ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КЛИЕНТ-СЕРВЕРНЫЕ ПРИЛОЖЕНИЯ БАЗ ДАННЫХ

Методические указания к лабораторным работам

Рязань 2010

УДК 004.655.3 (078.8)

Клиент-серверные приложения баз данных: Методические указания к лабораторным работам / Рязан. гос. радиотехн. универ.; Сост. А.Н. Андреев, А.В. Благодаров, Н.Н. Гринченко, Рязань, 2010. 16 с.

Содержат рекомендации для проведения лабораторных работ, посвященных изучению взаимодействия клиентских приложений с базами данных.

Предназначены для студентов очной и заочной форм обучения специальностей 230101 «Вычислительные машины, комплексы, системы и сети», 230104 «Системы автоматизированного проектирования», 230105 «Программное обеспечение вычислительной техники и автоматизированных систем», 080801 «Прикладная информатика», 090102 «Компьютерная безопасность» по дисциплинам «Базы данных», «Системы управления базами данных», «Проектирование баз данных», «Клиент-серверные приложения баз данных», «Проектирование распределенных баз данных».

Клиент-серверные технологии баз данных

Составители
А н д р е е в Александр Николаевич
Б л а г о д а р о в Андрей Витальевич
Г р и н ч е н к о Наталья Николаевна

ВВЕДЕНИЕ

Одним из важнейших аспектов при разработке информационных систем является разработка клиентских приложений баз данных (БД). Основной функцией таких приложений является представление информации в удобном для пользователя виде. Для разработки клиентских приложений могут с успехом использоваться современные системы программирования общего назначения, такие как: Microsoft Visual Studio, Borland Delphi, Borland C++ Builder и др.

Большинство систем программирования общего назначения содержат развитые средства взаимодействия с БД, с помощью которых можно осуществлять доступ к практически любым реляционным базам данных.

В методических указаниях рассматриваются примеры разработки клиентских приложений с указанием фрагментов кода, написанных с использованием популярной системы программирования Microsoft Visual C# 2005. В данной системе доступ к базам данных осуществляется с помощью технологии ADO.NET.

В качестве СУБД используется MS SQL Server 2000.

Лабораторная работа №1
РАЗРАБОТКА ПРОСТЕЙШИХ ПРИЛОЖЕНИЙ БАЗ ДАННЫХ НА ЯЗЫКЕ C#

Цель работы: Ознакомиться с технологией доступа к БД ADO.NET при разработке клиентских приложений баз данных с использованием системы программирования Microsoft Visual C# 2005.

ADO.NET соединения

Для доступа к БД в системе программирования Microsoft Visual C# используется универсальная технология доступа к данным ADO.NET. С ее помощью можно единообразно работать практически с любыми СУБД.

Чтобы организовать взаимодействие между базой данных и приложением, прежде всего необходимо создать соединение с БД. В ADO.NET можно создавать и управлять соединениями, используя *объекты соединения* (connection objects):

- **SqlConnection** – объект, который управляет соединением к MS SQL Server 7.0 и выше, определен в пространстве имен System.Data.SqlClient;
- **OleDbConnection** - объект, который управляет соединением к любым хранилищам данных, доступных через технологию OLE DB, определен в пространстве имен System.Data.OleDb ;
- **OdbcConnection** - объект, который управляет соединением к любым базам данных, доступных через технологию ODBC, определен в пространстве имен System.Data.Odbc;
- **OracleConnection** - объект, который управляет соединением к базам данных Oracle, определен в пространстве имен System.Data.Oracle.

В рамках лабораторных работ будем использовать СУБД MS SQL Server 2000 и, соответственно, объекты типа SqlConnection. Хотя класс SqlConnection специфичен для СУБД MS SQL Server, многие свойства, методы и события этого класса такие же, как у классов OleDbConnection и OdbcConnection. Следовательно, навыки, полученные при выполнении работы пригодятся для работы с любыми другими СУБД.

