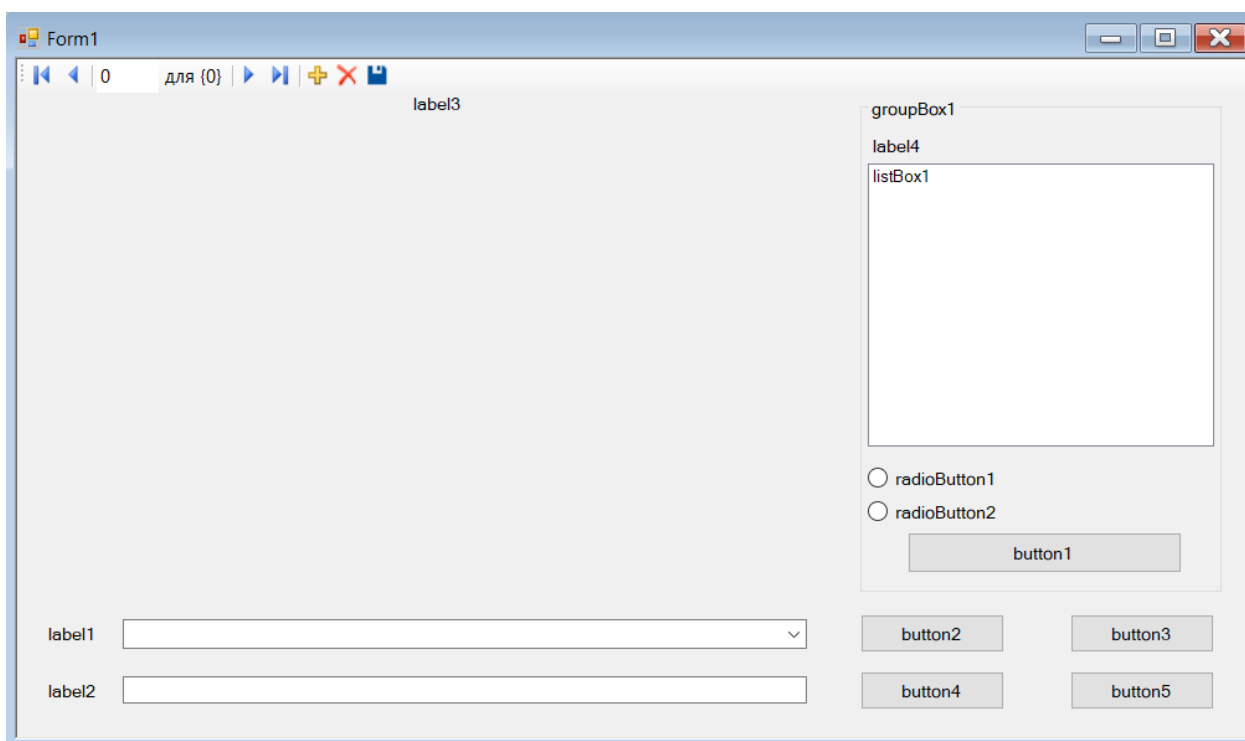
инструментов Источники данных на добавленное поле «Сумма заказа». Удалите кнопку «Рассчитать».

Создание табличных форм

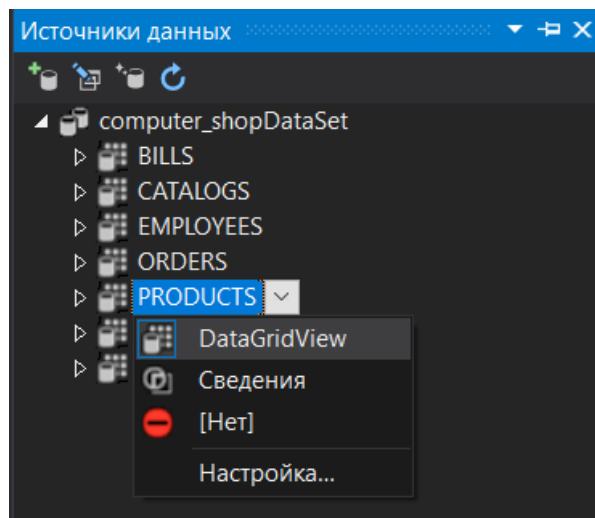
Перейдём теперь к созданию табличных форм для отображения данных. В данной главе также затрагиваются вопросы фильтрации и сортировки данных, а также реализуется поиск информации в таблице.

