

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ, МОЛОДІ І СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ**

**Г.Г. ШВАЧИЧ, О.В. ОВСЯННИКОВ, Л.М. ПЕТРЕЧУК**

**КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ В ДІЛОВОДСТВІ  
Розділ 2. IntraWeb технологія**

Затверджено на засіданні Вченої ради академії  
як навчальний посібник

**Дніпропетровськ НМетАУ 2012**

УДК 004 (075.8)

Швачич Г.Г., Овсянніков О.В., Петречук Л.М. Комп'ютерні технології в діловодстві. Розділ 2. IntraWeb технологія: Навч. посібник (російською мовою). – Дніпропетровськ: НМетАУ, 2012. – 48 с.

Викладена реалізація концепції IntraWeb технології та її застосування при створенні додатків у середовищі Delphi.

Призначений для студентів напрямку 6020105 – документознавство та інформаційна діяльність.

Іл. 72. Табл. 8. Бібліогр.: 7 найм.

Друкується за авторською редакцією.

Відповідальний за випуск                      Г.Г. Швачич, канд. техн. наук, проф.

Рецензенти:                      О.Г. Холод, канд. техн. наук, проф. (Дніпропетровський університет А. Нобеля)  
   І.Г. Бутенко, канд. техн. наук, доц. (УДХТУ)

© Національна металургійна академія  
України, 2012

© Швачич Г.Г., Овсянніков О.В.,  
Петречук Л.М., 2012

# 1. ТЕХНОЛОГИЯ INTRAWEB

## 1.1. Общие положения

**IntraWeb** — товарный знак *программного обеспечения* компании *AtoZed Software*, реализующий технологию быстрой разработки приложений с помощью библиотеки компонентов *IW* для Интернет и мобильных приложений на языке *Delphi/C++*, также существуют версии для других языков программирования, таких как *Java*, *C++ Builder* и *Delphi for .net*.

Компоненты *IntraWeb* позволяют визуально проектировать *Web* серверные приложения.

Библиотека компонентов *IntraWeb* (также известная как *VCL* для *Web*) установлена и использована в системах разработки *Delphi* (начиная с версий 7.0) и *C++ Builder*. Главное отличие от стандартной библиотеки *VCL* заключается в том, что она предназначена исключительно для пользовательских интерфейсов *Web*. Так же, как и другие библиотеки *VCL* или *CLX*, библиотека *IW* имеет схожую структуру и метод программирования, но добавляет для компонентов и модулей уникальную приставку *IW*, к примеру, *Button* — *IWButton* или *Forms* — *IWAppForm*.

*IntraWeb*, как средство *RAD*, направлена на сокращение труда и экономии времени разработчиков прикладных программ. *RAD* (от англ. *rapid application development* — быстрая разработка приложений) — концепция создания средств разработки программных продуктов, уделяющая особое внимание скорости и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы.

Технология *RAD* предусматривает активное привлечение заказчика уже на ранних стадиях – обследование организации, выработка требований к системе.

Принципы *RAD* технологии направлены на обеспечение трех основных ее преимуществ: высокой скорости разработки, низкой стоимости и высокого качества.

## 1.2. Структура IntraWeb-приложения

Реализованная в Delphi технология IntraWeb представляет собой набор VCL-компонент, расширяющих возможности Delphi-разработчика при проектировании приложений, работающих в сети с использованием WEB-браузеров на клиентских рабочих местах. Создание такого рода приложений напоминает разработку стандартного RAD-проекта: создание формы (WEB-формы), добавление компонент, запись кода для обработчиков событий и так далее. Тем не менее IntraWeb сильно отличается от других средств разработки WEB-приложений. В данной технологии нет необходимости явно применять HTML-редакторы (хотя возможность совместной работы WEB-дизайнера и программиста существует и здесь), задавать обработчики на JavaScript, как это делается, например, в WEBSnap-приложениях. IntraWeb-приложения содержат обыкновенный Pascal-код, но выполняются в окне WEB-браузера. Все преобразования в код, “понятный” WEB-браузеру (java script), производит сервер IntraWeb приложения.

Говоря о структуре **IntraWeb-приложения** выделяют клиентскую и серверную части.

**Клиентская часть** – это готовое IntraWeb-приложение, доступное клиентам как обычная Web-страница, загруженная в обозреватель Internet Explorer, Mozilla Firefox или Netscape Navigator.

**Серверная часть** IntraWeb-приложения может быть реализована как:

- сервис операционной системы
- Windows-приложение
- расширение функциональности IIS (ISAPI-расширение)

Во всех вариантах реализации IntraWeb-сервер поддерживает:

- SSL (англ. Secure Sockets Layer — уровень защищённых сокетов, криптографический протокол, который обеспечивает установление безопасного соединения между клиентом и сервером. SSL изначально разработан компанией Netscape Communications.)
  - хранение состояния с идентификатором сессии
  - средства аутентификации пользователя
  - средства для работы с JavaScript-кодом

### 1.3. Управление формами в IntraWeb

Управление формами в приложениях IntraWeb очень напоминает управление формами в обычных приложениях Delphi, но при этом имеет несколько ограничений:

1. Только одна форма может быть видимой в любой момент времени. Это связано с тем, что формы показываются в Web-браузере.
2. Нельзя использовать модальные формы. Тем не менее, в связи с тем, что только одна форма может быть видимой в любой момент времени, по существу, все формы являются модальными.
3. Формы должны принадлежать WebApplication.

IntraWeb хранит список форм как часть пользовательского сеанса (session). Способ хранения форм в списке напоминает кучу (stack), при этом только что показанные формы добавляются к верху. Когда формы скрыты или освобождены (released), этот список используется для определения формы, которая должна быть показана, если явно не указано показать другую форму через вызов метода `.Show` формы. Обычно список форм никогда напрямую не взаимодействует с пользователем, вместо этого вызываются методы форм. Тем не менее, бывают случаи, когда непосредственное взаимодействие со списком форм может быть необходимо. Для этих случаев `TIWApplication` содержит несколько методов для взаимодействия со списком форм, и они задокументированы в файле справки.

#### Отображение форм

Общий формат отображения формы следующий:

```
TFormType.Create(WebApplication).Show.
```

Владельцем формы является `WebApplication` и одно из отличий от обычных приложений заключается в том, что в обычных приложениях форма показывается немедленно после того, как метод `.Show` вызван. В случае с приложением IntraWeb вызов метода `.Show` просто устанавливает форму в качестве следующей активной формы, которая будет показана после того, как событие (event) завершится и вернется к IntraWeb. Только после выполнения события форма будет показана.

Экземпляр (instance) каждой формы может быть показан много раз посредством перемещения его наверх. В этом случае экземпляр формы будет представлен в списке форм в нескольких местах.

### **Скрытие форм**

В обычном приложении форма может быть скрыта без уничтожения формы с помощью вызова метода `.Hide`. Та же функциональная возможность может быть реализована в IntraWeb с помощью вызова метода `.Hide`. Метод `.Hide` скрывает форму без её уничтожения, в отличие от метода `.Release`. Метод `.Hide` удаляет все ссылки в списке форм как и метод `.Release`, но не уничтожает форму. В связи с этим, необходимо сохранять ссылку на форму, если в дальнейшем она будет отображаться, в противном случае форма окажется «осиротевшей».

### **Уничтожение форм**

В обычном приложении, когда форма больше не нужна, она может быть уничтожена с помощью использования методов `.Free` или `.Destroy`. В приложении IntraWeb эти методы работают аналогично, тем не менее их нельзя вызывать напрямую. Вместо этого надо вызвать метод `.Release`. Фактически метод `.Release` не уничтожает форму сразу при вызове. Форма не будет уничтожена, пока процесс (event) не завершится и не передаст управление IntraWeb. Дело в том, что метод `.Release` обычно вызывается процессом из самой формы, хотя так бывает не всегда.

После того как вызван метод `.Release`, как и в обычном приложении, активной становится та форма, которая была активной до того, как разрушенная форма стала активной. Если пользователь не будет возвращаться к предыдущей форме, должен быть вызван метод `.Show` другой формы. Метод `.Show` может быть вызван как до так и после вызова метода `.Release`, поскольку вызов метода `.Release` не будет иметь никакого эффекта, пока управление не вернётся IntraWeb.

После того как форма уничтожена, все ссылки на неё удаляются из списка форм. Это вызывает изменение в порядке форм, которые будут показаны когда формы скрыты или уничтожены без явного вызова метода `.Show`.

### **Передача данных между формами**

Данные могут передаваться между формами как в обычных приложениях. Поскольку формы являются устойчивыми, информация может храниться в переменных - членах классов формы.

## 2. ПРИМЕР СОЗДАНИЯ INTRAWEB ПРИЛОЖЕНИЯ

### 2.1. Схема разрабатываемого приложения

Примером применения IntraWeb технологии будет разработка IntraWeb-приложения для корпоративной системы на примере Internet-магазина. Рассматриваемая система содержит СУБД, описывающую товары и услуги и IntraWeb-приложение, позволяющее представлять пользователю в Internet предлагаемые товары и оформлять заявки на их покупку (рис.2.1).

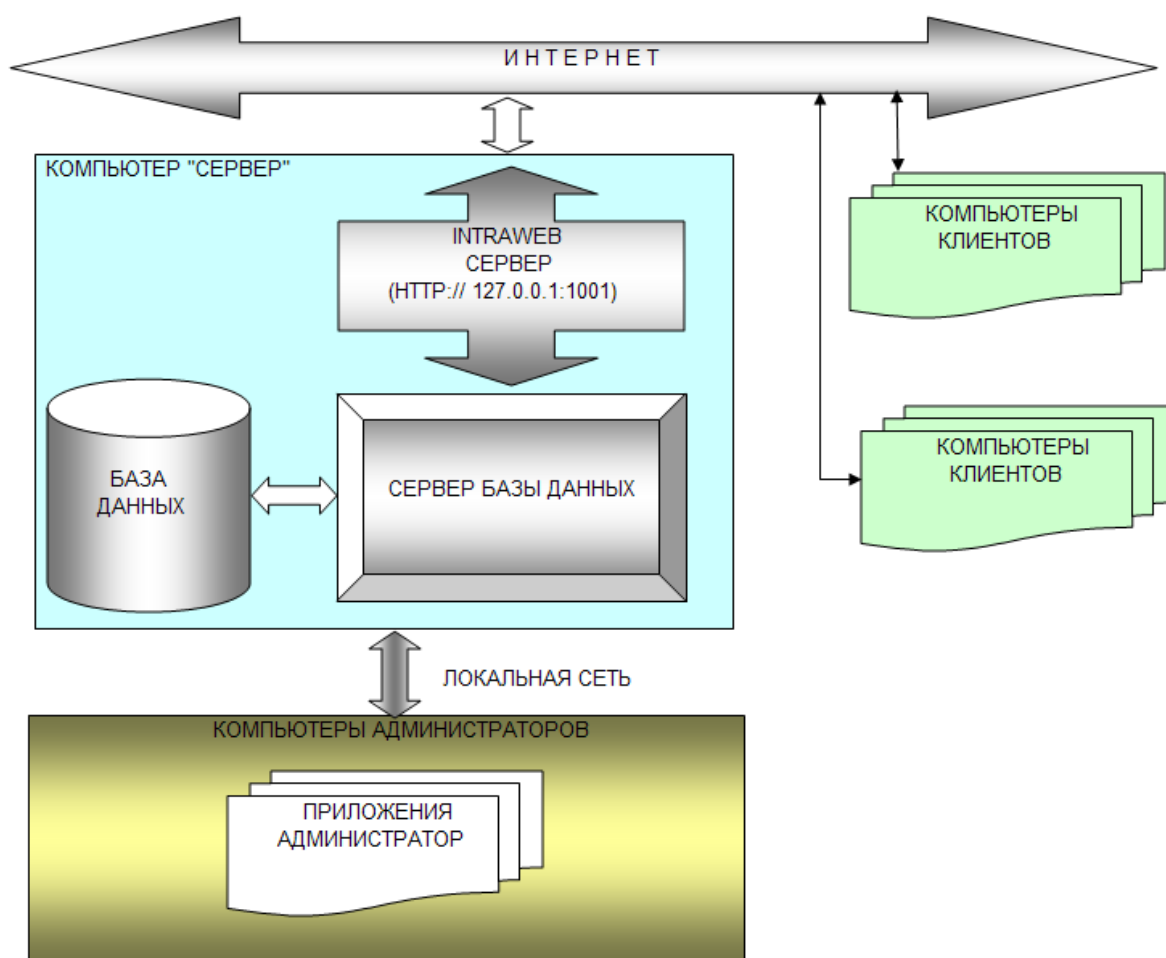


Рис. 2.1 Организация интернет магазина

Согласно приведенной на рисунке 2.1 схеме администраторы магазина обеспечивают в локальной сети ведение базы данных товаров и выполняют оформление продаж, на основе поступающих через Internet заказов от клиентов.

## 2.2. Схема базы данных

Базу данных, демонстрирующую работу интернет магазина, можно представить в виде двух пар связанных таблиц с отношением 1:M. Первая пара таблиц будет представлять предлагаемый перечень товаров и их описание (рис.2.2), а вторая пара описывать заявки клиентов и их заказы (рис.2.3).

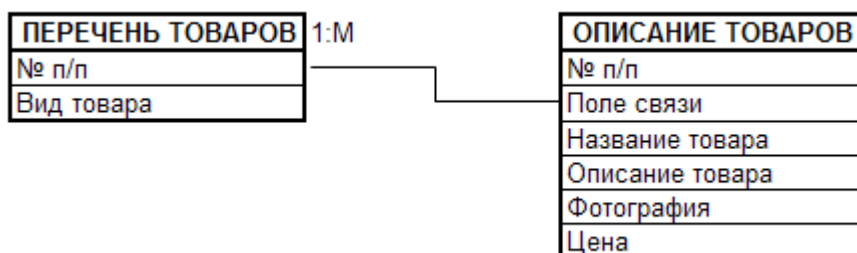


Рис. 2.2. Схема отношений таблиц описывающих товар

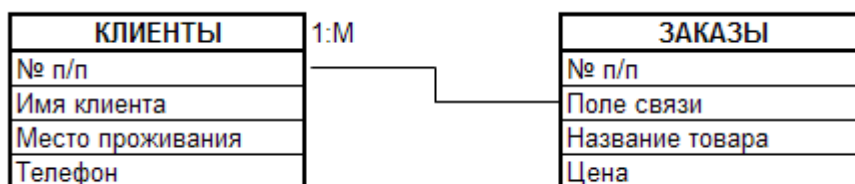


Рис. 2.3. Схема отношений таблиц описывающих заказы

**Примечание:** структура рассматриваемой базы данных предельно упрощена, так как нашей целью является демонстрация механизма транзакций.

## 2.3 Порядок разработки системы

На основании приведенных выше схем организации работы интернет магазина (рис.2.1) и схем базы данных (рис.2.2, рис.2.3) разработка системы состоит из следующих этапов:

- подготовительная стадия;
- создание базы данных;
- разработка СУБД «Администратор»;
- создание IntraWeb приложения.



### 2.3.1. Подготовительная стадия

Перед началом разработки системы необходимо определить директорий, в котором приложение будет сохранено. Для этого создадим в Windows папку с именем WEB\_MAGAZINE, в которой будет размещен комплексный проект. В папке WEB\_MAGAZINE создадим три папки:

ADMINISTRATOR – для размещения файлов проекта ADMIN;  
WEBMAGAZINE – для размещения файлов WEB приложения;  
PRODBASE – для размещения файлов базы данных.

### 2.3.2. Создание базы данных

Работу предлагается выполнять в следующем порядке:

1. В качестве сервера баз данных будем использовать **BDE** (IDAPI), а таблицы будем создавать в формате **paradox**. Приступим к созданию таблиц.

Используя утилиту *Database Desktop*, создадим алиас TO\_MAG к папке PRODBASE, в которой будут сохранены файлы базы данных (рис.2.4).

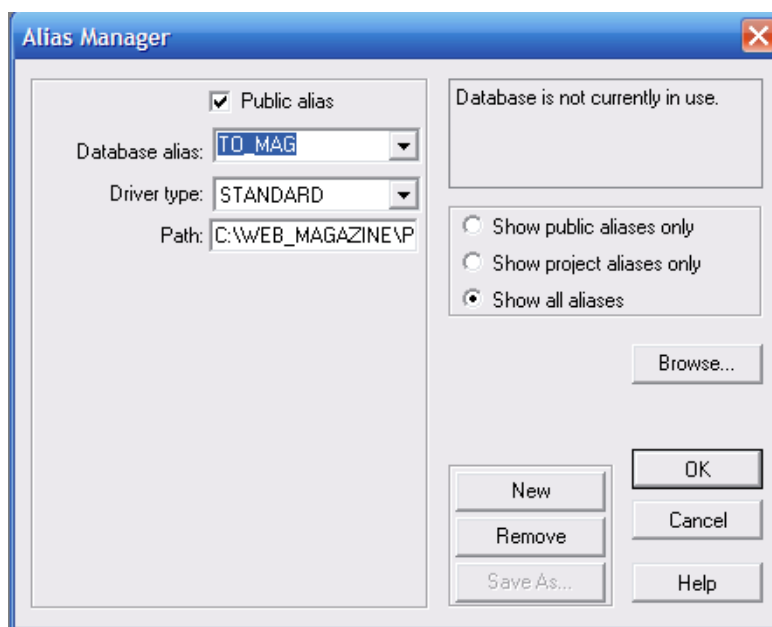


Рис.2.4. Создание алиаса

2. Создадим в *Database Desktop* главную таблицу PRODACTS (рис.2.5), описывающую название товаров, имеющих в магазине и сохраним ее по алиасу TO\_MAG под именем PRODACTS.

