

MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài thực hành : Lập trình dùng Databinding để chỉ viết code khởi tạo

I. Mục tiêu :

- Giúp SV làm quen với việc xây dựng ứng dụng dùng kỹ thuật databinding của VC# để truy xuất database mà không cần viết code truy xuất database, chỉ cần viết code khởi tạo nhỏ, nhờ đó hạn chế tối đa công sức, thời gian lập trình đồng thời hạn chế tối đa những rủi ro, lỗi của chương trình.

II. Nội dung :

- Xây dựng chương trình nhỏ cho phép xem thông tin chi tiết về tình hình mua hàng của khách hàng dùng kỹ thuật databinding để chỉ cần viết code khởi tạo nhỏ, chứ không cần viết code truy xuất database chi tiết.
- Form có giao diện như hình vẽ dưới đây.

III. Chuẩn đầu ra :

- Sinh viên nắm vững và xây dựng thành thạo các form truy xuất database dùng kỹ thuật databinding mà không cần viết code truy xuất database, chỉ cần viết code khởi tạo nhỏ.

IV. Qui trình :

Bài thực hành này sẽ viết chương trình có form giao diện sử dụng như sau :


The screenshot shows a Windows application window titled "Xem thông tin đơn hàng". The window has a blue title bar with standard Windows controls. Inside, there's a form with the following elements:

- Buttons: "Tới" (Go) and "Lùi" (Back).
- Fields:
 - Tên khách hàng (Customer Name): A dropdown menu showing "Alfreds Futterkiste".
 - Thông tin liên hệ (Contact Information): A text box containing "Maria Anders".
 - Số Phone (Phone Number): A text box containing "030-0074321".
 - Số Fax (Fax Number): A text box containing "030-0076545".
- Table: "Danh sách các đơn đặt hàng của khách hàng đang được chọn:" (List of orders for the selected customer). The table has columns: OrderID, CustomerID, EmployeeID, OrderDate, and RequiredDate. It contains four rows of data.

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10643	ALFKI	6	8/25/1997	9/22/1997
10692	ALFKI	4	10/3/1997	10/31/1997
10702	ALFKI	4	10/13/1997	11/24/1997
10835	ALFKI	1	1/15/1998	2/12/1998
- Table: "Chi tiết của đơn đặt hàng đang được chọn:" (Details of the selected order). The table has columns: OrderID, ProductID, UnitPrice, Quantity, and Discount. It contains three rows of data.

OrderID	ProductID	UnitPrice	Quantity	Discount
10643	28	45.6	15	0.25
10643	39	18	21	0.25
10643	46	12	2	0.25

- Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
- Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Windows, chọn icon "Windows Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (td. SellView), click button OK để tạo Project theo các thông số đã khai báo.

3. Form đầu tiên của ứng dụng đã hiển thị trong cửa sổ thiết kế, việc thiết kế form là quá trình lặp 4 thao tác tạo mới/xóa/hiệu chỉnh thuộc tính/tạo hàm xử lý sự kiện cho từng đối tượng cần dùng trong form.
4. Nếu cửa sổ ToolBox chưa hiển thị chi tiết, chọn menu View.Toolbox để hiển thị nó (thường nằm ở bên trái màn hình). Click chuột vào button  (Auto Hide) nằm ở góc trên phải cửa sổ ToolBox để chuyển nó về chế độ hiển thị thường trực. Thay đổi kích thước của form lớn ra theo yêu cầu.
5. Duyệt tìm phần tử Button (trong nhóm Common Controls), chọn nó, dời chuột về vị trí thích hợp trong form và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính Text = "Tôi" và thuộc tính (Name) = btnToi.
6. Lặp lại bước 5 để vẽ Button thứ 2 với thuộc tính Text = "Lùi" và thuộc tính (Name) = btnLui.
7. Duyệt tìm phần tử Label (trong nhóm Common Controls), chọn nó, dời chuột về vị trí thích hợp trong form và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính Text = "Tên khách hàng :".
8. Lặp lại bước 7 để vẽ 3 Label còn lại với thuộc tính Text tuần tự là "Địa chỉ liên hệ :", "Số Phone :", "Số Fax :".
9. Duyệt tìm phần tử ComboBox (trong nhóm Common Controls), chọn nó, dời chuột về bên phải Label "Tên khách hàng :" và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính (Name) = cbCust.
10. Duyệt tìm phần tử TextBox (trong nhóm Common Controls), chọn nó, dời chuột về vị trí bên phải Label "Địa chỉ liên hệ :" và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính (Name) = txtContact.
11. Lặp lại bước 10 để vẽ 2 TextBox còn lại với thuộc tính (Name) tuần tự là txtPhoneNo, txtFaxNo.
12. Duyệt tìm phần tử GroupBox (trong nhóm Common Controls), chọn nó, dời chuột về vị trí ngay dưới Label "Số Fax :" và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính Text = "Danh sách các đơn đặt hàng của khách hàng đang được chọn :".
13. Duyệt tìm phần tử DataGridView (trong nhóm Data), chọn nó, dời chuột vào trong GroupBox vừa vẽ và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính (Name) = grdOrders.
14. Lặp lại bước 12 và 13 để vẽ GroupBox có thuộc tính Text = "Chi tiết của đơn đặt hàng đang được chọn :" và DataGridView bên trong có thuộc tính (Name) = grdOrderDetails.
15. Tạo hàm xử lý sự kiện cho 2 button btnToi và btnLui rồi viết code cho chúng như sau :

```
//hàm xử lý Click chuột trên button "Tôi"
private void btnToi_Click(object sender, EventArgs e) {
    CurrencyManager cm = (CurrencyManager)this.BindingContext [dsView, "Customers"];
    //nếu không phải khách hàng cuối thì tiến tới 1 khách hàng
    if (cm.Position < cm.Count - 1) cm.Position++;
}
//hàm xử lý Click chuột trên button"Lùi"
private void btnLui_Click(object sender, EventArgs e) {
    //nếu không phải khách hàng đầu tiên thì lùi 1 khách hàng
    if (this.BindingContext[dsView, "Customers"].Position > 0)
        this.BindingContext[dsView, "Customers"].Position--;
}
```
16. Thêm lệnh using sau vào đầu file đặc tả class Form :

```
using System.Data.OleDb;
```