Рассмотрим создание табличной формы на примере формы, отображающей таблицу «Товары». Добавьте в проект новую форму и на неё поместите следующие объекты:

- четыре надписи (Label);
- пять кнопок (Button);
- выпадающий список (ComboBox);
- текстовое поле ввода (TextBox);
- группирующую рамку (GroupBox);
- список (ListBox);
- два переключателя (RadioButton).



Добавим на форму таблицу для отображения данных (DataGridView) из таблицы «Товары». Для этого на панели «Источники данных» (Data Sources), нажмите кнопку, расположенную справа от таблицы «Products». В появившемся списке объектов для отображения всей таблицы выберите «DataGridView»:



Перетащите таблицу «Products» из панели «Источники данных» на форму. Форма примет следующий вид:

The image shows a Windows application window titled 'Form1'. The form contains a DataGridView table with columns: ID_PRODUCT, NAME, PRICE, COUNT, MARK, and DESCRIP. The first row has a '*' in the first column. Below the table are two labels, 'label1' and 'label2', each followed by a text input field. To the right of the table is a 'groupBox1' containing a 'label4', a 'listBox1', two radio buttons ('radioButton1' and 'radioButton2'), and five buttons ('button1', 'button2', 'button3', 'button4', 'button5'). The 'label3' text is positioned above the table.

Обратите внимание на то, что на форме появилась таблица для отображения данных, подключённая к таблице «Товары». Также появились объекты связи и панель навигации.

Теперь перейдём к настройке свойств объектов. Начнём с настройки

свойств формы.

Задайте свойства формы следующим образом:

- `FormBorderStyle` (Стиль границы формы): `Fixed3D`;
- `MaximizeBox` (Кнопка развёртывания формы во весь экран): `False`;
- `MinimizeBox` (Кнопка свёртывания формы на панель задач): `False`;
- `Text` (Текст надписи в заголовке формы): Товары

(Табличный вид). Задайте свойства надписей (`Label1`, `Label2`, `Label3` и `Label4`) как:

- `AutoSize` (Авторазмер): `False`;
- `Text` (Текст надписи): «Товары (Табличный вид)», «Поле для сортировки», «Наименование» и «Критерий».

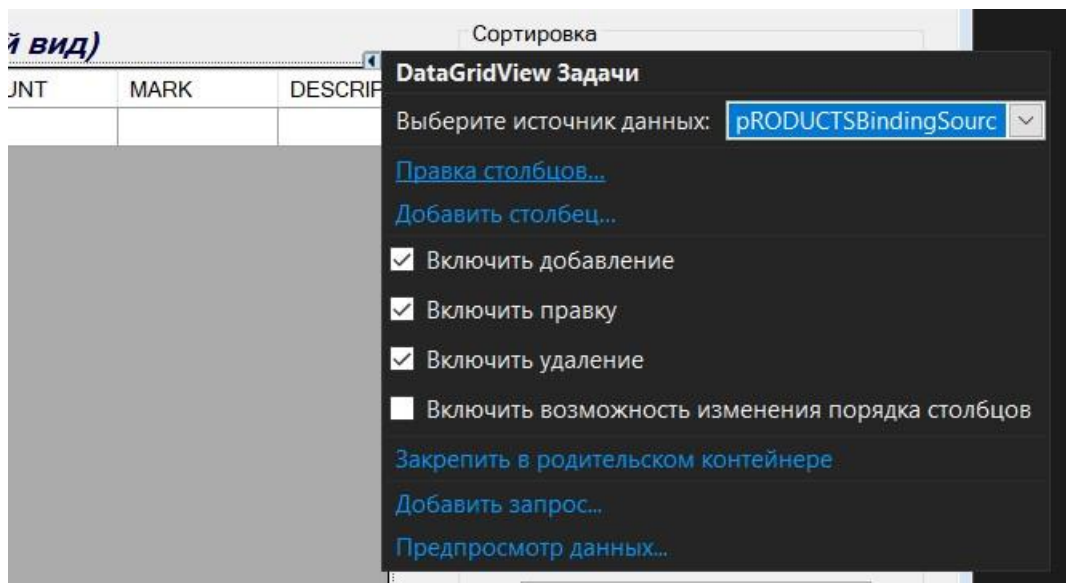
Задайте надписи на кнопках как: «Сортировать», «Фильтровать», «Показать все», «Найти» и «Закрыть». Для того чтобы нельзя было произвести сортировку не выбрав поля изначально заблокируем кнопку «Сортировать» (свойство `Enabled = false`).

