

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG TRUNG CẤP KINH TẾ - KỸ THUẬT BÌNH THUẬN

GIÁO TRÌNH
MÔ ĐUN: LẬP TRÌNH WEB
NGÀNH: LẬP TRÌNH MÁY TÍNH
TRÌNH ĐỘ: TRUNG CẤP

*(Ban hành kèm theo Quyết định số:/QĐ-TC .ngày....tháng...năm 202...
của Trường Trung cấp Kinh tế - Kỹ thuật Bình Thuận)*

Bình Thuận, năm 2023

(Lưu hành nội bộ)

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Trong hệ thống kiến thức chuyên ngành trang bị cho sinh viên nghề Lập trình máy tính, mô đun góp phần cung cấp những nội dung liên quan đến việc xây dựng các ứng dụng về cơ sở dữ liệu.

Các nội dung chính được trình bày trong tài liệu này gồm các bài:

- Tổng quan về asp.net
- Web server control
- Cơ bản về lập trình c# lập trình trong trang asp.net
- Sql server
- Truy cập và xử lý cơ sở dữ liệu với .net
- Các điều khiển liên kết dữ liệu

Bên cạnh đó, giáo trình cũng không thể tránh khỏi những sai sót nhất định. Nhóm tác giả rất mong được nhận những ý kiến đóng góp, phản hồi từ quý đồng nghiệp, các bạn, người học và bạn đọc.

Trân trọng cảm ơn.

Bình Thuận, ngày tháng năm 202

Tham gia biên soạn

1.
2.
3.
4.
5.

MỤC LỤC

TRANG

Bài 1. Tổng quan về ASP.NET.....	Error! Bookmark not defined.
1. Tổng quan về lập trình ứng dụng Web.....	5
1.1. HTTP và HTML. Nền móng của Kỹ thuật lập trình web	5
1.2. Cơ bản về HTML	5
1.3. Cơ bản về CSS.....	7
2. Web Server.....	12
2.1. Internet Information Services.....	12
2.2. Cài đặt Web Server	13
2.3. Cấu hình Internet Information Services	15
2.4. Tạo các ứng dụng web trên IIS	16
3. Tạo ứng dụng Web đầu tiên.....	17
3.1. Khởi động MS Visual Studio .Net	17
3.2. Tạo mới ứng dụng Web	18
Bài 2. WEB SERVER CONTROL.....	20
1. HTML Control	20
2. ASP.Net Web Control	22
Bài 3. CƠ BẢN VỀ LẬP TRÌNH ASP.NET	34
1. Khai báo biến	34
2. Mảng	34
3. Làm việc với mảng (Working with Arrays)	35
Bài 4. SQL SERVER	36
1. Tổng quan về SQL	36
2. Tổng quan về CSDL quan hệ.....	41
3. Câu lệnh truy vấn	43
Bài 5. TRUY CẬP VÀ XỬ LÝ CƠ SỞ DỮ LIỆU VỚI.NET.....	45
1. Tổng quan về ADO.NET.....	45
2. Phương thức của đối tượng Connection.....	46
3. Đối tượng SqlCommand.....	48
Bài 6. CÁC ĐIỀU KHIỂN LIÊN KẾT DỮ LIỆU	51
1. Điều khiển DataGrid.....	51
2. Điều khiển DataList.....	53
3. Điều khiển Repeater	54

GIÁO TRÌNH MÔ ĐUN LẬP TRÌNH WEB

Tên mô đun: LẬP TRÌNH WEB

Mã mô đun: MĐ21

Thời gian thực hiện mô đun: 90 giờ (Lý thuyết: 25 giờ; Thực hành: 61 giờ, Kiểm tra: 4 giờ)

I. VỊ TRÍ, TÍNH CHẤT CỦA MÔ ĐUN

** Vị trí:*

- Đây là mô đun đầu tiên lập trình trên môi trường web trong chương trình đào tạo Trung cấp lập trình máy tính, sử dụng môi trường Microsoft Visual Studio.NET làm nền tảng phát triển web. Vì vậy mô đun này trình bày các cấu trúc điều khiển của ngôn ngữ ASP.NET hiện đại thay vì sử dụng ngôn ngữ ASP cổ điển.

** Tính chất:*

- Cung cấp cho người học một kiến thức về lập trình mạng chạy trên Intranet/Internet.

II. MỤC TIÊU MÔ ĐUN

** Về kiến thức:*

- Trình bày được kiến trúc hạ tầng .NET.
- Vận dụng được cú pháp của ngôn ngữ lập trình ASP.NET.
- Sử dụng được phương pháp lập trình hướng đối tượng trong ngôn ngữ lập trình ASP.NET.
- Khai thác được các tính năng tiên tiến trong ASP.NET.
- Tạo được các ứng dụng web kết nối với cơ sở dữ liệu.
- Tạo được báo cáo.
- Lập trình mạng ở cấp độ cơ bản.

** Về kỹ năng:*

- Sử dụng được tính năng tiên tiến trong ASP.NET.
- Tạo được các ứng dụng WEB.
- Tạo được các ứng dụng cơ sở dữ liệu.
- Tạo được báo cáo.
- Lập trình mạng ở cấp độ cơ bản.

** Về tinh năng lực tự chủ và trách nhiệm:*

- Rèn luyện kỹ năng lập trình cẩn thận, chu đáo.

