

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Белорусский национальный технический университет

---

Кафедра «Системы автоматизированного проектирования»

БАЗЫ ЗНАНИЙ И ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ  
В САПР

Лабораторный практикум  
для студентов специальности 1-40 01 02-01 «Информационные  
системы и технологии в проектировании и производстве»

В 3 частях

Часть 1

Минск  
БНТУ  
2012

УДК 004.65+658.512.22-027.44(076.5)

ББК 32.81я7

Б17

С о с т а в и т е л и :

*В. А. Кочуров, Ю. О. Герман*

Р е ц е н з е н т ы :

доцент кафедры ПОВТ БНТУ, канд. техн. наук *В. В. Бугай*;

доцент кафедры ИТАС БГУИР, канд. техн. наук *О. В. Герман*

**Базы знаний и поддержка принятия решений в САПР : лабораторный практикум для студентов специальности 1-40 01 02-01 «Информационные системы и технологии в проектировании и производстве : в 3 ч. / сост.: В. А. Кочуров, Ю. О. Герман. – Минск : БНТУ, 2012. – Ч. 1. – 2012. – 27 с.**

ISBN 978-985-525-907-8 (Ч. 1).

Материал предназначен для использования в качестве методических указаний при изучении дисциплины «Базы знаний и поддержка принятия решений в САПР». Издание содержит лабораторные работы, которые проведут Вас шаг за шагом через процесс подключения и использования базы данных на платформе Microsoft Visual Studio 2008–2010 и поможет:

- создать и настроить набор данных в приложении на основе объектов базы данных;
- создать элементы управления с привязкой к данным;
- создать и использовать хранимые процедуры;
- создать запросы LINQ и т. д.

УДК 004.65+658.512.22-027.44(076.5)

ББК 32.81я7

ISBN 978-985-525-907-8 (Ч. 1)

ISBN 978-985-525-909-2

© Белорусский национальный  
технический университет, 2012

## СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1. ПОДКЛЮЧЕНИЕ К ДАННЫМ В БАЗЕ ДАННЫХ.....	4
ЛАБОРАТОРНАЯ РАБОТА № 2. СОЗДАНИЕ НАБОРА ДАННЫХ С ПОМОЩЬЮ КОНСТРУКТОРА НАБОРОВ ДАННЫХ.....	10
ЛАБОРАТОРНАЯ РАБОТА № 3. СОЗДАНИЕ ОТНОШЕНИЙ МЕЖДУ ТАБЛИЦАМИ ДАННЫХ.....	15
ЛАБОРАТОРНАЯ РАБОТА № 4. ОТОБРАЖЕНИЕ СВЯЗАННЫХ ДАННЫХ НА ФОРМЕ В ПРИЛОЖЕНИИ WINDOWS.....	24

## ЛАБОРАТОРНАЯ РАБОТА № 1

### ПОДКЛЮЧЕНИЕ К ДАННЫМ В БАЗЕ ДАННЫХ

В данной лабораторной работе рассматриваются следующие задачи:

- создание нового проекта Приложение Windows;
- Создание и настройка набора данных в приложении на основе объектов базы данных при помощи мастера настройки источника данных.

#### Создание проекта

Для создания нового проекта:

1. В меню Файл создайте новый проект.
2. Выберите Приложение Windows и нажмите ОК.
3. Проект создается и добавляется в обозреватель решений.

#### Подключение к базе данных "Borei"

Можно подключиться к базе данных Borei в версии SQL Server или в версии Microsot Access. Каждый процесс описан последовательно в следующих двух процедурах.

#### Подключение к базе данных "Borei" в SQL Server

1. В меню *Данные* выберите команду *Показать источники данных* или команду *Добавить новый источник данных*, чтобы запустить Мастер настройки источника данных (рис. 1).
2. Щелкните *Далее* на странице приветствия мастера настройки источника данных.
3. На странице Выбор типа источника данных выберите *База данных* (рис. 2).
4. Может потребоваться создать новое или выбрать существующее подключение к источнику данных. Если подключение к образцу базы данных "Борей" доступно в раскрывающемся списке, то выберите его, либо выберите элемент *Новое соединение* для создания нового соединения с данными (рис. 3).