У группирующей рамки задайте заголовок «Сортировка». У переключателей (Объекты `RadioButton1` и `RadioButton2`) задайте надписи как «Сортировка по возрастанию» и «Сортировка по убыванию», а у переключателя «Сортировка по возрастанию» (`RadioButton1`) задайте свойство `Checked` (Включён) равное `True` (Истина).

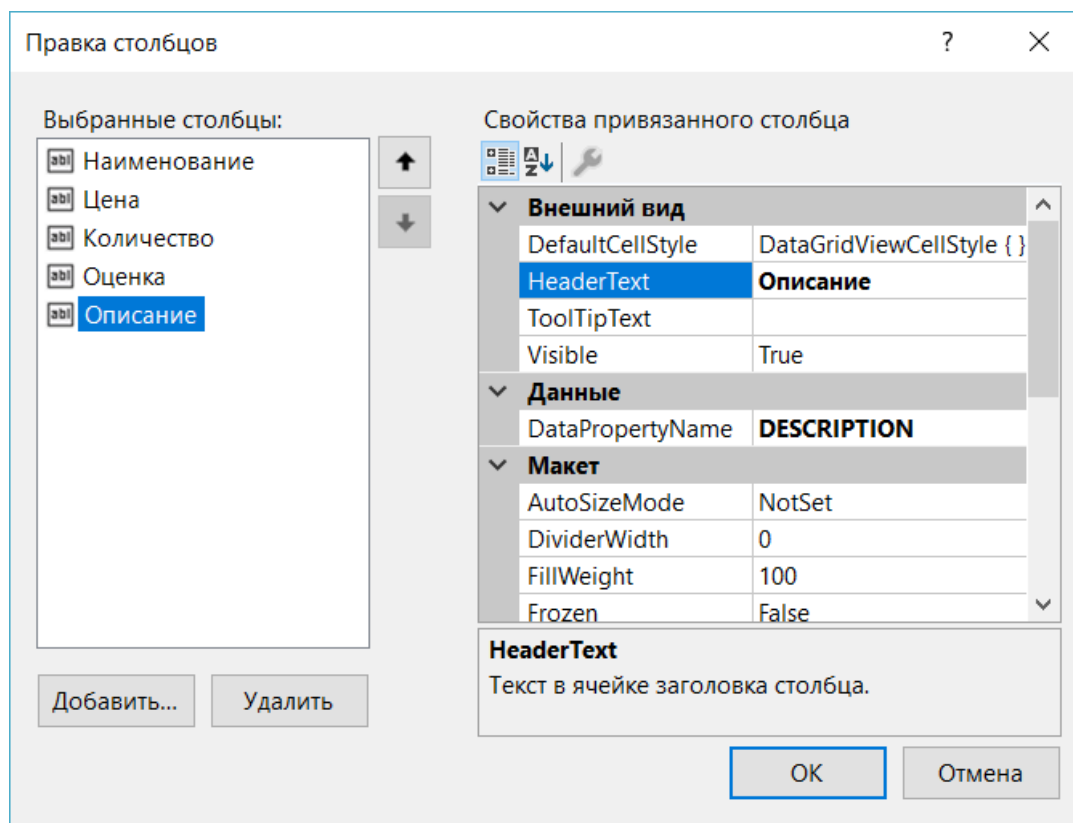
Заполните список (`ListBox1`) значениями:

