

Разработка Windows и Web решений с ADO.NET и MS SQL Server

Что такое ADO.NET?

- Это компонент доступа к данным в составе .NET Framework
- В его составе присутствует провайдер для соединения с внешними данными
- Используются объекты **Dataset** для локальной обработки в буфере обмена информации из БД и преобразования ее в формат XML
- ADO.NET можно использоваться:
 - в Windows приложениях для создания решений по доступу к данным в корпоративных настольных системах
 - в WEB приложениях для создания решений по доступу к данным через браузер
 - Для создания приложений для мобильных устройств и создания WEB сервисов

Условия применения ADO.NET

- **ОС** – Windows 2000 (SP2), Windows NT (SP 6a), Windows XP
- .NET Framework
(msdn.microsoft.com/netframework/downloads/howtoget.asp, .NET runtime, .NET SDK)
- MDAC - Microsoft Data Access Components (www.microsoft.com/data) - OLE DB, ADO, и ODBC

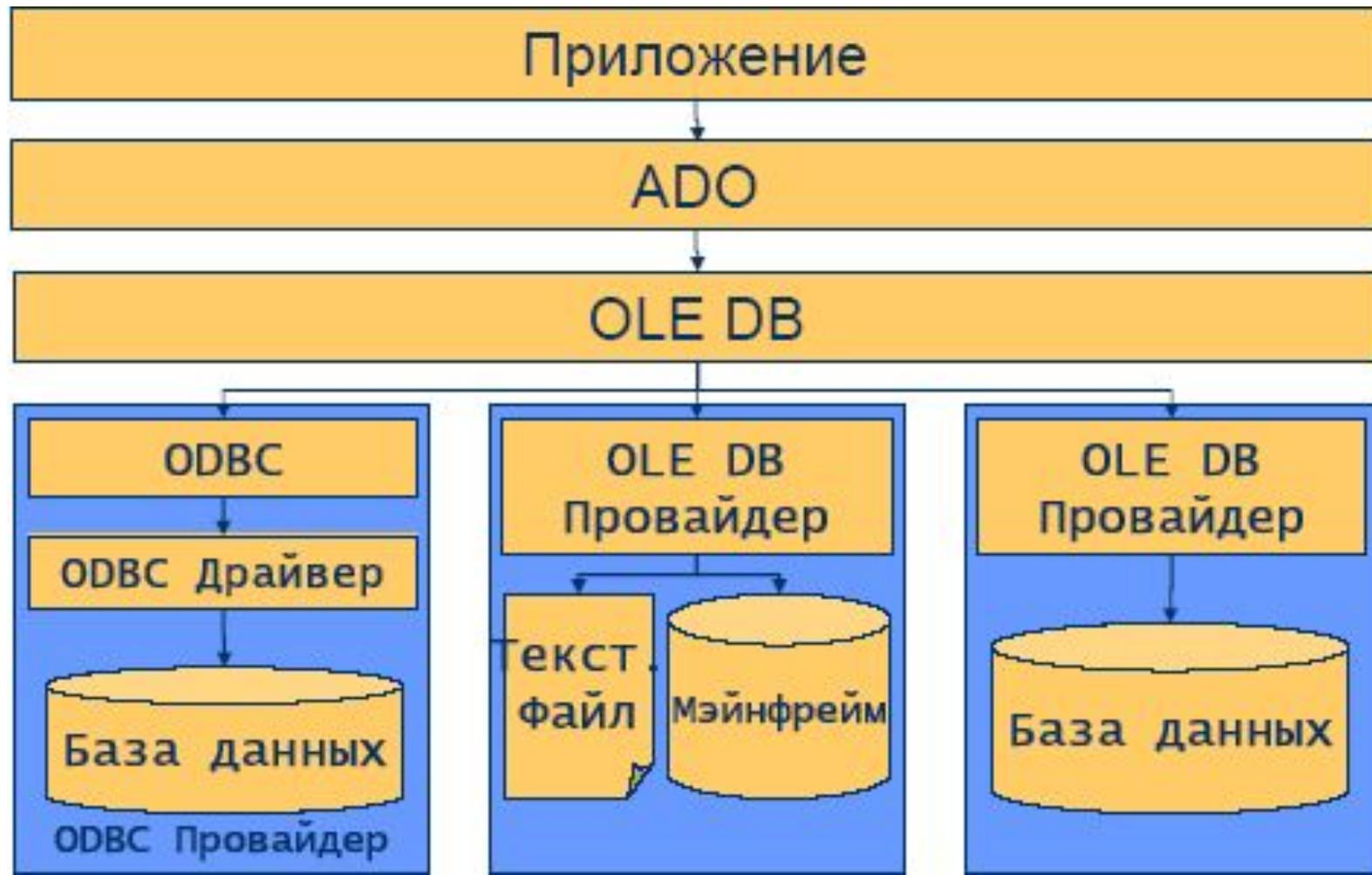
Обзор ADO.NET

Предшествующие технологии

- ODBC (Open Database Connectivity)
 - Взаимодействует с большим числом СУБД