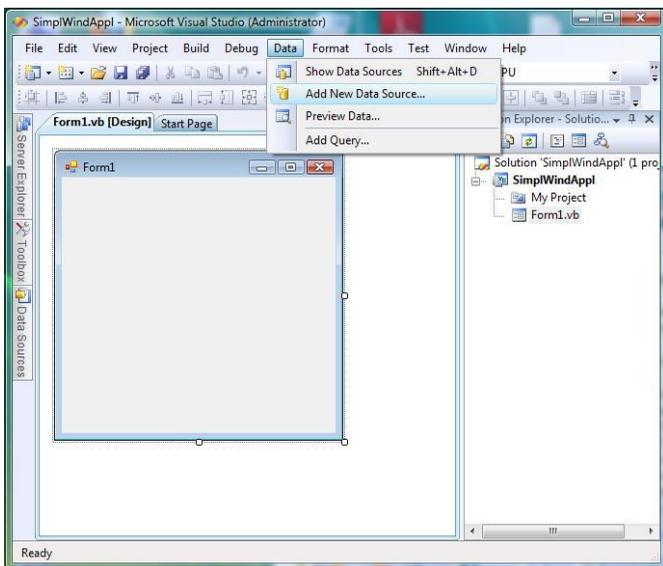


Рис. 1. Добавление нового источника данных

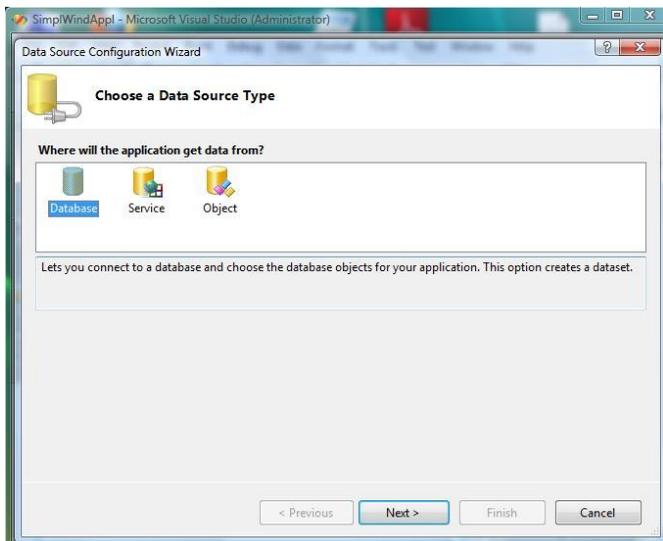


Рис. 2. Выбор типа источника

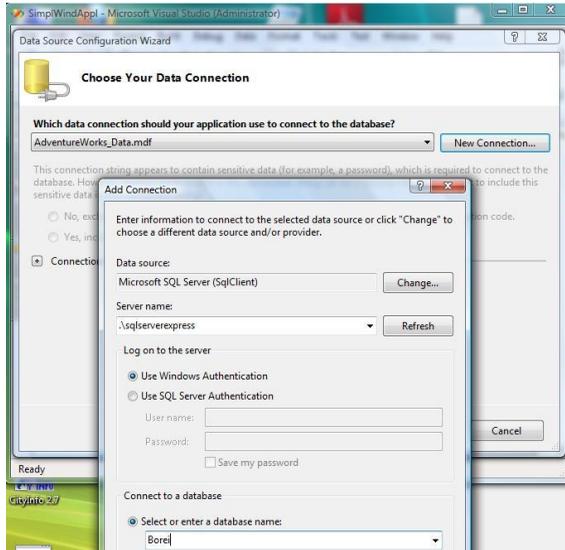


Рис. 3. Выбор подключения к источнику данных

Если откроется окно *Добавить подключение*, а *Источник данных* не имеет значение *Microsoft SQL Server*, нажмите кнопку *Изменить*, чтобы открыть диалоговое окно *Выбрать/изменить источник данных* (рис. 4).

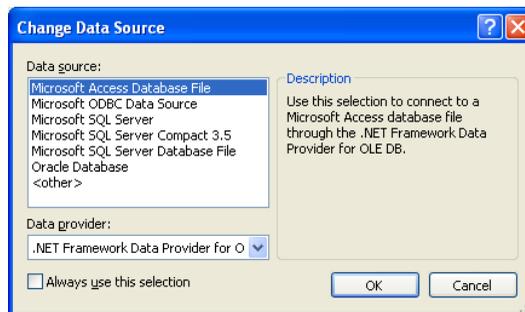


Рис. 4. Окно изменения источника данных

5. В диалоговом окне **Выбор источника** данных выберите Microsoft SQL Server и нажмите кнопку ОК.

Выберите имя сервера из раскрывающегося списка или введите имя сервера, в котором находится нужная база данных.

6. В зависимости от требований базы данных или приложения выберите элемент **Проверка подлинности Windows** либо используйте указанные имя пользователя и пароль для входа на SQL Server (Проверка подлинности SQL Server).

7. Выберите из раскрывающегося списка базу данных, к которой необходимо подключиться и нажмите кнопку ОК.

8. Щелкните **Далее** на странице **Сохранить строку подключения в файле конфигурации приложения**.

9. Разверните узел **Таблицы** на странице **Выбор объектов базы данных** (рис. 5).

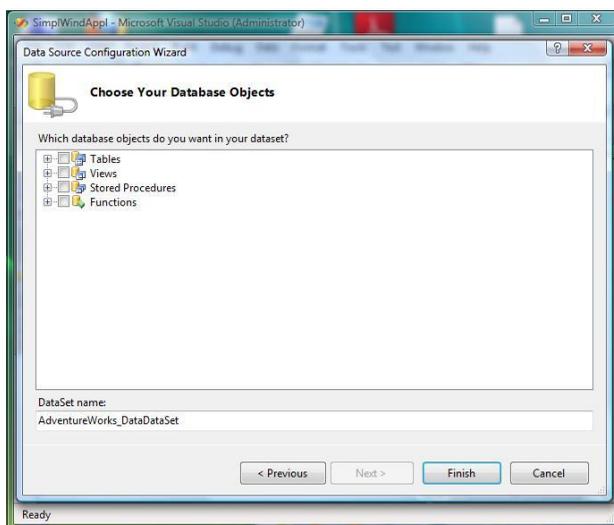


Рис. 5. Окно выбора объектов базы данных

10. Делаем выбор желаемых объектов, например, таблицы **Suppliers u Goods** (рис. 6).