- Наименование
- Цена
- Количество
- Оценка
- Описание
- Товарная группа

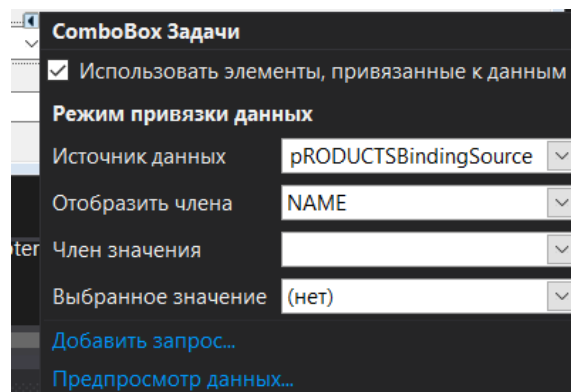
Настроим таблицу для отображения данных, удалив из неё поля с кодами. Выделите таблицу на форме и отобразите её меню задач, кликнув ЛКМ по кнопке, расположенной в верхнем правом углу таблицы. В меню действий выберите пункт «Правка столбцов...»:



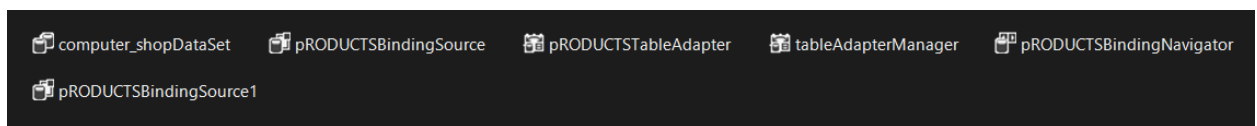
В окне «Edit Columns» из списка полей удалите поля «ID_PRODUCT» и «ID_CATALOG», выделив их и нажав кнопку Удалить. Также измените свойство HeaderText, чтобы наименование столбцов было на русском языке. Список полей примет вид, показанный на рисунке:



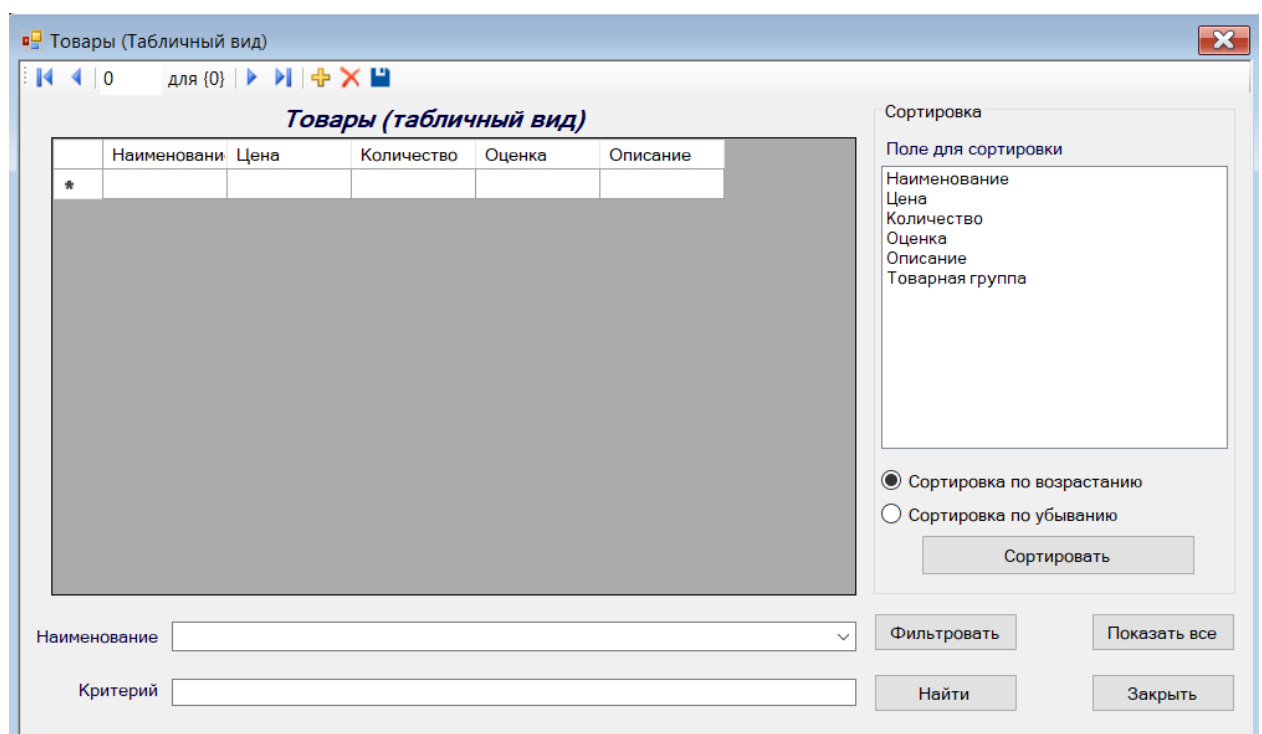
Настроим заполнение выпадающего списка наименованием товаров из таблицы Товары. Отобразите меню действий выпадающего списка. Включите опцию «Использовать элементы, привязанные к данным». В строке Источник данных выберите Другие источники данных, Источники данных проекта, Computer_ShopDataSet, ProductBindingSource. В строке Отобразить члена выберите Name:



На панели невидимых объектов появится дополнительный объект связи «ProductBindingSource1», предназначенный для заполнения выпадающего списка.



После настройки всех вышеперечисленных свойств объектов новая форма примет вид как показано на рисунке:



На этом мы заканчиваем настройку свойств объектов и переходим к

написанию кода обработчиков событий объектов.

Работу с кодом начнём написания кода для разблокирования кнопки «Сортировать», при выборе пункта списка (ListBox1). Для создания процедуры события я дважды щёлкните ЛКМ по списку. Появится процедура обработки события, происходящего при выборе пункта списка (ListBox1_SelectedIndexChanged). В процедуре наберите команду разблокировки кнопки «Сортировать» (Button1): Button1.Enabled = True (листинг 8).

Листинг 8 – Разблокировка кнопки

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    button1.Enabled = true;
}
```

Теперь перейдём к созданию кода сортирующего нашу таблицу в зависимости от выбранного поля и порядка сортировки при нажатии кнопки «Сортировать». Дважды щёлкните ЛКМ по кнопке «Сортировать». Появится процедура «Button1_Click», выполняемая при щелчке ЛКМ по кнопке. В процедуре наберите код, представленный в листинге 9.

Листинг 9 – Сортировка

```
private System.Windows.Forms.DataGridColumn COL;

private void button1_Click(object sender, EventArgs e)
{
    //создает переменную COL для хранения имени выбранного
    столбца таблицы COL = new
    System.Windows.Forms.DataGridColumn();

    //блок switch, присваивающий в
    //переменную Col имя выбранного столбца таблицы в зависимости от
    номера
    //выбранного пункта списка (ListBox1.SelectedIndex). Если выбран
    первый пункт
    //списка, то в переменную Col записывается столбец
```

```

        //DataGridViewTextBoxColumn2,        если        второй,        то        -
DataGridViewTextBoxColumn3

        //и так далее. Хотелось бы отметить тот факт, что нумерация пунктов
списка

        //начинается с нуля, а нумерация столбцов с единицы. Первый столбец
«Наименование»

        //носит имя DataGridViewTextBoxColumn2, так как имя
//DataGridViewTextBoxColumn1 имеет столбец заголовков строк;

switch (listBox1.SelectedIndex)
{
    case 0:
        COL    =    dataGridViewTextBoxColumn2;
break;
    case 1:
        COL    =    dataGridViewTextBoxColumn3;
break;
    case 2:
        COL    =    dataGridViewTextBoxColumn4;
break;
    case 3:
        COL    =    dataGridViewTextBoxColumn5;
break;
    case 4:
        COL    =    dataGridViewTextBoxColumn6;
break;
}

        //Блок If выполняет следующую операцию: если включён
        //переключатель «Сортировка по возрастанию» (RadioButton1), то
отсортировать

        //таблицу по полю заданному в переменной Col по возрастанию
        //(pPRODUCTSDataGridView.Sort                                (Col,
System.ComponentModel.ListSortDirection.
        //Ascending)), иначе по убыванию (pPRODUCTSDataGridView.Sort (Col,
System.
        //ComponentModel.ListSortDirection. Descending)).

```