 - Имеет набор функций API, который широко используются
 - Использует язык SQL для доступа к данным
- DAO (Data Access Objects)
 - Программный интерфейс для баз данных JET/ISAM
 - Использует ActiveX и OLE automation
- RDO (Remote Data Objects)
 - Тесно связан с ODBC
 - Более подходит к архитектуре клиент/сервер (по сравнению с DAO)
- OLE DB
 - Предоставляет доступ к реляционным и другим типам данных
 - Построен на основе технологии COM
 - Допускает и другие, кроме языка SQL, способы извлечения данных
 - Может использовать драйверы ODBC
 - Имеет низкоуровневый (C++) интерфейс
- ADO (ActiveX Data Objects)
 - Простой объектно-ориентированный интерфейс
 - Обеспечивает программный доступ к OLE DB вне C++ (из других языков программирования)

Цели ADO.NET

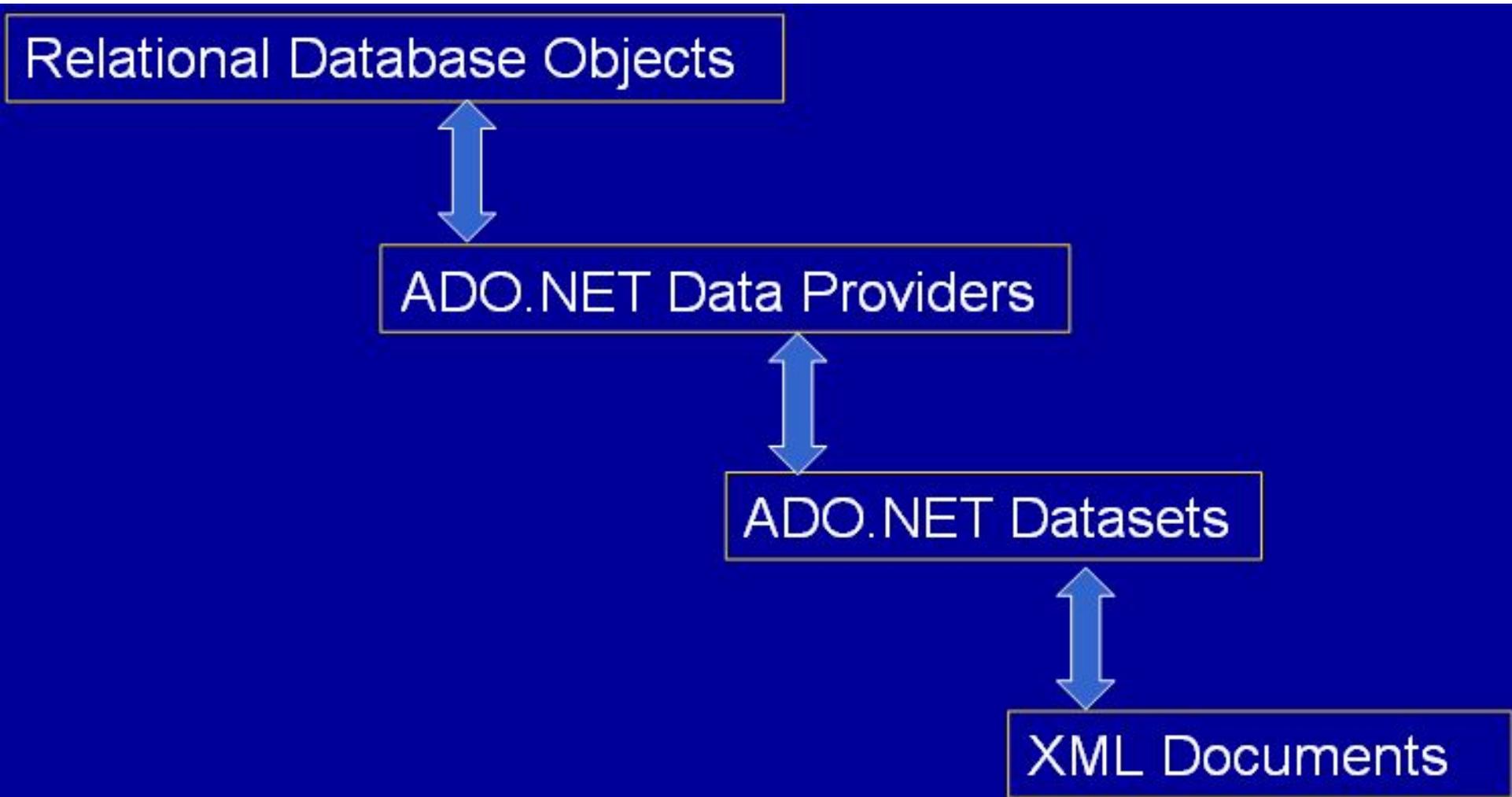
- Оптимальная структура классов
- Масштабируемость
- Поддержка XML (иерархические и реляционные данные)
- Доступ к данным по протоколу HTTP
- Поддержка знакомой программой модели ADO
- ADO по прежнему доступна через интерфейс .NET COM



