

## Практическое занятие №8

### Разработка прототипа информационной системы в Visual Studio 2019

**Цель занятия:** получить практические навыки разработки прототипа информационной системы в интегрированной среде разработки Visual Studio 2019.

#### Теоретическое введение

Для разработки прототипа информационной системы, работающей с базами данных, будем использовать язык C# – объектно-ориентированный язык программирования. Этот язык был разработан в 1998-2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft.NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. C# относится к семье языков с С-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Переняв многое от своих предшественников – языков C++, Java, Delphi, Modula и Smalltalk – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# не поддерживает множественное наследование классов (в отличие от C++).

Рассмотрим 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 называются 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, предназначаются 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-интерфейсе 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.

### **Подключение файла данных к проекту.**

Теперь рассмотрим создание пользовательского интерфейса БД в среде Visual Studio. Пользовательский интерфейс будет создаваться для базы данных Computer Shop (Компьютерный магазин). Пример создания интерфейса показан именно для этой базы данных. В практическом задании необходимо будет разработать базу данных в программе Microsoft Access 2016 для своего магазина с такими же таблицами, но другим наполнением.

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 Создание проекта, далее выберите язык программирования C# и приложение Windows Forms и нажмите кнопку «Далее» (рисунок 1). После введите имя проекта или оставьте по умолчанию и нажмите кнопку «Создать».

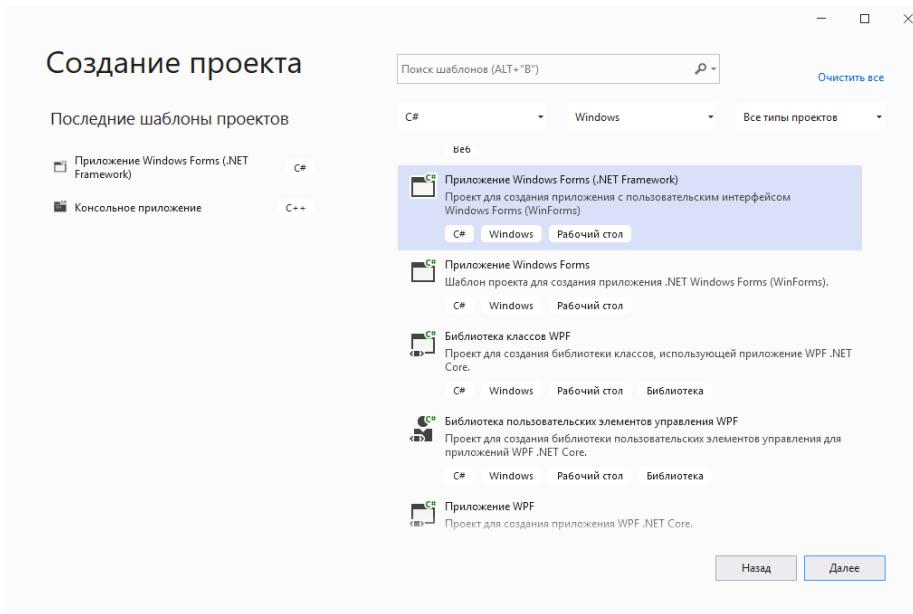


Рисунок 1

Окно разработки (рисунок 2) имеет следующую структуру:

1 Оконное меню – содержит полный набор команд для управления средой разработки.

2 Панель инструментов – содержит кнопки с наиболее часто используемыми командами среды разработки.

3 Панель элементов (Toolbox) – содержит кнопки классов для создания различных объектов (Элементов управления).

4 Обозреватель решений (Solution Explorer) – в зависимости от активизированной в нижней части данной панели вкладки, отображает обозреватель проекта или источники данных, подключённые к проекту. Обозреватель решений (Solution Explorer) отображает все файлы, входящие в проект и позволяет переключаться между ними.

5 Панель свойств (Properties) – отображает и позволяет изменять свойства выбранного объекта.

6 Рабочая область – в зависимости от выбранной вкладки, расположенной в верхней части области, отображает область дизайна формы, код формы или стартовую страницу.

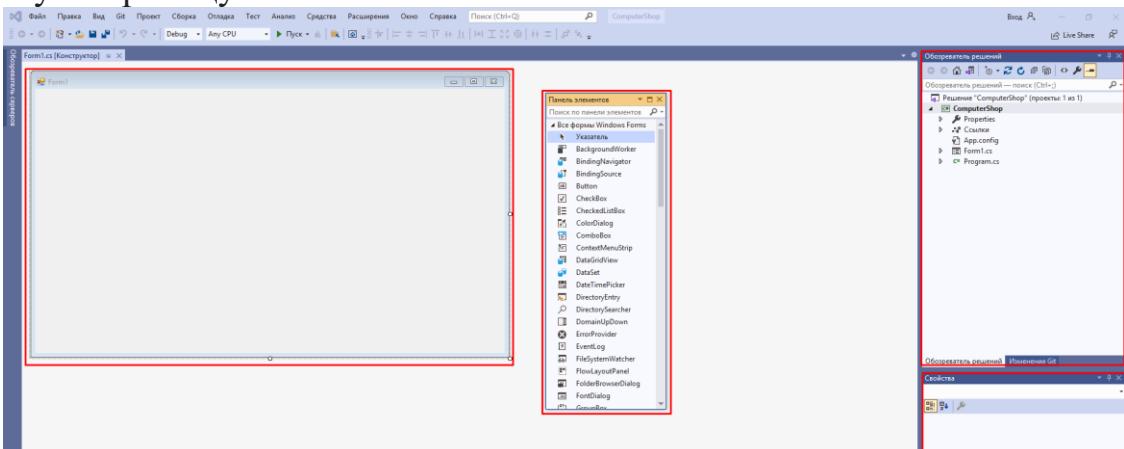


Рисунок 2

После создания нового проекта необходимо подключить к проекту созданную ранее в Microsoft Access 2016 БД «Computer Shop». Для подключения БД к проекту в оконном меню среды разработки выберите Проект -> Добавить новый источник данных (рисунок 3).

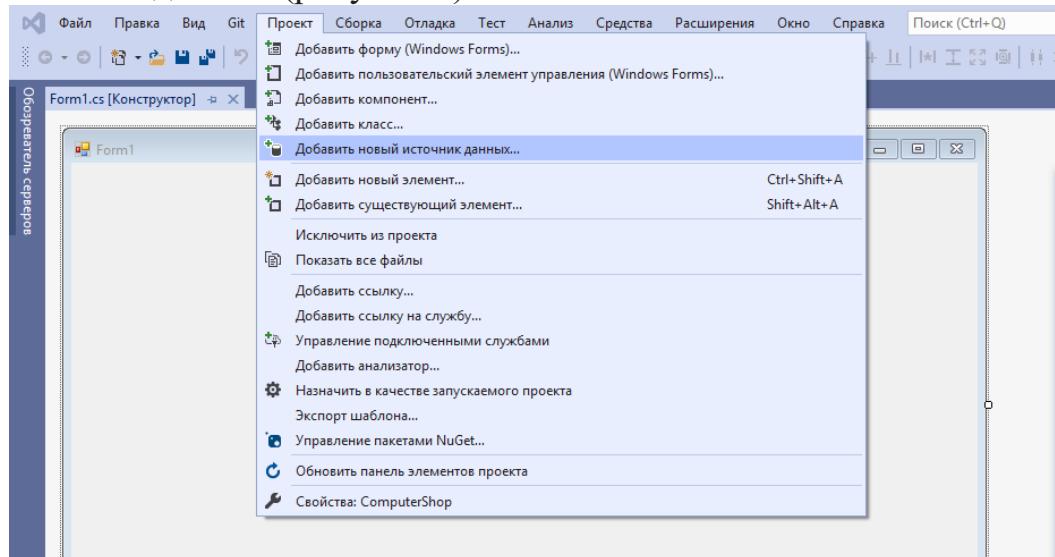


Рисунок 3

В окне Выбор типа источника данных (рисунок 4) 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 (рисунок 6).