```

        if (radioButton1.Checked)
pRODUCTSDataGridView.Sort (COL,
        System.ComponentModel.ListSortDirecti
on.Ascending); else
        pRODUCTSDataGridView.Sort (COL,
        System.ComponentModel.ListSortDirection.Descending);
    }

```

Рассмотрим код обработчика события нажатия кнопки «Фильтровать». Дважды щёлкните по кнопке «Фильтровать» и в процедуре обработки события «Button2_Click» наберите код (листинг 10).

Листинг 10 – Фильтрация

```

private void button2_Click(object sender, EventArgs e)
{
    pRODUCTSBindingSource.Filter = "NAME='" + comboBox1.Text + "'";
}

```

У объекта pRODUCTSBindingSource имеется текстовое свойство Filter, которое определяет условие фильтрации. Условие фильтрации имеет синтаксис: “<Имя поля><Оператор>’<Значение>”. В нашем случае значение поля «NAME» приравнивается к значению, выбранному в выпадающем списке (ComboBox1.Text).

Теперь перейдём к кнопке «Показать всё», отменяющей фильтрацию записей. Дважды щёлкните по вышеперечисленной кнопке. Появится процедура Button3_Click. В появившейся процедуре наберите команду, как показано в листинге 11.

Листинг 11 – Отмена фильтрации

```

private void button3_Click(object sender, EventArgs e)
{
    pRODUCTSBindingSource.Filter = "";
}

```

Заметим, что если присвоить свойству «Filter» значение пустой строки (“”), то его действие будет отменено.

Далее рассмотрим реализацию поиска информации в таблице. Дважды щёлкните по кнопке «Найти». В появившейся процедуре обработки нажатия кнопки «Button4_Click» наберите код, как показано в листинге 12.

Листинг 12 – Поиск по ячейкам

```
private void button4_Click(object sender, EventArgs e)
{
    //перебирает все ячейки таблицы и
    //устанавливает в них белый цвет фона и чёрный цвет текста, то есть,
    //отменяет результаты предыдущего поиска
    for (int i = 0; i < pPRODUCTSDataGridView.ColumnCount - 1; i++)
    {
        for (int j = 0; j < pPRODUCTSDataGridView.RowCount - 1; j++)
        {
            pPRODUCTSDataGridView[i, j].Style.BackColor = Color.White;
            pPRODUCTSDataGridView[i, j].Style.ForeColor = Color.Black;
        }
    }
    //перебирает все ячейки таблицы и если они
    //содержат текст, введённый в поле ввода (TextBox1), то устанавливает
    в них
    //голубой цвет фона и синий цвет текста, чем выделяет
    искомые ячейки. for (int i = 0; i <
    pPRODUCTSDataGridView.ColumnCount - 1; i++)
    {
        for (int j = 0; j < pPRODUCTSDataGridView.RowCount - 1; j++)
        {
            if (pPRODUCTSDataGridView[i,
            j].Value.ToString().IndexOf(textBox1.Text) != -1)
            {
                pPRODUCTSDataGridView[i, j].Style.BackColor = Color.AliceBlue;
                pPRODUCTSDataGridView[i, j].Style.ForeColor = Color.Blue;
            }
        }
    }
}
```

```
}  
}
```

Наконец рассмотрим код для кнопки «Закрыть». Дважды щёлкните ЛКМ по этой кнопке и в появившейся процедуре «Button5_Click» наберите код, как показано в листинге 13.

Листинг 13 – Закрытие формы