.NET Framework

- v.2.6+ для SQL Server или OLE DB .NET Data Provider
- v2.7+ для ODBC .NET Data Provider
- Visual Studio .NET автоматически устанавливается (в версии 2.7+)

ADO.NET, БД и XML



Возможности провайдера ADO.NET

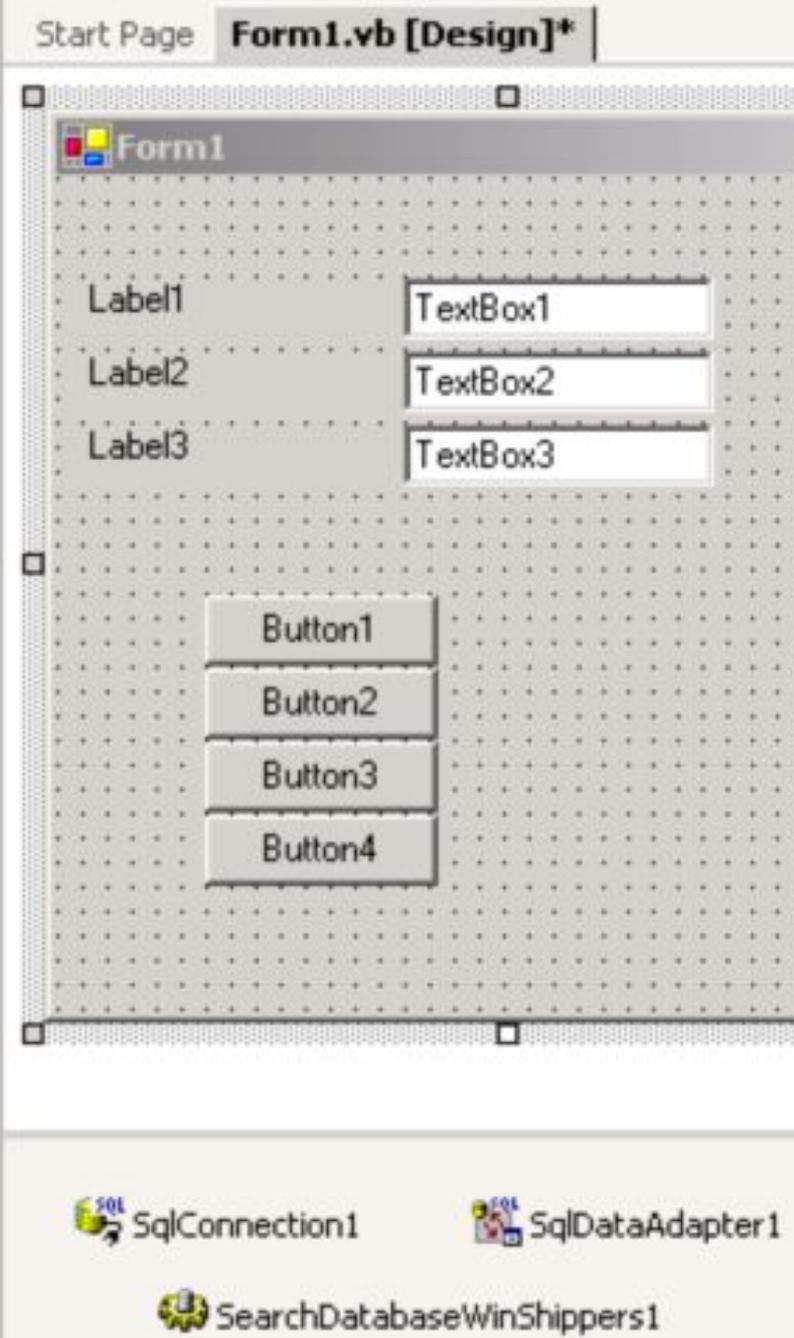
- Провайдер предназначен для “захвата” приложением ADO.NET внешних данных
- Классы объектов ADO.NET провайдера помогают выполнять следующие задачи:
 - соединение с источником данных
 - захват данных
 - обработку данных
 - переопределение данных
- Каждый .NET провайдер (SQL Server, Oracle, Access и т.д.) имеет свой набор классов объектов ADO.NET, включая
 - Connection – соединение с БД
 - Command – для выполнения команд SELECT, INSERT, UPDATE, DELETE
 - DataReader – возврат данных, предназначенных только для чтения
 - DataAdapter - заполняет временный набор данных в буфере обмена с БД и обновляет базу данных значениями из временного набора данных