Объект SqlConnection

Рассмотрим основные свойства и методы объекта *SqlConnection*.

Для указания параметров соединения с БД в классе *SqlConnection* доступно свойство **ConnectionString**, которое содержит строку подключения. Строка соединения может включать имя сервера баз данных, имя базы данных, имя пользователя, пароль и некоторую другую информацию, например:

```
"Data Source=dc.vpm.rrtu\\stud;"+
"Initial Catalog=743_Supply;"+
"User ID=sa;"+
"Password=sa;";
```

Порядок заполнения строки соединения будет рассмотрен далее

Для открытия соединения с источником данных предназначен метод ***Open***, который использует информацию, заданную в свойстве *ConnectionString*.

Метод ***Close*** закрывает соединение. Не следует пренебрегать операцией закрытия соединения, так как многие источники данных поддерживают только ограниченное число открытых соединений, и открытые соединения отнимают системные ресурсы.

Объект соединения может быть создан либо на этапе разработки проекта с помощью средства под названием *Server Explorer*, либо программным способом во время выполнения приложения. Далее рассмотрим оба этих способа.

Подключение к источникам данных во время разработки приложений с помощью Server Explorer

Средство *Server Explorer* обеспечивает подключение к источникам данных во время разработки приложения. Это позволяет просматривать доступные источники данных, а также редактировать и создавать таблицы и другие объекты баз данных. Приложение не использует это соединение. Информация, получаемая с помощью соединения во время разработки, используется для установления свойств нового соединения, которое автоматически добавляется в приложение.

Например, во время разработки приложения можно использовать *Server Explorer* для создания подключения к БД, расположенной на некотором сервере баз данных. Позже, при проектировании пользовательского интерфейса (например, формы) можно просматривать экземпляр базы данных, выбирать поля из таблиц и перетаскивать их на форму. При этом автоматически создается адаптер

данных (data adapter) для формы. Также создается новое соединение, определенное для формы, копирующее информацию строки соединения из соединения, созданного во время работы с *Server Explorer*, в новое соединение. Новое соединение будет использоваться для работы с источником данных при запуске приложения.

Информация о созданных во время разработки соединениях хранится на локальном компьютере. Следовательно, установив однажды такое соединение, можно его использовать в дальнейшем. Соединения доступны через окно *Server Explorer*.

Для вызова *Server Explorer* следует выполнить пункт меню *Tools – Connect to Database*. Далее рассмотрим процесс создания нового соединения в *Server Explorer*.

1. Создайте новый проект, выбрав пункт меню *File – New – Project* и задав следующие настройки:

Project Type (тип проекта): Visual C# - Windows
Template (шаблон): Windows Application
Name (название проекта): SampleApplication

Укажите также путь к папке для хранения проекта, например, d:\student\743\Brigada1\Lab1 (рисунок 1).