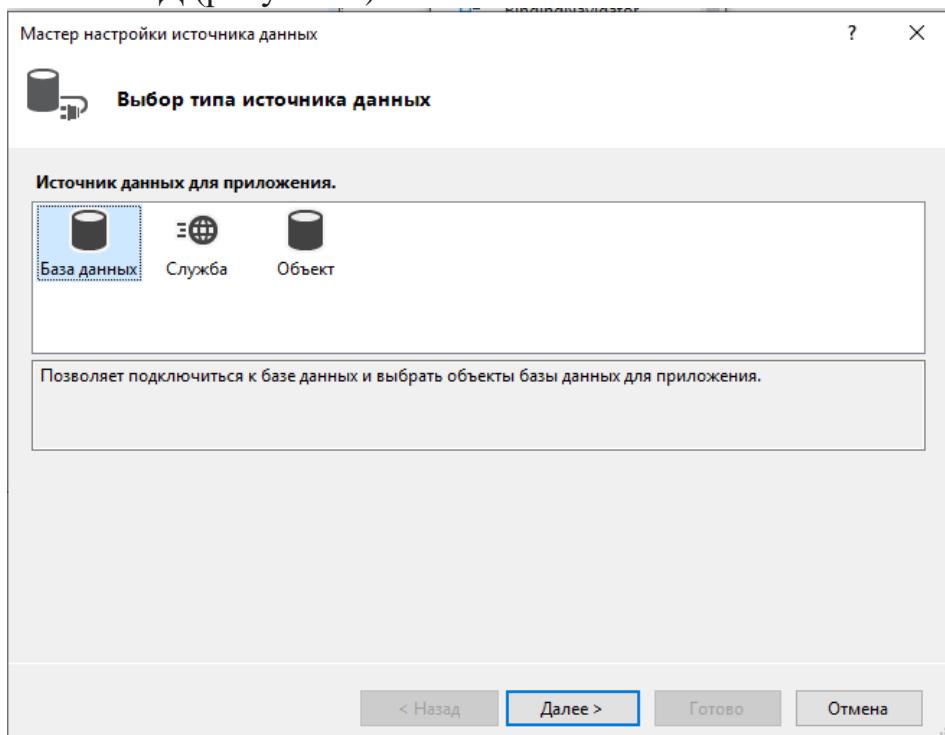


Рисунок 4

Дальше необходимо указать тип модели БД, которую следует использовать (Рисунок 5). Выберите «Набор данных».

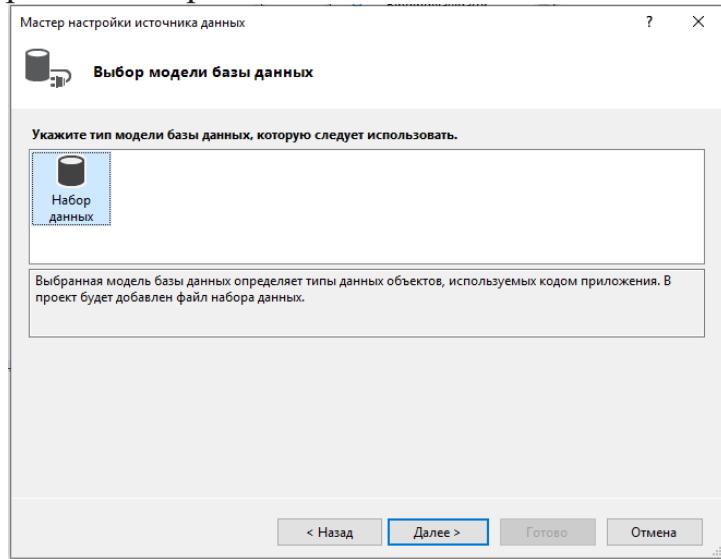


Рисунок 5

В окне выбора подключения к БД (рисунок 6), для создания нового подключения нажмите кнопку «Создать подключение». Появится окно добавления нового источника данных (рисунок 7).

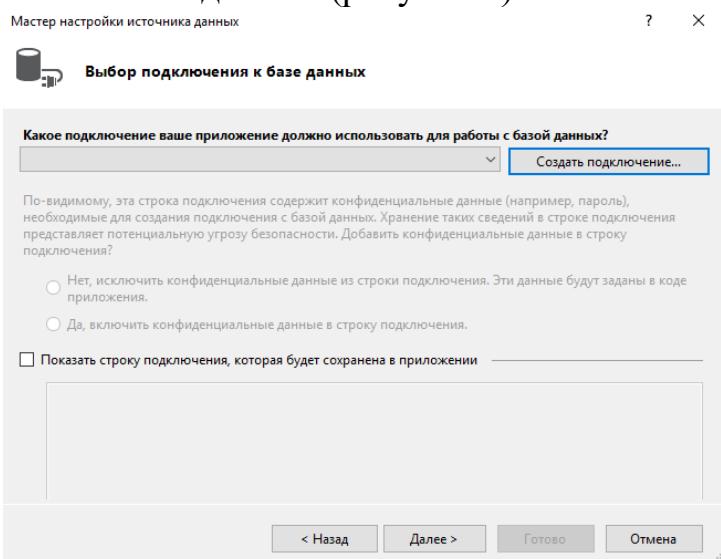


Рисунок 6

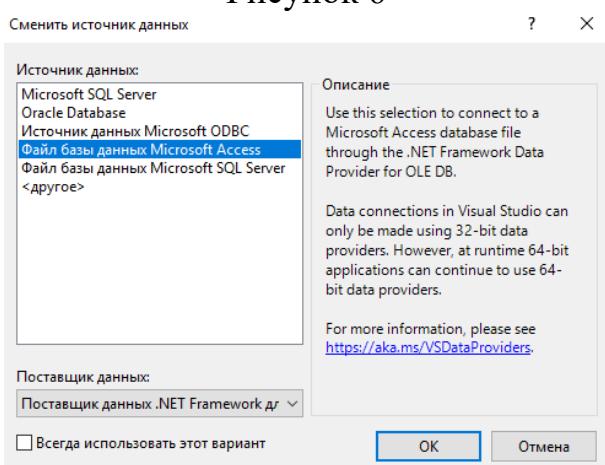


Рисунок 7

Нажмите Обзор и выберите файл базы данных (рисунки 8, 9).

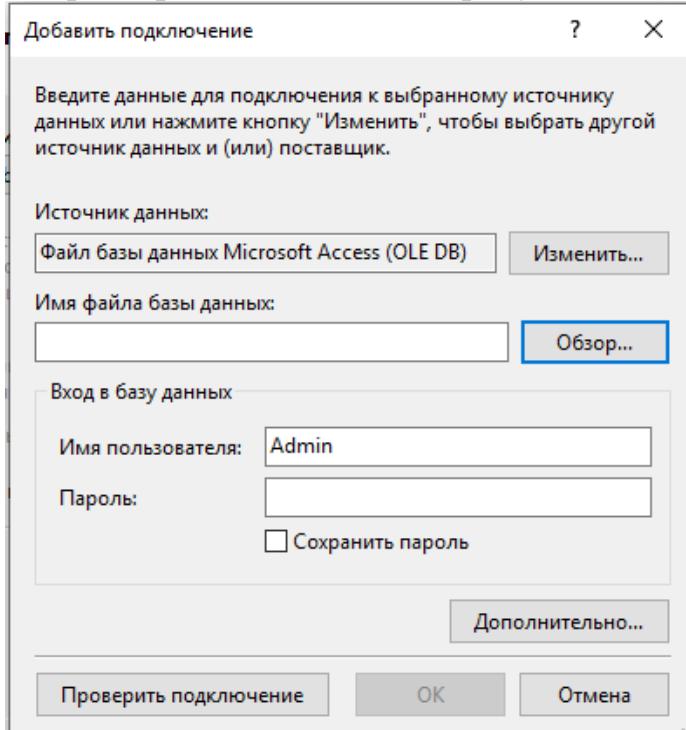


Рисунок 8

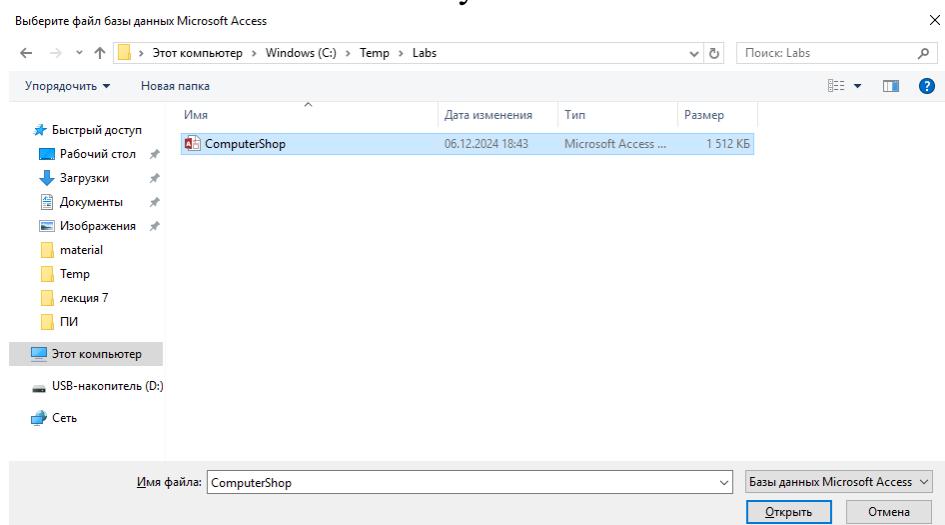


Рисунок 9

Для проверки работоспособности создаваемого соединения нажмите кнопку «Проверить подключение». Появится сообщение Проверка подключения выполнена (рисунок 10), говорящее о том, что соединение работоспособно.