Взаимодействие ADO.NET с БД

- Чтение данных с помощью объекта **DataReader**
 - Используются классы объектов Connection и DataReader
 - Соединение с БД разрывается, пока открыт объект DataReader
 - Большая скорость разрыва соединения
- Использование классов объектов Connection, DataAdapter и **Dataset** позволяет выполнить
 - Изменение БД
 - Масштабирование БД

Что есть Windows приложение .NET

- Разрабатывается для локальной установки на ПК пользователя
- Может иметь преимущество перед локальными ресурсами (файлами, Microsoft Office и т.п.)
- Windows приложение .NET использует классы объектов *Windows Form* и *Window Form control* (элемент формы)



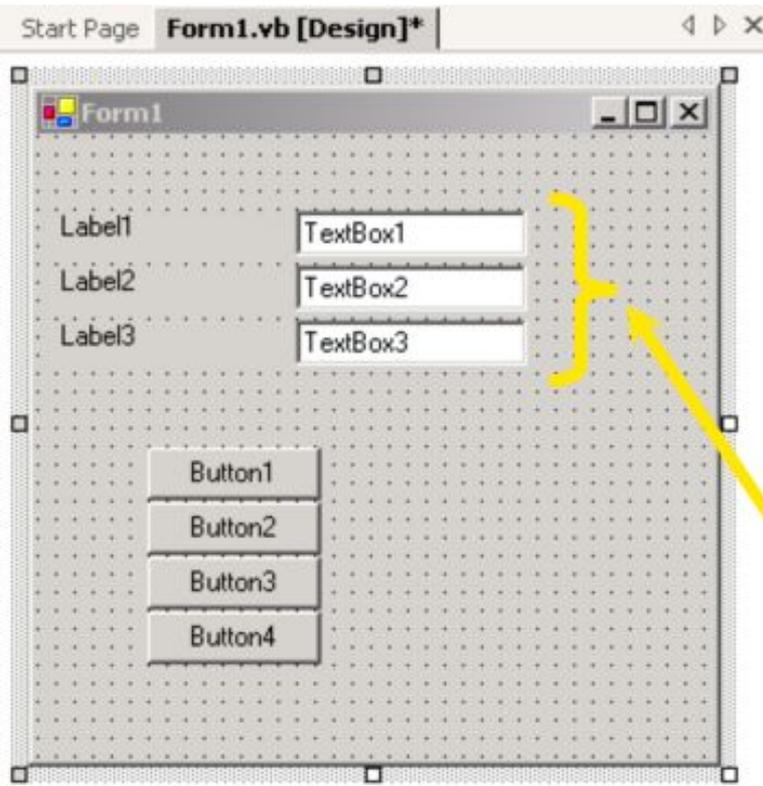
- На стартовой странице выполнить команду **FILE - NEW PROJECT**, выбрать шаблон *Windows Application* из папки *Visual Basic*, ввести имя проекта *SearchDatabaseWin*, **OK**
- С панели инструментов методом “**Drag and Drop**” трижды перенести на форму *Form1* элементы *TextBox* (текстовое поле) и *Labels* (метки) и четырежды - элемент *Button* (кнопка)
- По команде **VIEW – OPEN SERVER EXPLORER** откройте окно с БД, перенесите из БД *Northwind MS SQL Server* таблицу *Shippers* на форму. При этом в системный трей (под формой) добавятся объекты *SqlConnection1* и *SqlDataAdapter1*
- В контекстном меню объекта *SqlDataAdapter1* выполните команду **GENERATE DATASET**, указав имя *SearchDatabaseWinShippers* объекта *DataSet*. Нажмите **OK** для размещения объекта с именем *SearchDatabaseWinShippers1* в системном трее