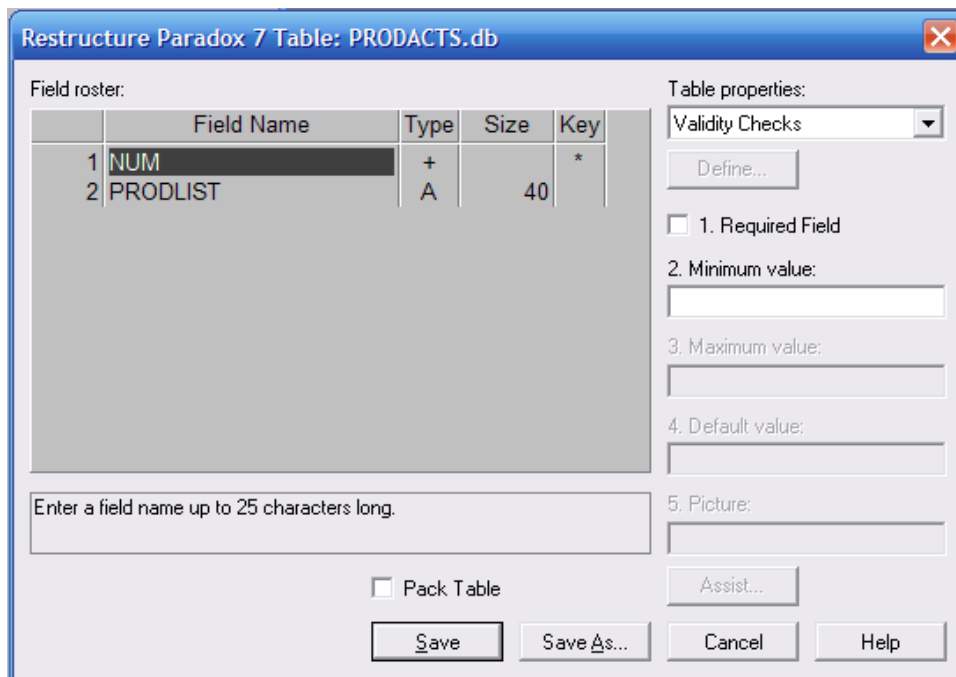


Рис.2.5. Структура таблицы PRODUCTS

3. Создадим связанную таблицу DESCRIPTION (рис.2.6), описывающую спецификации товаров.

**Примечание.** Поле LINK должно иметь вторичный индекс ID.

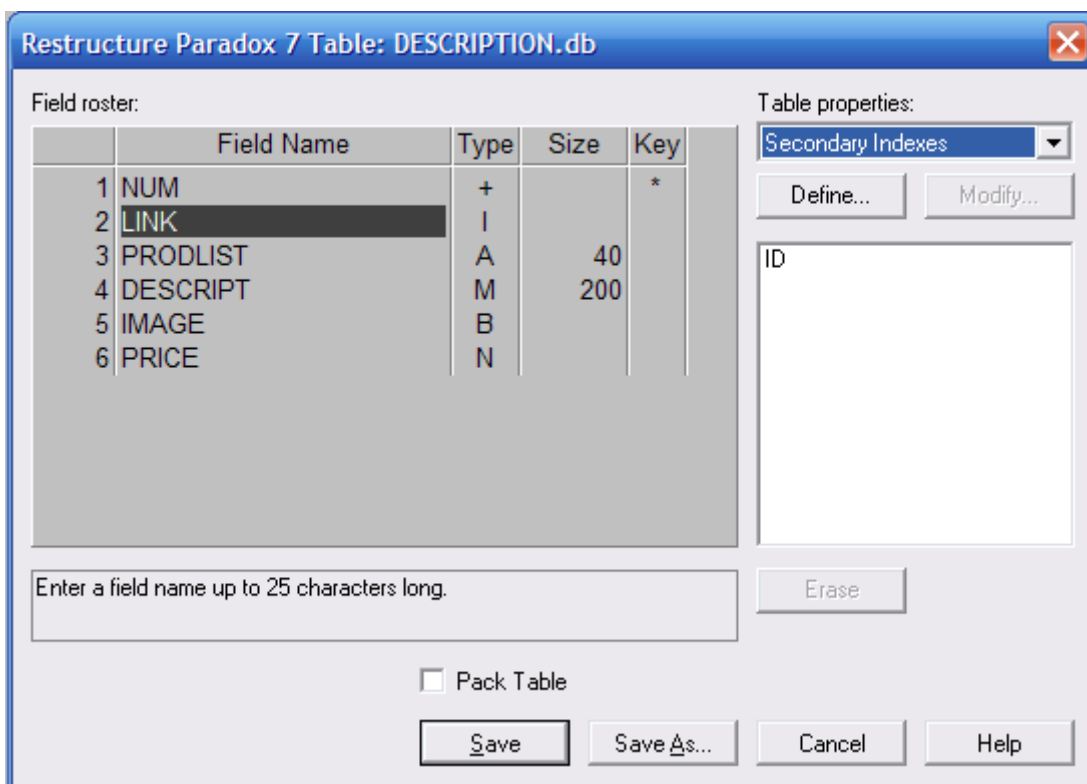


Рис.2.6. Структура таблицы DESCRIPTION

4. Сохраним созданную таблицу по алиасу TO\_MAG под именем DESCRIPTION.

5. Создадим главную таблицу CLIENT\_LIST (рис.2.7), описывающую регистрационные данные покупателя и сохраним ее по алиасу TO\_MAG под именем CLIENT\_LIST.

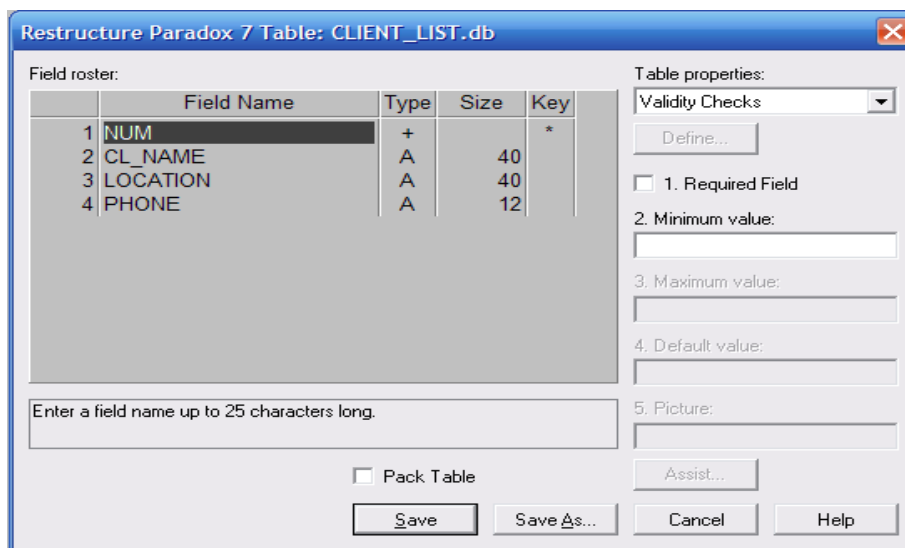


Рис.2.7. Структура таблицы CLIENT\_LIST

6. Создадим связанную таблицу CLIENT\_ORD (рис.2.8), описывающую заявку покупателя.

**Примечание.** Поле LINK должно иметь вторичный индекс ID.

7. Сохраним созданную таблицу по алиасу TO\_MAG под именем CLIENT\_ORD.

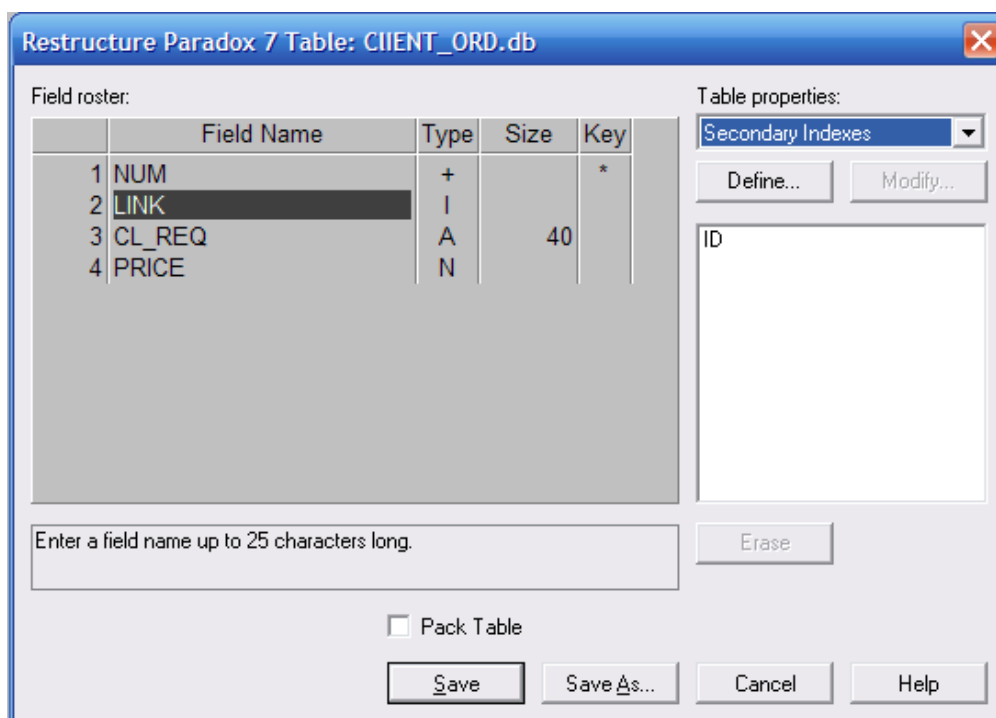


Рис.2.8. Структура таблицы CLIENT\_ORD

### 2.3.3. Разработка СУБД «Администратор»

Работу предлагается выполнять в следующем порядке:

1. Разработку СУБД начнем с создания нового проекта, в котором изменим имя формы на MainForm. Сохраним проект в папку ADMINISTRATOR как: модуль → Unit1, project → ADMIN. Далее создадим модуль данных, используя репозиторий Delphi: File/New/Other (рис.2.9) и сохраним созданный модуль под именем Unit2 в папку проекта.

2. Добавим сохраненный модуль данных к проекту, выполнив команду меню Project/ Add Project. В модуле Unit1 главной формы необходимо определить использование модуля данных Unit2 (рис.2.10). В завершении выполним команду Save All, сохраняющую все файлы проекта.

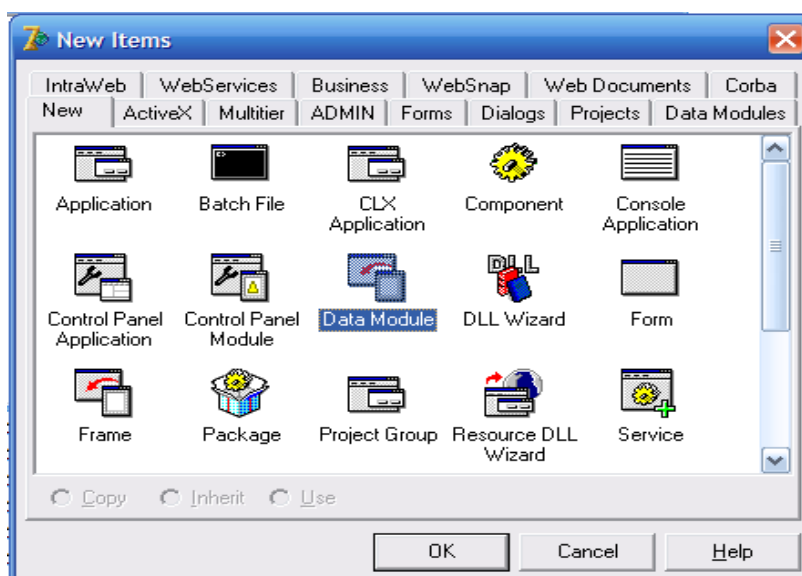


Рис.2.9. Выбор Data Module в репозитории Delphi

```
private
    { Private declarations }
public
    { Public declarations }
end;

var
    MainForm: TMainForm;

implementation

uses Unit2;

{$R *.dfm}

end.
```

Рис.2.10. Использование Unit2

3. Разместим в модуле следующие компоненты набора данных и компоненты связи с данными в соответствии с рисунком 2.11 и изменим их имена, согласно значениям, приведенным в таблице 2.1.

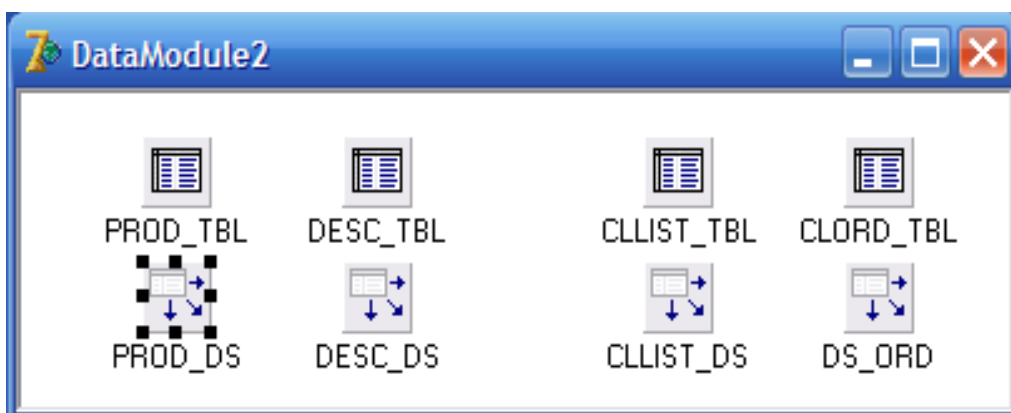


Рис.2.11. Размещение компонентов в модуле данных

Таблица 2.1

Значения свойства Name компонентов

№ п/п	Имя компонента	Класс компонента
1	PROD_TBL	TTable
2	DESC_TBL	TTable
3	CLLIST_TBL	TTable
4	CLORD_TBL	TTable
5	PROD_DS	TDataSource
6	DESC_DS	TDataSource
7	CLLIST_DS	TDataSource
8	DS_ORD	TDataSource

4. Выполним одноименные соединения таблиц и компонентов связи (таблица 2.2), используя свойство DataSet компонентов класса TDataSource:

Таблица 2.2

Значения свойств DataSet компонентов класса TDataSource

№ п/п	Имя компонента	Значение свойства DataSet
1	PROD_DS	PROD_TBL
2	DESC_DS	DESC_TBL
3	CLLIST_DS	CLLIST_TBL
4	DS_ORD	CLORD_TBL

5. В свойстве DataBaseName всех компонентов набора данных класса TTable необходимо выбрать алиас TO\_MAG, а в свойстве TableName компонентов набора данных выбрать значения, приведенные в таблице 2.3.

Таблица 2.3

Значения свойств TableName компонентов класса TTable

№ п/п	Имя компонента	Значение свойства TableName
1	PROD_TBL	PRODACTS.db
2	DESC_TBL	DESCRIPTION.db
3	CLLIST_TBL	CLIENT_LIST.db
4	CLORD_TBL	CIIENT_ORD.db

6. Теперь необходимо создать отношения 1:M между таблицами PRODACTS.db : DESCRIPTION.db, для чего в свойстве MasterSource компонента DESC\_TBL выбрать значение PROD\_DS, а значение свойства MasterField создать с помощью редактора свойства Field Link Designer, выбрав в списке Available Index значение ID (рис.2.12,а) и далее, в окнах DetailFields MasterFields соответствующие значения LINK и NUM. Для установления отношения выполнить команду Add (рис.2.12,б).