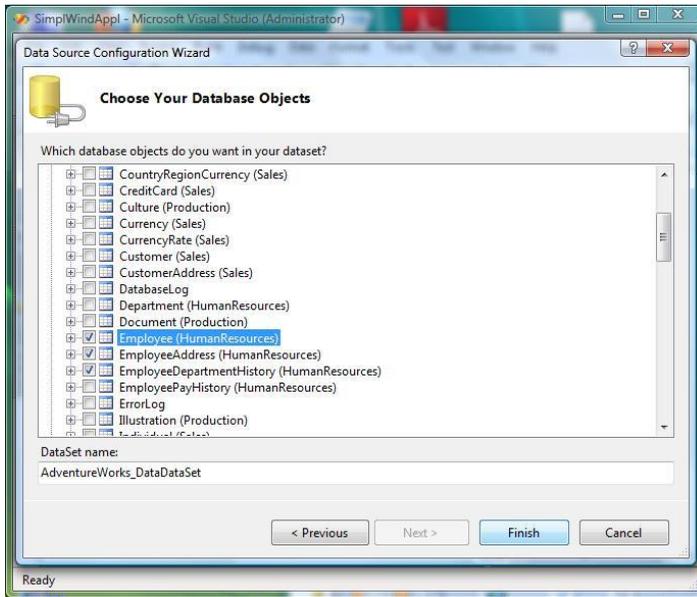


Рис. 6. Выбор объектов в окне источников данных

11. Завершите работу мастера, нажав кнопку **Готово**, и типизированный набор данных добавится в проект (рис. 7).

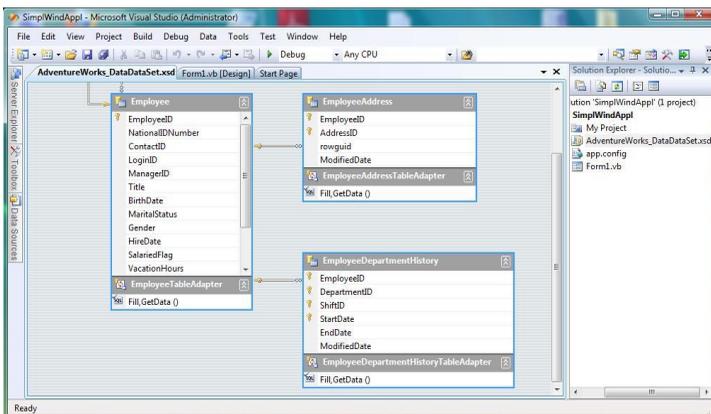


Рис. 7. Созданный набор данных

BoreiDataSet добавляется в проект, и таблицы Suppliers (Поставщики) и Goods (Товары) отображаются в окне **Источники данных**.

### **Подключение к Microsoft Access-версии базы данных Borei**

1. В диалоговом окне **Выбор источника данных** выберите элемент **Файл базы данных Microsoft Access** и щелкните ОК. Если открылось диалоговое окно **Добавление соединения и Источник данных** не является **Файлом базы данных Microsoft Access**, то щелкните **Изменить** для открытия диалогового окна **Выбор/изменение источника данных**.

2. Введите путь к файлу базы данных Borei.mdb или нажмите кнопку **Обзор**, чтобы найти файл базы данных Borei.mdb.

3. Нажмите кнопку ОК.

4. Щелкните **Далее** на странице **Сохранить строку подключения в файле конфигурации приложения**.

5. Разверните узел **Таблицы** на странице **Выбор объектов базы данных**.

6. Выберите таблицы Suppliers и Goods и нажмите **Готово**.

7. BoreiDataSet добавляется в проект, а таблицы Suppliers и Goods отображаются в окне **Источники данных**.

8. Теперь можно создавать элементы управления с привязкой к данным, перетаскивая элементы из окна **Источники данных** на форму.

### **Добавление элементов управления на форму**

Можно создавать элементы управления с привязкой к данным, перетаскивая элементы из окна **Источники данных** на форму.

Для создания элементов управления с привязкой к данным на форме перетащите главный узел Customers из окна **Источники данных** на форму.

Элементы управления с привязкой к данным *BoreiDataSet*, *CustomersTableAdapter*, **BindingSource** и **BindingNavigator** появляются в области компонентов на форме вместе с панелью инструментов (**BindingNavigator**) для управления записями.

### **Выполнение приложения**

Для выполнения приложения нажмите клавишу F5.

## ЛАБОРАТОРНАЯ РАБОТА № 2

### СОЗДАНИЕ НАБОРА ДАННЫХ С ПОМОЩЬЮ КОНСТРУКТОРА НАБОРОВ ДАННЫХ

В данной лабораторной работе рассматриваются следующие задачи:

- создание нового проекта Приложение Windows;
- добавление в проект пустого элемента **Набор данных**;
- создание и настройка источника данных в приложении путем создания набора данных с помощью **Конструктора наборов данных**;
- создание подключения к базе данных "Борей" в **Обозревателе серверов**;
- создание таблиц при помощи адаптеров таблиц в наборе данных на основе таблиц базы данных.

Для выполнения этого руководства потребуется доступ к учебной базе данных "Борей" (версия для SQL Server или Access).

#### Создание проекта приложения Windows

Для создания проекта приложения Windows см. Лабораторную работу № 1.

