**Управление базами данных в Visual Basic 1.**

**Технология ADO.NET**

**VB.NET** позволяет отображать, анализировать и изменять информацию в базах данных. Для этой цели имеется модель доступа к данным, которая называется **ADO.NET**. Эта модель позволяет работать с мощными базами данных, созданными одной из СУБД, например, **Microsoft Access**, **Microsoft FoxPro**, **Paradox**, **Oracle** и **Microsoft SQL Server**. Далее работа с базой данных будет рассмотрена на примере базы данных СУБД **MS SQL Sever**.

В **ADO.NET** для управления информацией в базе данных необходимо:

1. Подключиться (**Tools – Connect to Database**) к базе данных.

2. Создать набор данных (**Dataset**), содержащий копию данных одной или нескольких таблиц, с которыми планируется работать.

3. Создать и связать с набором данных элементы управления **DataGridView**, предназначенные для отображения и редактирования информации в таблицах набора данных.

Для начала работы выполните следующие действия:

1. Создайте свою рабочую папку.

3. Откройте приложение **Microsoft Visual Studio**.

4. Создайте новый проект.

5. Добавьте в проект еще две формы Form2 и Form3. Для добавления каждой формы следует выполнить команду меню **Project - Add Windows Form**. В открывшемся окне **Add New Item** в поле **Templates** следует выбрать шаблон **Windows Form** и нажать на кнопку **Add**.

6. Действуя аналогично, добавьте в проект модуль **Module1**. Отличие лишь в том, что вместо шаблона **Windows Form** в этом случае следует выбрать шаблон **Module**.

7. Введите код двух инструкций модуля:

|  |
| --- |
| **Module** Module1  Dim frm2 As New Form2  Dim frm3 As New Form3  End **Module** |

Назначение каждой из этих двух инструкций состоит в том, что каждая из них создает экземпляр соответствующей формы. Экземпляр формы Form1 создается системой автоматически.

8. Задайте свойствам **Text** форм **Form1**, **Form2** и **Form3** соответствующие значения: **«Ф.И.О»**, **«Категория обучения»** и **«ВУЗ»**.

9. Поместите на форме **Form1** кнопку **Button1** и кнопку **Button2**. После запуска проекта отображается форма **Form1**. Кнопка **Button1** позволит отобразить форму **Form2**, кнопка **Button2** позволит отобразить форму **Form3**.

10. Поместите на форме **Form2** кнопку **Button1**. Эта кнопка позволит после отображения формы Form2 при необходимости скрыть ее.

11. Поместите на форме **Form3** кнопку **Button1**. Эта кнопка позволит после отображения формы Form3 при необходимости скрыть ее.

12. Свяжите с кнопками **Button1** и **Button2** формы **Form1** код:

|  |
| --- |
| Private Sub Button1\_Click(ByVal sender As System.Object, \_ ByVal e As System.EventArgs) Handles Button1.Click  Form2.Show()  End Sub  Private Sub Button2\_Click(ByVal sender As System.Object, \_ ByVal e As System.EventArgs) Handles Button2.Click  Form3.Show()  End Sub |

13. Свяжите с кнопками **Button1** каждой из форм **Form2** и **Form3** код:

|  |
| --- |
| Private Sub Button1\_Click(ByVal sender As System.Object, \_ ByVal e As System.EventArgs) Handles Button1.Click  Form1.Show()  End Sub |

* 1. **Соединение с базой данных**

Для соединения с базой данных нужно:

1. Выполнить команду меню **Tools – Connect to Database**.

2. В открывшемся окне **Choose Data Source** выберите **Microsoft SQL Server**, **Data Provider** - **.NET Framework Data Provider for SQL Server**.

3. В строке Server name введите IP-адрес сервера с базой данных. В разделе **Log in to the server** выберите **Use SQL Server Authentication**, а в строках **User name** и **Password** введите логин и пароль, которые вам выдал преподаватель. В разделе **Connect to database**, введите или выберите в строке **Select or enter a database name** название той базы данных, к которой вы хотите подключиться.

4. Выберите те таблицы и элементы базы данных, которые вы хотите использовать в проекте (в случае примера это таблицы **poss**, **kat\_obuch** и **vuz**) и нажмите OK.

**2. Отображение информации с помощью объекта DataGridView**

Элемент управления **DataGridView** отображает информацию таблицы в виде строк и столбцов. Применим три объекта **DataGridView** для отображения таблицы **poss**, таблицы **kat\_obuch**, и таблицы **vuz** обучения базы данных (**MPEI13** в случае примера).

1. Разместите на каждой из трех форм элемент **DataGridView**. Затем, щелкните по стрелке в правом верхнем углу элемента и в списке **Data Source** выберите соответствующую таблицу. В результате этих действий:

* на форме **Form1** появится навигатор **possBindingNavigator**, а также сетка данных **possDataGridView**;
* на форме **Form2** появится навигатор **kat\_obuchBindingNavigator**, а также сетка данных **kat\_obuchDataGridView**;
* на форме **Form3** появится навигатор **vuzBindingNavigator**, а также сетка данных **vuzDataGridView**.

2. Сохраните проект. Запустите приложение, нажав на кнопку **Start Debugging** панели управления. Убедитесь, что теперь уже можно видеть содержащуюся в базе данных информацию.

3. Остановите выполнение проекта. Предпримите следующие меры для повышения удобства дальнейшей работы с базой данных:

3.1. Скройте поле **Code** во всех трех таблицах. Значение кода, который является первичным ключом таблицы, устанавливается базой данных автоматически. В его отображении нет необходимости. Чтобы скрыть поле таблицы, нужно выделить связанный с ней объект **DataGridView**, раскрыть стрелку в правом верхнем углу этого объекта, щелкнуть на ссылке **Edit Columns**, выбрать название столбца **Code**, свойству **Visible** задать значение **False** и закрыть окно **Edit Columns** кнопкой ОК. Проделайте это для всех трех объектов **DataGridView**.