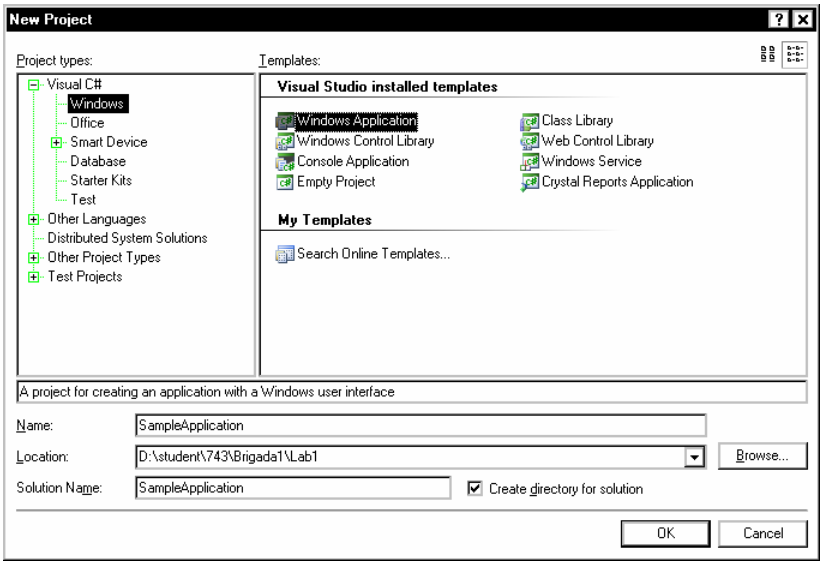

Рисунок 1 – Создание нового проекта

Не рекомендуется сохранять проект на сетевом диске, так как это может существенно увеличить время компиляции. Если

особенности компьютерного класса, где выполняются лабораторные работы, предусматривают хранение информации на сетевых дисках, то можно сначала создать и отладить проект на локальном диске, а уже затем скопировать его на сетевой диск.

2. Откройте окно *Server Explorer*, если оно еще не отображено. Для этого выберите пункт меню *Tools – Connect to Database*. При этом появится окно добавления нового соединения, которое нужно закрытьНажав кнопку *Cancel*. В окне *Server Explorer* отображены соединения с базами данных под общим узлом *Data Connections*.

3. Добавьте новое соединение, выполнив пункт контекстного меню Add Connection для узла *Data Connections* (рисунок 2).

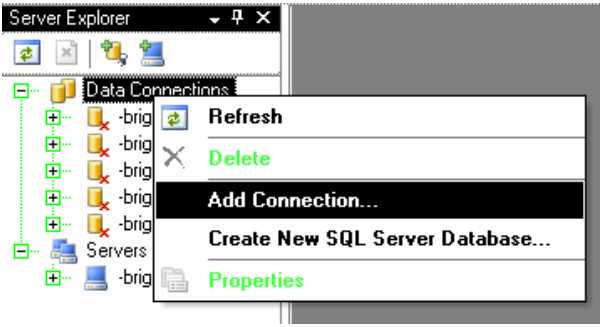

Рисунок 2 – Добавление нового соединения

4. Выберите источник данных, нажав в открывшемся окне кнопку *Change*. В качестве источника данных (*Data Source*) в нашем случае (для СУБД MS SQL Server 2000) это будет Microsoft SQL Server, а в качестве провайдера (*Data Provider*) – *.NET Framework Data Provider for SQL Server* (рисунок 3). Нажмите OK.

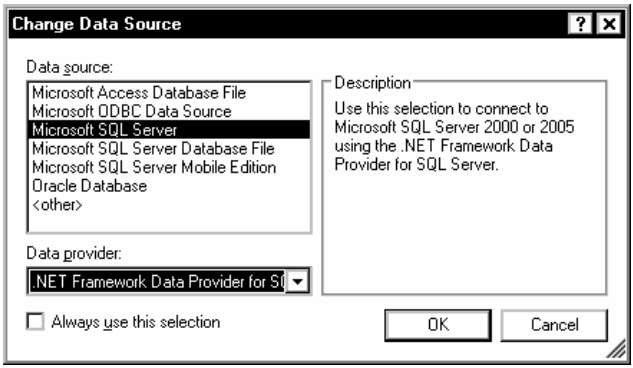

Рисунок 3 – Выбор источника данных

5. Укажите имя сервера (например, dc.vpm.rrtu\stud), способ аутентификации (в нашем случае, *Use SQL Server Authentication*), имя пользователя и пароль (в нашем случае sa и sa), а также и имя вашей базы данных (например 743_Supply) (рисунок 4). Проверьте соединение, нажав кнопку *Test connection*.