#### Добавление нового набора данных в приложение

1. В меню **Проект** выберите команду **Добавить новый элемент** (Add New Item). Открывается диалоговое окно **Добавление нового элемента**.

2. В поле **Шаблоны** диалогового окна **Добавление нового элемента** выберите **Набор данных** (DataSet).

3. Назовите набор данных *BoreiDataset* и нажмите кнопку **Добавить**.

Visual Studio добавит файл с именем *BoreiDataset.xsd* в проект и откроет его в **Конструкторе наборов данных**.

#### Создание подключения к данным в обозревателе серверов

1. В меню **Вид** выберите команду **Обозреватель серверов**.

2. В **Обозревателе серверов** нажмите на кнопку **Подключиться к базе данных**.

3. Создайте соединение с учебной базой данных "Борей".

*Для создания подключения к базе данных SQL Express*

1. В меню **Данные** щелкните **Показать источники данных**.
2. В окне **Источники данных** выберите **Добавить новый источник данных**.
3. На странице **Выбор типа источника данных** выберите **База данных** и нажмите кнопку **Далее**.
4. На странице **Выбор модели базы данных** выберите **Набор данных** и нажмите кнопку **Далее**.
5. На странице **Выбор подключения базы данных** выберите **Новое подключение** для создания нового подключения к данным.

*Примечание.* Если параметру **Источник данных** присвоено значение, отличное от **Файл базы данных Microsoft SQL Server**, выберите **Изменить** для вызова диалогового окна **Выбрать/Сменить источник данных**

6. В диалоговом окне **Выбор источника данных** выберите **Файл базы данных Microsoft SQL Server**. Нажмите **Продолжить**. Поставщиком по умолчанию является **Поставщик данных .NET Framework для SQL Server**.

7. Введите путь к файлу базы данных, к которой требуется доступ, или нажмите кнопку **Обзор**, чтобы найти файл базы данных.

*Примечание.* В данном пошаговом руководстве выберите файл *Борей.mdf*, использованный в *Лабораторной работе 1*.

8. Выберите режим проверки подлинности Windows или введите учетные данные, необходимые для вашей базы данных.

9. Нажмите кнопку **ОК**, а затем кнопку **Далее**.

*Примечание.* Файлы локальной базы данных могут быть включены в проект в виде файла в решении. При создании подключений к файлам локальной базы данных можно выбрать между созданием копии базы данных в проекте или подключением к существующему файлу базы данных в текущем местоположении.

10. Нажмите кнопку **Да** для копирования файла базы данных в проект.

11. Щелкните **Далее** на странице **Сохранить строку подключения в файле конфигурации приложения**.

## Создание таблиц в наборе данных

В этом разделе будет объяснено, как добавлять таблицы в набор данных. Для создания таблицы *Suppliers*:

1. Разверните узел соединения с данными, созданный в **Обозревателе серверов**, а затем узел **Таблицы**.
2. Перетащите таблицу Suppliers из **Обозревателя серверов** в **Конструктор наборов данных**.

Таблица *Suppliers* и *SuppliersTableAdapter* добавляются в наборе данных.

Для создания таблицы *Goods* перетащите таблицу *Goods* из **Обозревателя серверов** в **Конструктор наборов данных**. К набору данных добавляются таблица *Goods*, *GoodsTableAdapter* и отношения данных между таблицами Suppliers и Goods.

## Создание элементов управления для отображения данных из таблицы Suppliers

Можно создавать элементы управления с привязкой к данным, перетаскивая элементы из окна Источники данных на форму Windows Forms.

Чтобы создать элементы управления для отображения данных поставщиков (родительские записи):

1. Выберите таблицу Suppliers в окне **Источники данных** и щелкните стрелку раскрывающегося списка (рис. 8).
2. Выберите **Сведения** из списка управления таблицы Suppliers.
3. Перетащите главный узел Suppliers из окна **Источники данных** на форму Form1.

Привязанные к данным элементы управления с подписями описания отображаются на форме. Следующие компоненты отображаются в области компонентов:

- BoreiDataSet – типизированный набор данных, содержащий таблицы Suppliers и Goods.
- SuppliersBindingSource – элемент управления привязки к данным, который привязывается к таблице данных Suppliers в BoreiDataSet.

- SuppliersBindingNavigator – панель навигатора, который используется для обхода записей в таблице Suppliers.
- SuppliersTableAdapter. TableAdapter, который обеспечивает связь между базой данных и BoreiDataSet.
- TableAdapterManager. Компонент TableAdapterManager, который используется для управления порядком вставки, обновления и удаления для всех компонентов TableAdapter в наборе данных.

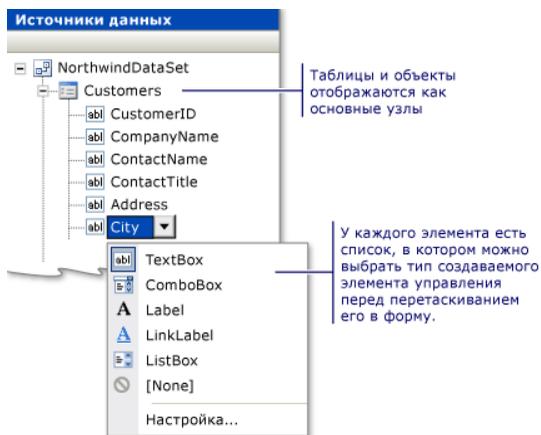


Рис. 8. Выбор типа элемента управления

### Создание элементов управления для отображения данных из таблицы Goods

Чтобы создать элементы управления для отображения товаров для каждого поставщика (дочерних записей):

- в окне Источники данных разверните узел Suppliers, выберите последний столбец в таблице Suppliers (это разворачиваемый узел Goods ниже столбца Fax) и перетащите его на нижнюю часть Form1. (Этот узел в иерархии таблицы Suppliers представляет связанные товары для поставщика, в отличие от основного узла Goods, который представляет все записи в таблице Orders, а не заказы отдельных клиентов.)