3.2. Сделайте так, чтобы в столбцах таблицы **poss** вместо системных имен отображались понятные стороннему пользователю названия. Они должны называться **Ф.И.О.**, **Категория обучения** и **ВУЗ**. Выделите связанный с таблицей объект **DataGridView**, раскройте стрелку в правом верхнем углу этого объекта, щелкните на ссылке **Edit Columns**, выберите столбец и его свойству **HeaderText** задайте соответствующее значение. Аналогичные действия проведите с остальными заголовками.

4. Чтобы выпадающий список фамилий был отсортирован по алфавиту задайте свойству **Sort** объекта **fioBindingSource** формы **Form1** значение Фамилия.

5. Для проверки последствий сделанных изменений запустите проект и обратите внимание на отображение таблицы.

6. Остановите выполнение проекта.

**4. Фильтрация данных**

Фильтровать информацию – это иметь возможность отбирать в базе данных информацию, удовлетворяющую некоторым требованиям. Например, нужно увидеть всех студентов определенного ВУЗа. Для выполнения фильтрации данных необходимо:

1. Поместить над каждым полем таблицы, по которому предполагается выполнять фильтрацию, текстовое поле **TextBox** или комбинированный список **ComboBox**. Это фильтры, которые помещены над соответствующим полем таблицы, по которому предполагается выполнение фильтрации данных.

2. Связать объект **ComboBox1** с источником данных, откуда будет взят выпадающий список. Для этого выделите этот объект. Щелчком на стрелке, появившейся над правым верхним углом, откройте окно **ComboBox Tasks**. В этом окне поставьте галочку **Use data bound items**. В открывшемся отделе **Data Binding Mode** из выпадающих списков следует задать значения полей **Data Source**, **Display Member** и **Value Member**. При задании из выпадающего списка значения поля **Data Source** следует последовательно раскрыть узлы **OtherDataSources**, **ProjectDataSources**, **MPEI13DataSet** и выбрать таблицу **kat\_obuch**. Значения полей, которые нужно установить для объекта **ComboBox1**, приведены в таблице:

|  |  |
| --- | --- |
| **Название поля** | **Значение поля** |
| Data Source | kat\_obuchBindingSource |
| Display Member | Код |
| Value Member | Код |

3. Действуя аналогично, свяжите объект **ComboBox2** с источником данных **vuzBindingSource**. Значения полей, которые нужно установить для объекта **ComboBox2**, приведены в таблице:

|  |  |
| --- | --- |
| Название поля | Значение поля |
| Data Source | ИздательстваBindingSource |
| Display Member | Код |
| Value Member | Код |

4. Связать с каждым из этих трех объектов код:

|  |
| --- |
| Private Sub TextBox1\_TextChanged(ByVal sender As Object, \_ ByVal e As System.EventArgs) Handles TextBox1.TextChanged  If TextBox1.Text = String.Empty Then  possBindingSource.RemoveFilter()  Else  possBindingSource.Filter = "Ф.И.О. like '%" + \_ TextBox1.Text + "%'"  End If  End Sub  Private Sub ComboBox1\_SelectedIndexChanged(ByVal sender \_ As System.Object, ByVal e As  System.EventArgs) Handles \_ ComboBox1.SelectedIndexChanged  If ComboBox1.Text = String.Empty Then  possBindingSource.RemoveFilter()  Else  possBindingSource.Filter = "Категория обучения = " & \_ ComboBox1.SelectedValue.ToString  End If  End Sub    Private Sub ComboBox2\_SelectedIndexChanged(ByVal sender \_ As System.Object, ByVal e As System.EventArgs) Handles \_ ComboBox2.SelectedIndexChanged  If ComboBox2.Text = String.Empty Then  possBindingSource.RemoveFilter()  Else  possBindingSource.Filter = "ВУЗ= " & \_ ComboBox2.SelectedValue.ToString  End If  End Sub |

Этот код содержит подпрограмму – обработчика события **TextChanged** (изменение значения свойства **Text** соответствующего текстового поля). Также он содержит две подпрограммы – обработчика события **SelectedIndexChanged** (изменение значение выделенного элемента списка). Если значение свойства **Text** после изменения не является пустой строкой, то к соответствующему объекту **BindingSource** применяется метод **Filter**, который выполняет фильтрацию записей. Справа от знака равенства в обращении к методу **Filter** должна стоять строка с проверкой соответствия проверяемого объекта шаблону. В тех случаях, когда должно быть установлено соответствие шаблону лишь части проверяемого объекта, для этого применяется инструкция **Like** языка **SQL**. Когда же требуется проверка точного равенства проверяемого объекта шаблону это можно сделать без применения инструкции **Like**. Поясним синтаксис инструкции **Like**: *проверяемый объект* ***Like******‘шаблон’***. Записанная на языке **SQL** инструкция **Like** выполняет проверку соответствия проверяемого объекта шаблону. Строка в **SQL** окружается символами ‘’. Символ % означает, что на этом месте в проверяемом объекте может стоять любое количество символов.

5. Запустите проект и проверите работу фильтров. Для сброса фильтра в выпадающем списке следует выбирать пустое поле.

6. Остановите выполнение проекта.

7. Измените код подпрограммы **TextBox1\_TextChanged**. Попробуйте в этой подпрограмме обойтись без оператора **Like**. По аналогии с подпрограммамой **ComboBox1\_SelectedIndexChanged** и **ComboBox2\_SelectedIndexChanged** используйте проверку точного равенства проверяемой строки значению фильтра.

8. Снова запустите проект и проверьте работу фильтров.