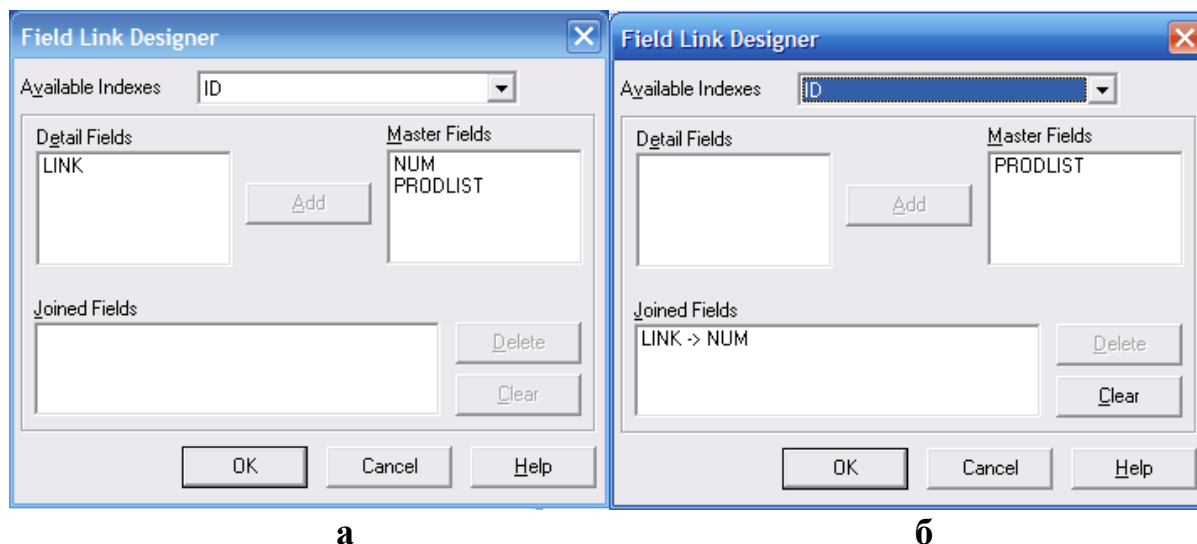
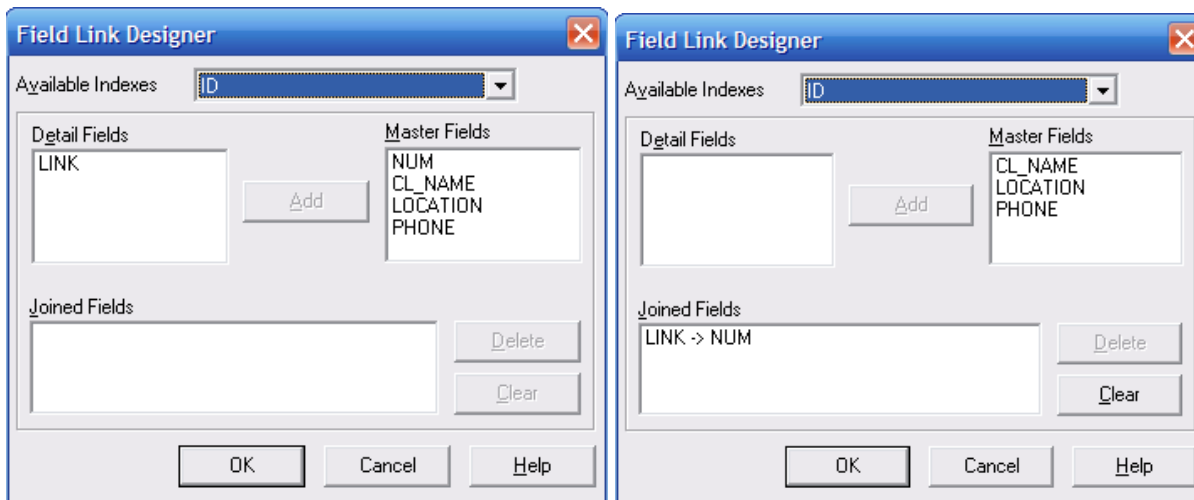


Рис.2.12. Создание отношения 1:M для таблиц PRODACTS.db : DESCRIPTION.db

7. Создадим отношения 1:M между таблицами CLIENT\_LIST.db : DESCRIPTION.db, для чего в свойстве MasterSource компонента CLORD\_TBL выберем значение CLLIST\_DS, а значение свойства MasterField создадим с по-

мощью редактора свойства Field Link Designer, выбрав в списке Available Index значение ID (рис.2.13,а), а в окнах DetailFields и MasterFields соответствующие значения LINK и NUM. Для установления отношения выполним команду Add (рис.2.13,б).



**а**

**б**

Рис.2.13. Создание отношения 1:М для таблиц CLIENT\_LIST.db :  
DESCRIPTION.db

8. Далее необходимо создать для каждой таблицы доменную структуру, для чего выполним поочередно двойной щелчок по компонентам набора данных PROD\_TBL, DESC\_TBL, CLLIST\_TBL и CLORD\_TBL. В открывшихся окнах редакторов выполним команду Add All Fields контекстного меню (рис.2.14).

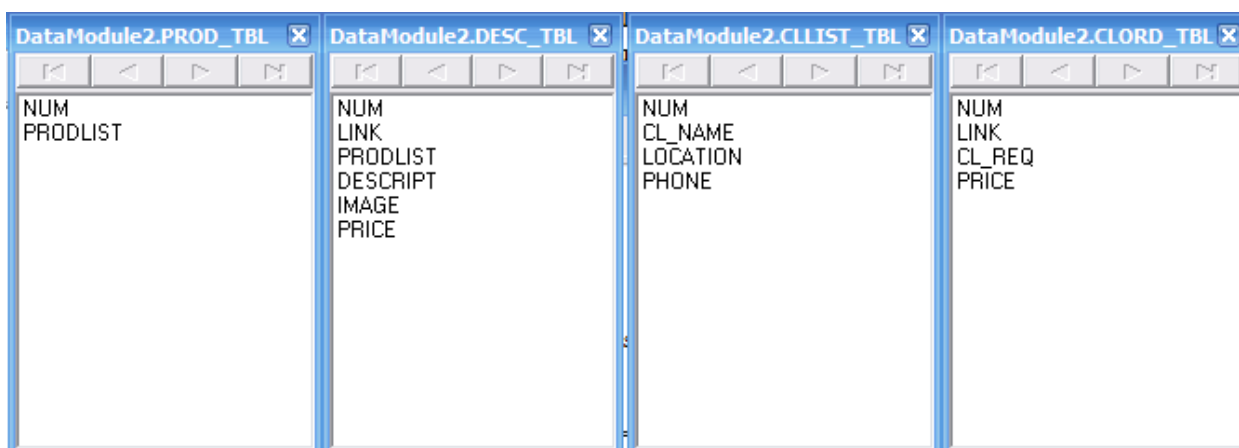


Рис.2.14. Создание доменных структур

9. Установим для всех компонентов набора данных PROD\_TBL, DESC\_TBL, CLLIST\_TBL и CLORD\_TBL значение свойства Active → True и выполним команду Save All, сохраняющую все файлы проекта.

10. Изменим для формы значения свойств:

- Caption := АДМИНИСТРАТОР;
- BorderStyle := bsSingle;
- Position := poScreenCenter.

11. Установим в форму компонент PageControl1 и создадим в нем две закладки TabSheet1 и TabSheet2, выполнив команду контекстного меню NewPage (рис.2.15). Значение свойства Align → alClient, а значение свойства Caption для закладок TabSheet и TabSheet2 установить равным “Товары” и “Заявки” соответственно.



Рис.2.15. Установка компонента PageControl1 в форму и создание закладок

12. Установим на закладку TabSheet1 компонента PageControl1 компоненты (рис.2.16):

- Panel1 (свойство Align := alLeft);
- Splitter1 (свойство Color установить по желанию);
- Panel2 (свойство Align := alLeft).



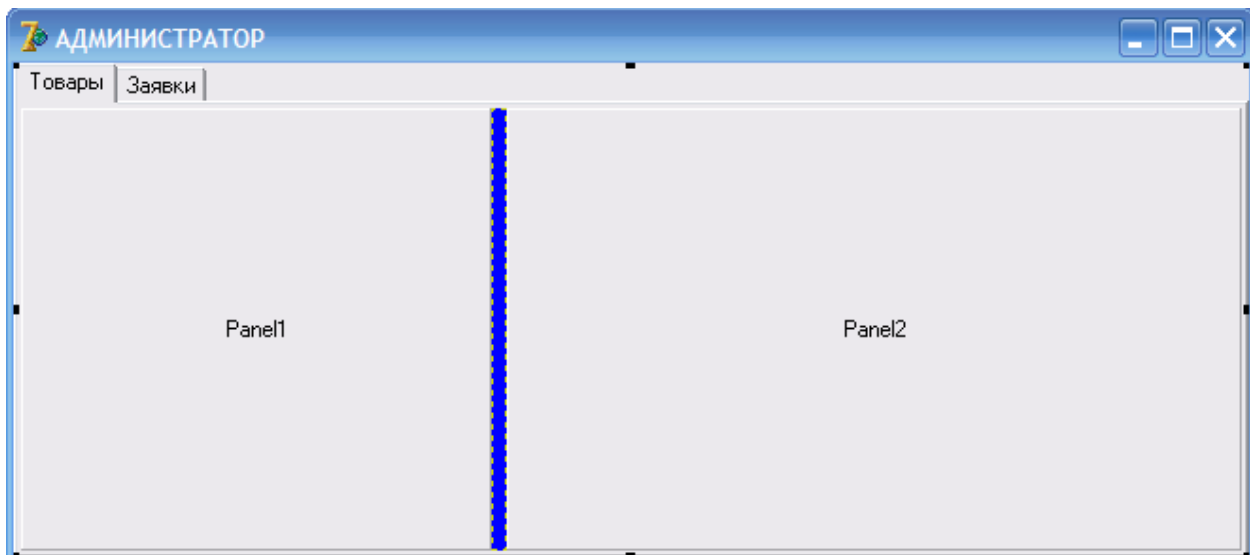


Рис.2.16. Расположение компонентов в форме

13. Установим на панелях Panel1 и Panel2 компоненты Panel3, Panel4, Panel5, Panel6 как показано на рисунке 2.17. Свойство Align для компонентов Panel3, Panel5 должно иметь значение alTop, а для компонентов Panel4, Panel6 должно иметь значение alBottom.

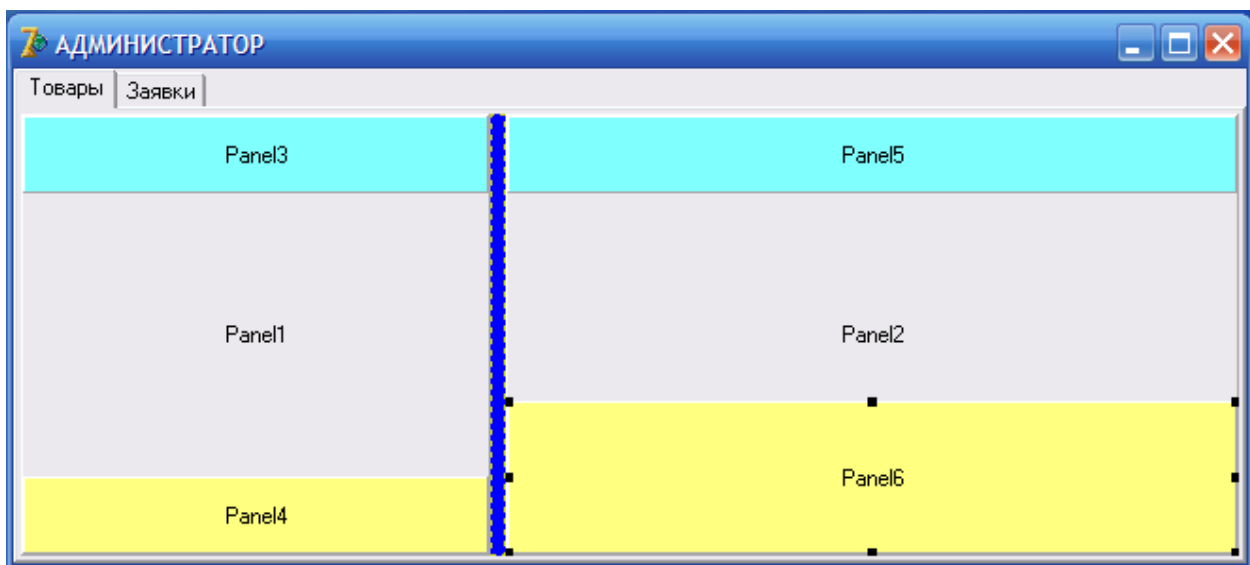


Рис.2.17. Установка компонентов

14. Далее необходимо установить на Panel1 компонент DBGrid1, а на Panel2 компоненты DBGrid2 и DBMemo1. Свойство Align компонентов DBGrid1 и DBGrid2 установить в значение alClient и alLeft соответственно, а для свойства Align компонента DBMemo1 выбрать значение alClient. Также установим на Panel4 и Panel6 компоненты DBNavigator1 и DBNavigator2 и изменить заголовки компонентов (свойство Caption) Panel3, Panel5 на значения “НАИМЕНОВАНИЕ ТОВАРА” и “СПИСОК ТОВАРОВ” соответственно, а

первоначальное значение свойства Caption компонентов Panel4 и Panel6 удалить.

15. Установить на компонент Panel6 кнопку Button1, изменив свойство Caption на “Загрузить” и компонент DBImage1 для свойства Stretch которого выбрать значение True. Также установить невидимый компонент диалога OpenPictureDialog1 (рис.2.18).

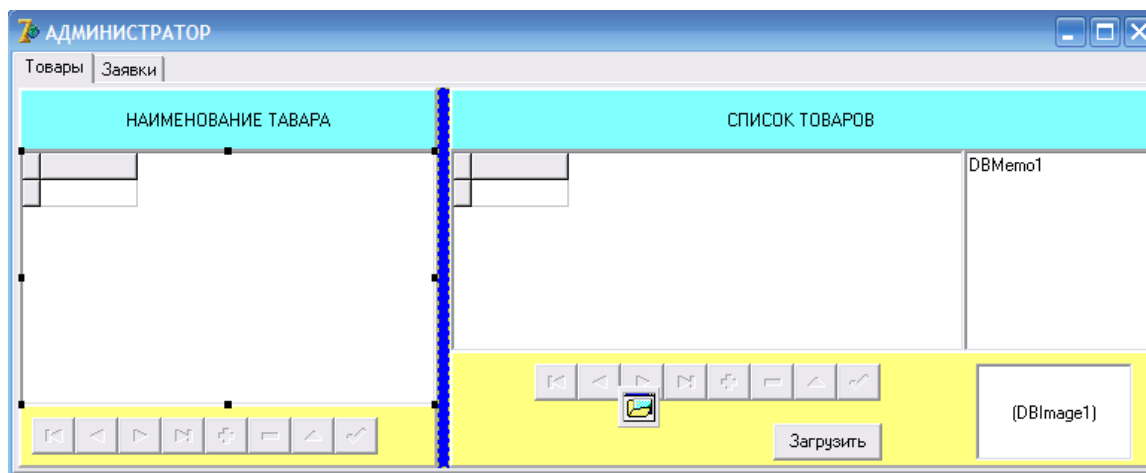


Рис.2.18. Установка компонентов

16. Подключим компоненты DBGrid1 и DBNavigator1 к источнику данных PROD\_TBL, выбрав значения свойства DataSource = DataModule2.PROD\_DS. Также подключим компоненты DBGrid2 и DBNavigator2 к источнику данных DESC\_TBL, выбрав значения свойства DataSource = DataModule2.DESC\_DS.

17. Для компонентов DBNavigator1 и DBNavigator2 установим невидимыми кнопки Cancel и Refresh выбрав значения false для nbCancel и nbRefresh в свойствах VisibleButtons.

После завершения указанных действий внешний вид закладки “Товары” должен приблизительно соответствовать рисунку 2.19.

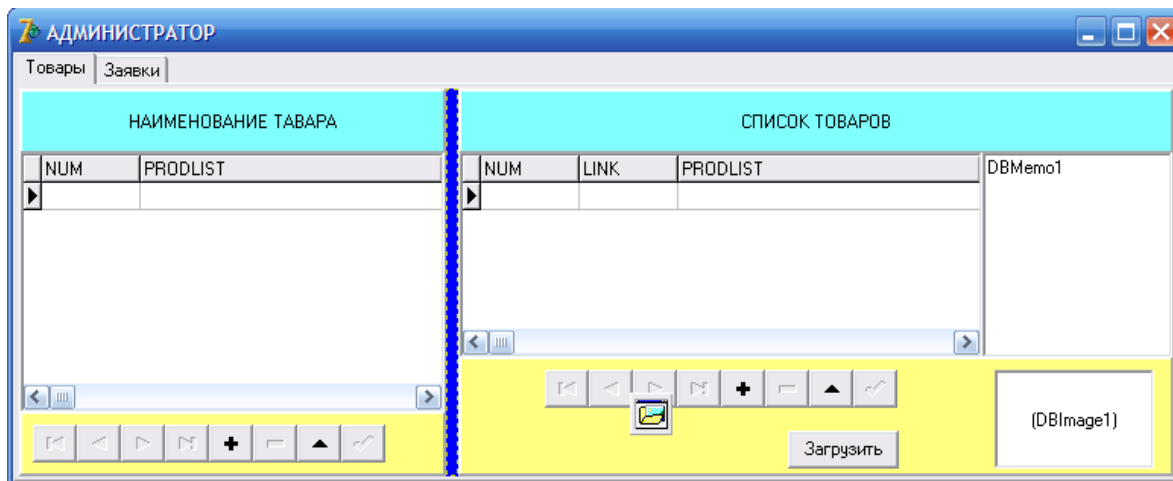


Рис.2.19. Вид закладки “Товары”

18. Для изменения названий заголовков столбцов и представления данных в компонентах DBGrid1 и DBGrid2 необходимо дважды щелкнуть мышью по указанным компонентам и далее выполнить команду Add All Fields в окнах редакторов полей (рис.2.20).