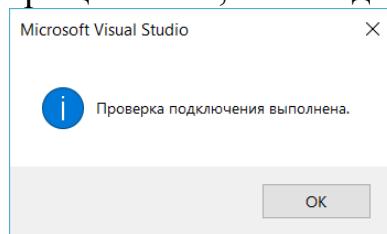


Рисунок 10

Закройте окно, представленное на рисунке 9, а затем в окне добавления нового соединения (рисунок 10) нажмите кнопку OK. Произойдёт возвращение к

окну выбора подключения к БД. Просмотрите созданную строку подключения, щёлкнув по знаку «+» в нижней части окна (рисунок 11).

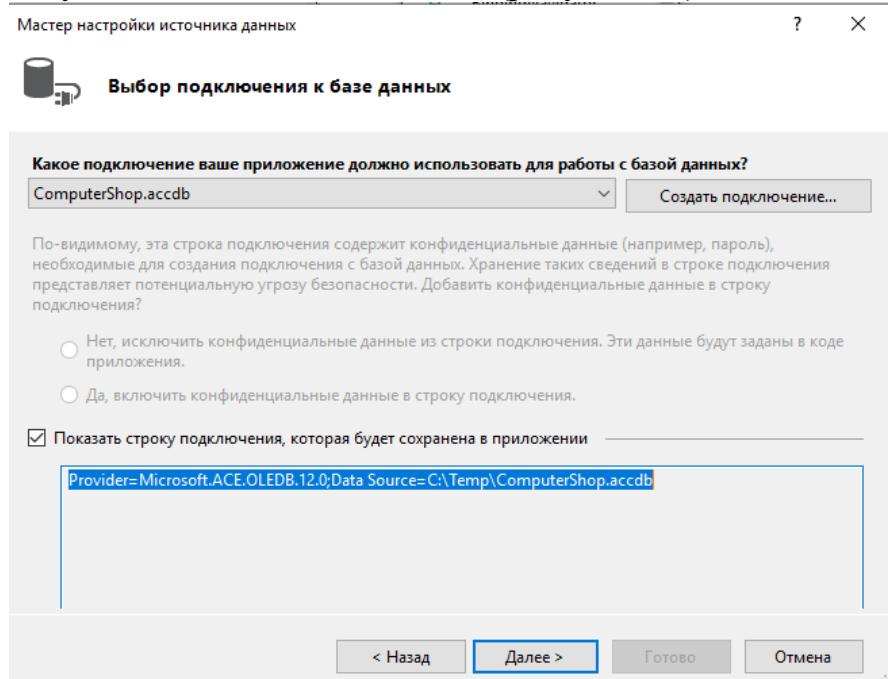


Рисунок 11

В следующем диалоговом окне (рисунок 12) выберите «Нет».

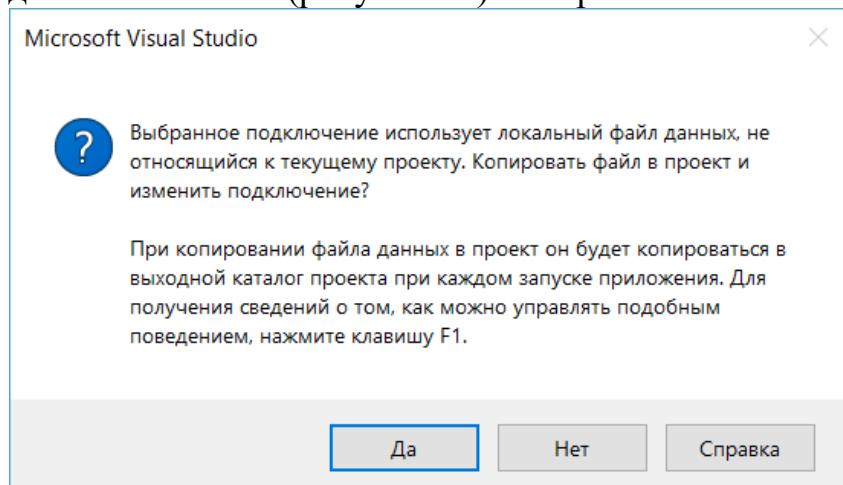


Рисунок 12

Появится окно с запросом о сохранении строки подключения в файле конфигурации приложения (рисунок 13). Для сохранения строки подключения включите опцию Да, сохранить подключение как и нажмите кнопку Далее. Появится окно выбора объектов подключаемой БД (рисунок 14). Выберите все объекты к и нажмите кнопку Готово. Подключение завершено.