Двойным щелчком по заголовку формы *Form1* в процедуре события Load (*открыть*) введите **выделенный** код:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    SqlDataAdapter1.Fill(SearchDatabaseWinShippers1, "Shippers")
```

```
End Sub, где Shippers - имя объекта DataTable набора данных DataSet
```



Заполните поле формы *TextBox1* данными. Для этого выделите его и в окне свойств в ветке *DataBindings* выберите свойство *Text*, в наборе *SearchDatabaseWinShippers1* разверните таблицу *Shippers* и укажите поле *ShipperID* таблицы для связывания его с полем *TextBox1* формы. Повторите настройки для полей *TextBox2* и *TextBox3* формы, связав их с полями *CompanyName* и *Phone* таблицы *Shippers*

Populate text boxes with dataset with the Text property within the Databindings property collection

Навигация по строкам в наборе данных

Используйте свойство *Position* объекта *BindingContext* для контроля навигации по строкам или столбцам, отображаемым в текстовых полях формы:

1. *Position.MinValue* и *Position.MaxValue* для перехода на первую и последнюю строку набора
2. “- = 1” и “+ = 1” для перехода на предыдущую или следующую строку

(во время отладки приложения по кнопке F5, можно проверить, что в БД результаты редактирования не вносятся. Они сохраняются в буфере обмена, который чистится после закрытия сессии, т.е. при повторном запуске приложения (<http://www.programmingmsaccess.com/mygb.htm>))

Процедуры событий нажатия на кнопки

Дважды щелкните на кнопке Button1 и в окне процедуры введите **выделенный** код

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Me.BindingContext(SearchDatabaseWinShippers1, "Shippers").Position = _  
Me.BindingContext(SearchDatabaseWinShippers1, _  
"Shippers").Position.MinValue
```

```
End Sub
```

Переход по записям

```
Private Sub Button1_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button1.Click  
    Me.BindingContext(SearchDatabaseWinShippers1, "Shippers").Position = _  
        Me.BindingContext(SearchDatabaseWinShippers1, _  
            "Shippers").Position.MinValue  
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button2.Click  
    Me.BindingContext(SearchDatabaseWinShippers1, "Shippers").Position -= 1  
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button3.Click  
    Me.BindingContext(SearchDatabaseWinShippers1, "Shippers").Position += 1  
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button4.Click  
    Me.BindingContext(SearchDatabaseWinShippers1, "Shippers").Position = _  
        Me.BindingContext(SearchDatabaseWinShippers1, _  
            "Shippers").Position.MaxValue  
End Sub
```

Что есть .NET Web приложение?

- ASP.NET проект
- .aspx это файл со страницей Web, которая состоит из Web-формы
- Приложение должно находиться на сервере приложений IIS (MS Internet Information Services).
- Страницы можно открыть с помощью любого браузера

Отличие ASP.NET от ASP (динамических серверных страниц)