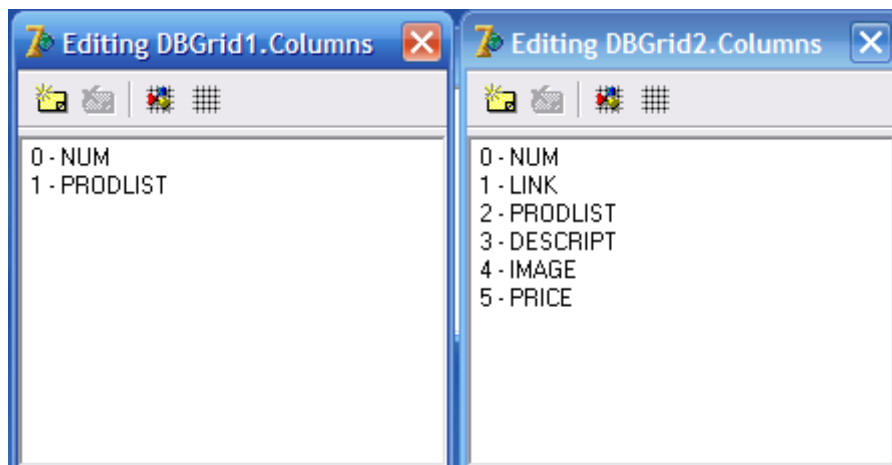


Рис.2.20. Вид редакторов полей компонентов DBGrid

19. Для того чтобы пользователю не выводиться значения служебных и бинарных полей достаточно удалить их из редактора, выбрав соответствующее поле и выполнив команду Delete контекстного меню. Для DBGrid1 необходимо удалить поле NUM, а для DBGrid2 поля NUM, LINK, DESCRIPT и IMAGE (рис.2.21).

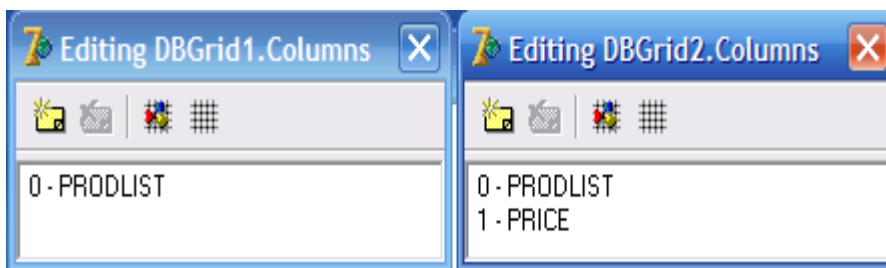


Рис.2.21. Вид редакторов полей компонентов DBGrid

20. В завершении дизайна закладки “Товары” выполним русификацию заголовков столбцов компонентов DBGrid1 и DBGrid2, а также свяжем компоненты DBMemo1 и DBImage1 с соответствующими полями набора данных, для чего:

- выберем в редакторе DBGrid1 поле PRODLIST и в инспекторе объектов установим соответствующие свойства в следующие значения – Title/Caption := “НАЗВАНИЕ ТОВАРА” и Title/Alingment := taCenter;

- выберем в редакторе DBGrid2 поочередно поля PRODLIST и PRICE в инспекторе объектов установим соответствующие свойств в следующие значения – для PRODLIST Title/Caption := “НАЗВАНИЕ ТОВАРА”, для PRICE := “ЦЕНА” и Title/Alingment обоих полей установим в значение taCenter.

Свяжем компоненты DBMemo1 и DBImage1 с соответствующими полями набора данных (таблица 2.4)

Таблица 2.4

Установление связи с полями

№ п/п	Имя компонента	Значение DataSource	Значение DataField
1	DBMemo1	DataModule2.DESC_DS	DESCRIPT
2	DBImage1	DataModule2.DESC_DS	IMAGE

Значение свойства ScrollBars компонента DBMemo1 установим в значение ssBoth. Завершенный дизайн страницы должен соответствовать виду, приведенному на рисунке 2.22.

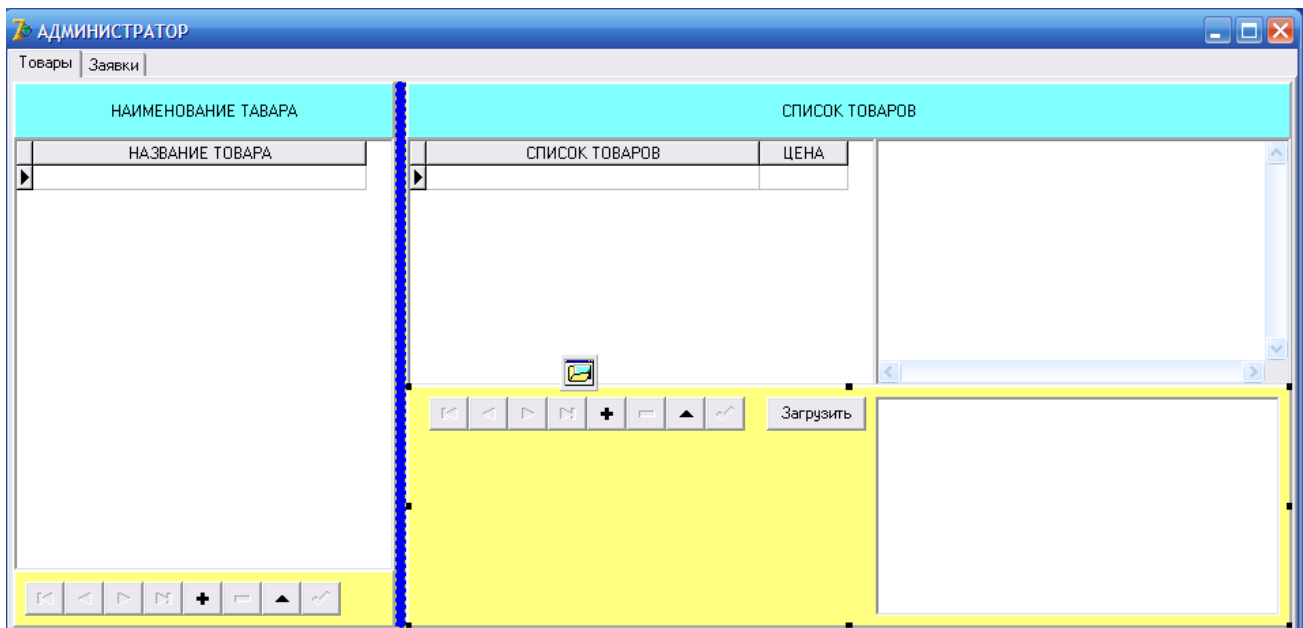


Рис.2.22. Дизайн страницы "Товары"

21. Единственным методом, который нам придется реализовать, является метод загрузки графического файла (картинки) в бинарное поле. Данный метод реализуем в обработчике события onClick кнопки "Загрузить" (рис.2.23). Перед написанием метода не забудьте установить в форму компонент диалога OpenFileDialog.

```

procedure TMainForm.Button1Click(Sender: TObject);
begin
  if DataModule2.DESC_TBL.RecordCount > 0 then // если имеются записи, то:
  begin
    DataModule2.DESC_TBL.Edit; // Переводим набор данных в редактирование
    if OpenFileDialog1.Execute then // Если диалог, то:
      DBImage1.Picture.LoadFromFile(OpenFileDialog1.FileName); //загружаем файл
    DataModule2.DESC_TBL.Post; // Сохраняем файл в бинарном поле БД
  end;
end;

```

Рис.2.23. Метод загрузки картинки

22. Теперь самостоятельно выполните дизайн страницы "Заявки", которая должна соответствовать виду, приведенному на рисунке 2.24.

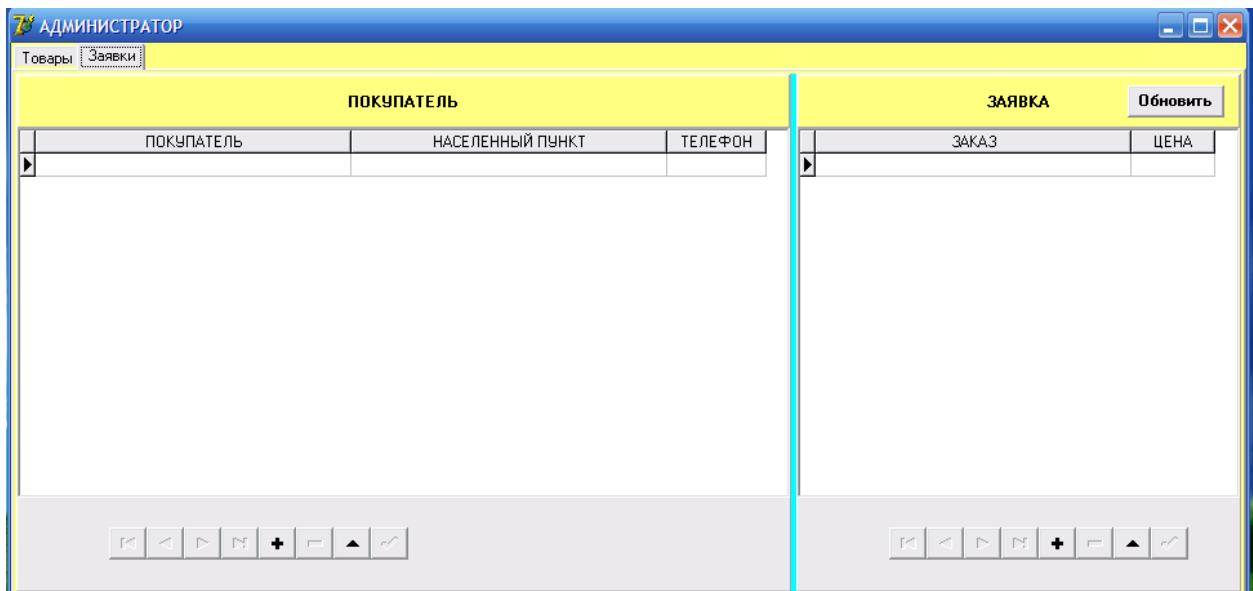


Рис.2.24. Дизайн страницы "Заявки"

23. Необходимо определить метод для события щелчка кнопки "Обновить". Это будет метод переоткрытия таблиц Refresh для обновления просмотра поступивших заказов (рис.2.25).

```

procedure TMainForm.Button2Click(Sender: TObject);
begin
  DataModule2.CLORD_TBL.Close;
  DataModule2.CLLIST_TBL.Close;
  DataModule2.CLLIST_TBL.Open;
  DataModule2.CLORD_TBL.Open;
end;

```

Рис.2.25. Метод Refresh

24. Теперь можно записать в базу данных несколько позиций товара и описать их характеристики (рис.2.26).

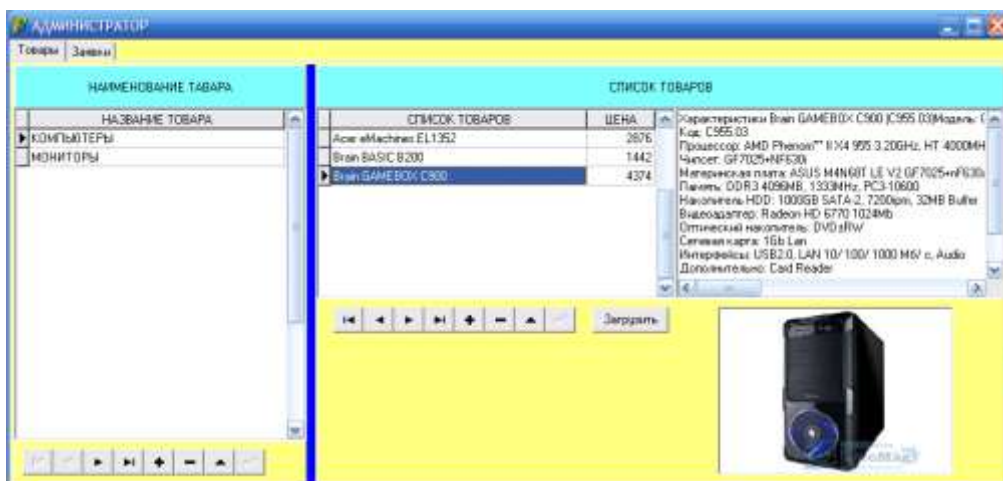


Рис.2.26. Запись в базу данных позиций товаров

### 2.3.4. Создание IntraWeb-приложения

Работу предлагается выполнять в следующем порядке:

1. Для создания IntraWeb приложения воспользуемся репозиторием Delphi, для чего выполним команду File/New/Other главного меню (рис.2.27) и выберем на странице IntraWeb проект типа Stand Alone Application with Data Module. Указанный тип проекта включает в себя *Сервер приложения, модуль главной формы и модуль данных.*

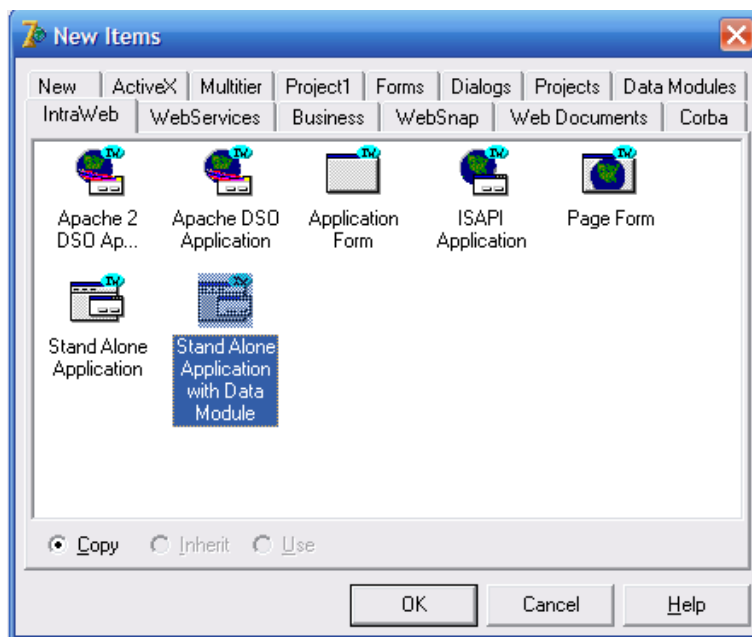


Рис.2.27. Выбор типа проекта в репозитории Delphi

2. В окне Select Directory необходимо выбрать путь к папке размещения файлов проекта. В нашем случае X:/WEB\_MAGAZINE/WEBMAGAZINE (рис.2.28).