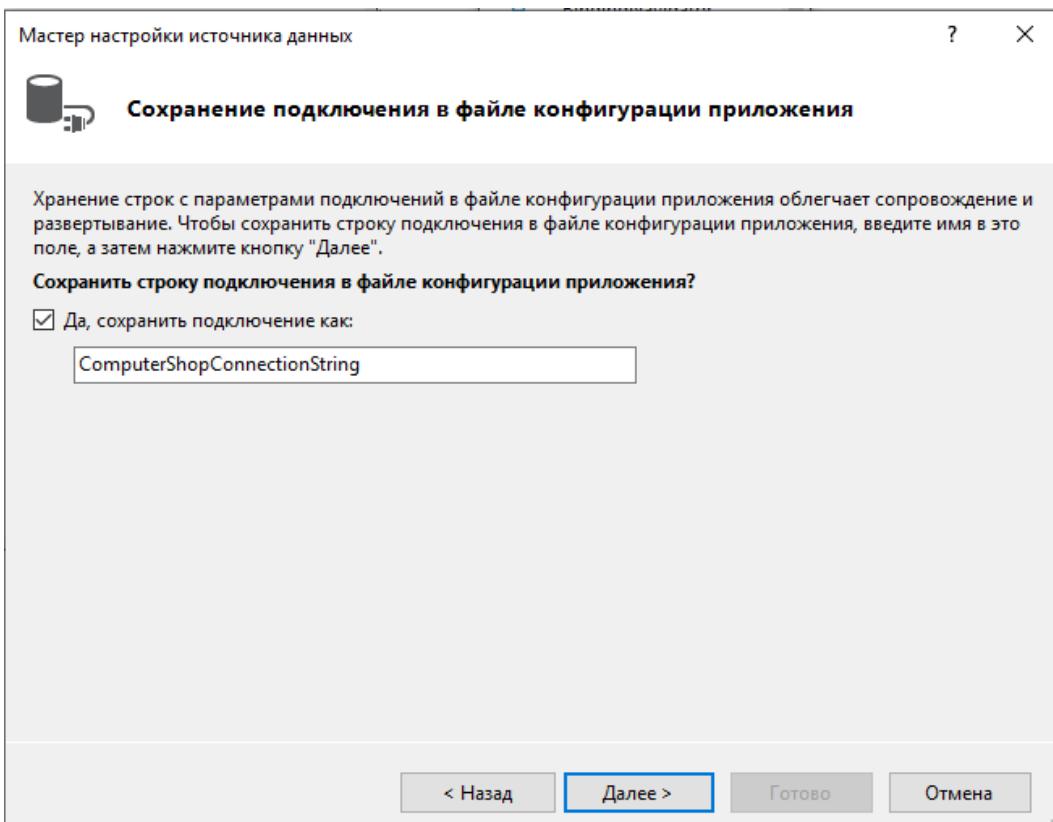


Рисунок 13

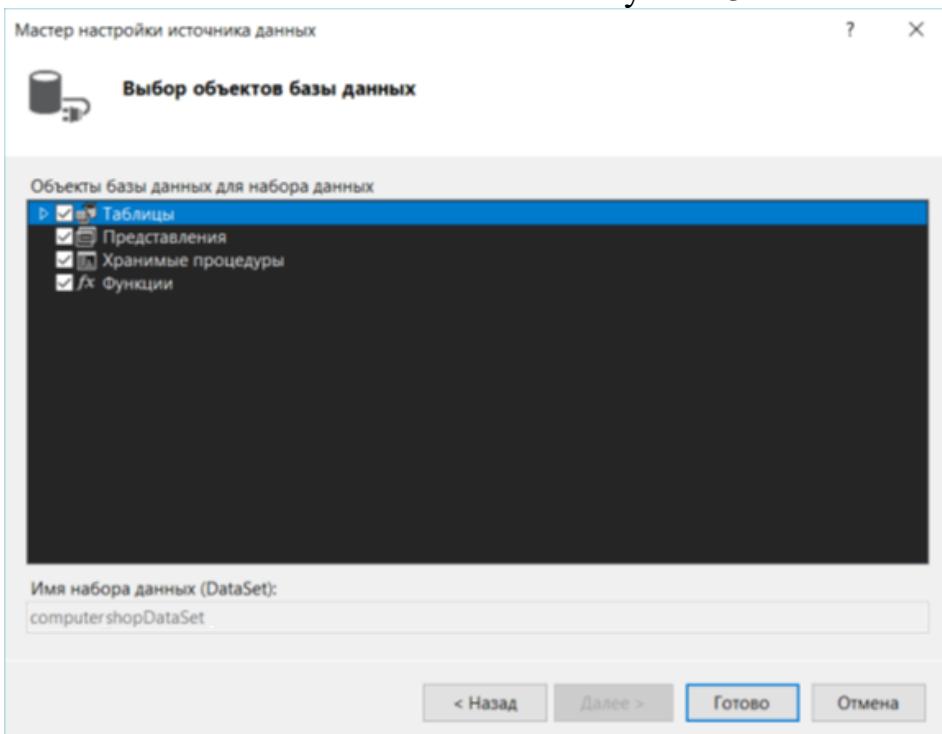


Рисунок 14

Рассмотрим работу с главной кнопочной формой и создание простых ленточных форм для работы с данными.

Поместите на форму надпись (Label) и четыре кнопки (Button) как показано на рисунке 15.

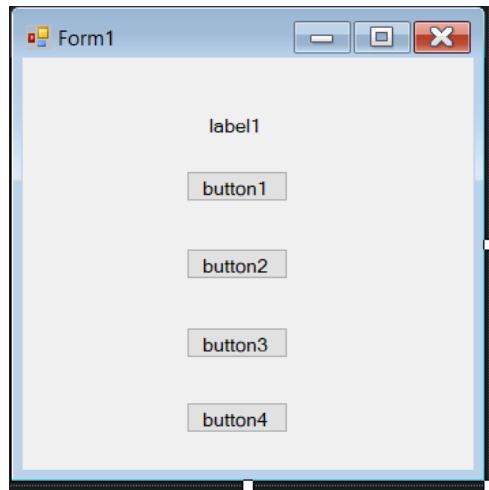


Рисунок 15

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 развертывания формы во весь экран): False;
- MinimizeBox (Кнопка сворачивания 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 16.

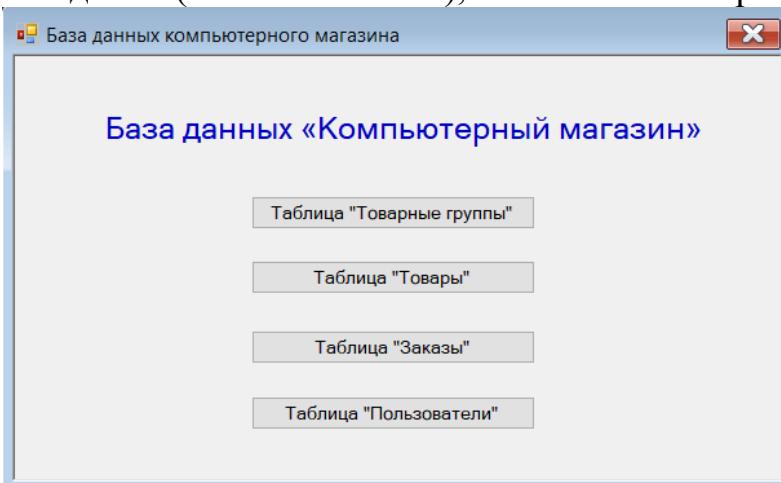


Рисунок 16

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 «Проект»-> «Добавить форму (Windows Forms)...». Появится окно «Добавление нового элемента - ComputerShop» (Добавить новый компонент) (рисунок 17).

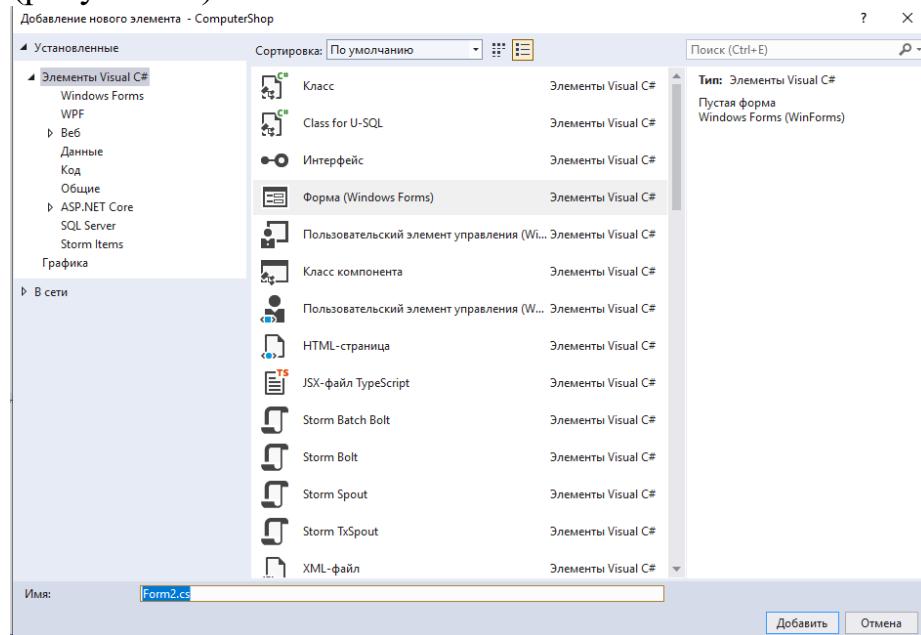


Рисунок 17

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 развертывания формы во весь экран): False;
- MinimizeBox (Кнопка сворачивания 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 (рисунок 18).

Под полями таблицы «Пользователи» в виде подтаблицы располагается таблица «Заказы» (рисунок 18). Подтаблица показывает, что таблица «Заказы» является вторичной по отношению к таблице «Пользователи».

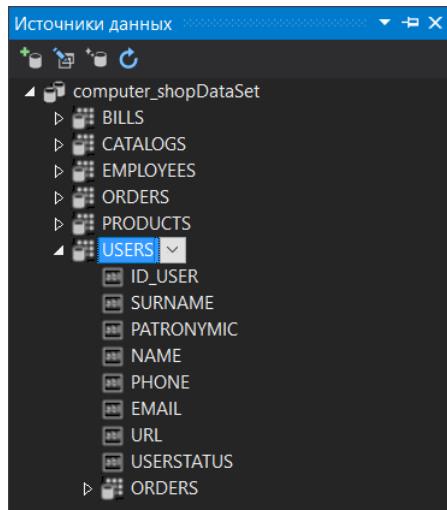


Рисунок 18 – Панель Источники данных

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 19.

Рисунок 19 – Форма пользователи

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 (рисунок 20). Рассмотрим функции этих объектов:

- COMPUTER\_SHOPDataSet (Набор данных COMPUTER\_SHOP) – обеспечивает подключение формы к 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.



Рисунок 20 – Объекты связи формы с БД

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 на главной 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 (рисунок 21).

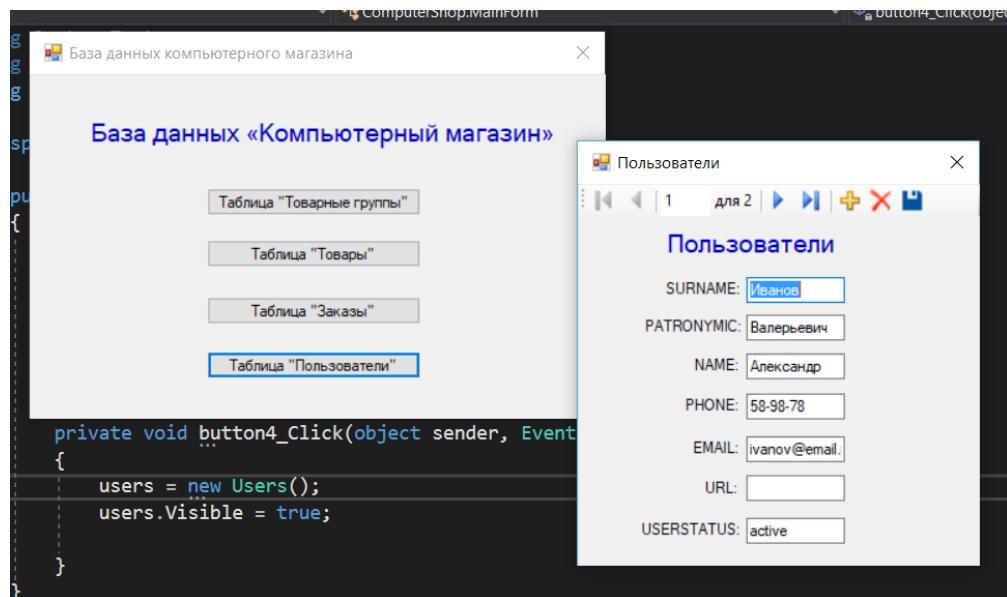


Рисунок 21 – Результат вызова формы

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, рисунки 22 – 23).