DataGridView добавляется на форму, а новые компоненты BindingSource (GoodsBindingSource) и объект TableAdapter (GoodsTableAdapter) добавляются в область компонентов.

***Примечание.** Откройте окно Свойства и выберите компонент GoodsBindingSource. Изучите свойства DataSource и DataMember, чтобы увидеть настройку привязки для отображения связанных записей. DataSource настроен на SuppliersBindingSource (BindingSource родительской таблицы), в отличие от таблицы Suppliers. Свойству DataMember присвоено значение FK\_Goods\_CodeSuppl, являющееся именем объекта DataRelation, связывающего таблицы друг с другом.*

### **Тестирование приложения**

1. Нажмите клавишу F5.
2. Выберите различных клиентов, чтобы проверить правильность отображения заказов в таблице на форме.
3. Измените одну или несколько записей.
4. Нажмите кнопку Сохранить (значок в виде дискеты).
5. Убедитесь, что изменения были сохранены в базе данных.

## **ЛАБОРАТОРНАЯ РАБОТА № 3**

### **СОЗДАНИЕ ОТНОШЕНИЙ МЕЖДУ ТАБЛИЦАМИ ДАННЫХ**

В этой лабораторной работе, в частности, рассматриваются следующие задачи:

- создание нового приложения Windows;
- добавление нового набора данных в приложение;
- добавление двух новых таблиц данных в набор данных;
- добавление столбцов в таблицы данных;
- задание первичного ключа для таблиц;
- связывание таблиц;

#### **Создание нового приложения Windows**

Чтобы создать проект приложения см. Лабораторную работу № 1.

#### **Добавление нового набора данных в приложение**

1. В меню **Проект** выберите **Добавить новый элемент**.  
Откроется диалоговое окно *"Добавление нового элемента"*.

2. В области **Шаблоны** выберите **Набор данных**.
3. Щелкните **Добавить**. Visual Studio добавит файл с именем *DataSet1.xsd* в проект и откроет его в **Конструкторе наборов данных**.

#### **Добавление двух новых таблиц данных в набор данных**

1. Перетащите DataTable из вкладки **Набор данных Панели элементов** в **Конструктор наборов данных**. К набору данных будет добавлена таблица с именем DataTable1.
2. Щелкните заголовок DataTable1 и переименуйте его в Suppliers.
3. Перетащите вторую DataTable из вкладки **Набор данных Панели элементов** в **Конструктор наборов данных**. К набору данных будет добавлена таблица с именем DataTable1.
4. Щелкните заголовок DataTable1 и переименуйте его в Goods.

#### **Добавление столбцов к пользовательским наборам данных**

##### **Добавление столбцов в таблицу "Suppliers"**

1. Щелкните правой кнопкой мыши таблицу *Suppliers*.
2. В открывшемся контекстном меню выберите **Добавить**, а затем щелкните **Столбец**.
3. Задайте имя столбца – *CodeSuppliers*.
4. В окне Свойства назначьте свойству *Data Type* значение *Int16*.
5. Повторите эту процедуру, чтобы добавить следующие столбцы:

<b>Столбец</b>	<b>Свойство Data Type</b>
----------------	---------------------------

Title	String
AddressTo	String
Post	String
Address	String
City	String
IIndex	String
Country	String
Telephone	String
Fax	String

## Добавление столбцов в таблицу "Goods"

1. Щелкните таблицу *Goods* правой кнопкой мыши.
2. В открывшемся контекстном меню выберите **Добавить**, а затем щелкните **Столбец**.
3. Задайте имя столбца – CodeGoods.
4. В окне Свойства задайте для свойства *DataType* значение *Int16*.
5. Повторите эту процедуру, чтобы добавить следующие столбцы:

Столбец	Свойство DataType
Mark	String
CodeSuppliers	Int16
Supplier	String
<b>CodeType</b>	Int16
<b>Category</b>	String
Price	Int16
InWarehouse	Int16
Expected	Int16
MinimalStock	Int16
DeliveriesStopped	String

## Задание первичного ключа для таблицы Suppliers

Уникальный идентифицирующий столбец для таблицы *Suppliers* – CodeSuppliers.

Для задания первичного ключа таблицы *Suppliers*:

1. Выделите щелчком таблицу *Suppliers*.
2. Щелкните правой кнопкой мыши столбец *CodeSuppliers* и выберите **Задать первичный ключ** в контекстном меню.

## Для задания первичного ключа таблицы Goods:

1. Выделите щелчком таблицу *Goods*.
2. Щелкните правой кнопкой мыши столбец *CodeGoods* и выберите **Задать первичный ключ** в контекстном меню.

## Добавление элементов управления к форме

Теперь создадим элементы управления с привязкой к данным, перетаскивая элементы из окна *Источники данных* на форму.