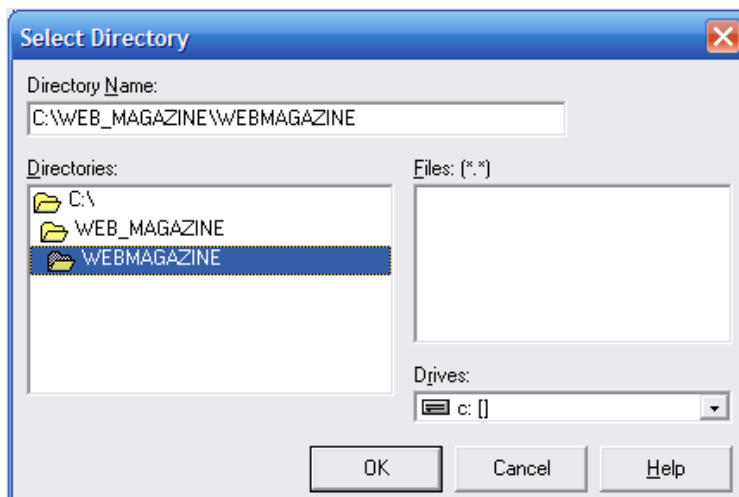


Рис.2.28. Выбор пути размещения проекта

2. Выполним команду меню View/Project Manager и из окна менеджера проектов (рис.2.29) покажем все три формы (рис.2.30)

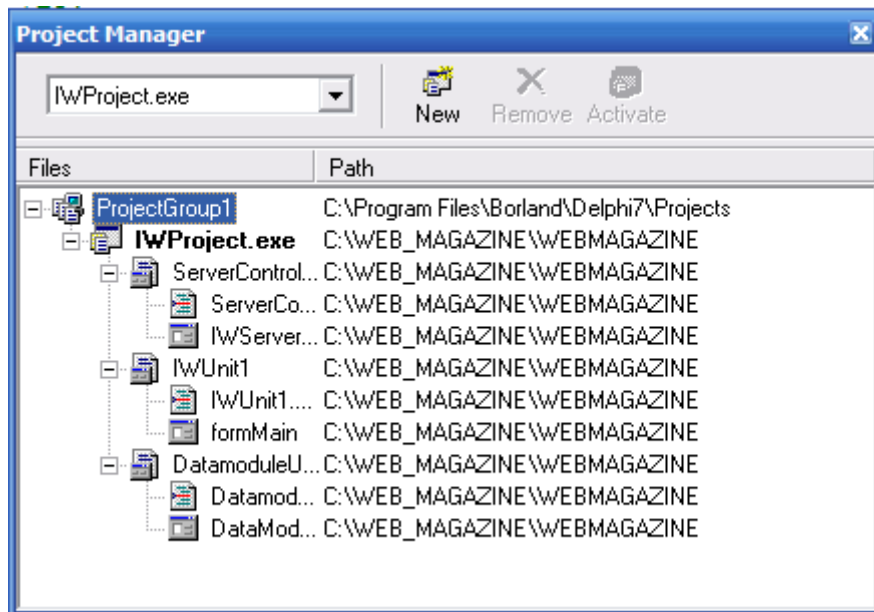


Рис.2.29. Окно менеджера проектов

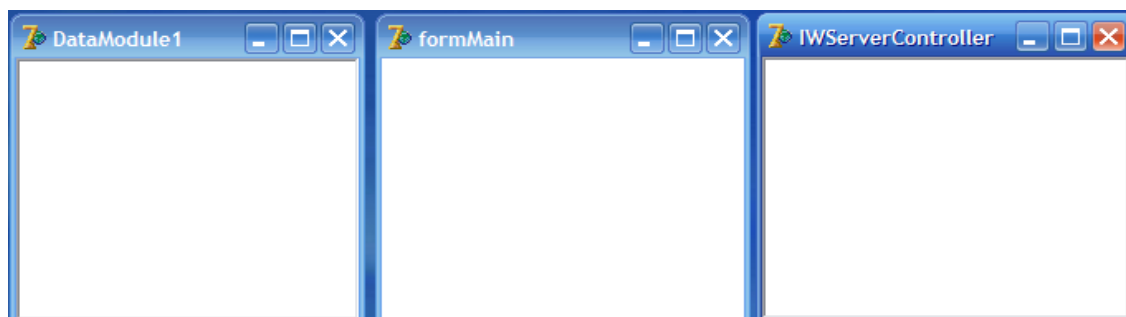


Рис.2.30. Вид форм проекта

3. Выполним команду RUN, которая вызовет окно сервера приложения (рис.2.31).

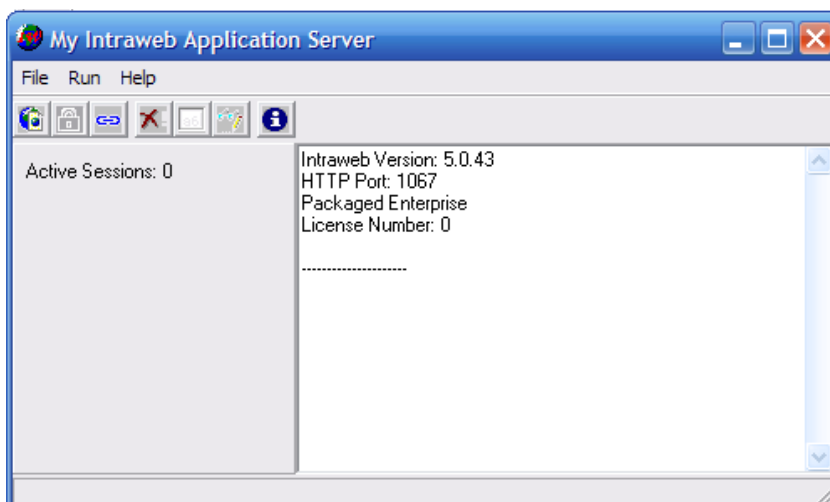


Рис.2.31. Вид окна сервера приложения



4. В окне сервера приложения выполним команду RUN → Execute, которая должна запустить Internet Explorer с пустой Web страницей (рис.2.32).

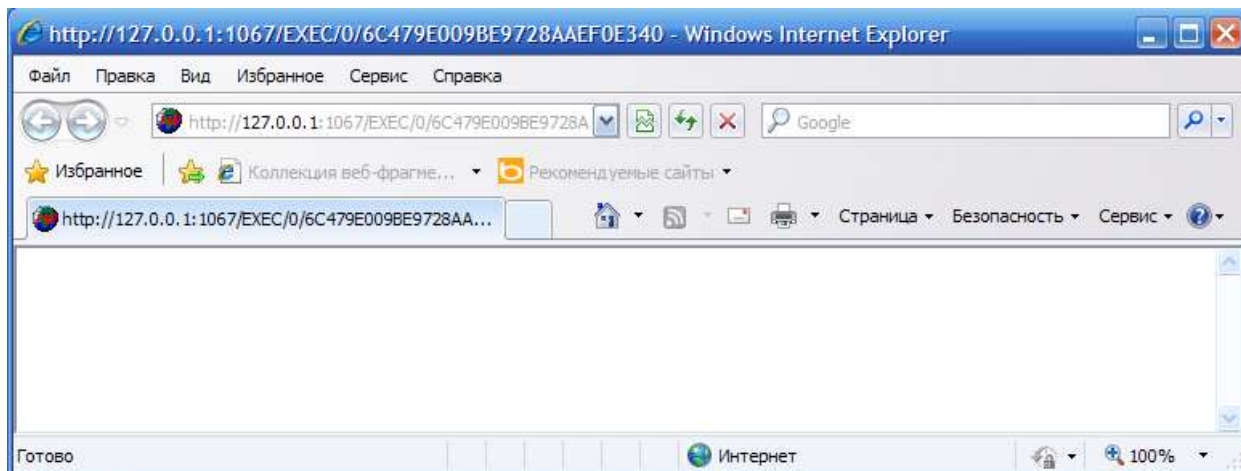


Рис.2.32. Окно Internet Explorer

5. Закроем Internet Explorer и сервер приложения,

6. Теперь приступим к реализации логики модуля данных. Установим в форму модуля данных по четыре компонента класса TTable и TDataSource (рис.2.33).

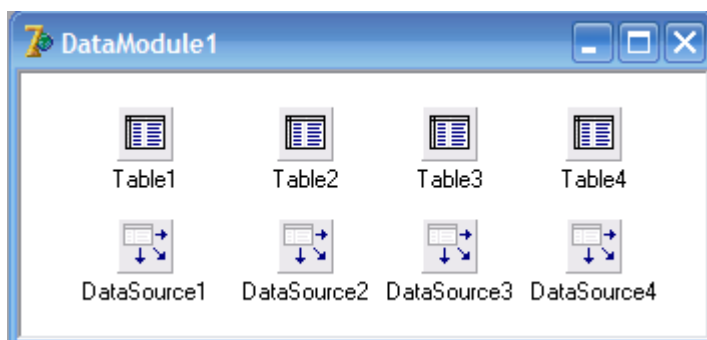


Рис.2.33. Установка компонентов доступа к данным

7. Переименуем установленные компоненты, как показано на рисунке 2.34 и посредством свойства DataSet компонентов TDataSource свяжем соответствующие пары (таблица 2.5).

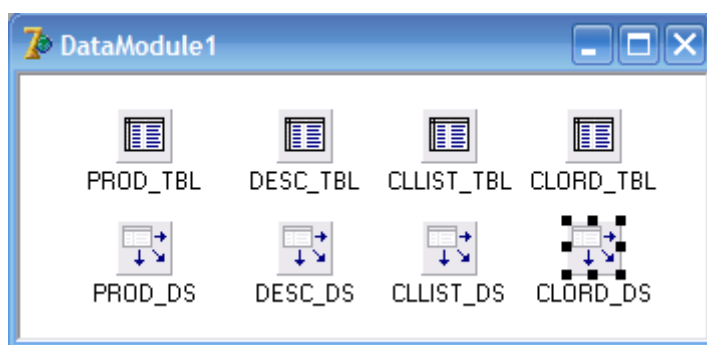


Рис.2.34. Переименование компонентов

Таблица 2.5

Значения свойств компонентов

№ п/п	Имя компонента	Значение свойства DataSet
1	PROD_DS	PROD_TBL
2	DESC_DS	DESC_TBL
3	CLLIST_DS	CLLIST_TBL
4	CLORD_DS	CLORD_TBL

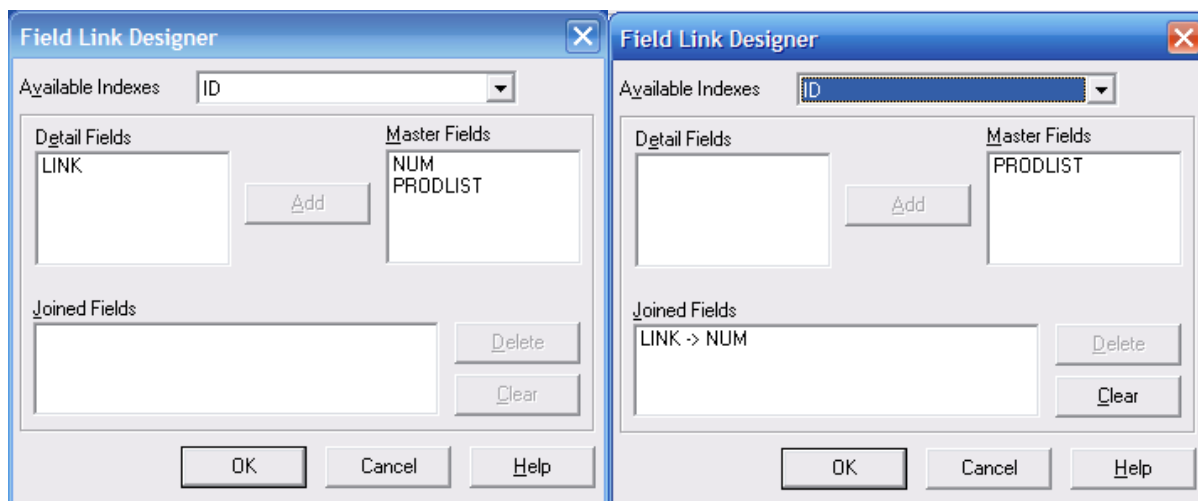
8. Для всех компонентов набора данных класса TTable выбрать алиас TO\_MAG в свойстве DataBaseName. В свойстве TableName компонентов набора данных выбрать значения, приведенные в таблице 2.6.

Таблица 2.6

Значения свойств TableName компонентов класса TTable

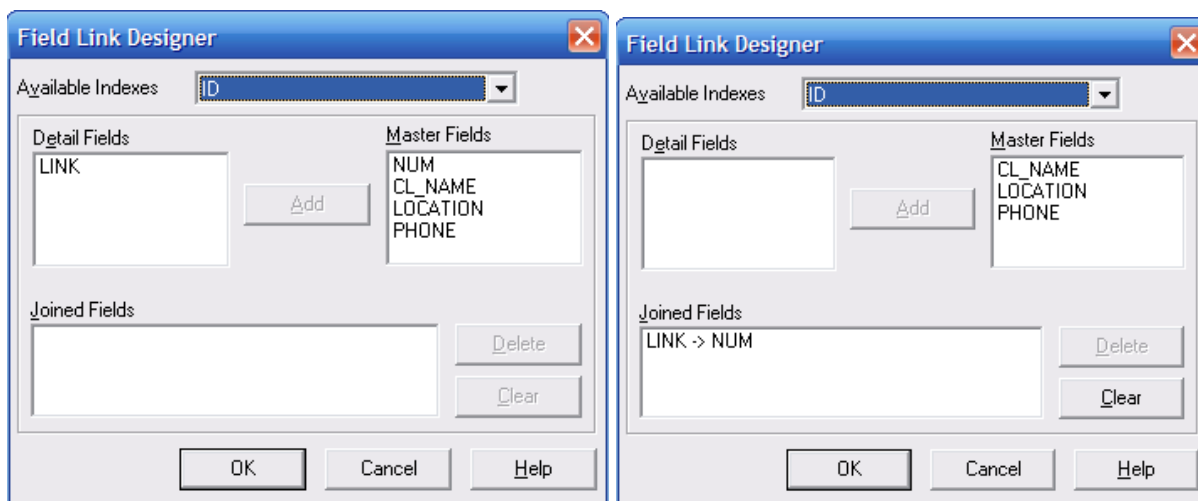
№ п/п	Имя компонента	Значение свойства TableName
1	PROD_TBL	PRODUCTS.db
2	DESC_TBL	DESCRIPTION.db
3	CLLIST_TBL	CLIENT_LIST.db
4	CLORD_TBL	CLIENT_ORD.db

9. Для обеспечения отношения 1:M между таблицами PRODUCTS.db : DESCRIPTION.db, в свойстве MasterSource компонента DESC\_TBL необходимо выбрать значение PROD\_DS, а значение свойства MasterField создать с помощью редактора свойства Field Link Designer, выбрав в списке Availabel Index значение ID (рис.2.35,а) и далее в окнах Detail Fields Master Fields соответствующие значения LINK и NUM. Для установления отношения выполнить команду Add (Рис.2.35,б).



**а** **б**  
 Рис.2.35. Создание отношения 1:М для таблиц PRODACTS.db :  
 DESCRIPTION.db

10. Создадим отношение 1:М между таблицами CLIENT\_LIST.db : DESCRIPTION.db, для чего в свойстве MasterSource компонента CLORD\_TBL выберем значение CLLIST\_DS, а значение свойства MasterField создадим с помощью редактора свойства Field Link Designer, выбрав в списке Available Index значение ID (рис.2.36,а) и далее в окнах Detail Fields Master Fields соответствующие значения LINK и NUM. Для установления отношения выполнить команду Add (рис.2.36,б).



**а** **б**  
 Рис.2.36. Создание отношения 1:М для таблиц CLIENT\_LIST.db :  
 DESCRIPTION.db

11. Создадим для каждой таблицы доменную структуру, для чего выполним поочередно двойной щелчок по компонентам набора данных PROD\_TBL, DESC\_TBL, CLLIST\_TBL и CLORD\_TBL. В открывшихся окнах редакторов выполним команду Add All Fields контекстного меню (рис.2.37).

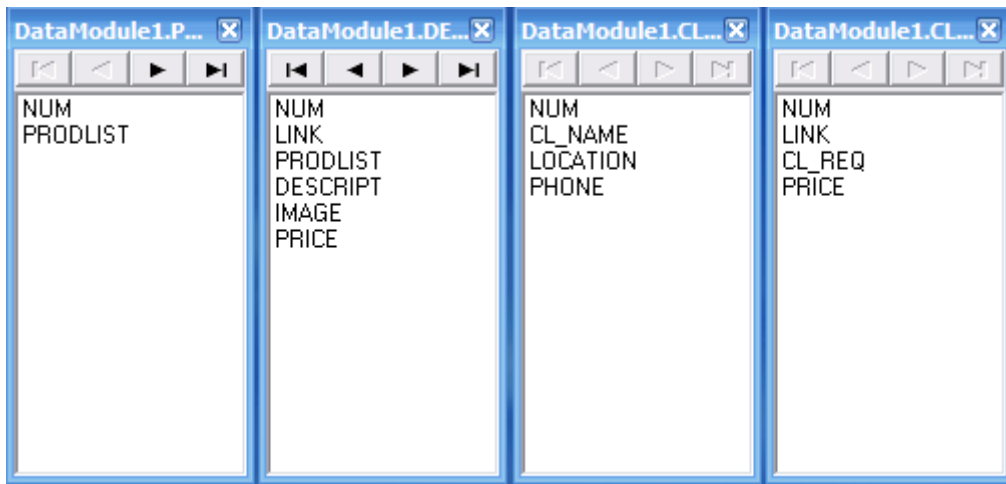


Рис.2.37. Создание доменных структур

12. Теперь необходимо установить для всех компонентов набора данных PROD\_TBL, DESC\_TBL, CLLIST\_TBL и CLORD\_TBL значение свойства **Active** равное **True**.

13. Далее выполним команду Save All, сохраняющую все файлы проекта.

14. Для того, чтобы модуль данных был доступен из главной формы проекта необходимо его объявить в разделе Uses модуля unit IWUnit1. Также для выполнения вычислений необходимо добавить в раздел модуль SysUtils. После объявления модуля необходимо выполнить команду Run (рис.2.38).

```
implementation
{$R *.dfm}

uses
  ServerController, DatamoduleUnit, SysUtils;
```

Рис.2.38. Объявление модулей в разделе **uses**

15. Теперь можно приступить к дизайну главной формы приложения. Установим в форму formMain такие компоненты из палитры IWData (рис.2.39):

- IWDBGrid1;
- IWDBGrid2;
- IWDBImage1;
- IWDBMemo1.