#### 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;
        }
    }
}
```

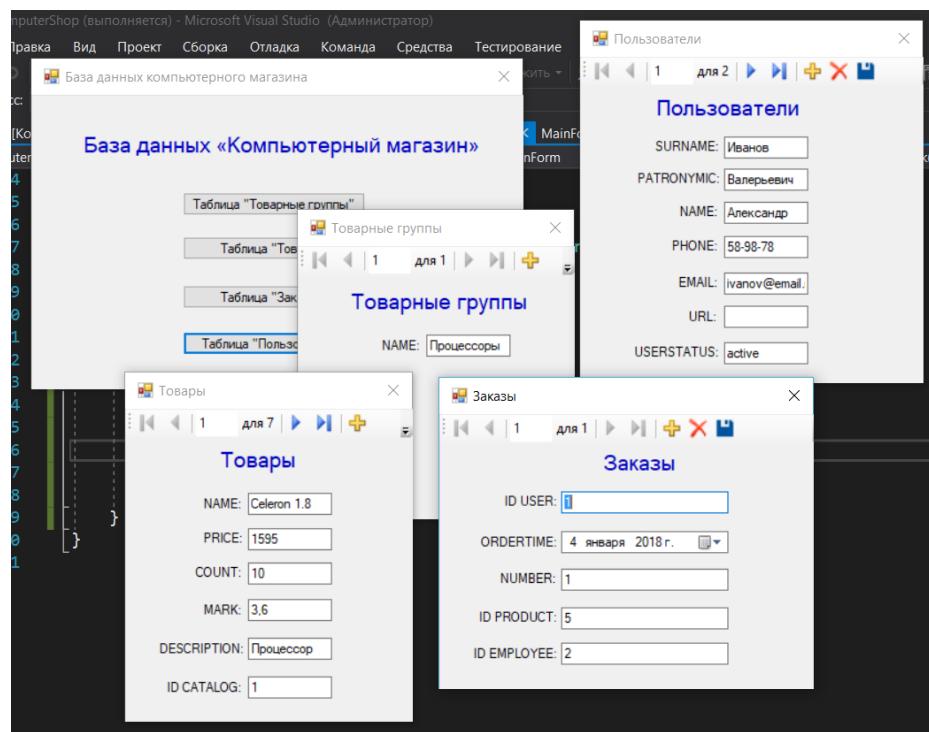


Рисунок 22

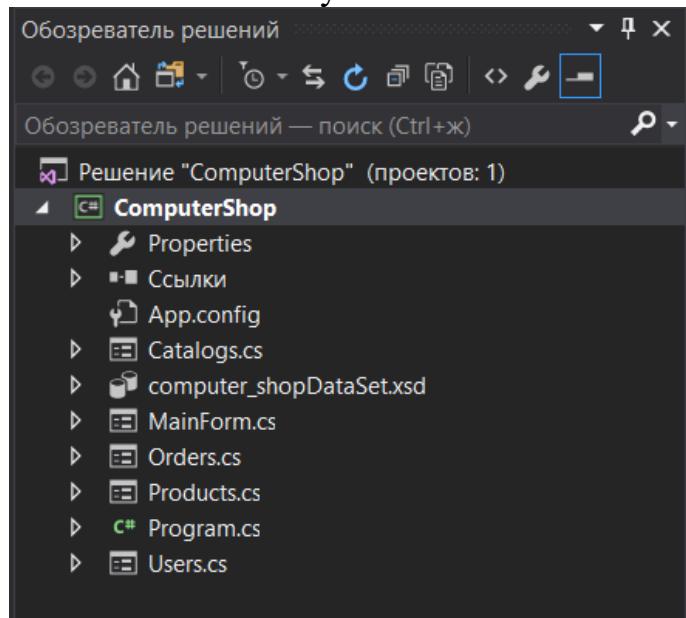


Рисунок 23 – Обозреватель решений

Рассмотрим теперь 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» (рисунок 24).

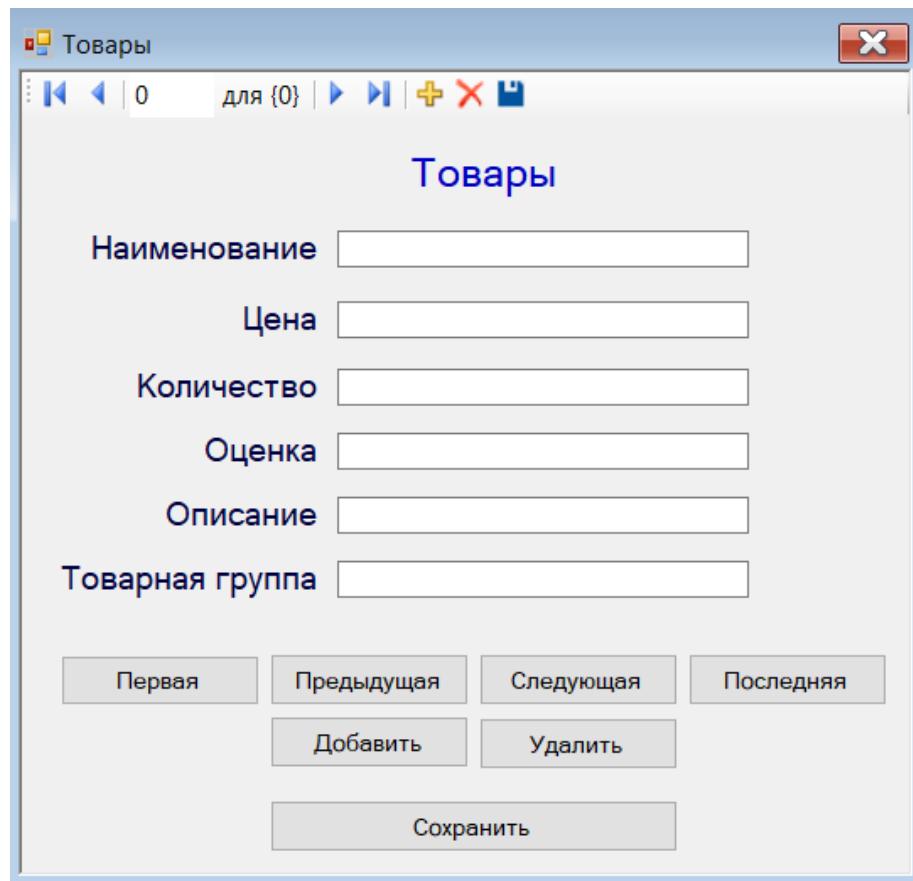


Рисунок 24 – Добавление кнопок

Дважды щёлкните ЛКМ по кнопке «Первая» и в появившемся окне кода формы «Products» в процедуре «Button1\_Click» наберите команду для перехода к первой записи «pRODUCTSBindingSource.MoveFirst()» (листинг 3). Подобным образом добавьте команды для остальных кнопок (листиング 4).

Листинг 3 – Команда для перехода к первой записи

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

Листинг 4 – Команды для остальных кнопок

```
private void button2_Click(object sender, EventArgs e)
{
    pRODUCTSBindingSource.MovePrevious();
}
private void button3_Click(object sender, EventArgs e)
{
    pRODUCTSBindingSource.MoveNext();
}
private void button4_Click(object sender, EventArgs e)
{
    pRODUCTSBindingSource.MoveLast();
}
private void button5_Click(object sender, EventArgs e)
{
    pRODUCTSBindingSource.AddNew();
}
private void button6_Click(object sender, EventArgs e)
{
    pRODUCTSBindingSource.RemoveCurrent();
}
private void button7_Click(object sender, EventArgs e)
{
    //проверяет введённые в поля данные на соответствие типам данных полей
}
```