Для создания элементов управления с привязкой к данным на форме перетащите главный узел *Suppliers* из окна *Источники данных* на форму.

Аналогичным образом перетащим на форму узел *Goods*.

Элементы управления с привязкой к данным *DataSet1*, *SuppliersBindingSource* и *GoodsBindingSource* появляются в области компонента на форме вместе с панелью инструментов (*BindingNavigator*) для управления записями.

Добавим на форму ещё две *DataGrid*: *DataGridView3* и *DataGridView4*, в которые будут выводиться дочерние и, соответственно, родительские строки.

Также добавим три кнопки:

- *Button1 (Заполнить)*. В обработчик событий *Click* этой кнопки поместим код для добавления строк к обеим таблицам;
- *Button2 (Найти товары)*. Обработчик событий *Click* этой кнопки будет содержать код вывода дочерних записей;
- *Button3 (Найти поставщиков)*. Обработчик событий *Click* этой кнопки будет содержать код вывода родительских записей.

Добавим также два текстовых поля для ввода значений *Title* и *Mark*, для которых будут выводиться дочерние и родительские записи.

## Добавление строк в объект *DataTable* посредством кода

Чтобы добавить новые записи в набор данных, нужно создать новую строку данных и добавить ее в коллекцию *DataRow (Rows)* набора данных (*DataTable*). Следующие процедуры показывают, как создать новую строку и вставить ее в *DataTable*. Примеры предназначены для нетипизированных наборов данных.

## Вставка новой записи в нетипизированный набор данных

В этом примере предполагается, что нетипизированный набор данных представлен *DataTable Suppliers*, который содержит 10 столбцов.

Нетипизированные наборы данных требуют знание имен столбцов или индексов при кодировании. В этом примере используются имена столбцов.

**Чтобы добавить запись в нетипизированный набор данных** следует вызвать метод *NewRow* объекта *DataTable* для создания новой пустой строки. Эта новая строка наследует структуру столбцов из таблицы данных *DataColumnCollection*.

Следующий код обработчика событий *Button1\_Click* создает новые строки, заполняет их данными и добавляет в коллекцию *Rows* таблицы.

```
private void button1_Click(object sender, EventArgs e)
{
    DataRow newSuppliersRow =
SuppliersGoods.Tables["Suppliers"].NewRow();
    newSuppliersRow["CodeSuppliers"] = 1;
    newSuppliersRow["Title"] = "BELSOFT";
    newSuppliersRow["AddressTo"] = "Вероника Кудрявцева";
    newSuppliersRow["Post"] = "Менеджер по закупкам";
    newSuppliersRow["Address"] = "ул. Большая Садовая, 12";
    newSuppliersRow["City"] = "Москва";
    newSuppliersRow["Index"] = "123456";
    newSuppliersRow["Country"] = "Россия";
    newSuppliersRow["Telephone"] = "(095)3252222";
    newSuppliersRow["Fax"] = "(095)3252222";
    SuppliersGoods.Tables["Suppliers"].Rows.Add(newSuppliersRow);

    DataRow newSuppliersRow1 =
SuppliersGoods.Tables["Suppliers"].NewRow();
    newSuppliersRow1["CodeSuppliers"] = 2;
    newSuppliersRow1["Title"] = "Tecnis";
    newSuppliersRow1["AddressTo"] = "Кривец Ольга";
    newSuppliersRow1["Post"] = "Менеджер по продажам";
    newSuppliersRow1["Address"] = "ул. Сурганова, 37";
    newSuppliersRow1["City"] = "Минск";
```

```
newSuppliersRow1["IIndex"] = "123456";
newSuppliersRow1["Country"] = "Беларусь";
newSuppliersRow1["Telephone"] = "(095)3252222";
newSuppliersRow1["Fax"] = "(095)3252222";
SuppliersGoods.Tables["Suppliers"].Rows.Add(newSuppliersRow1);
```

```
DataRow newSuppliersRow2 =
SuppliersGoods.Tables["Suppliers"].NewRow();
newSuppliersRow2["CodeSuppliers"] = 3;
newSuppliersRow2["Title"] = "Uni";
newSuppliersRow2["AddressTo"] = "Дмитрий Сидоров";
newSuppliersRow2["Post"] = "Координатор";
newSuppliersRow2["Address"] = "Бостон 78934";
newSuppliersRow2["City"] = "Новый Орлеан";
newSuppliersRow2["IIndex"] = "123456";
newSuppliersRow2["Country"] = "США";
newSuppliersRow2["Telephone"] = "(095)3252222";
newSuppliersRow2["Fax"] = "(095)3252222";
SuppliersGoods.Tables["Suppliers"].Rows.Add(newSuppliersRow2);
```

```
DataRow newGoodsRow =
SuppliersGoods.Tables["Goods"].NewRow();
newGoodsRow["CodeGoods"] = 1;
newGoodsRow["Mark"] = "Genius SP-G06";
newGoodsRow["CodeSuppliers"] = 1;
newGoodsRow["Supplier"] = "BELSOFT";
newGoodsRow["CodeType"] = 1;
newGoodsRow["Category"] = "Колонки, наушники, микрофоны";
newGoodsRow["Price"] = 500;
newGoodsRow["InWarehouse"] = 5;
newGoodsRow["Expected"] = 0;
newGoodsRow["MinimalStock"] = 0;
newGoodsRow["DeliveriesStopped"] = " ";
SuppliersGoods.Tables["Goods"].Rows.Add(newGoodsRow);
```

```

DataRow newGoodsRow1 =
SuppliersGoods.Tables["Goods"].NewRow();
newGoodsRow1["CodeGoods"] = 2;
newGoodsRow1["Mark"] = "Наушники Dialog M-
750HV+микрофон";
newGoodsRow1["CodeSuppliers"] = 2;
newGoodsRow1["Supplier"] = "Tecnis";
newGoodsRow1["CodeType"] = 1;
newGoodsRow1["Category"] = "Жесткие диски";
newGoodsRow1["Price"] = 500;
newGoodsRow1["InWarehouse"] = 5;
newGoodsRow1["Expected"] = 0;
newGoodsRow1["MinimalStock"] = 0;
newGoodsRow1["DeliveriesStopped"] = " ";
SuppliersGoods.Tables["Goods"].Rows.Add(newGoodsRow1);
DataRow newGoodsRow2 =
SuppliersGoods.Tables["Goods"].NewRow();
newGoodsRow2["CodeGoods"] = 3;
newGoodsRow2["Mark"] = "Samsung";
newGoodsRow2["CodeSuppliers"] = 3;
newGoodsRow2["Supplier"] = "Uni";
newGoodsRow2["CodeType"] = 1;
newGoodsRow2["Category"] = "Флоппи-дисководы";
newGoodsRow2["Price"] = 50;
newGoodsRow2["InWarehouse"] = 5;
newGoodsRow2["Expected"] = 0;
newGoodsRow2["MinimalStock"] = 0;
newGoodsRow2["DeliveriesStopped"] = " ";
SuppliersGoods.Tables["Goods"].Rows.Add(newGoodsRow2);
//Выполним привязку наборов к dataGridView1 и dataGridView2
dataGridView1.DataSource =
SuppliersGoods.Tables["Suppliers"].DefaultView;
dataGridView2.DataSource =
SuppliersGoods.Tables["Goods"].DefaultView;
}