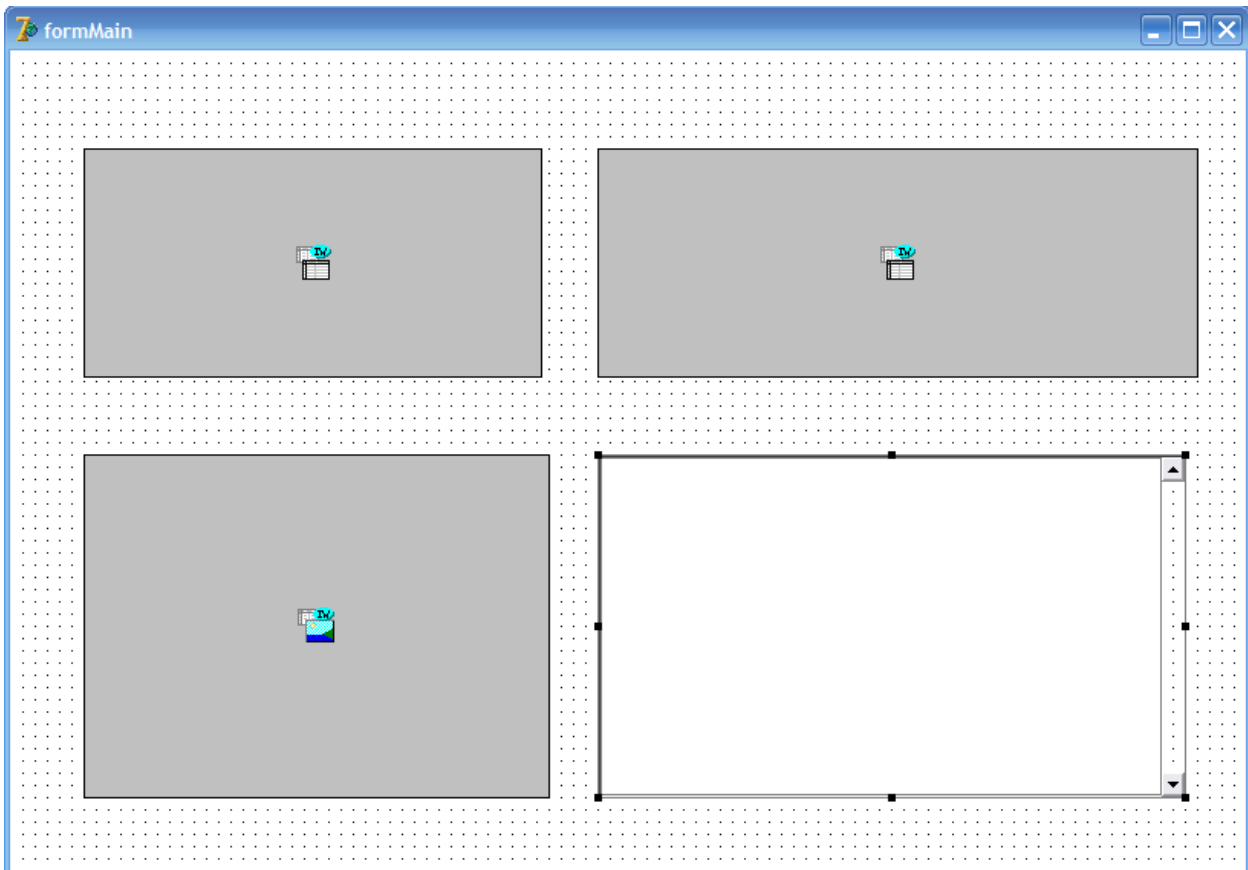


Рис.2.39. Установка компонентов в форму

16. Далее свяжем установленные элементы управления с соответствующими наборами данных модуля данных (таблица 2.7).

Таблица 2.7

Связь компонентов

№ п/п	Имя компонента	Свойство	Значение
1	IWDBGrid1	DataSource	DataModule1.PROD_DS
2	IWDBGrid2	DataSource	DataModule1.DESC_DS
3	IWDBImage1	DataSource	DataModule1.DESC_DS
4	IWDBImage1	DataField	IMAGE
5	IWDBMemo1	DataSource	DataModule1.DESC_DS
6	IWDBMemo1	DataField	DESCRIPT

17. Теперь можно проверить, как будут отображаться данные на нашей WEB странице. Выполним команды RUN Delphi и запущенного сервера приложения. После выполнения команды RUN сервера в Internet Explorer откроется Web страница, вид которой должен приблизительно соответствовать виду, приведенному на рисунке 2.40.

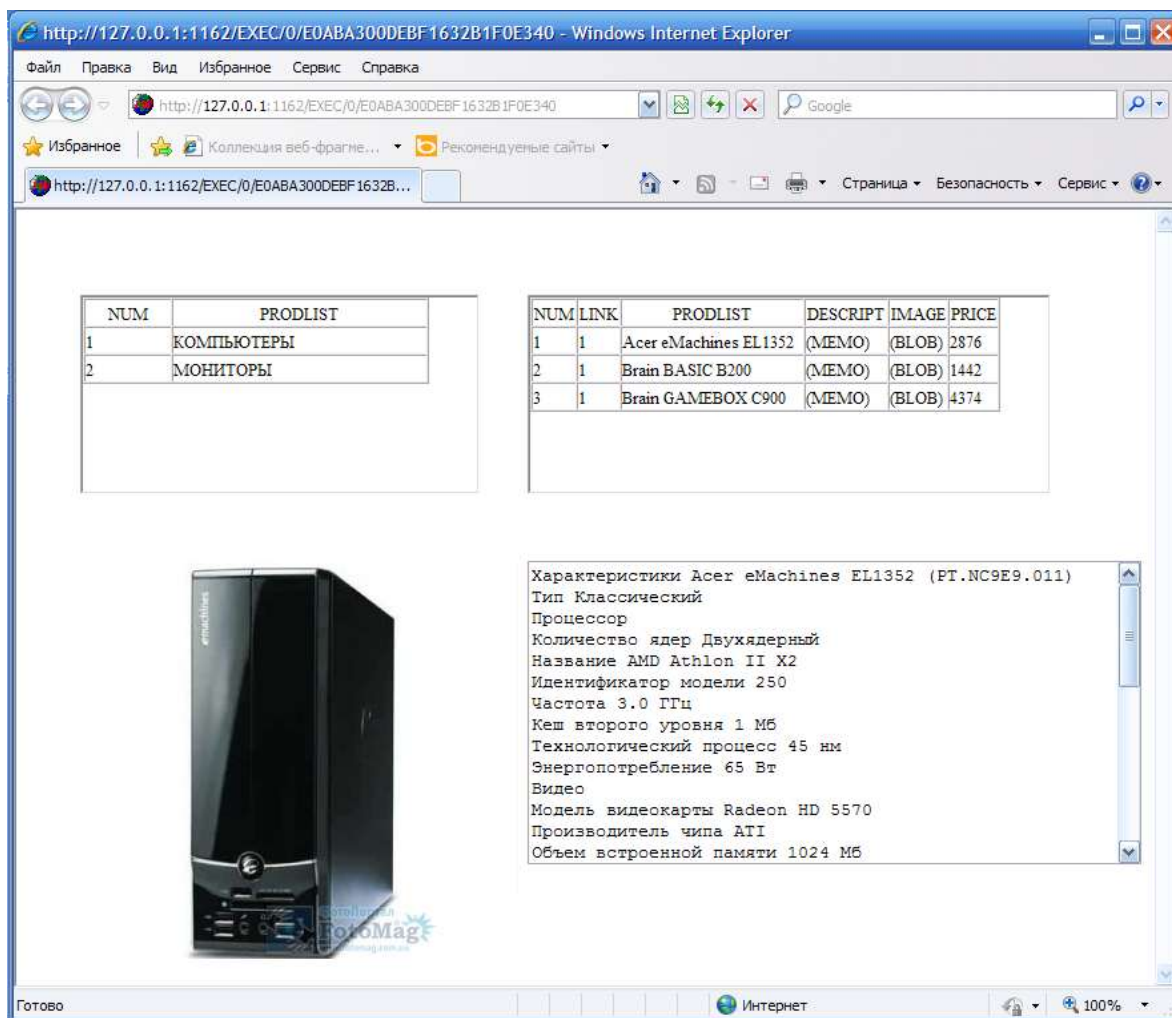


Рис.2.40. Вид web-страницы

18. Закроем приложение и Internet Explorer.

19. Приступим к настройке IWDBGrid1. Установим свойства Lines в состоянии `tlRows`, а свойство `Options/dgIndicator` в состояние `true`. Далее в свойстве `Columns` вызовем редактор, в котором выполним команду `ADD` контекстного меню (рис.2.41). В свойстве редактора `DataField` выберем значение `PRODLIST`, а в свойстве `LinkField` значение `NUM`. Изменим в свойстве `Title/Text` текущее значение на товары, а в свойстве `Title/BGColor` текущее значение например на `clAqua`.

20. Так, как компонент `IWDBGrid1` не является компонентом `DBAWARE` нам придется самостоятельно описать его взаимодействие с набором данных.

Объявим раздел **protected**, в котором продекларируем процедуру `GoIndNoProd` (рис.2.42).

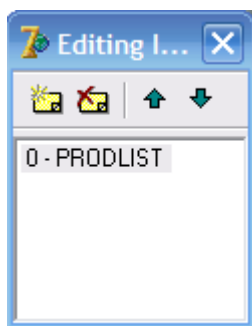


Рис.2.41. Редактор свойств Columns

```

type
  TFormMain = class (TIWAppForm)
    IWDBGrid1: TIWDBGrid;
    IWDBGrid2: TIWDBGrid;
    IWDBMemo1: TIWDBMemo;
    IWDBImage1: TIWDBImage;
    procedure IWDBGrid1Columns0Click(ASender: TObject;
      const AValue: String);
  protected
    {Декларирование процедуры "Переход к записи в таблице PROD_TBL"}
    procedure GoIndNoProd(const AIndNo: integer);
  public

  end;

implementation
{$R *.dfm}

```

Рис.2.42. Декларирование процедуры

21. Тело процедуры опишем в секции **implementation** (рис.2.43).

```

implementation
{$R *.dfm}

uses
  ServerController, DatamoduleUnit, SysUtils;

{Описание процедуры "Переход к записи в таблице PROD_TBL"}
procedure TFormMain.GoIndNoProd(const AIndNo: integer);
begin
  DataModule1.PROD_TBL.Locate('NUM', AIndNo, []);
end;

```

Рис.2.43. Тело процедуры GoIndNoProd

22. Создадим для обработчика события OnClick созданного поля PRODLIST, в котором вызовем созданный выше метод (рис.2.44).

```

procedure TFormMain.IWDBGrid1Columns0Click(ASender: TObject;
  const AValue: String);
begin
  GoIndNoProd(StrToInt(AValue)); // Вызов процедуры GoIndNoProd
end;

```

Рис.2.44. Обработчик события поля PRODLIST

23. В исполняемом **IE** приложении должна обрабатываться связь между главным и подчиненным представлением данных (рис.2.45)

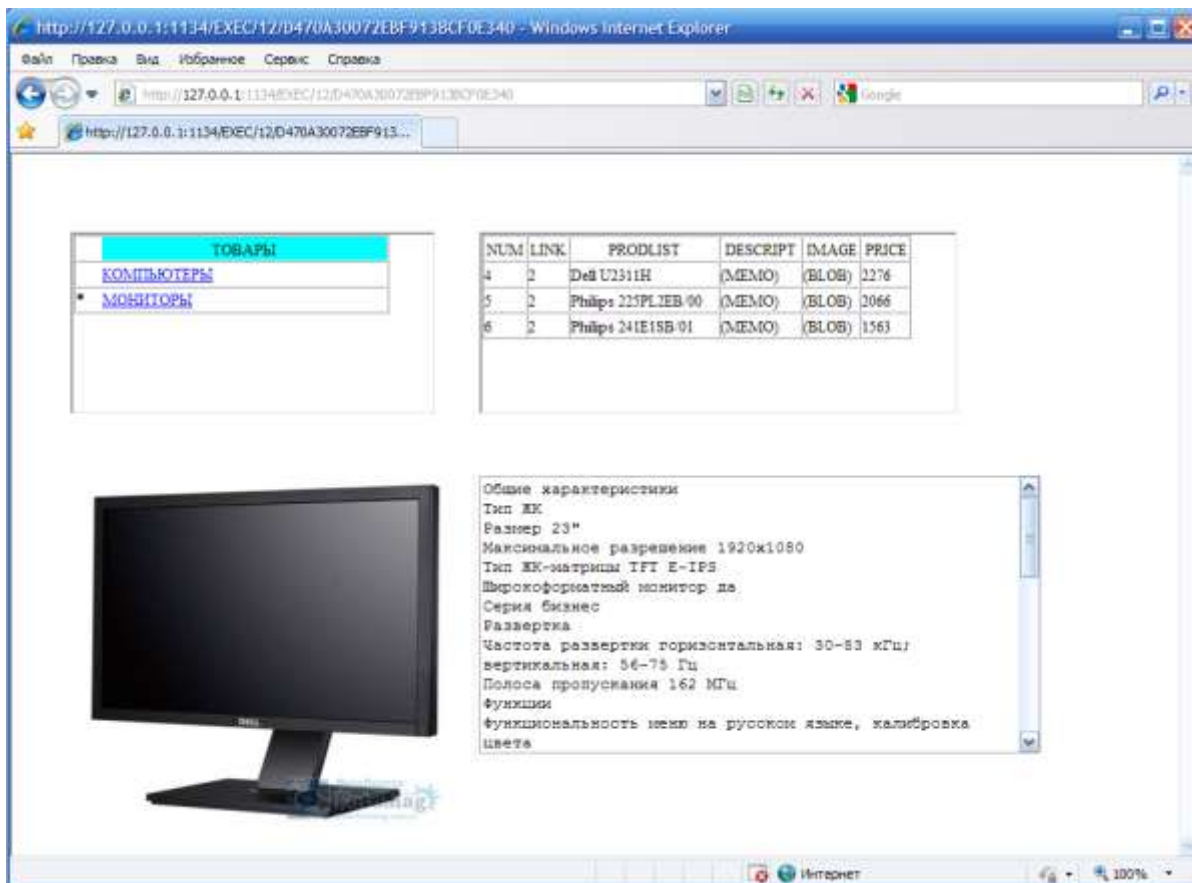


Рис.2.45. Отработка связанных наборов данных

24. Для того чтобы выбранная запись в гриде, отображающем перечень наименований товаров, имела цветовую подсветку необходимо описать метод управления цветом. Для этого необходимо определить в разделе uses использование модуля управления графикой (рис.2.46), а в обработчике события OnRenderCell компонента IWDBGrid1 создать соответствующий метод (рис.2.47).



```

implementation
{$R *.dfm}

uses
  ServerController, DatamoduleUnit, SysUtils, Graphics;

```

Рис.2.46. Описание используемых модулей

```

procedure TFormMain.IWDBGrid1RenderCell(ACell: TIWGridCell; const ARow,
  AColumn: Integer);
begin
  {Цветовое выделение выбранной записи. $00BBE2FD - цвет подсветки}
  if IWDBGrid1.RowIsCurrent then ACell.BGColor := $00BBE2FD;
end;

```

Рис.2.47. Обработчик события OnRenderCell

25. Приступим к настройке управления компонента IWDBGrid2.

Установим свойства компонента IWDBGrid2 аналогично свойствам IWDBGrid1.

Свойства Lines установим в состояние tlRows, а свойство Options/dgIndicator в состояние true. Далее в свойстве Columns вызовем редактор, в котором выполним поочередно две команды ADD контекстного меню (рис.2.48). В свойстве редактора DataField для первого созданного поля выберем значение PRODLIST, а в свойстве LinkField значение NUM. Для второго созданного поля выберем значение DataField равное PRICE. Для поля PRODLIST изменим в свойстве Title/Text текущее значение на НАИМЕНОВАНИЕ, а в свойстве Title/BGColor текущее значение например на clGradientActiveCaption. Аналогично для поля PRICE определим значение Title/Text равное ЦЕНА (грн.) а значение свойства Title/BGColor установим равным \$00C1C1C1. Также установим значение свойства Algnment в состояние taCenter.