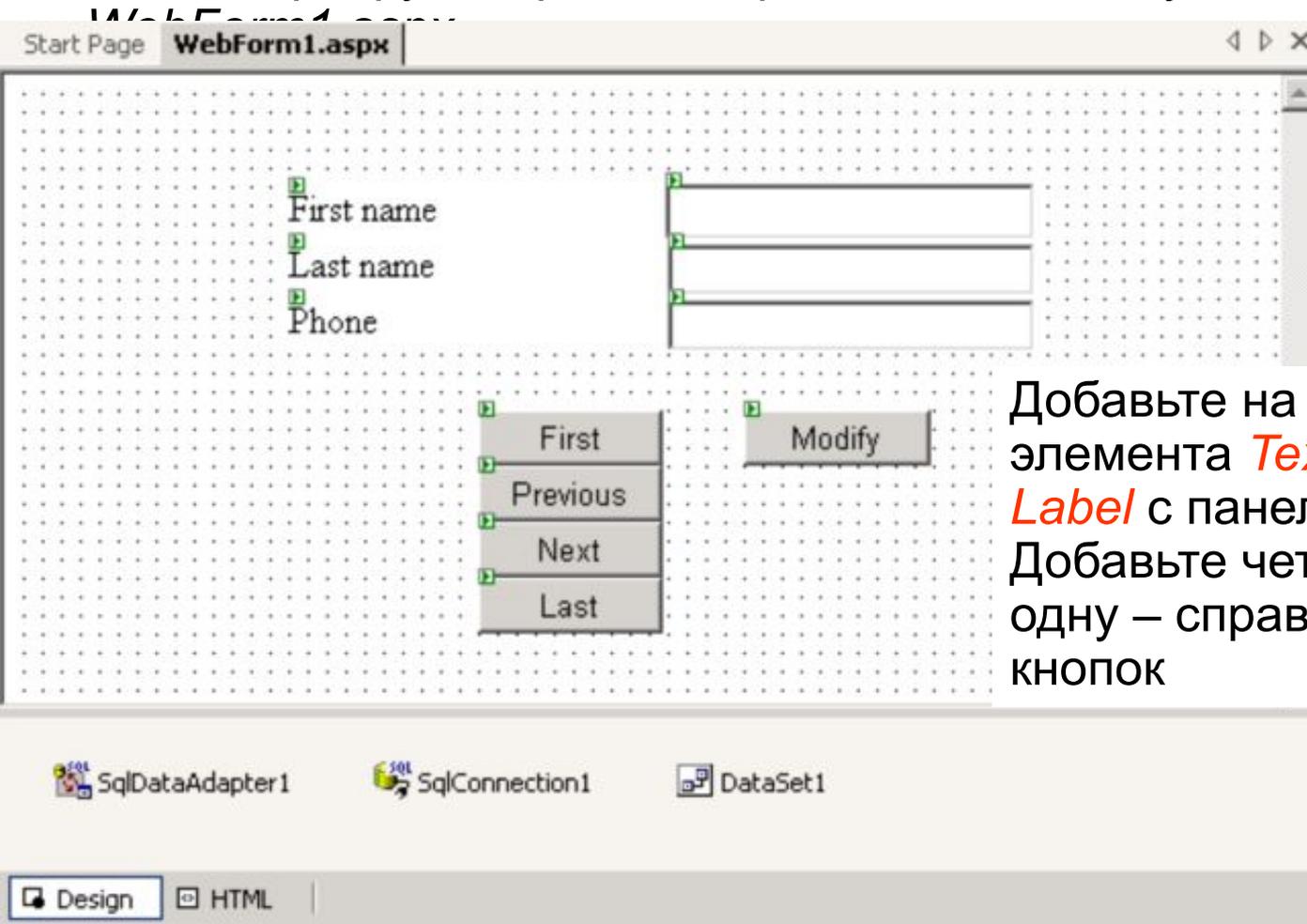
- ASP.NET и ASP могут выполняться на одном и том же сервере
- ASP.NET страницы компилируются, поэтому они выполняются быстрее и масштабируются легче
- ASP.NET страницу можно конструировать, как Windows приложение, без ввода кода HTML and Java скрипта

Особенности применения ADO.NET с ASP.NET

- Web формы обязательно отсоединяются от источника данных в отличие от Windows форм, которые отсоединяются при выполнении определенных условий
- Заполняют элементы формы данными из БД, могут отобразить как отсоединенные данные так и модифицированный набор данных

Создание ASP.NET приложения

- На стартовой странице выполните команду FILE - New Project. Выберите ASP.NET Web Application в папке Visual Basic Projects. Введите имя папки для размещения на WEB сервере <http://localhost/SearchDatabaseWeb>, OK
- При желании можно ввести user name, password, и domain для доступа к Web серверу. Стартовая страница имеет по умолчанию имя `WebForm1.aspx`



Добавьте на форму три элемента *Textbox* и три метки *Label* с панели инструментов. Добавьте четыре кнопки и еще одну – справа от основных кнопок