```

this.Validate();
//закрывает подключение с сервером
this.pRODUCTSBindingSource.EndEdit();
//обновляет данные на сервере
this.tableAdapterManager.UpdateAll(this.computer_shopDataSet);
}

```

Для проверки работы созданных кнопок запустите проект откроите 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...» (рисунок 25).

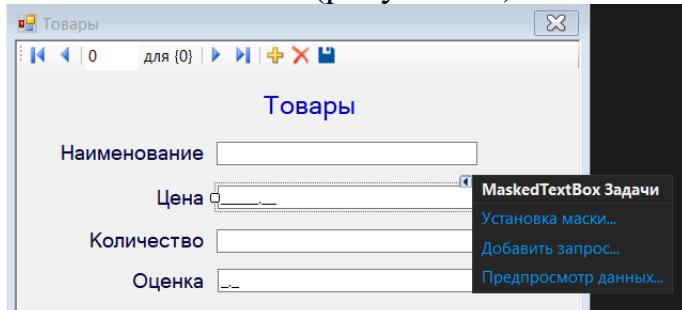


Рисунок 25 – Установка маски

После этого на экране появится окно задания маски «Маска ввода» (рисунок 26). Символы для создания масок ввода приведены в таблице 1.

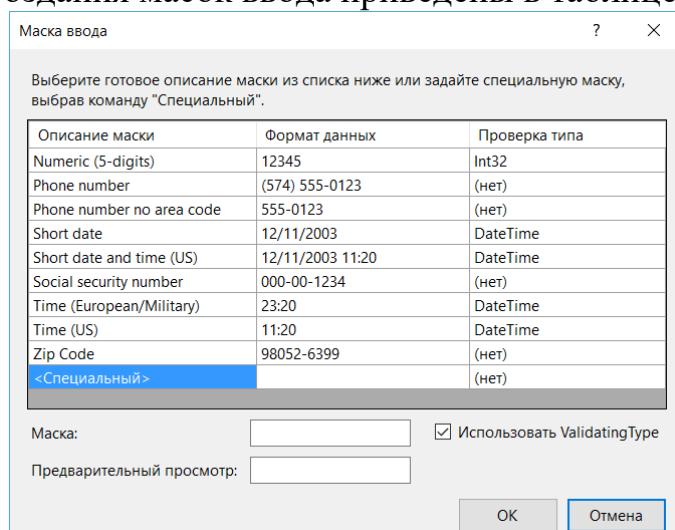


Рисунок 26 – Окно Маска ввода

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 «OK». 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». Проделайте 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 . После создания выпадающего списка подключим его к полю «Код товарной 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» (рисунок 27).

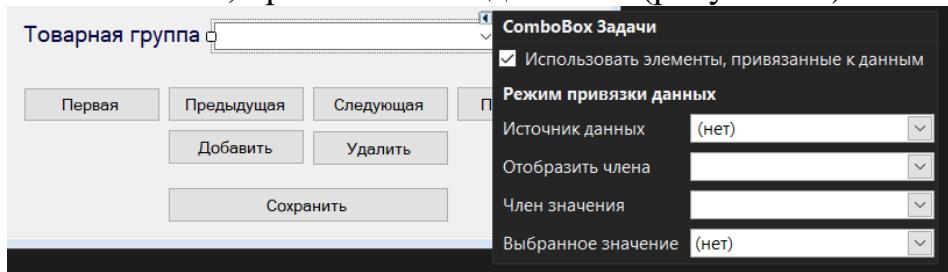


Рисунок 27 – Создание Выпадающего списка

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 (рисунок 28).

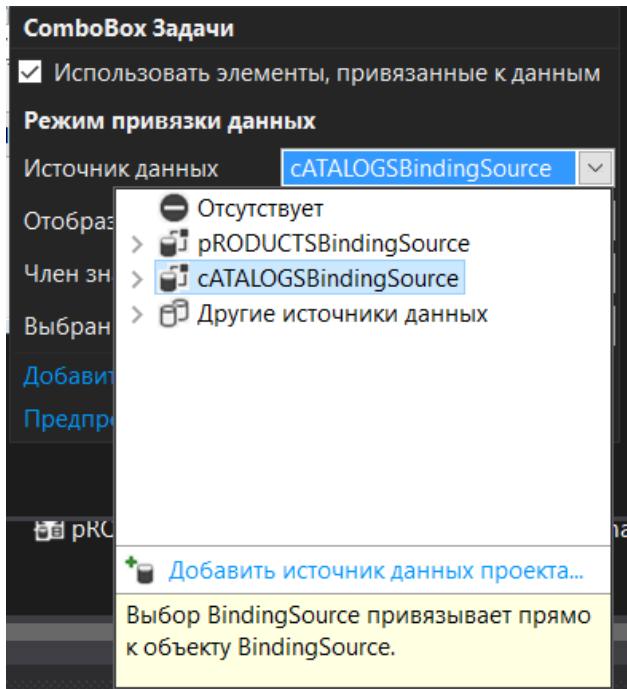


Рисунок 28 – Выбор источника данных

Заполните параметры согласно рисунка 29.

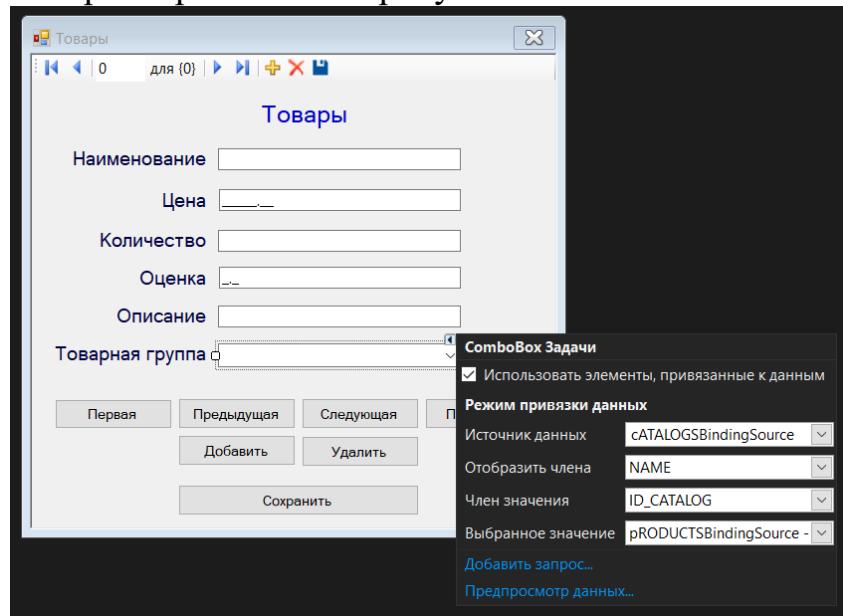


Рисунок 29 – Заполнение параметров поля «Выпадающий список»

Рассмотрим на примере поля «Статус» в таблице «Пользователи» создание выпадающего списка с предопределенными значениями. Для этого удалите поле ввода справа от строки «Статус» на форме «Пользователи». 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 (рисунок 30). На рисунке 31 показан результат добавления поля типа выпадающий список.

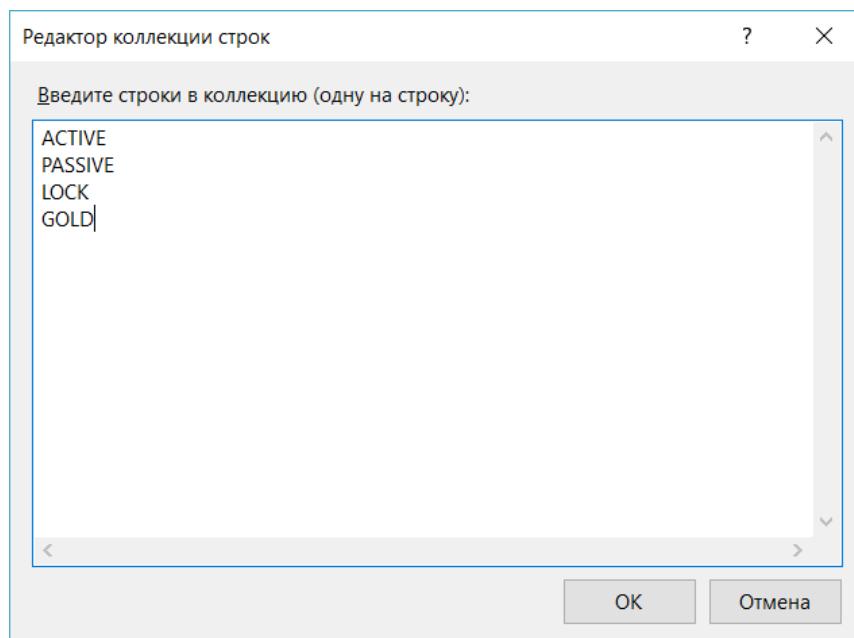


Рисунок 30 – Редактор коллекции строк

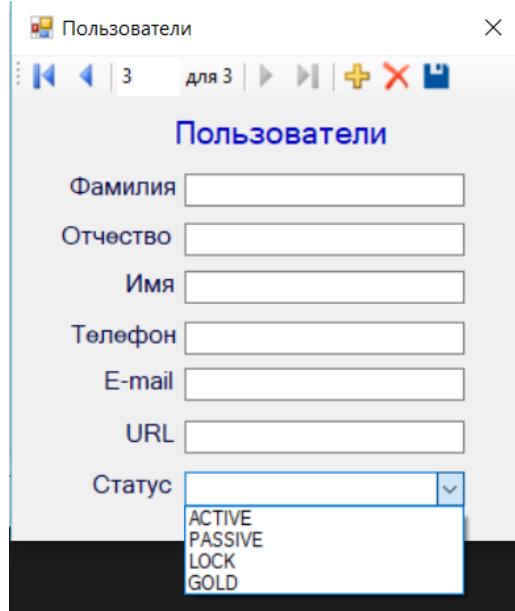


Рисунок 31 – Результат добавления поля с выпадающим списком  
Добавим поле, в котором будет высчитываться сумма заказа. При этом сумма  
заказа рассчитывается по нажатию кнопки «Рассчитать» (рисунок 32).



Рисунок 32 – Кнопка «Рассчитать»