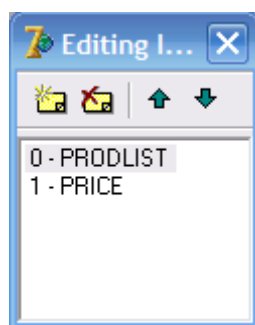


Рис.2.48. Определение отображаемых полей

29. В разделе **protected** объявим защищенный метод GoIndNoDesc (рис.2.49), тело процедуры (рис.2.50, а) и вызов метода опишем в обработчике события OnClick поля PRODLIST (рис.2.50, б).

```
protected
  {Декларирование процедуры "Переход к записи в таблице PROD_TBL"}
  procedure GoIndNoProd(const AIndNo: integer);
  {Декларирование процедуры "Переход к записи в таблице DESC_TBL"}
  procedure GoIndNoDesc(const AIndNo: integer);
public
end;
```

Рис.2.49. Декларирование метода

а) 

```
{Описание процедуры "Переход к записи в таблице DESC_TBL"}
procedure TFormMain.GoIndNoDesc(const AIndNo: integer);
begin
    DataModule1.DESC_TBL.Locate('NUM', AIndNo, []);
end;
```

б)

```
procedure TFormMain.IWDBGrid2Columns0Click(ASender: TObject;
  const AValue: String);
begin
    GoIndNoDesc(StrToInt(AValue)); // Вызов процедуры GoIndNoDesc
end;
```

Рис.2.50. а) процедура GoIndNoDesc, б) обработчик события OnClick поля PRODLIST

30. Создадим метод подсветки выбранного товара. Данный метод опишем в обработчике события OnRenderCell компонента IWDBGrid2 (рис.2.51).

```
procedure TFormMain.IWDBGrid2RenderCell(ACell: TIWGridCell; const ARow,
  AColumn: Integer);
begin
  {Цветовое выделение выбранной записи. $00BBE2FD - цвет подсветки}
  if IWDBGrid2.RowIsCurrent then ACell.BGColor := $00BBE2FD;
end;
```

Рис.2.51 Обработчик события OnRenderCell

31. Скомпилируем приложение и проверим его работу. При выборе товаров в IWDBGrid1 и IWDBGrid2 их представления и вид должны изменяться (рис.2.52).

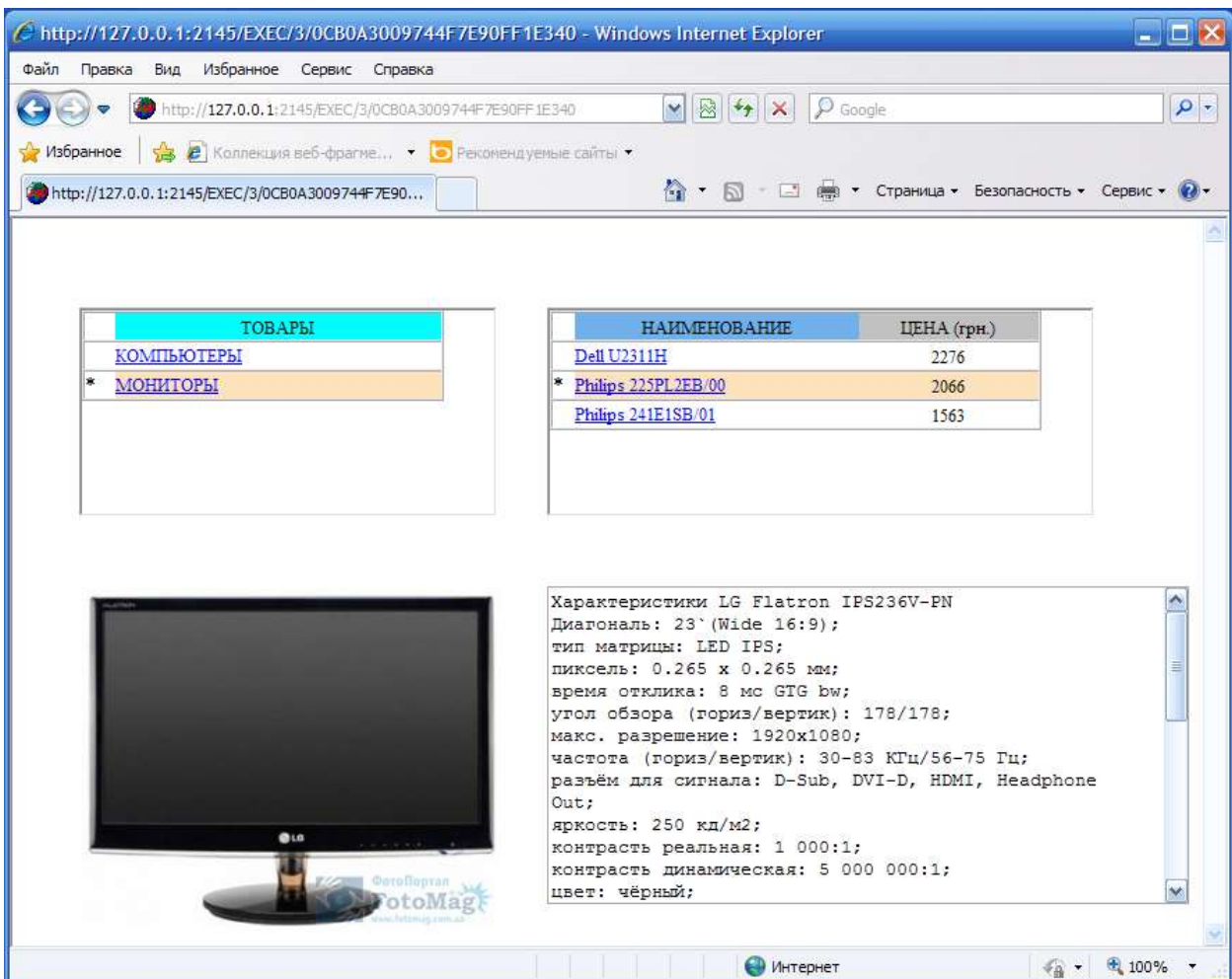


Рис.2.52. Управление выбором товара

32. Сохраним проект и приступим к разработке корзины покупателя.

Установим в модуль данных два компонента ClientDataSet1 и DataSource1 (рис.2.53), которые соответственно переименуем в REQ\_CDS и REQ\_DS (рис.2.54).

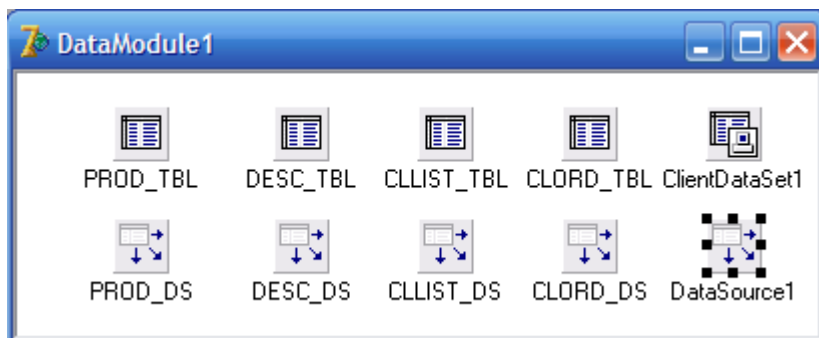


Рис.2.53. Установка компонентов в модуль данных

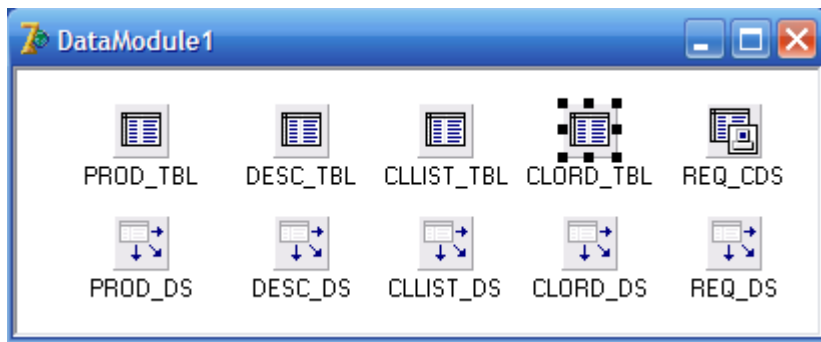


Рис.2.54. Именованние компонентов в модуле данных

Свяжем компонент REQ\_DS с компонентом REQ\_CDS посредством свойства DataSet.

33. Создадим структуру виртуального набора данных для компонента REQ\_CDS, который будет содержать записи покупок. Для создания виртуального набора данных необходимо дважды щелкнуть мышкой по компоненту и в открывшемся редакторе используя контекстное меню выполнить команду New Field (рис.2.55) и создать поля, соответственно рисункам 2.56, 2.57.

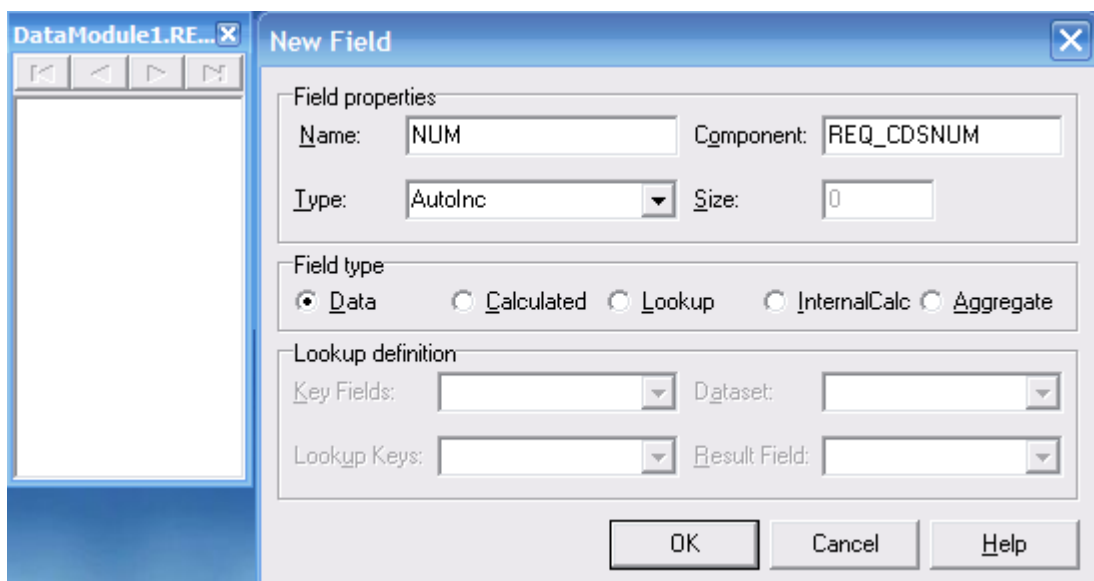


Рис.2.55. Создание поля NUM



Рис.2.56. Создание поля CL\_REQ



Рис.2.57. Создание поля PRICE

34. Выберем в модуле данных компонент REQ\_CDS и создадим для него DataSet выполнив одноименную команду контекстного меню Create DataSet.

35. Теперь продолжим разработку главной формы. Установим в форму компонент IWRectangle, четыре кнопки IWBUTTON, три поля ввода IWEDIT, три метки IWLabel и компонент класса TIWDBGrid (рис.58). Компоненты разместим в форме ниже ранее установленных 4-х компонентов. Определим свойства компонентов согласно таблице 2.8.



Рис.2.58. Размещение компонентов в форме.

Таблица 2.8

Значения свойств компонентов

№ п/п	Класс компонента	Свойство	Значение
1	TIWBUTTON	NAME	SHOWBTN
		CAPTION	КУПИТЬ
2	TIWBUTTON	NAME	GOBTN
		CAPTION	ОТПРАВИТЬ ЗАКАЗ
		VISIBLE	FALSE
3	TIWBUTTON	NAME	ADDBTN
		CAPTION	ДОБАВИТЬ В КОРЗИНУ
4	TIWBUTTON	NAME	DELBTN
		CAPTION	УДАЛИТЬ
5	TIWLABEL	CAPTION	ИМЯ
6	TIWLABEL	CAPTION	ГОРОД
7	TIWLABEL	CAPTION	ТЕЛЕФОН
8	TIWEDIT	NAME	NameEdit
		TEXT	Пусто
		VISIBLE	FALSE
9	TIWEDIT	NAME	SityEdit
		TEXT	Пусто
		VISIBLE	FALSE
10	TIWEDIT	NAME	PhoneEdit
		TEXT	Пусто
		VISIBLE	FALSE

36. Настроим компонент IWDBGrid3, установив свойства DataSource := DataModule1.REQ\_DS, Option/Indicator := true и Lines := tlRows. В редакторе полей Columns добавим соответствующие поля (рис.2.59). Выберем для первого

поля значение DataField := NUM, для второго DataField := CL\_REQ и LinkField := NUM, для третьего DataField := PRICE.

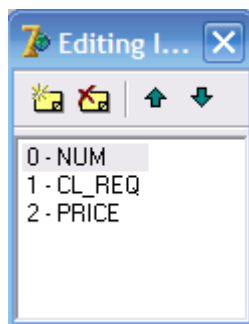


Рис.2.59. Вид редактора полей Columns

37. Русифицируем и оформим заголовки полей, как показано на рисунке 2.60.

№	ТОВАР	ЦЕНА
*		

Рис.2.60. Вид сетки, отображающей корзину заказов

38. Протестируем процедуру GoIndNoREQ (рис.2.61) и опишем ее (рис.2.62).

```
protected
  {Декларирование процедуры "Переход к записи в таблице PROD_TBL"}
  procedure GoIndNoProd(const AIndNo: integer);
  {Декларирование процедуры "Переход к записи в таблице DESC_TBL"}
  procedure GoIndNoDesc(const AIndNo: integer);
  {Декларирование процедуры "Переход к записи в корзине REQ_CDS"}
  procedure GoIndNoREQ(const AIndNo: integer);
public
end;
```

Рис.2.61. Декларирование процедуры GoIndNoREQ

```
{Описание процедуры "Переход к записи в корзине REQ_CDS"}
procedure TFormMain.GoIndNoREQ(const AIndNo: integer);
begin
  DataModule1.REQ_CDS.Locate('NUM', AIndNo, []);
end;
```

Рис.2.62. Описание процедуры GoIndNoREQ

- a) **procedure** TFormMain.IWDBGrid3Columns1Click(ASender: TObject;  
**const** AValue: String);  
**begin**  
 GoIndNoREQ(StrToInt(AValue)); // Вызов процедуры GoIndNoREQ  
**end;**
- б) **procedure** TFormMain.IWDBGrid3RenderCell(ACell: TIWGridCell; **const** ARow,  
 AColumn: Integer);  
**begin**  
 {Цветовое выделение выбранной записи. \$00BBE2FD - цвет подсветки}  
**if** IWDBGrid3.RowIsCurrent **then** ACell.BGColor := \$00BBE2FD;  
**end;**

39. Напишем обработчики событий onClick кнопки ADDBTN “ДОБАВИТЬ В КОРЗИНУ” и кнопки DELBTN “УДАЛИТЬ” (рис.2.63, рис.2.64).

```

procedure TFormMain.ADDBTNClick(Sender: TObject);
begin
  DataModule1.REQ_CDS.Append; // Добавить запись в корзину
  DataModule1.REQ_CDSCL_REQ.Value := DataModule1.DESC_TBLPRODLIST.Value;
  DataModule1.REQ_CDSPRICE.Value := DataModule1.DESC_TBLPRICE.AsInteger;
  DataModule1.REQ_CDS.Post; // Сохранить запись в корзине
end;

```

Рис.2.63. Обработчик события onClick кнопки ADDBTN

```

procedure TFormMain.DELBTNClick(Sender: TObject);
begin
  if DataModule1.REQ_CDS.RecordCount > 0 then // если к-во записей > 0 то:
    DataModule1.REQ_CDS.Delete; // удаляем выбранную запись
end;

```

Рис.2.64. Обработчик события onClick кнопки DELBTN

40. Запустим приложение и проверим его функциональность (рис.2.65).