Rèn luyện kỹ năng suy luận logic

III. NỘI DUNG MÔ ĐUN

1. Chương trình khung

Mã MH, MĐ	Tên môn học/ mô đun	Số tín chỉ	Thời gian đào tạo (giờ)				Học kỳ			
			Tổng số	Trong đó			1	2	3	4
				Lý thuyết	Thực hành/ thực tập/ bài tập/ thảo luận	Thi/ kiểm tra				
I	Các môn học chung	15	316	116	183	17	151	165	0	0
MH01	Giáo dục Chính trị	2	30	15	13	2		30		
MH02	Pháp luật	1	15	9	5	1	15			
MH03	Giáo dục thể chất	1	30	4	24	2	30			
MH04	Giáo dục QP - AN	2	45	21	21	3		45		
MH05	Tin học	2	45	15	28	2	45			
MH06	Tiếng Anh	4	90	30	56	4		90		
MH07	GD SKSS, SK tình dục và phòng chống HIV/AIDS	1	16	7	9		16			
MH08	Kỹ năng mềm	2	45	15	27	3	45			
II	Các môn học, mô đun đào tạo bắt buộc	47	1.410	332	1.028	50	315	240	405	450
II.1	Môn học, mô đun cơ sở	10	255	72	172	11	255	0	0	0
MH09	An toàn lao động	2	30	27	1	2	30			
MĐ10	Tin học văn phòng	3	75	15	57	3	75			
MH11	Lập trình căn bản	3	90	15	72	3	90			
MĐ12	Phần cứng máy tính	2	60	15	42	3	60			
II.2	Môn học, mô đun chuyên môn	37	1.155	260	856	39	60	240	405	450
MH13	Cấu trúc dữ liệu và giải thuật	2	60	25	32	3		60		
MH14	Lập trình hướng đối tượng	2	60	25	32	3			60	
MH15	Cơ sở dữ liệu	3	90	25	62	3		90		
MĐ16	Đồ họa ứng dụng	3	90	15	72	3		90		
MH17	Mạng căn bản	2	60	25	33	2	60			
MH18	Tiếng Anh chuyên ngành	2	60	25	32	3			60	
MĐ19	Thiết kế Web	3	90	15	71	4			90	
MĐ20	Lập trình Windows	4	105	25	76	4			105	
MĐ21	Lập trình Web	3	90	25	61	4			90	
MĐ22	Lập trình quản lý	3	90	25	61	4				90
MĐ23	PHP & MySQL	3	90	15	72	3				90
MĐ24	Quản trị mạng	3	90	15	72	3				90
MĐ25	Thực tập nghề nghiệp tại cơ sở	4	180		180					180
Tổng cộng		62	1.726	448	1.211	67	466	405	405	450

2. Chương trình chi tiết mô đun

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra*
1	ASP.NET và kiến trúc.NET	8	2	6	
2	Lập trình trong môi trường.NET	10	2	8	
3	Cơ bản về ASP.NET	15	2	13	
4	Hướng đối tượng với ASP.NET	10	2	6	2
5	Những tính năng tiên tiến trong ASP.NET	8	2	6	
6	ASP.NET và các lớp cơ sở	8	2	6	
7	Truy cập và xử lý cơ sở dữ liệu với.NET	10	1	9	
8	Ajax	6	1	5	
9	Lập báo cáo với crystal report	15	1	12	2
	Cộng:	90	15	71	4

IV. ĐIỀU KIỆN THỰC HIỆN MÔ ĐUN

* *Vật liệu:*

- Xưởng thực hành đạt chuẩn.
- Các mô hình trên giấy in hoặc trên power point.

* *Dụng cụ:*

- Máy tính, máy chiếu.
- Bài giảng soạn trên phần mềm dạy học.
- Bộ phần mềm Microsoft Visual Studio .NET 2003 trở lên.

V. NỘI DUNG VÀ PHƯƠNG PHÁP ĐÁNH GIÁ

* *Về kiến thức:*

- Đánh giá thông qua bài kiểm tra trắc nghiệm hay kiểm tra viết đạt các yêu cầu sau:
- Hiểu được kiến trúc hạ tầng .NET.
- Trình bày được cú pháp của ngôn ngữ lập trình ASP.NET.
- Sử dụng được phương pháp lập trình hướng đối tượng trong ngôn ngữ lập trình ASP.NET.

* *Về kỹ năng:*

- Đánh giá thông qua bài tập thực hành đạt các yêu cầu sau:
- Sử dụng được tính năng tiên tiến trong ASP.NET.
- Tạo được các ứng dụng WEB.
- Tạo được các ứng dụng cơ sở dữ liệu.
- Tạo được báo cáo.
- Lập trình mạng ở cấp độ cơ bản.

* *Về tinh thần tự chủ và trách nhiệm:*

- Rèn luyện kỹ năng lập trình cẩn thận, chu đáo.
- Rèn luyện kỹ năng suy luận logic.

VI. HƯỚNG DẪN THỰC HIỆN MÔ ĐUN

1. Phạm vi áp dụng chương trình

- Chương trình được sử dụng để giảng dạy cho trình độ trung cấp.