Кликните 2 раза на кнопку «Рассчитать» и наберите команду для события «button1\_Click» как показано в листинге 5.

#### Листинг 5 – 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».

Рассмотрим теперь создание 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).

Расположите объекты как показано на рисунке 33.

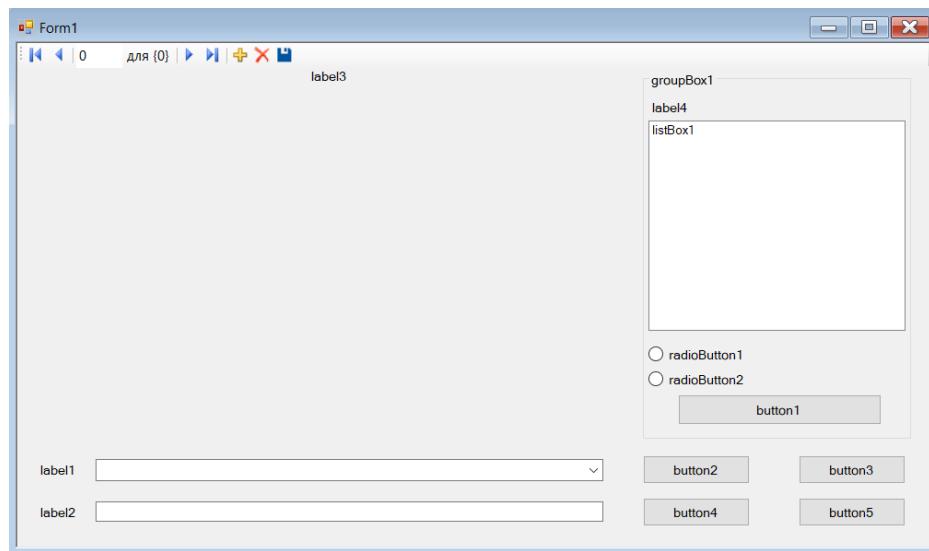


Рисунок 33 – Создание новой формы

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» (рисунок 34).

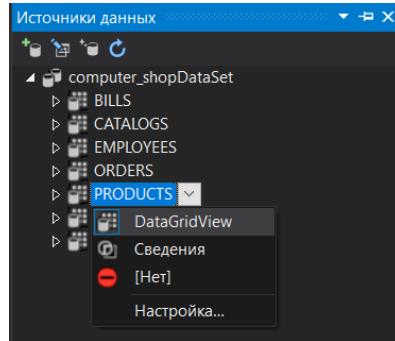


Рисунок 34 – Выбор DataGridView

Перетащите таблицу «Products» из панели «Источники данных» на форму. Форма примет следующий вид (рисунок35).

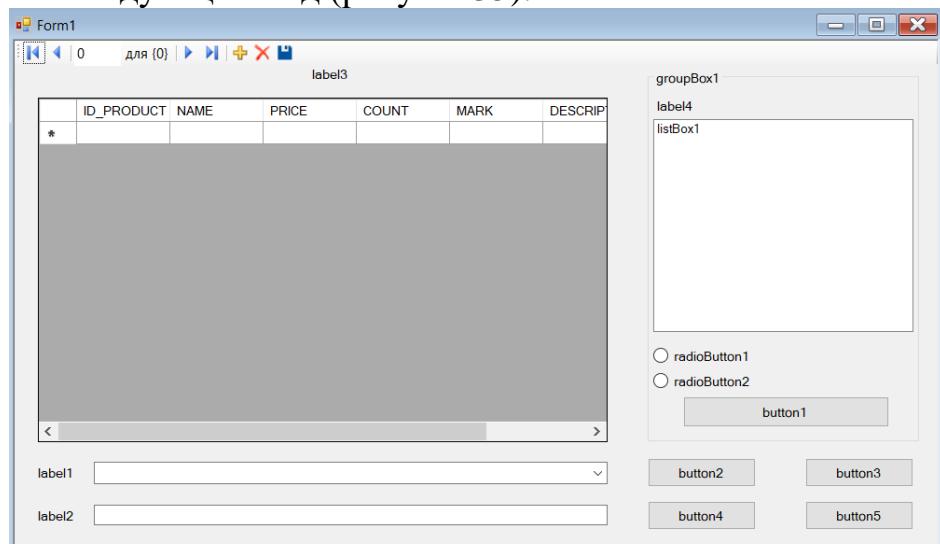


Рисунок 35 – Добавление таблицы Products на форму

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 развертывания формы во весь экран): False;
- MinimizeBox (Кнопка сворачивания 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...» (рисунок 36).

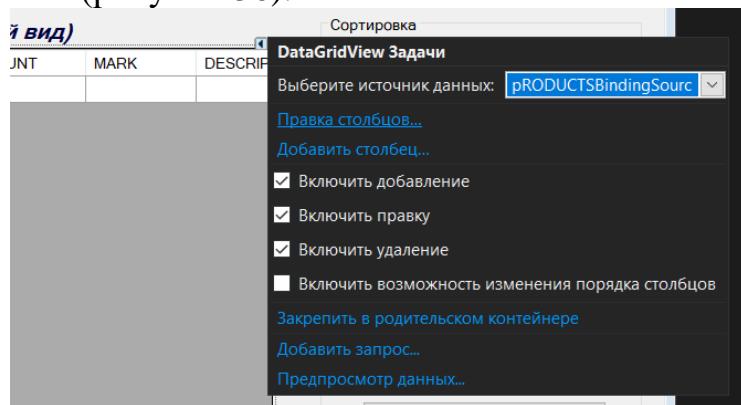


Рисунок 36 – Меню задачи объекта DataGridView

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 37.

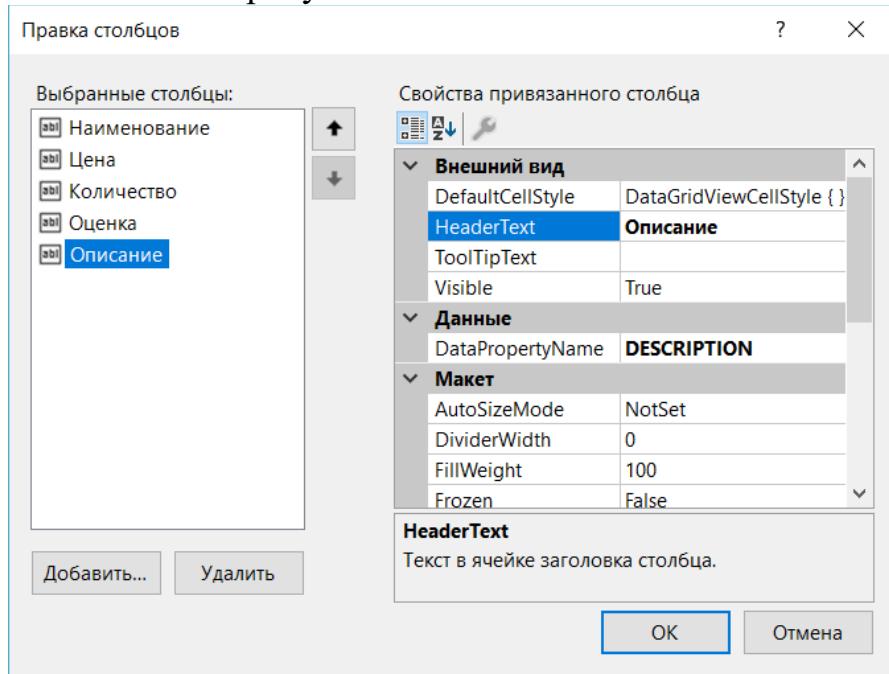


Рисунок 37 – Окно Правка столбцов

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 (рисунок 38).

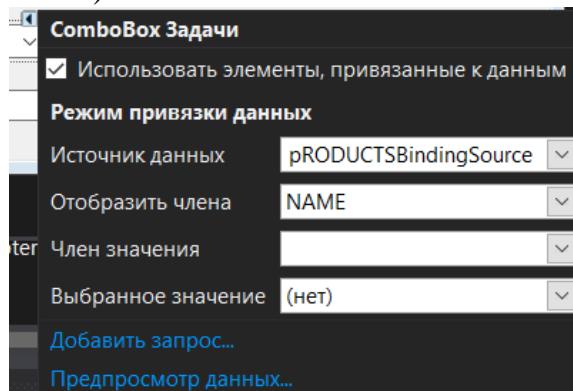


Рисунок 38 – Настройка ComboBox

На панели невидимых объектов появится дополнительный объект связи «ProductBindingSource1», предназначенный для заполнения выпадающего списка (рисунок 39).

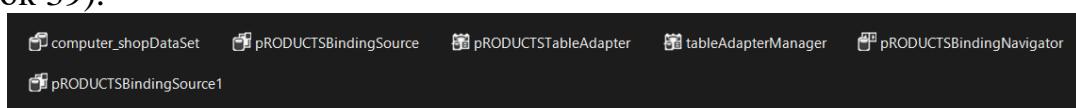


Рисунок 39 – Панель невидимых объектов

После настройки всех вышеперечисленных свойств объектов новая форма примет вид как показано на рисунке 40.

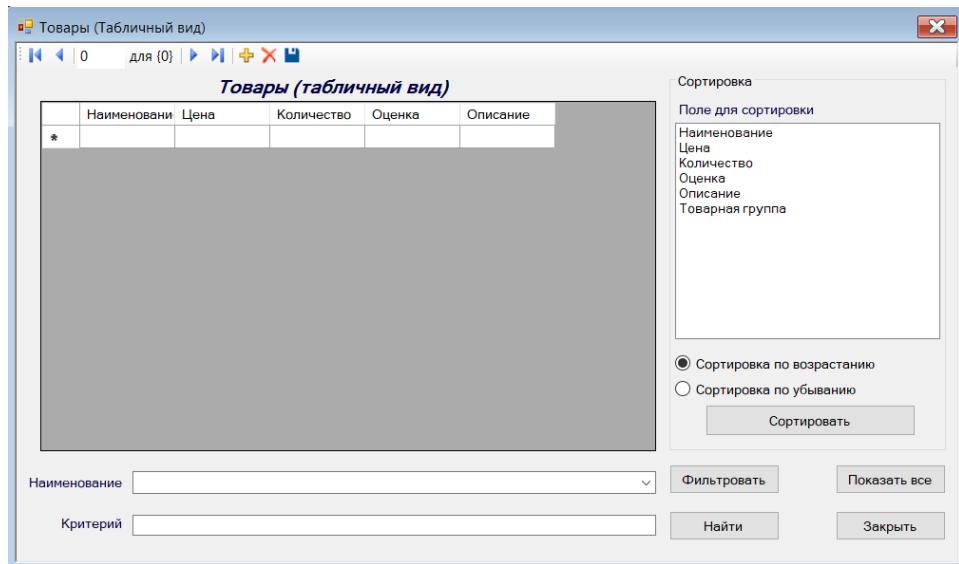


Рисунок 40 – Внешний вид формы Товары (Табличный вид)

На этом настройка свойств объектов закончена. Перейдем 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 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 (листиング 6).

#### Листинг 6 – 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 7.

#### Листинг 7 – Сортировка