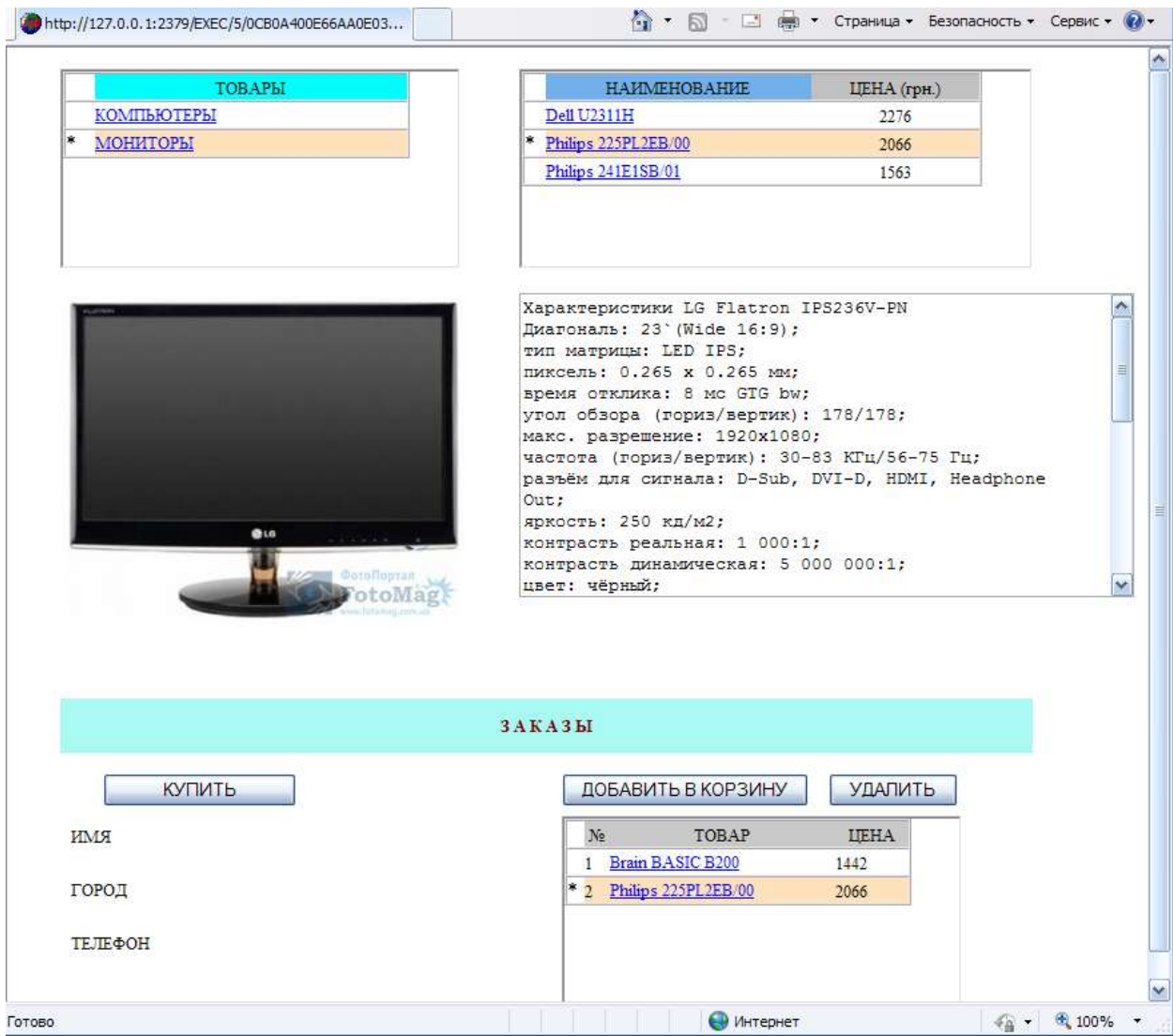


Рис.2.65. Проверка функциональности приложения

41. Напишем обработчик события onClick для кнопки SHOWBTN “КУПИТЬ” (рис.2.66).

```

procedure TFormMain.SHOWBTNClick(Sender: TObject);
begin
  {Установка видимости элементов регистрации покупателя}
  NameEdit.Visible := true;  SityEdit.Visible := true;
  PhoneEdit.Visible := true; GOBTN.Visible := true;
end;

```

Рис.2.66. Обработчик события onClick для кнопки SHOWBTN

42. Транзакции базы данных при покупке товара и формирование пакета заказа напишем в личном методе TransPack (рис.2.67, рис.2.68).

```

protected
  {Декларирование процедуры "Переход к записи в таблице PROD_TBL"}
  procedure GoIndNoProd(const AIndNo: integer);
  {Декларирование процедуры "Переход к записи в таблице DESC_TBL"}
  procedure GoIndNoDesc(const AIndNo: integer);
  {Декларирование процедуры "Переход к записи в корзине REQ_CDS"}
  procedure GoIndNoREQ(const AIndNo: integer);
public
  {Декларирование процедуры "Передача пакета заказа"}
  procedure TransPack;
end;

```

Рис.2.67. Декларирование процедуры TransPack

43. В обработчике события onClick для кнопки GOBTN “ОТПРАВИТЬ ЗАКАЗ” (рис.2.69) опишем всю логику управления процедуры оформления заказа.

44. Теперь можно скомпилировать проект, проверить его функциональность. Выполним полную проверку функциональность системы, для чего запустим из рабочих папок WEB-приложение и приложение Администратор (Admin.exe). При покупке товара должны быть заполнены все поля и в корзине должен быть выбран покупаемый товар. Иначе при выполнении покупки будет выведено соответствующее сообщение-предостережение.

```

{Описание процедуры "Передача пакета заказа"}
procedure TFormMain.TransPack;
var
i: integer;
begin
    {Переоткрытие наборов данных}
    DataModule1.CLLIST_TBL.Close; DataModule1.CLORD_TBL.Close; // Закрытие таблиц
    DataModule1.CLLIST_TBL.Open; DataModule1.CLORD_TBL.Open; // Открытие таблиц
    {Регистрация покупателя}
    DataModule1.CLLIST_TBL.Insert; // Добавляем запись в таблицу CLLIST_TBL
    DataModule1.CLLIST_TBLCL_NAME.Value := NameEdit.Text; // Передаем ИМЯ
    DataModule1.CLLIST_TBLLOCATION.Value := SityEdit.Text; // Передаем ГОРОД
    DataModule1.CLLIST_TBLPHONE.Value := PhoneEdit.Text; // Передаем ТЕЛЕФОН
    DataModule1.CLLIST_TBL.Post; // Сохраняем запись в таблице CLLIST_TBL

    {Передача заказа}
    DataModule1.REQ_CDS.First; // Переходим к 1-й записи корзины
    for i := 1 to DataModule1.REQ_CDS.RecordCount do // сначала до конца записей
    begin
        DataModule1.CLORD_TBL.Insert; // Добавляем запись в таблицу CLORD_TBL
        DataModule1.CLORD_TBLCL_REQ.Value := DataModule1.REQ_CDSCL_REQ.Value;
        DataModule1.CLORD_TBLPRICE.Value := DataModule1.REQ_CDSPRICE.Value;
        DataModule1.CLORD_TBL.Post; //Сохраняем запись в таблице CLORD_TBL
        DataModule1.REQ_CDS.Next; // Переходим к следующей записи в корзине
    end;
end;

```

Рис.2.68. Реализация метода TransPack

```

procedure TFormMain.GOBTNClick(Sender: TObject);
begin
    {Если заполнены все поля и заполнена корзина то выполняем}
    if (NameEdit.Text <> '') and (SityEdit.Text <> '') and
        (PhoneEdit.Text <> '') and (DataModule1.REQ_CDS.RecordCount > 0) then
    begin
        TransPack; // Передача пакета заказа

        WebApplication.ShowMessage('СПОСИБО ЗА ПОКУПКУ'); // Вывод сообщения
        NameEdit.Visible := false; // Не видимость поля
        SityEdit.Visible := false; // Не видимость поля
        PhoneEdit.Visible := false; // Не видимость поля
        NameEdit.Text := ''; // Очистка содержимого поля
        SityEdit.Text := ''; // Очистка содержимого поля
        PhoneEdit.Text := ''; // Очистка содержимого поля
        GOBTN.Visible := false; // Не видимость кнопки
        DataModule1.REQ_CDS.Close; // Закрытие DataSet
        DataModule1.REQ_CDS.CreateDataSet; // Очистка DataSet (Очистка корзины)
        DataModule1.REQ_CDS.Open; // Открытие DataSet
    end else // в противном случае
    begin
        if DataModule1.REQ_CDS.RecordCount <= 0 then
            WebApplication.ShowMessage('НЕ ВЫБРАН ТОВАР') // Вывод сообщения
        else
            WebApplication.ShowMessage('ЗАПОЛНИТЕ ВСЕ ПОЛЯ'); // Вывод сообщения
        end;
        DataModule1.REQ_CDS.Open; // Открытие DataSet
    end;
end;

```

Рис.2.69 Обработчик события onClick кнопки GOBTN “ОТПРАВИТЬ ЗАКАЗ”

45. Сделайте несколько покупок от имени разных покупателей, а затем выполните команду “Обновить” в приложении Администратор. На странице заявки Вы увидите заявки на покупки, сделанные покупателями (рис.2.70, рис.2.71).

The screenshot shows a web browser window with the following content:

- Navigation:** Top bar with "Страница", "Безопасность", and "Сервис" menus.
- Left Panel:** A sidebar with a "ТОВАРЫ" (Goods) section containing links for "КОМПЬЮТЕРЫ" (Computers) and "МОНИТОРЫ" (Monitors).
- Main Product List:** A table with columns "НАИМЕНОВАНИЕ" (Name) and "ЦЕНА (грн.)" (Price in UAH).
 

НАИМЕНОВАНИЕ	ЦЕНА (грн.)
<a href="#">Dell U2311H</a>	2276
<a href="#">Philips 225PL2EB/00</a>	2066
* <a href="#">Philips 241E1SB/01</a>	1563
- Product Image:** A photograph of a Philips 241E1SB/01 monitor.
- Product Characteristics:** A text block titled "Характеристики Philips 241E1SB/01" listing technical specifications:
  - Тип ЖК-панели TFT-ЖКД
  - Размер панели 23,6 дюйма / 59,95 см
  - формат изображения 16:9
  - Шаг пикселей 0,272 x 0,272 мм
  - Угол просмотра 176° (Г) / 170° (В), @ C/R > 5
  - Яркость 250 кд/м²
  - SmartContrast 25 000:1
  - Коефф. контрастности (тип.) 1000:1
  - Время отклика (типич.) 5 мс
  - Оптимальное разрешение 1920 x 1080 при 60 Гц
  - Цвета дисплея 16,7 М
  - Частота сканирования 30–83 кГц (В) / 56–75 Гц (Ш)
- Order Section:** A section titled "ЗАКАЗЫ" (Orders) containing:
  - Buttons: "КУПИТЬ" (Buy), "ДОБАВИТЬ В КОРЗИНУ" (Add to cart), "УДАЛИТЬ" (Delete).
  - Order Form: Fields for "ИМЯ" (Name) with value "АРТЕМ", "ГОРОД" (City) with value "ЛУГАНСК", and "ТЕЛЕФОН" (Phone) with value "55-555-55". A button "ОТПРАВИТЬ ЗАКАЗ" (Send order) is below.
  - Shopping Cart: A table with columns "№" (No.), "ТОВАР" (Goods), and "ЦЕНА (грн.)" (Price in UAH).
 

№	ТОВАР	ЦЕНА (грн.)
1	<a href="#">Brain GAMEBOX C900</a>	4374
* 2	<a href="#">Philips 241E1SB/01</a>	1563

Рис.2.70. Выполнение покупки товара

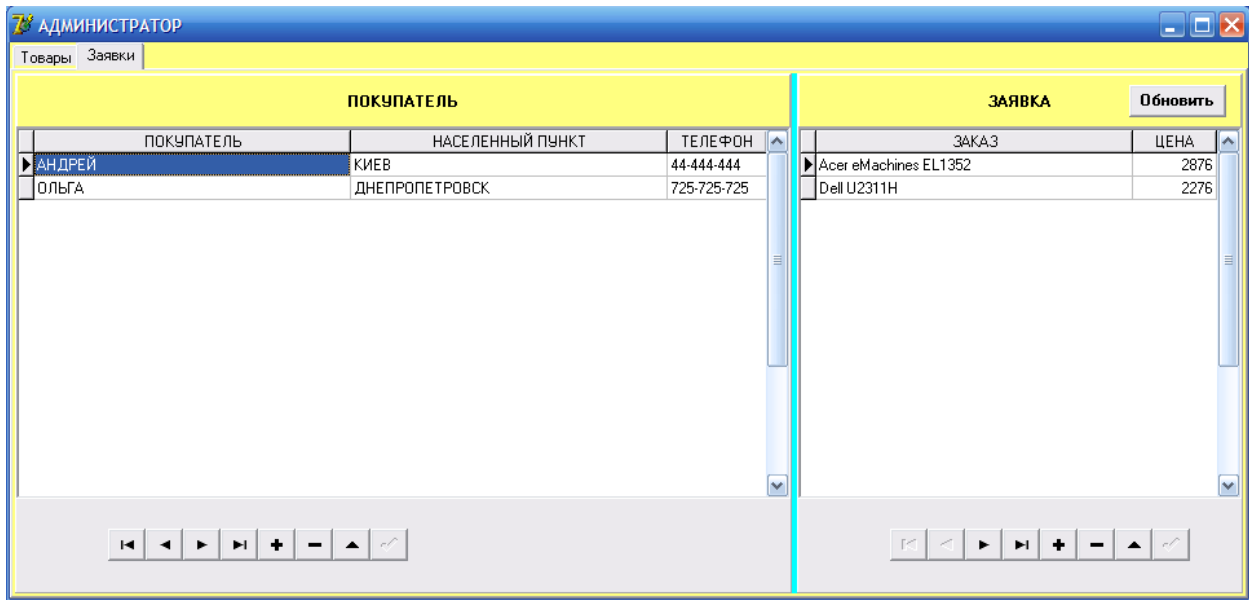


Рис.2.71. Поступление заявок

46. В завершении работы определим реальный IP адрес и порт, который будет идентифицировать наш сайт. Для этого в свойстве BoundIP модуля сервера приложения IWServerController укажем IP адрес **127.0.0.1**, в свойстве порт - значение порта равное **1001**, а в свойстве StartCmd значение **start**.

Теперь запустим сервер приложения и отдельно запустим Интернет эксплорер Windows, в командном поле которого введем адрес **http://127.0.0.1:1001/start** (рис.2.72) и выполним команду **перейти**.

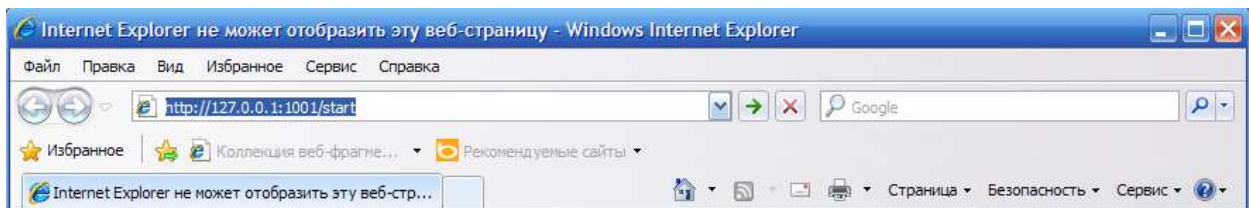


Рис.2.72. Ввод адреса в командную строку

После выполнения команды в окне IE откроется созданный нами сайт.

**Примечание.** Вследствие того, что в Delphi 7 установлена **trial** версия IntraWeb, другие IP адреса использовать нельзя. Для полноценной разработки IntraWeb приложений необходимо приобрести и установить коммерческую версию IntraWeb библиотеки.

## ЛИТЕРАТУРА

1. А. Архангельский. Программирование в Delphi 7. – К.: Каравела, 2008. – 832 с.
2. С. Бобровский., Delphi 7. Учебный курс. – Ростов на Дону: Феникс, 2008. – 452 с.
3. М. Фленов. Библия Delphi (3-е издание). – СПб.: – ВHV, 2010. – 740 с.
4. А. Хомоненко, В. Гофман, Е. Мещеряков, В. Никифоров. Delphi 7. Наиболее полное руководство. – СПб.:БХВ-Петербург, 2010. – 616 с.
5. IntraWeb Documentation. Manual and Tutorials. Copyright© 2003–2006, Nevrona Designs.
6. С. Бобровский. Технология IntraWeb. Разработка приложений для бизнеса. Учебный курс. – Харьков: Фолио, 2011. – 322 с.
7. IntraWeb – VCL для Web, <http://www.atozed.com/intraweb/index.ru.aspx>.

## СОДЕРЖАНИЕ

	стр.
1. ТЕХНОЛОГИЯ INTRAWEB.....	3
1.1. Общие положения.....	3
1.2. Структура IntraWeb-приложения.....	4
1.3. Управление формами в IntraWeb.....	5
2. ПРИМЕР СОЗДАНИЯ INTRAWEB ПРИЛОЖЕНИЯ.....	7
2.1. Схема разрабатываемого приложения.....	7
2.2. Схема базы данных.....	8
2.3. Порядок разработки системы.....	8
2.3.1. Подготовительная стадия.....	9
2.3.2. Создание базы данных.....	9
2.3.3. Разработка СУБД «Администратор».....	12
2.3.4. Создание IntraWeb-приложения.....	23
ЛИТЕРАТУРА.....	46

Навчальне видання

Швачич Геннадій Григорович  
Овсянніков Олександр Васильович  
Петречук Ліна Миколаївна

# **КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ В ДІЛОВОДСТВІ**

## **Розділ 2. IntraWeb технологія**

Навчальний посібник

Тем. план 2012, поз.240

Підписано до друку 26.06.2012. Формат 60x84 1/16. Папір друк. Друк плоский.  
Облік.-вид. арк. 2,82. Умов. друк. арк. 2,79. Тираж 100 пр. Замовлення № 100.

Національна металургійна академія України  
49600, м. Дніпропетровськ-5, пр. Гагаріна, 4

---

Редакційно-видавничий відділ НМетАУ