2. Hướng dẫn một số điểm chính về phương pháp giảng dạy mô đun:

- Thực hiện môn học này bằng phương pháp thuyết trình, đàm thoại. Sử dụng các nguồn tài nguyên soạn trên phần mềm dạy học, giảng dạy sử dụng máy tính và máy chiếu. Chủ yếu thao tác mẫu khi thực hiện xây dựng các ứng dụng ban đầu.
- Giáo viên có thể lựa chọn một trong hai ngôn ngữ C#.NET hoặc VB.NET để giảng dạy theo nhu cầu của thị trường.

3. Những trọng tâm chương trình cần chú ý:

- Sử dụng thành thạo các cấu trúc điều khiển của ngôn ngữ Web
- Sử dụng các đối tượng ADO để kết nối cơ sở dữ liệu và thao tác dữ liệu qua các lệnh SQL.
- Xây dựng trang web có tính bảo mật như trang đăng nhập, xác thực người dùng,...

4. Tài liệu cần tham khảo:

- [1] PHẠM HỮU KHANG, *Lập Trình ASP.NET 2.0*, Nhà Xuất Bản Lao động - Xã hội, 2006
- [2] DƯƠNG QUANG THIÊN, *Lập Trình Web dùng ASP.NET và C#*, Nhà Xuất Bản Tổng Hợp TP. Hồ Chí Minh, 2007

BÀI 1. TỔNG QUAN VỀ ASP.NET

Mã bài: MD21-B01

Mục tiêu

- Trình bày được định nghĩa, cú pháp và các thành phần cơ bản của HTML;

- Sử dụng được các thành phần cơ bản của HTML.

Nội dung chính:

1. Tổng quan về lập trình ứng dụng Web

1.1. HTTP và HTML. Nền móng của Kỹ thuật lập trình web

Kỹ thuật cơ bản của lập trình ứng dụng web khởi đầu là HyperText Transfer Protocol (HTTP), đó là một giao thức cho phép các máy tính trao đổi thông tin với nhau qua mạng máy tính. HTTP được xác định qua URLs (Uniform Resource Locators), với cấu trúc chuỗi có định dạng như sau:

```
http: // <host> [: <port>] [ <path> [? <query>]]
```

Sau tiền tố http://, chuỗi URL sẽ chứa tên host hay địa chỉ IP của máy server (có thể có số cổng đi kèm), tiếp theo là đường dẫn dẫn đến tập tin server được yêu cầu. Tuy chọn sau cùng là tham số, còn được gọi là **query string** (chuỗi tham số/chuỗi truy vấn).

Vi dụ:

Phân tích địa chỉ http://www.comersus.com/comersus6/store/index.asp
Trang web index.asp được lưu trữ trong thư mục **/comersus6/store** tại Web Server với host là **www.comersus.com**

Một số thuật ngữ:

Internet: là một hệ thống gồm nhiều máy tính ở khắp nơi trên thế giới nối lại với nhau;

WWW: World Wide Web (mạng toàn cầu), thường được dùng khi nói về Internet;

Web Server: Máy tính lưu trữ các trang web;

Web Client: Máy tính dùng để truy cập các trang web;

Web Browser: Phần mềm dùng để truy cập web;

Một số web browser phổ biến: Internet Explorer, Netscape Navigator, Avant Browser, Opera,

1.2. Cơ bản về HTML

a.Tag cấu trúc

HTML bao gồm 3 tag để xác định cấu trúc của trang web bao gồm:

```
<html></html> <head></head> <body></body>
```

b.Tag định dạng văn bản

Mặc dù có rất nhiều tag để định dạng văn bản, những tag sau đây là những tag cơ bản nhất mà gần như bất cứ một trang web nào cũng phải sử dụng:

<p>: bắt đầu một đoạn văn

bản mới
: xuống dòng

<h1></h1>, <h2></h2>,....: đặt dòng văn bản nằm trong cặp tag là tiêu đề (heading).

c.Tag ghi chú

Cũng như các ngôn ngữ lập trình, để cho phép người viết trang web đặt những ghi chú dành riêng cho mình vào trong trang web, HTML cung cấp tag ghi chú. Đây là tag đặc biệt so với những tag khác:

```
<!-- nội dung ghi chú -->
```

d.Định dạng kiểu chữ

Trong các tài liệu, văn bản chúng ta thường sử dụng các kiểu chữ **đậm**, nghiêng, gạch dưới,... ví dụ sau minh họa các tag được dùng định dạng kiểu chữ:

```
<html> <body>
```

```
<b>In đậm</b> <br>
```

```
<strong>In rất đậm </strong><br>
```

```
<font size="4">c</font><big>hữ lớn </big><br> <em>nhân  
mạnh</em><br>
```

```
<i>in nghiêng </i><br>
```

e. Font chữ, màu sắc và canh lề

```
<font face="..." size="..." c
```

```
olor="#HHHHHH">...</font>
```

```
<p align="left/right/center">
```

Ví dụ:

```
<p align="center">
```

```
<font face="Algerian" size="5">Computer Joke </font> <p><u>Kỹ thuật  
viên</u>:
```

```
<font face="Arial">Máy tính của anh có ổ đĩa mềm chứ ?</font>
```

```
<p><u>Khách</u>:
```

```
<i><font face="Times New Roman">Tôi không nhìn thấy bên trong. Có
```

chữ " Intel PentiumInside
";</i>
</p>

Thuộc tính của một tag

Một thông tin định dạng có thể gồm nhiều chi tiết, trong ví dụ trên, font chữ sẽ hiển thị cho một chuỗi văn bản được chỉ định qua tag tuy nhiên, font chữ lại gồm nhiều chi tiết như: tên font, kích thước, màu sắc,...