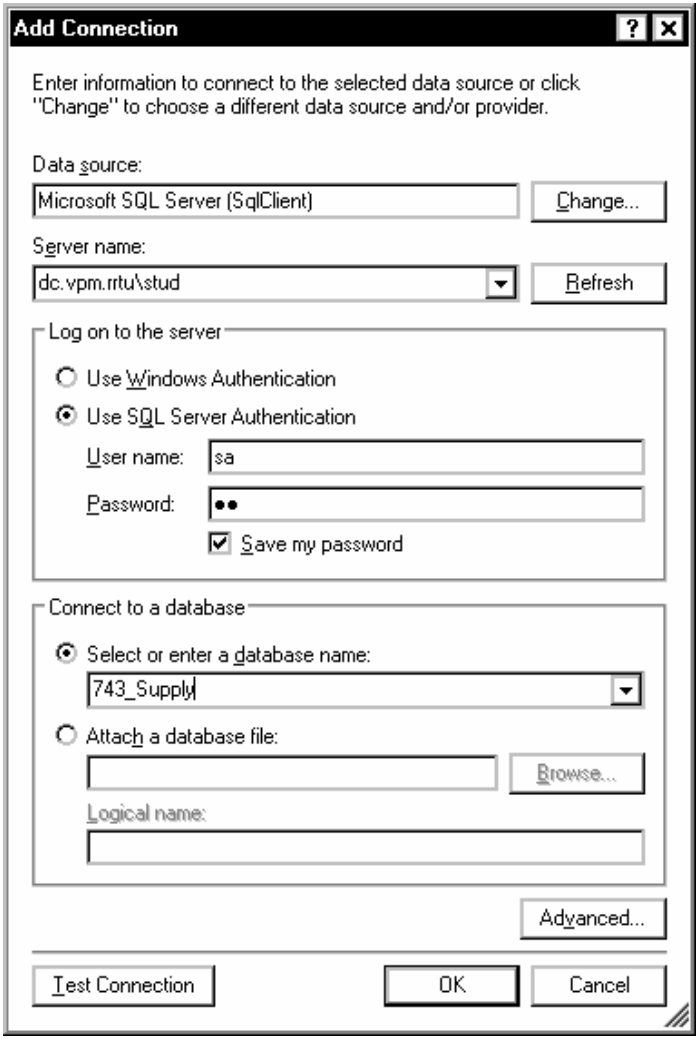

Рисунок 4 – Настройка соединения

После нажатия кнопки *OK*, в окне *Server Explorer* появится новое соединение.

С помощью *Server Explorer* можно просматривать данные, тестировать запросы, создавать и редактировать объекты БД и т.д.

Для просмотра данных таблицы, например, таблицы *Товар*, необходимо в контекстном меню таблицы *Товар* выбрать опцию *Show Table Data*.

Для просмотра сведений о структуре таблицы в контекстном меню следует выбрать действие *Open table Definition*.

Для отладки запросов следует выбрать пункт меню *New Query*. Пусть требуется вывести наименования всех товаров. Для этого можно воспользоваться визуальным конструктором запросов.

В контекстном меню любой таблицы выберем действие *New Query*. В появившемся диалоговом окне выберем интересующую нас таблицу таблицы *Товар.* В окне конструктора запросов отметим поля *Код* и *Наименование* и выполним запрос. Результаты выполнения будут видны в соответствующем окне.

Созданное соединение можно будет использовать и в других проектах. Для удаления соединения необходимо в контекстном меню соединения выбрать опцию *Delete* и подтвердить удаление.

Создание нового соединения программным способом

Можно создавать объект *SqlConnection* непосредственно в коде приложения.

В созданный ранее проект добавим код, который создает объект типа *SqlConnection*, устанавливает свойство *ConnectionString*, открывает и закрывает соединение.

Переключитесь в режим редактирования кода приложения, выполнив пункт меню *View – Code*.

Добавьте пространство имен *System.Data.SqlClient* с помощью оператора:

```
using System.Data.SqlClient;
```

Для класса формы добавьте следующий метод:

```
public void ConnectToSql()
{
  SqlConnection conn = new SqlConnection();
  conn.ConnectionString =
   "Data Source=dc.vpm.rrtu\\stud;"+
   "Initial Catalog=743_Supply;"+
   "User ID=sa;"+
   "Password=sa;";
```

```
    try {
     conn.Open();
    }
    catch (Exception ex) {
     MessageBox.Show(ex.Message);
    }
    finally {
     conn.Close();
    }
  }
```

Для проверки работоспособности, добавьте на форму объект *ListBox,* задав имя *ListOfConnectionTest*, и объект *Button* с именем *btnRun* и свойством *Text = "Выполнить"*.