```
private System.Windows.Forms.DataGridViewColumn COL;
private void button1_Click(object sender, EventArgs e)
{
    //создает переменную COL для хранения имени выбранного столбца таблицы
    COL = new System.Windows.Forms.DataGridViewColumn();
    //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)
pPRODUCTSDataGridView.Sort (COL, System.ComponentModel.ListSortDirection.Ascending);
else
pPRODUCTSDataGridView.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 8).

#### Листинг 8 – Фильтрация

```

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

```

У объекта pPRODUCTSBindingSource 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 9.

#### Листинг 9 – Отмена фильтрации

```

private void button3_Click(object sender, EventArgs e)
{
pPRODUCTSBindingSource.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 10.

#### Листинг 10 – 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 < pRODUCTSDataGridView.ColumnCount - 1; i++)
{
for (int j = 0; j < pRODUCTSDataGridView.RowCount - 1; j++)
{
pRODUCTSDataGridView[i, j].Style.BackColor = Color.White;
pRODUCTSDataGridView[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 < pRODUCTSDataGridView.ColumnCount - 1; i++)
{
for (int j = 0; j < pRODUCTSDataGridView.RowCount - 1; j++)
{
if (pRODUCTSDataGridView[i, j].Value.ToString().IndexOf(textBox1.Text) != -1)
{
pRODUCTSDataGridView[i, j].Style.BackColor = Color.AliceBlue;
pRODUCTSDataGridView[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 41.

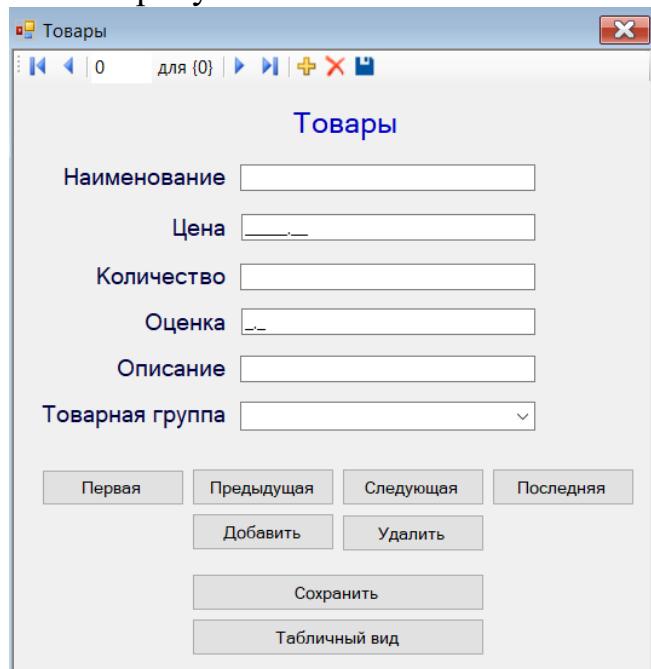


Рисунок 41 – Форма Товары

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 11.

Листинг 11 – 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 (рисунок 42).

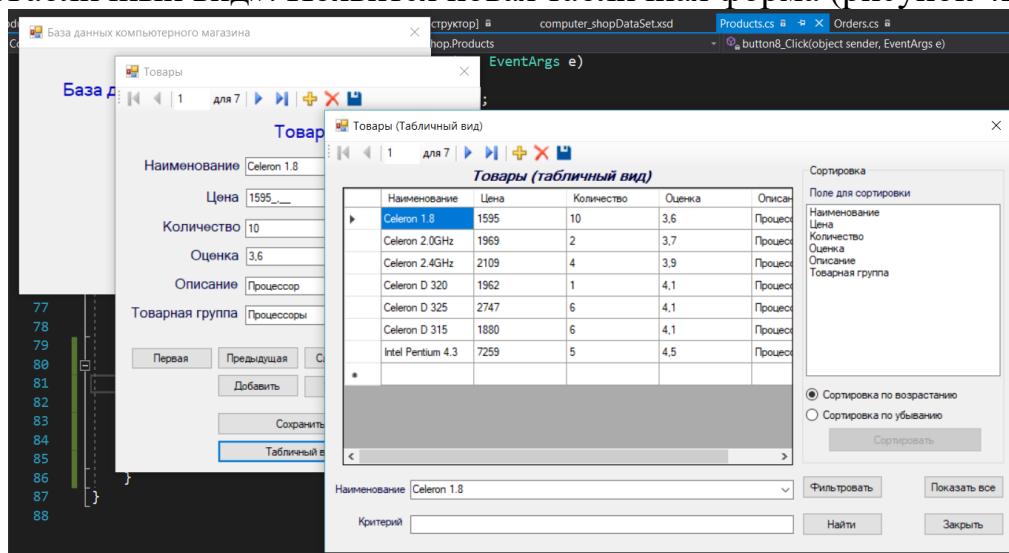


Рисунок 42 – Форма Товары (табличный вид)

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.

## Практическое задание

1 Создайте базу данных магазина. Четные варианты – магазин одежды, нечетные варианты – магазин бытовой техники. Содержание структуры БД должно быть аналогично приведенным в примере.

2 Для своего варианта создайте интерфейс с применением форм, как показано в примере и проверьте его работу для поиска, сортировки и фильтрации данных.