Các thông tin chi tiết được gọi là các thuộc tính của tag. Một tag có thể có nhiều thuộc tính. Bạn nên đặt giá trị của thuộc tính trong dấu ngoặc kép.

Định dạng trước nội dung văn bản Web browser sẽ không quan tâm đến cách bạn trình bày đoạn code HTML trong file .html mà chỉ dựa vào các tag để trình bày nội dung trang web.

Tag <pre> được dùng khi bạn muốn yêu cầu web browser "tôn trọng" các khoảng trắng và xuống dòng trong đoạn code HTML của mình.

1.3. Cơ bản về CSS

1.3.1. Giới thiệu CSS

a. CSS là gì

CSS: Cascading Style Sheets

Các Style định nghĩa cách trình duyệt hiển thị các đối tượng HTML

Các Style được lưu trong Style Sheet

Các Style Sheet độc lập được lưu trong file CSS riêng biệt

Các Style Sheet độc lập có thể tiết kiệm nhiều thời gian cho bạn

Nhiều định nghĩa Style cho cùng một loại đối tượng sẽ được sử dụng theo lớp.

b. Style giúp bạn giải quyết nhiều vấn đề

HTML tag được thiết kế để định dạng cách hiển thị nội dung của một trang Web bằng cách định nghĩa như "đây là phần header", "đây là một đoạn", "đây là một bảng",... Mỗi trình duyệt hiển thị nội dung trang Web theo cách riêng của mình dựa trên những định nghĩa đó.

Các trình duyệt thông dụng như Internet Explorer hay Netscape liên tục thêm thắt các tag HTML mới của riêng mình vào danh sách các HTML tag chuẩn của W3C làm cho việc tạo lập các văn bản Web để hiển thị độc lập trên mọi trình duyệt ngày càng khó khăn.

Để giải quyết vấn đề này, W3C (World Wide Web consortium- tổ chức chịu trách nhiệm tạo lập các chuẩn trên Web) tạo ra các STYLE cho HTML 4.0

Cả Netscape 4.0 và Internet Explorer 4.0 đều hỗ trợ Cascading Style Sheets.

1. Style Sheet tiết kiệm nhiều công sức thiết kế

Các Style trên HTML 4.0 định nghĩa cách mà các thành phần HTML được hiển thị. Các Style thường được lưu trong các file độc lập với trang Web của bạn. Các file CSS độc lập cho phép bạn thay đổi hình thức thể hiện và khuôn dạng của tất cả các trang trong Website thống nhất mà chỉ phải thực hiện thay đổi một lần.

2. Style nào sẽ được dùng?

Ta có thể nói rằng, các Style sẽ được sử dụng theo "lớp" (cascade) ưu tiên khi nhiều Style định nghĩa một thành phần HTML được tham chiếu trong một file HTML. Thứ tự ưu tiên được sắp xếp từ cao xuống thấp:

Style cho thành phần HTML cụ thể

Style trong phần HEAD

Style trong file CSS

Mặc nhiên theo trình duyệt

1.3.2. Cú pháp CSS

1. Lý thuyết liên quan:

Cú pháp của CSS gồm 3 phần: đối tượng, thuộc tính và giá trị:

Đối tượng { thuộc tính: giá trị }

Đối tượng thường là các tag HTML mà bạn muốn định nghĩa cách hiển thị. Thuộc tính là thuộc tính hiển thị của đối tượng đó. Giá trị là cách mà bạn muốn một thuộc tính hiển thị như thế nào. Cặp { thuộc tính: giá trị } được đặt trong dấu { }.

Body { color: black }

Nếu giá trị gồm nhiều từ, đặt chúng trong dấu nháy đôi:

p { font-family: "sans serif" }

Nếu bạn muốn định nghĩa nhiều thuộc tính của một đối tượng, phân cách các cặp thuộc tính: giá trị bằng dấu (;).

p { text-align: center; color: red }

Để định nghĩa Style được dễ đọc hơn:

```
P {  
text-align:  
center; color: black;  
font-family:  
arial }
```

1.1. Nhóm nhiều đối tượng

Bạn có thể định nghĩa một Style cho nhiều đối tượng cùng một lúc:

```
h1, h2, h3, h4, h5, h6  
{ color: green  
}
```

1.2. Thuộc tính Class

Với thuộc tính Class, bạn có thể định nghĩa nhiều Style khác nhau cho cùng một đối tượng. Ví dụ, bạn muốn có hai Style cho cùng một tag <P>, nếu tag <P> nào có class=right sẽ canh lề bên phải, class=center sẽ canh giữa:

```
p.right {text-align:  
right} p.center {text-align:  
center}
```

Trong trang HTML:

```
<p class="right">
```

Đoạn này sẽ được canh
phải. </p>

```
<p class="center">
```

Đoạn này sẽ được canh giữa.

```
</p>
```

Bạn cũng có thể bỏ qua tên đối tượng để định nghĩa kiểu Style cho tất cả các thành phần có Class mà bạn định nghĩa. Ví dụ:

```
.center {  
text-align:  
center; color: red  
}
```

Trong trang HTML sau, cả H1 và đoạn văn bản đều được canh giữa:

```
<h1 class="center">
```

Tiêu đề này sẽ được canh
giữa. </h1>

```
<p class="center">
```

Đoạn này sẽ được canh giữa. </p>

1.3. Thuộc tính ID

Thuộc tính ID có thể dùng định nghĩa Style theo hai cách:

Tất cả các thành phần HTML có cùng một ID.