```

## Связывание таблиц

Связь создается между общими столбцами из каждой таблицы – в данном случае используется столбец *CodeSuppliers*.

1. Перетащите объект **Связь** из вкладки **Набор данных Панели элементов** в таблицу *Goods*. Откроется диалоговое окно "**Отношение**".

2. В поле **Родительская таблица** выберите *Suppliers*.

3. В поле **Дочерняя таблица** выберите *Goods*.

4. В поле **Столбцы** таблицы *Suppliers* задайте значение ключа равным *CodeSuppliers*.

5. В поле **Столбцы** таблицы *Goods* задайте значение внешнего ключа равным *CodeSuppliers*.

6. Нажмите кнопку ОК, чтобы создать связь; линия отношения отобразится в конструкторе между двумя таблицами.

7. В главном меню **Данные** выберите команду **Показать подписи отношений**.

## Навигация в DataRelations (ADO.NET)

Одно из основных назначений объекта *DataRelation* состоит в обеспечении переходов от одного объекта *DataTable* к другому в пределах *DataSet*. Это позволяет получать все связанные объекты *DataRow* в одном объекте *DataTable* при указании единственного значения *DataRow* из связанного объекта *DataTable*. Например, после установления связи *DataRelation* между таблицей заказчиков и таблицей заказов можно получить все строки с заказами для конкретной строки заказчика с помощью метода *GetChildRows*.

В следующем примере кода создается связь *DataRelation* между таблицей *Suppliers* и таблицей *Goods* набора данных *DataSet1*

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;
```

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

```
namespace laba3
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        static DataSet1 SuppliersGoods = new DataSet1();
```

```
        DataRelation Relation =
```

```
            SuppliersGoods.Relations.Add("Relation",
```

```
SuppliersGoods.Tables["Suppliers"].Columns["CodeSuppliers"],
            SuppliersGoods.Tables["Goods"].Columns["CodeGoods"]);
```

### **Получение доступа к записям в связанных объектах**

Если набор данных содержит связанные таблицы, объект *DataRelation* может сделать доступными подчинённые записи в другой таблице.

Можно использовать объект *DataRelation* для поиска связанных записей путем вызова метода *GetChildRows* класса *DataRow* в родительской таблице; этот метод возвращает массив связанных дочерних записей. Или можно вызвать метод *GetParentRow* класса *DataRow* в дочерней таблице; этот метод возвращает одну *DataRow* из родительской таблицы.

### **Доступ к подчинённым записям**

Для возврата дочерних записей выбранной родительской записи вызывается метод *GetChildRows* для определенной строки данных *Suppliers*. При этом возвращаются все заказы для каждого заказчика в виде массива строк из таблицы *Goods*. Это действие выполняется при нажатии кнопки *button2*.

```
private void button2_Click_1(object sender, EventArgs e)
```

```
{
```

```
    DataSet1.SuppliersRow[] rez = (DataSet1.SuppliersRow[]) SuppliersGoods.Suppliers.Select("Title = '" + textBox1.Text + "'");
```

```
22
```

```

foreach (DataSet1.SuppliersRow Row in rez)
{
this.dataGridView3.DataSource =(DataSet1.GoodsRow[])
Row.GetChildRows("Relation");
}
}

```

### **Возврат родительской записи для выбранной дочерней записи**

При нажатии кнопки `button3` вызывается метод *GetParentRow* для определенной строки таблицы *Goods*. При этом возвращается одна строка из таблицы *Suppliers*.