Также измените заголовок формы приложения, задав для нее свойство *Text = "Проверка соединения"*.

В блок *try* добавьте код:

```
ListOfConnectionTest.Items.Add("Соединение открыто");
```

В блок *catch* добавьте код:

```
ListOfConnectionTest.Items.Add("Ошибка соединения");
```

В блок *finally* добавьте код:

```
ListOfConnectionTest.Items.Add("Соединение закрыто");
```

Для кнопки добавьте код:

```
ConnectToSql();
```

Запустите приложение. Если при запуске появится ошибка (рисунок 5), следует зайти в пункт меню *Project – <Имя проекта> Properties* и на вкладке *Debug* установить флажок для опции *Enable SQL Server debugging*.

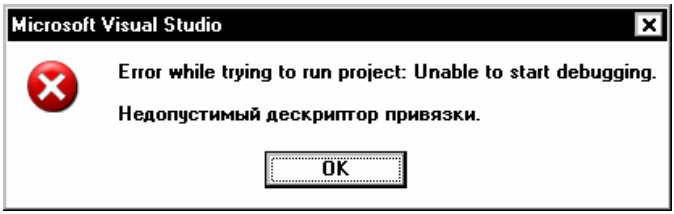

Рисунок 5 – Ошибка при запуске приложения

После запуска приложения нажмите на кнопку. В случае успешного соединения вы увидите в окне сообщения как на рисунке 6.

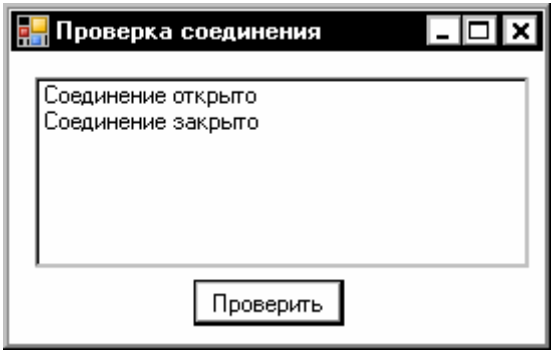

Рисунок 6 – Успешная проверка соединения

Можно сымитировать ошибку открытия соединения, задав в строке соединения несуществующее имя сервера. В этом случае вы увидите в окне соответствующее сообщение.

Создание простейшего приложения БД

Создадим простейшее приложение баз данных, позволяющее просматривать информацию в таблице Товары.

1. Создайте новый проект с именем SampleGridView.

2. Измените свойство формы *Text = "Форма для просмотра таблицы"*. Перетащите с вкладки *Data* панели инструментов объект *DataGridView* и задайте для него свойство *Text = "dgViewТовар"*.

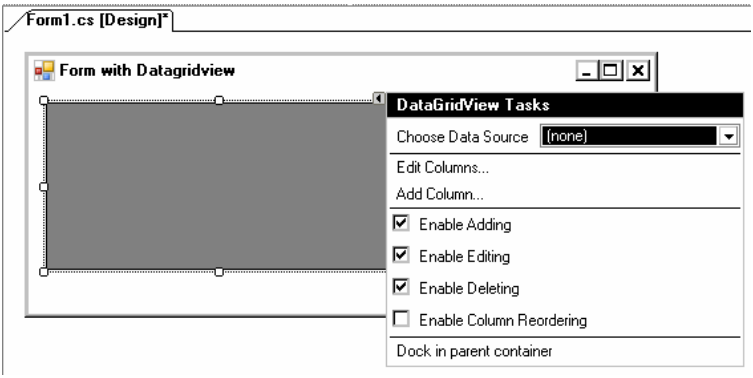

Рисунок 7 – Выбор источника данных

3. Укажите источник данных, выполнив пункт *Choose data source* в открывшемся меню (рисунок 7). Можно выбрать существующий или создать новый. В данном примере создадим новый источник.