Chỉ một thành phần HTML nào đó có ID được định nghĩa.

Ví dụ sau, Style dùng cho tất cả các thành phần HTML có ID là "intro":

```
#intro {  
  font-  
size:110%; font-  
weight:bold;  
color:#0000ff;  
  background-  
color:transparent }
```

Ví dụ sau, Style chỉ dùng cho thành phần <P> nào có ID là "intro" trong trang Web.

```
p#intro {  
  font-  
size:110%;  
font-  
weight:bold;  
color:#0000ff;  
  background-  
color:transparent }
```

1.4. Ghi chú trong CSS

CSS dùng cách ghi chú tương tự như ngôn ngữ C: các đoạn ghi chú bắt đầu bằng /* và kết thúc bởi */. Ví dụ:

```
/* Đây là phần ghi chú */  
p {  
  text-align: center;  
  /* Đây là phần ghi chú  
  */ color: black;  
font-family: arial }
```

2. Trình tự thực hiện:

Bước 1: Tạo file style.css

Bước 2: Viết hàm css cho thẻ cần làm

Bước 3: Chạy thử kiểm tra xem được chưa

1.3.3. Sử dụng CSS trong trang HTML

1. Lý thuyết liên quan:

1.1. Làm thế nào chèn vào một Style Sheet

Khi trình duyệt đọc một Style, nó sẽ định dạng nội dung trang Web theo

Style đó. Có 3 cách để sử dụng Style trong một trang HTML.

1.2. Dùng file CSS riêng

File CSS độc lập nên dùng khi Style được áp dụng cho nhiều trang. Mỗi trang sử dụng Style định nghĩa trong file CSS sẽ phải liên kết đến file đó bằng tag <link> đặt trong phần HEAD:

```
<head>
  <link rel="stylesheet" type="text/css"
href="tên_file.css" >
</head>
```

Ví dụ một file CSS: **Style.css**

```
hr {color: sienna}
```

```
p {margin-left: 20px}
```

```
body {background-image: url("images/back40.gif")}
```

1.3. Định nghĩa các Style trong phần HEAD

Các Style định nghĩa trong phần HEAD có thể dùng cho nhiều thành phần HTML trong trang Web đó. Bạn sử dụng tag <Style> để định nghĩa Style:

```
<head>
  <style
type="text/css">      hr
  {color: sienna}
  p {margin-left: 20px}
  body
  {background-image: url("images/back40.gif")}
</style>
</head>
```

Ghi chú: Trình duyệt thường bỏ qua các tag HTML mà nó không biết, do đó để các trình duyệt không hỗ trợ CSS không hiển thị phần định nghĩa Style, bạn nên đặt trong tag ghi chú của HTML: <!-- ... -->

```
<head>
  <style type="text/css">
  <!--
  hr {color: sienna}
  p {margin-left: 20px}
  body {background-image: url("images/back40.gif")} -->
</style> </head>
```

1.4. Dùng Style cho một thành phần HTML cụ thể

Style cho một tag HTML cụ thể gần như không tận dụng được các lợi điểm của CSS ngoại trừ cách hiển thị đối tượng. Bạn dùng thuộc tính Style để định nghĩa Style cho thành phần HTML.

```
<p style="color: sienna; margin-left:
20px"> Đây là đoạn văn bản
</p>
```

1.5. Nhiều Style cho một đối tượng

Nếu một đối tượng được định nghĩa nhiều Style, nó sẽ sử dụng Style cụ thể nhất. Ví dụ, một file CSS định nghĩa tag H3 như sau:

```
h3 {
    color: red;
    text-
align: left;
font-size: 8pt
}
```

2. Trình tự thực hiện:

Bước 1: Tạo file style.css

Bước 2: Viết hàm css cho thẻ cần làm

Bước 3: Chạy thử kiểm tra xem được chưa

Bước 4: Trong thẻ <head> của file html thêm đoạn sau

```
<link rel="stylesheet" type="text/css" media="screen" href="style.css">
```

Bước 5: Chạy thử và kiểm tra xem được chưa

3. Sai hỏng thường gặp, nguyên nhân và biện pháp phòng tránh/ khắc phục

a) Sai hỏng thường gặp

CSS không chạy

b) Nguyên nhân

Sai cấu trúc

Gọi file css sai

c) Biện pháp phòng tránh/ khắc phục

Kiểm tra lại cấu trúc file css

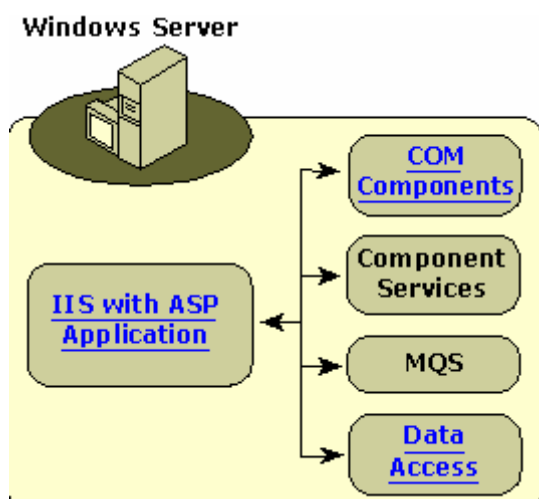
Kiểm tra lại dòng lệnh gọi style.css

2. Web Server

2.1. Internet Information Services

IIS có thể được sử dụng như một Web server, kết hợp với ASP để xây

dựng các ứng dụng Web tận dụng các điểm mạnh của Server-side Script, COM component, ... theo mô hình Client/Server.