Заполнение системного трей с помощью элементов ADO.NET

- Откройте окно the Server Explorer, выберите на сервере БД *Pubs*. Разверните список таблиц и перетащите на форму таблицу *Authors*.
- При этом в системный трей добавятся объекты **SqlConnection1** и **SqlDataAdapter1**
- Перетащите на форму объект **Dataset** из вкладки Data панели инструментов. Укажите его свойство *Untyped*, ОК. Это добавит объект **Dataset1** в системный трей

Программный код для страницы **WebForm1.aspx**

- В окне *Solution Explorer* (проводник решений проекта) нажмите правую кнопку на объекте *WebForm1.aspx* и выберите команду *View Code*
- Это позволит открыть окно *WebForm1.aspx.vb* с пустым событием **Page_Load** открытия страницы.
- Заполните набор данных **Dataset1** с помощью объекта **SqlDataAdapter1** и укажите на отображение первой строки набора. Отредактируйте таблицу через форму.

```
SqlDataAdapter1.Fill(DataSet1, "authors")
```

‘ отображаем первую строку набора

```
If Not Me.IsPostBack Then
```

```
    Session("MyRowID") = 0
```

```
    MoveRow()
```

```
End If
```

‘ Замена набора данных информацией со страницы

```
SqlDataAdapter1.UpdateCommand = New SqlClient.SqlCommand( _  
    "UPDATE authors SET phone = " & "@NewPhone WHERE au_fname = " & _  
    "@MyFname AND au_lname = @MyLname", SqlConnection1)
```

‘ Задание параметров @MyFname и @MyLname для операции Update

```
SqlDataAdapter1.UpdateCommand.Parameters.Add("@NewPhone",  
    SqlDbType.Char, 12, "phone")
```

```
Dim prm1 As SqlClient.SqlParameter =  
    SqlDataAdapter1.UpdateCommand.Parameters.Add( _  
    "@MyFname", SqlDbType.VarChar, 20, "au_fname")  
prm1.SourceVersion = DataRowVersion.Original
```

```
Dim prm2 As SqlClient.SqlParameter =  
    SqlDataAdapter1.UpdateCommand.Parameters.Add( _  
    "@MyLname", SqlDbType.VarChar, 40, "au_lname")  
prm2.SourceVersion = DataRowVersion.Original
```

```
Sub MoveRow()
```

```
‘ Выделенная строка
```

```
Dim MyRow As DataRow
```

```
MyRow = DataSet1.Tables("authors").Rows(Session("MyRowID"))
```

```
‘ размещение в полях формы полей таблицы
```

```
TextBox1.Text = MyRow("au_fname")
```

```
TextBox2.Text = MyRow("au_lname")
```

```
TextBox3.Text = MyRow("phone")
```

```
End Sub
```

Установка переменной Session в значение 0 для перехода на первую строку

```
Private Sub Button1_Click(ByVal sender As System.Object, _
```

```
ByVal e As System.EventArgs) Handles Button1.Click
```

```
    Session("MyRowID") = 0
```

```
    MoveRow()
```

```
End Sub
```

Перемещение на последнюю строку

```
Private Sub Button4_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button4.Click  
    Session("MyRowID") = DataSet1.Tables("authors").Rows.Count - 1  
    MoveRow()  
End Sub
```

Перемещение на следующую строку

```
Private Sub Button3_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button3.Click  
    If Session("MyRowID") < DataSet1.Tables("authors").Rows.Count - 1 Then  
        Session("MyRowID") += 1  
        MoveRow()  
    End If  
End Sub
```

Перемещение на предыдущую строку

```
Private Sub Button2_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button2.Click  
    If Session("MyRowID") > 0 Then  
        Session("MyRowID") -= 1  
        MoveRow()  
    End If  
End Sub
```

Возврат значения в набор Dataset1 и применение метода Update к объекту SqlDataAdapter1

```
Private Sub Button5_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button5.Click  
    'Point a datarow at the current row in the datatable  
    Dim MyRow As DataRow  
    MyRow = DataSet1.Tables("authors").Rows(Session("MyRowID"))  
    'Populate the phone column value in the datarow with the phone text box value  
    MyRow(3) = TextBox3.Text  
    'Invoke the update method  
    SqlDataAdapter1.Update(DataSet1, "authors")  
End Sub
```

Тестирование Web приложения

Выполнить команду Build. Выполнить на объекте WebForm1.aspx в окне Solution Explorer команду View in Browser. Протестировать кнопки (после применения кнопки Modify закрыть и открыть приложение вновь для проверки факта изменения данных)