4. При создании нового источника данных появится окно мастера (рисунок 8)

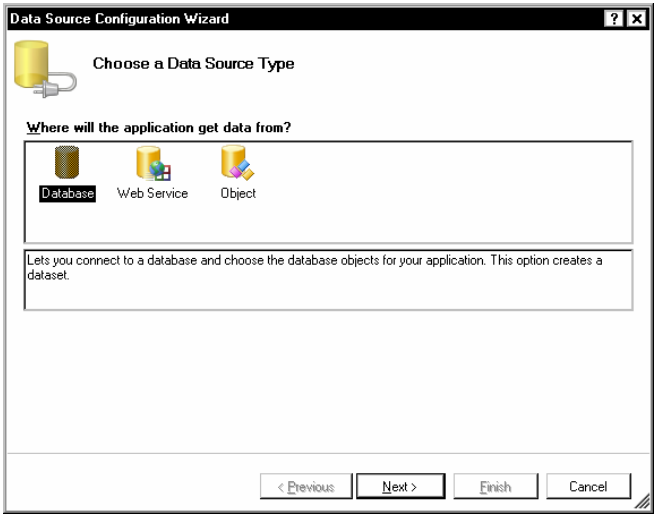

Рисунок 8 – Выбор типа источника данных

Выберите *Database* и нажмите *Next*.

5. На следующем шаге мастер предложит выбрать существующее подключение либо создать новое. Ранее мы создавали соединение к базе данных *743_Supply*. Выберите его и нажмите *Next*.

6. На следующем шаге мастер предложит сохранить строку подключения в файле конфигурации. Оставьте без изменений и нажмите *Next*.

7. На следующем шаге мастера необходимо выбрать объект базы данных (рисунок 9). Выберите таблицу *Товар*. Измените название набора данных: *GoodsDataSet* и нажмите *Finish*.

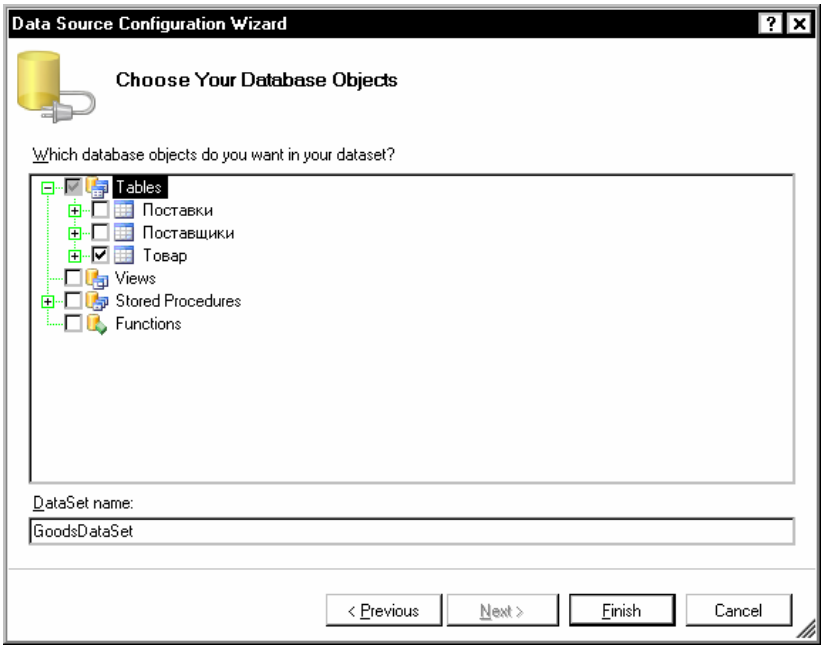

Рисунок 9 – Выбор объекта базы данных

8. В результате (рисунок 10) добавится 3 компонента: *goodsDataSet*, товарBindingSource, товарTableAdapter.