IIS có rất nhiều phiên bản, đầu tiên được phát hành rời trong bản Service pack của WinNT.

– Các phiên bản Windows 2000 đã có tích hợp IIS 5.0. – Windows XP tích hợp IIS 5.5

– Windows XP .NET Server tích hợp IIS 6 hỗ trợ các tính năng dành cho .NET của ASP.NET và Web Service.

2.2. Cài đặt Web Server

1. Cài đặt Web Server trên Windows 2000/Windows XP Professional

Windows 2000 tích hợp sẵn IIS nhưng không tự động cài đặt do đó, bạn phải tự cài IIS nếu hệ thống đã được cài rồi.

Bước 1. Chọn Control Panel | Add/Remove programs.

Bước 2. Add/Remove Windows Components.

Bước 3. Đánh dấu vào mục Internet Information Services (IIS).

Bước 4. Chọn nút Details để chọn các mục chi tiết.

Bước 5. Chọn các mục cần cài đặt trong đó bạn nhớ chọn:

 f FrontPage 2000 Server Extensions

 f Internet Information Services Snap-In

 f Internet Services Manager (HTML)

Bước 6. Click OK để hệ thống tự cài đặt.

2. Cài đặt Web Server trên Windows Server 2003

Cài đặt Web Server trên Windows Server 2003 cũng tương tự như cài đặt Web Server trên Windows 2000.

Bước 1. Chọn Control Panel | Add/Remove programs. Bước 2.

Add/Remove Windows Components.

Bước 3. Đánh dấu vào Application Server.

Bước 4. Chọn nút Details để chọn các mục chi tiết.

Các bước còn lại thực hiện như trên Windows 2000 (từ bước 3 đến bước

6)

3. Kiểm tra kết quả cài đặt Web Server

Sau khi đã cài đặt IIS, bạn có thể kiểm tra xem Web Server đã làm việc hay chưa. Mở web browser (Internet Explorer) và gõ: //localhost trong phần địa chỉ. Một khi Web server đã được cài đặt, một trang web mẫu sẽ được hiển thị.

– Localhost là địa chỉ của máy cục bộ mà bạn đang làm việc. Nếu máy của bạn đang kết nối vào mạng LAN và có một địa chỉ IP, bạn có thể dùng địa chỉ này thay cho localhost.

Để xác định địa chỉ IP của máy mình: f Vào menu Start|Run và gõ lệnh: command hoặc cmd

f Trên màn hình DOS, gõ lệnh: ipconfig và xem phần IP Address

– Khi gõ //localhost, bạn sẽ thấy trong thanh địa chỉ tự động đổi thành: http://localhost. HTTP là giao thức mặc định được dùng trên Internet. Vì HTTP là một giao thức thuộc bộ TCP/IP, bạn cần có địa chỉ IP để các máy tính khác trong mạng có thể truy cập được đến trang web của bạn.

– Sau khi cài đặt Web Server, mặc định trên ổ đĩa C:\ sẽ có sẵn thư mục C:\inetpub\wwwroot. Đây là thư mục mà Web Server mặc định ánh xạ vào //localhost, do đó, các trang web đặt trong wwwroot có thể được truy cập bởi các máy tính khác.

Để kiểm tra, hãy tạo một trang web và đặt vào c:\inetpub\wwwroot. Trên IE, gõ địa chỉ: /localhost/<tên file html>

2. Trình tự thực hiện

Bước 1. Chọn Control Panel | Add/Remove programs.

Bước 2. Add/Remove Windows Components.

Bước 3. Đánh dấu vào mục Internet Information Services (IIS).

Bước 4. Chọn nút Details để chọn các mục chi tiết.

Bước 5. Chọn các mục cần cài đặt trong đó bạn nhớ chọn:

FrontPage 2000 Server Extensions

Internet Information Services Snap-In

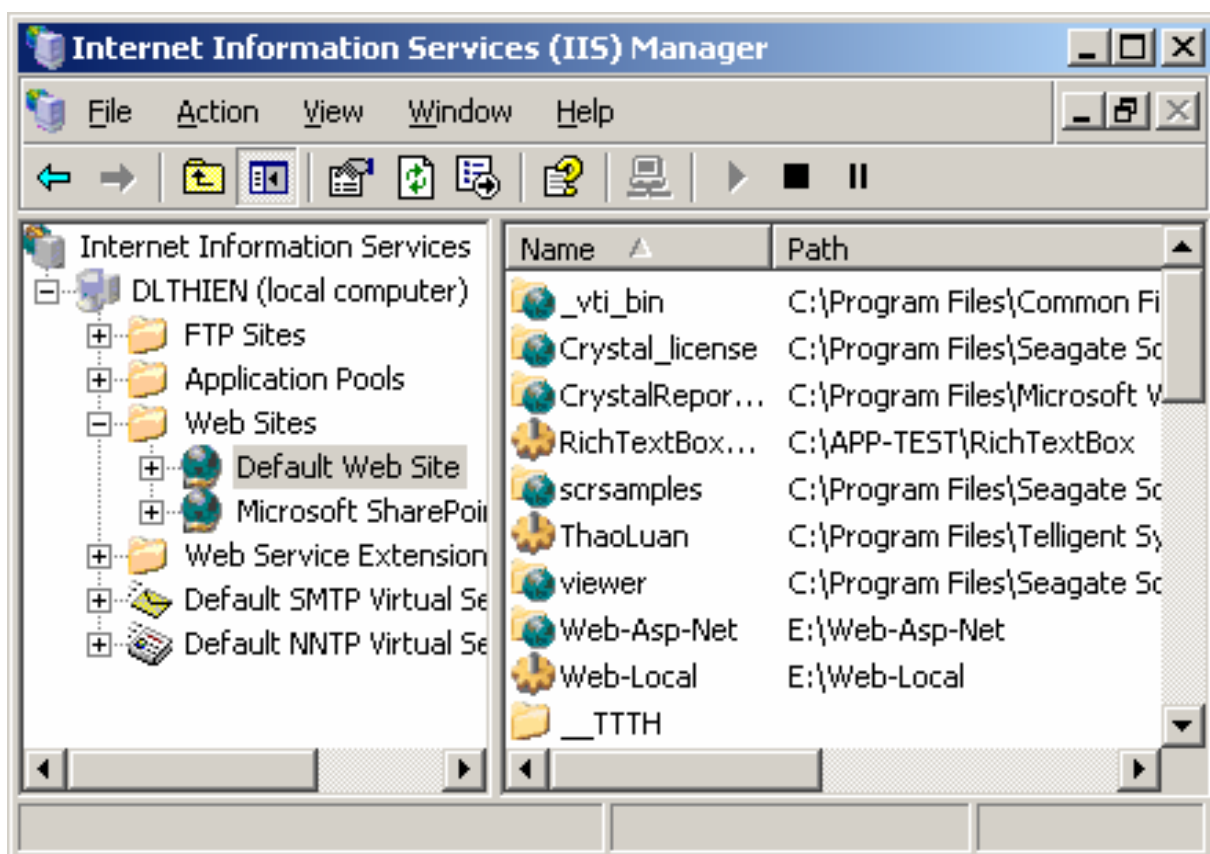
Internet Services Manager (HTML)

Bước 6. Click OK để hệ thống tự cài đặt.

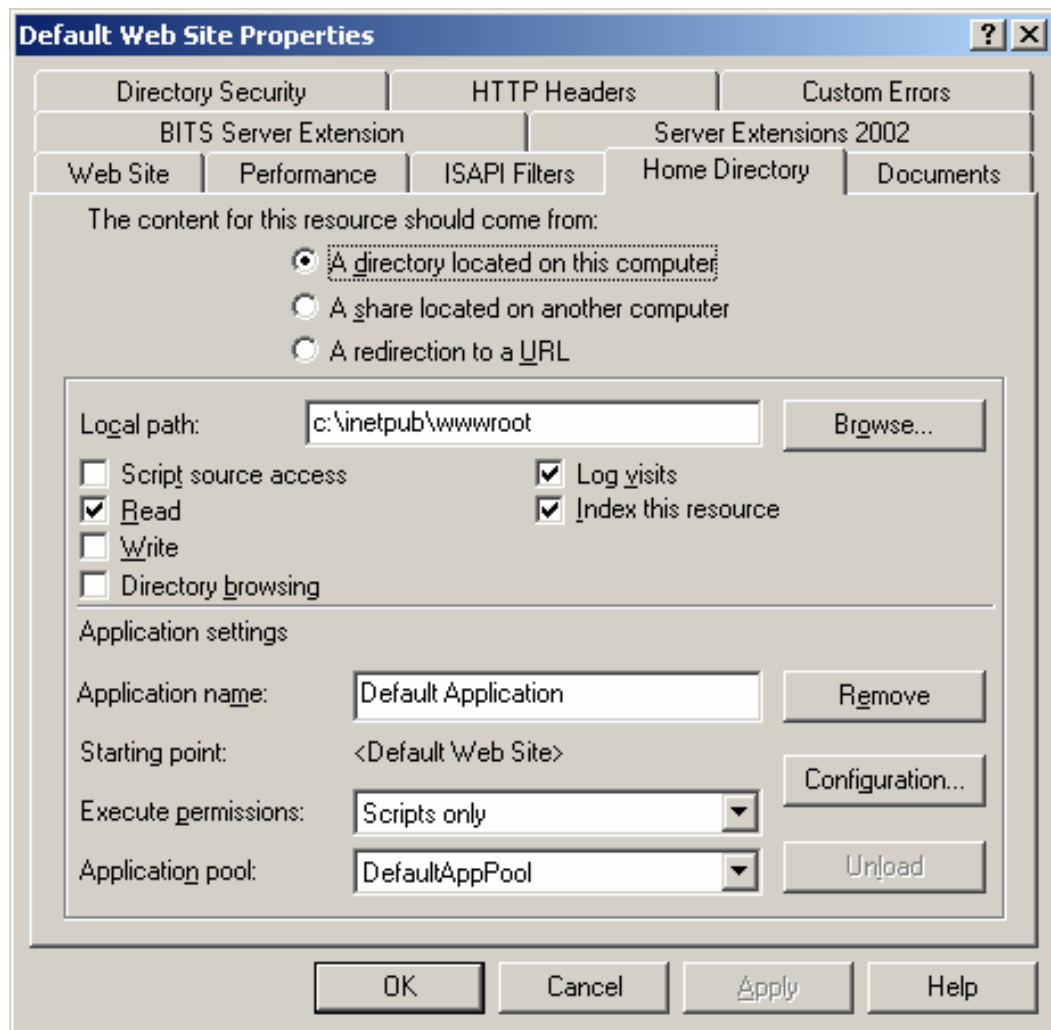
2.3. Cấu hình Internet Information Services