```
private void button5_Click(object sender, EventArgs e)  
{  
    this.Close();  
}
```

В заключение создадим кнопку на ленточной форме, отображающей таблицу «Товары», для отображения соответствующей табличной формы. Откройте ленточную форму для таблицы «Товары» и поместите на неё новую кнопку, как это показано на рисунке:

The screenshot shows a Windows application window titled "Товары". The window has a standard Windows title bar with a close button. Below the title bar is a ribbon interface. The ribbon has a tab labeled "Товары". Below the ribbon, there are several input fields: "Наименование", "Цена", "Количество", "Оценка", "Описание", and "Товарная группа" (a dropdown menu). At the bottom of the window, there are several buttons: "Первая", "Предыдущая", "Следующая", "Последняя", "Добавить", "Удалить", "Сохранить", and "Табличный вид".

Подключим к кнопке «Табличный вид» созданную ранее табличную форму. Для этого дважды щёлкните ЛКМ по кнопке «Табличный вид» и в появившейся процедуре

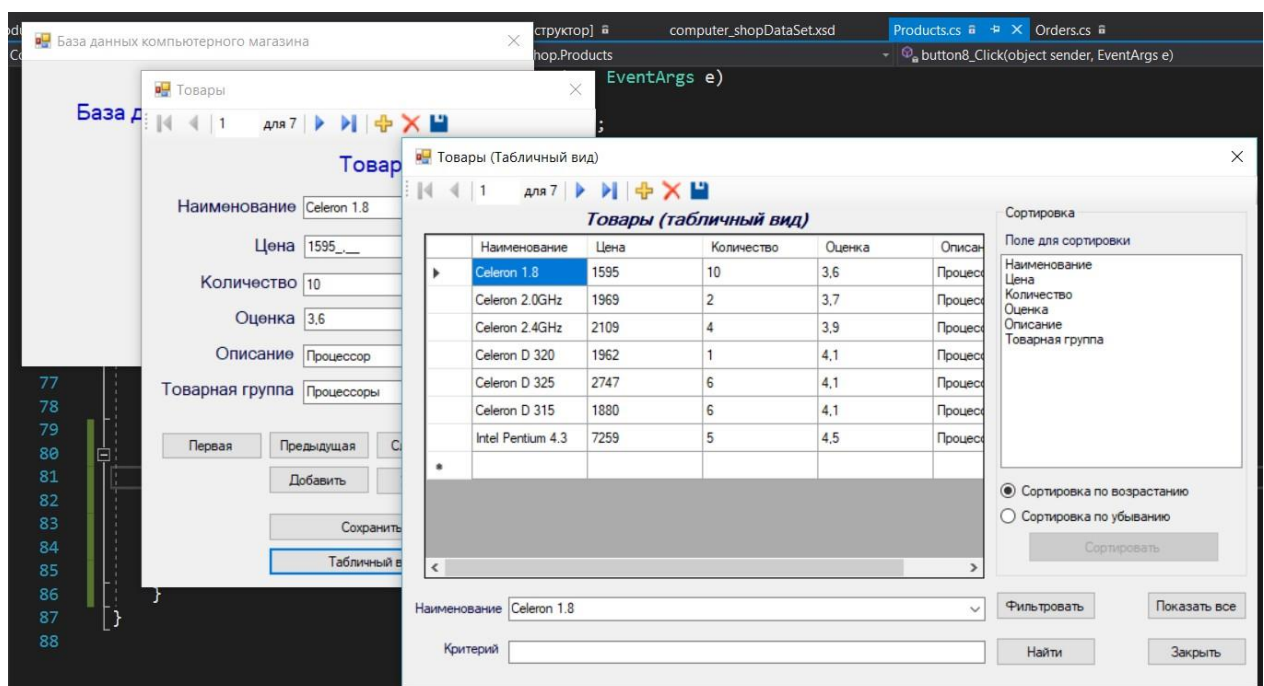
«Button8_Click» наберите код, как показано в листинге 14.

Листинг 14 – Вызов формы Товары (табличный вид)

```
private ProductsTable productstable;

private void button8_Click(object sender, EventArgs e)
{
    productstable = new
ProductsTable(); productstable.Visible
= true;
}
```

Теперь проверим работоспособность созданной табличной формы. Запустите проект и на главной кнопочной форме нажмите кнопку «Товары». На появившейся ленточной форме, отображающей таблицу «Товары» нажмите кнопку «Табличный вид». Появится новая табличная форма:



Проверьте, как работает поиск, фильтрация и сортировка записей в таблице, нажимая на соответствующие кнопки. После проверки работы формы для возвращения в среду разработки просто закройте все формы.

ЗАДАНИЕ:

Создать интерфейс пользователя для собственной БД.