<http://www.programmingmsaccess.com/mygb.htm>

СОТРУДНИКИ

таб_номер

ФИО

тел_дом

первая следующая предыдущая последняя

сохранить

OleDbDataAdapter1 OleDbConnection1 DataSet11

Web-форма БД Access

Address <http://localhost/DS/WebForm1.aspx>

СОТРУДНИКИ

таб_номер

ФИО

тел_дом

Private Sub **Page_Load** (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

OleDbDataAdapter1.Fill(DataSet11, "сотрудники")

If Not Me.IsPostBack Then

 Session("таб_номер") = 0

 MoveRow()

End If

' Замена набора данных информацией со страницы

' Задание параметров @таб_номер, @ФИО, и @тел_дом для операции Update

 OleDbDataAdapter1.UpdateCommand = New OleDb.OleDbCommand("UPDATE сотрудники SET тел_дом=?

WHERE таб_номер=?", OleDbConnection1)

 OleDbDataAdapter1.UpdateCommand.Parameters.Add("@тел_дом", OleDb.OleDbType.Char, 50, "тел_дом")

 Dim prm1 As OleDb.OleDbParameter = OleDbDataAdapter1.UpdateCommand.Parameters.Add("@таб_номер",

 OleDb.OleDbType.Char, 50, "таб_номер")

' указание на ID -поле

 prm1.SourceColumn = "таб_номер"

 prm1.SourceVersion = DataRowVersion.Original

 Dim prm2 As OleDb.OleDbParameter = OleDbDataAdapter1.UpdateCommand.Parameters.Add("@ФИО",

OleDb.OleDbType.Char, 50, "ФИО")

 prm2.SourceVersion = DataRowVersion.Original

 Dim prm3 As OleDb.OleDbParameter = OleDbDataAdapter1.UpdateCommand.Parameters.Add("@тел_дом",

 OleDb.OleDbType.Char, 50, "тел_дом")

 prm3.SourceVersion = DataRowVersion.Original

End Sub

Private Sub **Button1_Click** (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

‘кнопка ПЕРВАЯ запись. Нумерация записей функцией Session, начиная с нулевой (т.е. первой).

Session("таб_номер") = 0

MoveRow()

End Sub

Private Sub **Button2_Click** (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

‘кнопка ПРЕДЫДУЩАЯ запись

If Session("таб_номер") > 0 Then

Session("таб_номер") -= 1

MoveRow()

End If

End Sub

Private Sub **Button3_Click** (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

‘кнопка СЛЕДУЮЩАЯ запись

If Session("таб_номер") < DataSet11.Tables("сотрудники").Rows.Count - 1 Then

Session("таб_номер") += 1

MoveRow()

End If

End Sub

Private Sub **Button4_Click** (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

‘кнопка ПОСЛЕДНЯЯ запись

Session("таб_номер") = DataSet11.Tables("сотрудники").Rows.Count - 1

MoveRow()

End Sub

Private Sub **Button5_Click**(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

‘кнопка СОХРАНИТЬ

'Point a datarow at the current row in the datatable

Dim MyRow As DataRow

MyRow = DataSet11.Tables("сотрудники").Rows(Session("таб_номер"))

'Populate the phone column value in the datarow with the phone text box value

MyRow(2) = TextBox3.Text

'Invoke the update method

Me.OleDbDataAdapter1.Update(DataSet11)

End Sub

Sub **MoveRow**()

'Выделенная строка

Dim MyRow As DataRow

MyRow = DataSet11.Tables("сотрудники").Rows(Session("таб_номер"))

' размещение в полях формы полей таблицы

TextBox1.Text = MyRow("таб_номер")

TextBox2.Text = MyRow("ФИО")

TextBox3.Text = MyRow("тел_дом")

End Sub

ProgrammingMSAccess.com

ProgrammingMSAccess.com Guest Registry



We no longer operate a guess book at this site. However, you are invited to view the content accessible from our home page and the menu on the left.

[Home](#)



Заключение

- ADO.NET
 - работает с любой БД, соединяясь с помощью драйвера ODBC
 - Имеет специальные драйвера для SQL Server и Oracle и использует OLE DB .NET и ODBC .NET провайдеры для других типов БД
- Visual Studio .NET использует метод “drag and drop” для конструирования ADO.NET приложения
- Возможности конструктора можно расширить кодом на Visual Basic .NET