```

private void button3_Click(object sender, EventArgs e)
{
DataSet1.GoodsRow[] rez1 = (DataSet1.GoodsRow[]) SuppliersGoods.Goods.Select("Mark = '" + textBox2.Text + "'");
DataSet1.SuppliersRow[] mass = new DataSet1.SuppliersRow[10];
Int32 i = 0;
foreach (DataSet1.GoodsRow row in rez1)
{
mass[i] = (DataSet1.SuppliersRow)row.GetParentRow("Relation");
i++;
}
this.dataGridView4.DataSource = mass;
}

```

### **Сохранение проекта**

В меню *Файл* выберите *Сохранить все*.

### **Тестирование приложения**

1. Нажмите клавишу F5.
2. Выберите имена поставщиков, чтобы проверить правильность отображения товаров, и наоборот, выберите товар, для которого Вы хотите узнать поставщика.

## ЛАБОРАТОРНАЯ РАБОТА № 4

### ОТОБРАЖЕНИЕ СВЯЗАННЫХ ДАННЫХ НА ФОРМЕ В ПРИЛОЖЕНИИ WINDOWS

В данной лабораторной работе рассматриваются следующие задачи:

- создание нового проекта Приложение Windows;
- создание источника данных;
- создание элементов управления для отображения данных из таблицы
- фильтрация записей таблицы Suppliers.

**Создание проекта** рассмотрено в Лабораторной работе №1.

#### **Создание источника данных**

На этом шаге создается набор данных на основе таблиц Suppliers и Goods учебной базы данных Borel.

*Для создания источника данных:*

1. В меню *Данные* выбрали команду *Показать источники данных*.
2. Чтобы запустить *Мастер настройки источника данных*, выбрали элемент *Добавили новый источник данных* в окне *Источники данных*.
3. На странице *Выбор типа источника данных* выбрали элемент *База данных* и нажали *Далее*.
4. На странице *Выбор подключения базы данных* выполнили следующее действие:
  - Выбрали Новое подключение для открытия диалогового окна *Добавили/изменили подключение*.
5. Щелкнули *Далее* на странице Сохранить строку подключения в файле конфигурации приложения.
6. Развернули узел **Таблицы** на странице *Выбор объектов базы данных*.
7. Выбрали таблицы *Suppliers* и *Goods* и нажали *Готово*.

Набор данных *DataSet1* будет добавлен к проекту, и таблица *Suppliers* появится в окне *Источники данных*.

### **Создание элементов управления для отображения данных из таблицы *Suppliers***

***Чтобы создать элементы управления для отображения данных клиентов (родительские записи):***

1. Выбрали таблицу *Suppliers* в окне *Источники данных* и щелкнули стрелку раскрывающегося списка.
2. Выбрали *Сведения* в меню.
3. Перетащили главный узел *Suppliers* из окна *Источники данных* на *Form1*.

### **Создание элементов управления для отображения данных из таблицы *Goods***

***Чтобы создать элементы управления для отображения заказов для каждого клиента (дочерних записей):***

- В окне *Источники данных* развернули узел *Suppliers*, выбрали последний столбец в таблице *Suppliers* (это разворачиваемый узел *Goods*) и перетащили его на нижнюю часть *Form1*.

### **Тестирование приложения**

***Чтобы проверить работу приложения:***

1. Нажали клавишу F5 для запуска приложения.
2. Выбрали различных клиентов с помощью *SuppliersBindingNavigator* для проверки корректности отображения заказов на *DataGridView*.

Далее необходимо произвести фильтрацию записей *Suppliers* путем добавления параметризации к таблице *Suppliers*. Чтобы сделать это, выбрали любой элемент управления, который отображает данные из таблицы *Suppliers*, щелкнули на смарт-тег и выбрали *Добавить запрос*. Выполнили диалоговое окно "Построитель условий поиска".

## ЛИТЕРАТУРА

1. Solid Quality Learning Microsoft SQL Server 2005. Реализация и обслуживание. Учебный курс Microsoft / Пер. с англ. – М.: «Русская Редакция», СПб.: «Питер», 2007. – 768 с.: ил.
2. Электронная документация по SQL Server 2008 [Электронный ресурс] – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/ms365325.aspx>, свободный. – Загл. с экрана.
3. Библиотека MSDN [Электронный ресурс] – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/default.aspx>, свободный. – Загл. с экрана.

Учебное издание

**БАЗЫ ЗНАНИЙ И ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ  
В САПР**

Лабораторный практикум  
для студентов специальности 1-40 01 02-01 «Информационные  
системы и технологии в проектировании и производстве»

В 3 частях

Часть 1

С о с т а в и т е л и :

**КОЧУРОВ** Вадим Александрович  
**ГЕРМАН** Юлия Олеговна

Технический редактор *О. В. Песенько*

Подписано в печать 31.08.2012. Формат 60×84 <sup>1</sup>/<sub>16</sub>. Бумага офсетная. Ризография.

Усл. печ. л. 1,57. Уч.-изд. л. 1,23. Тираж 100. Заказ 409.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет. ЛИ № 02330/0494349 от 16.03.2009. Пр. Независимости, 65. 220013, г. Минск.