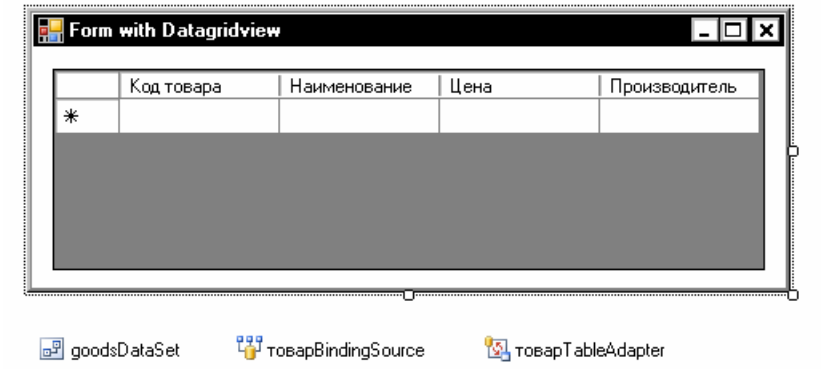

Рисунок 10 – Выбор объекта базы данных

Также будет сгенерирован следующий код:

```
private void Form1_Load(object sender, EventArgs e)
{
 this.товарTableAdapter.Fill(this.goodsDataSet.Товар);
}
```

9. Запустите приложение. Данные появятся сразу после загрузки формы (рисунок 11).

Рисунок 11 – Окно простейшего приложения БД

Настройка соединения в процессе выполнения приложения

При разработке клиентских приложений обычно нужно дать возможность пользователю настраивать параметры соединения с БД в процессе работы с программой. Рассмотрим, каким образом это можно сделать.

1. Создайте новый проект с именем SampleConnectionDialog.

2. Измените свойство формы Text = “Форма с вызовом диалога настройки соединия”. Перетащите с вкладки *Data* панели инструментов объект *Button*, задав свойства *Name = btnConnection*, *Text = “Выбрать соединение”,* и *TextBox* со свойством *Name = txtString*.

3. Добавьте ссылку на библиотеку *Microsoft.Data.ConnectionUI.Dialog.dll* (по умолчанию, C:\Program Files\Microsoft Visual Studio 8\Common7\IDE). Для этого в обозревателе решений (*Solution Explorer*) в контекстном меню *References* выберите *Add reference*. На вкладке *Browse* выберите нужную библиотеку. Нажмите *OK*.

4. Добавьте следующий код для кнопки.

```
Microsoft.Data.ConnectionUI.DataConnectionDialog _dialog =
  new Microsoft.Data.ConnectionUI.DataConnectionDialog();
Microsoft.Data.ConnectionUI.DataSource.
  AddStandardDataSources(_dialog);
Microsoft.Data.ConnectionUI.DataConnectionDialog.Show(_dialog);
```

```
txtString.Text = "ConnectionString: "+_dialog.ConnectionString;
```

5. Запустите приложение и настройте необходимое соединение. После настройки в текстовом поле вы увидите значение строки соединения, которое можно использовать для настройки объекта SqlConnection.

Задание

1. В соответствии с предметной областью, заданной вариантом задания (см. Приложение 1), разработать схему базы данных, которая будет включать в себя таблицы и связи между ними. Указать первичные и внешние ключи таблиц. Заполнить таблицы не менее чем двадцатью записями правдоподобной информации.

2. Разработать клиентское приложение для доступа к своей БД, позволяющее просматривать информацию в одной из таблиц-справочников. Приложение реализовать в двух вариантах: с программным созданием соединения и с созданием соединения с помощью Server Explorer. Для варианта с программным созданием соединения предусмотреть кнопку для вызова диалога настройки соединения.

СОДЕРЖАНИЕ