17. Thêm các lệnh định nghĩa các thuộc tính dữ liệu cần dùng sau đây vào ở vị trí đầu class đặc tả Form :

```
//định nghĩa các thuộc tính dữ liệu cần dùng
private String connectionString;
private DataViewManager dsView;
private DataSet ds;
private OleDbConnection cn;
```

18. Hiệu chỉnh lại hàm constructor của Form để có nội dung như sau :

```
public Form1() {
    InitializeComponent();
    //xây dựng chuỗi đặc tả database cần truy xuất
    connectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source =
c:\\NorthWind.mdb;";
    //tạo đối tượng Connection đến database
    cn = new OleDbConnection(connectionString);
    //tạo đối tượng DataSet
    ds = new DataSet("CustOrders");
    //tạo đối tượng DataAdapter quản lý danh sách các khách hàng
    OleDbDataAdapter da1 = new OleDbDataAdapter
        ("SELECT * FROM Customers", cn);
    //ánh xạ Tablename "Table" tới bảng dữ liệu "Customers"
    da1.TableMappings.Add("Table", "Customers");
    //chứa bảng Customers vào Dataset
    da1.Fill(ds);
    //tạo đối tượng DataAdapter quản lý danh sách các đơn đặt hàng
    OleDbDataAdapter da2 = new OleDbDataAdapter
        ("SELECT * FROM Orders", cn);
    //ánh xạ Tablename "Table" tới bảng dữ liệu "Orders"
    da2.TableMappings.Add("Table", "Orders");
    // chứa bảng Orders vào Dataset
    da2.Fill(ds);
    //tạo đối tượng DataAdapter quản lý danh sách các mặt hàng
    OleDbDataAdapter da3 = new OleDbDataAdapter
        ("SELECT * FROM [Order Details]", cn);
    //ánh xạ Tablename "Table" tới bảng dữ liệu "Orders"
    da3.TableMappings.Add("Table", "OrderDetails");
    // chứa bảng [Orders Details] vào Dataset
    da3.Fill(ds);
    //thiết lập quan hệ "RelCustOrd" giữa bảng Customers và Orders
    System.Data.DataRelation relCustOrd;
    System.Data.DataColumn colMaster1;
    System.Data.DataColumn colDetail1;
    colMaster1 = ds.Tables["Customers"].Columns["CustomerID"];
    colDetail1 = ds.Tables["Orders"].Columns["CustomerID"];
    relCustOrd = new System.Data.DataRelation
        ("RelCustOrd", colMaster1, colDetail1);
    //"add" quan hệ vừa tạo vào dataSet
    ds.Relations.Add(relCustOrd);
    //thiết lập quan hệ "relOrdDet" giữa bảng Orders & [Order Details]
    System.Data.DataRelation relOrdDet;
    System.Data.DataColumn colMaster2;
    System.Data.DataColumn colDetail2;
```

```

colMaster2 = ds.Tables["Orders"].Columns["OrderID"];
colDetail2 = ds.Tables["OrderDetails"].Columns["OrderID"];
relOrdDet = new DataRelation("RelOrdDet",colMaster2,colDetail2);
// "add" quan hệ vừa tạo vào dataSet
ds.Relations.Add(relOrdDet);
// Xác định DataViewManager của DataSet.
dsView = ds.DefaultViewManager;
// thiết lập Databinding giữa database với 2 DataGridView
grdOrders.DataSource = dsView;
grdOrders.DataMember = "Customers.RelCustOrd";
grdOrderDetails.DataSource = dsView;
grdOrderDetails.DataMember = "Customers.RelCustOrd.RelOrdDet";
// thiết lập Databinding giữa database với ComboBox
cbCust.DataSource = dsView;
cbCust.DisplayMember = "Customers.CompanyName";
cbCust.ValueMember = "Customers.CustomerID";
// thiết lập Databinding giữa database với 3 Textbox
txtContact.DataBindings.Add("Text",dsView,"Customers.ContactName");
txtPhoneNo.DataBindings.Add("Text",dsView,"Customers.Phone");
txtFaxNo.DataBindings.Add("Text",dsView,"Customers.Fax");
}

```

19. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Lúc đầu, form sẽ hiển thị thông tin về khách hàng đầu tiên trong bảng, khi bạn click vào button "Tới" hay "Lùi", thông tin khách hàng tương ứng sẽ tự được hiển thị. Bạn cũng có thể chọn 1 khách hàng tùy ý trong ComboBox "Tên khách hàng :" để chương trình tự hiển thị thông tin chi tiết về khách hàng đó.
20. Tóm lại Databinding trong VC# giúp ta giảm nhẹ rất nhiều công sức viết chương trình truy xuất database : chúng ta chỉ viết đoạn code thiết lập databinding giữa các đối tượng giao diện với dữ liệu tương ứng trong database chứ chúng ta không cần viết đoạn code cập nhật nội dung của các phần tử giao diện theo sự biến động của database, chúng ta cũng không cần viết code cập nhật database theo nội dung mà người dùng thay đổi trên các đối tượng giao diện.