1. Lý thuyết liên quan:

Để cấu hình IIS, vào Control Panel| Administrative Tools|Internet Services Manager. Trên các hệ điều hành Windows 2000/XP, Microsoft sử dụng công cụ Microsoft Management Console (MMC) để làm công cụ quản lý, do đó tất cả các thao tác đều sử dụng menu ngữ cảnh bằng cách nhấp chuột phải trên mục muốn chọn. Chọn Properties của mục Default Web Site, bạn có thể xem và cấu hình lại các thông tin dành cho trang web mặc định của mình.



- Trên tab Home Directory, bạn có thể thay đổi đường dẫn đến một thư mục khác trên ổ cứng nếu muốn



– Trên tab Documents, bạn có thể đặt trang web mặc định sẽ hiển thị khi Web Browser không chỉ định trang web cụ thể. Bạn sẽ thấy index.htm và default.htm được liệt kê trong phần này. Đây là lý do tại sao khi bạn gõ //localhost thì Web browser lại hiển thị được trang homepage. Thực ra, //localhost tương đương với //localhost/index.htm hay //localhost/default.htm.

– Trong tab Directory Security, bạn có thể định lại các chế độ kiểm tra người dùng truy cập vào web site.

2.4. Tạo các ứng dụng web trên IIS

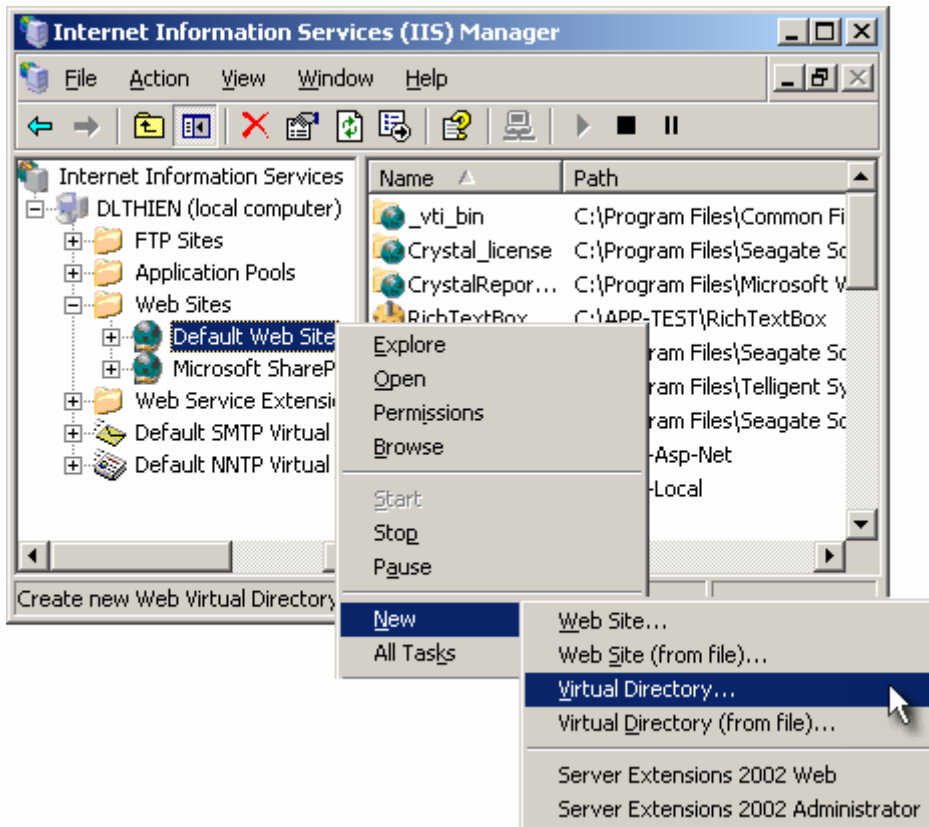
Một Web Server có thể quản lý nhiều ứng dụng Web đồng thời. Thông thường, bạn sẽ tổ chức một thư mục con trong wwwroot cho mỗi ứng dụng nhưng bạn cũng có thể tạo ánh xạ từ một thư mục khác.

– Nếu bạn đặt thư mục trong wwwroot, IIS sẽ tự động liệt kê nó trong mục Default Web Site.

– Nếu muốn tạo một thư mục nằm ngoài thư mục wwwroot thành một web site, chúng ta cần tạo Virtual Directory liên kết đến thư mục đó.

Để tạo một virtual directory:

- Chọn mục New | Virtual Directory trên menu ngữ cảnh.



- Nhập vào tên alias cho thư mục ảo.
- Chọn nút Browse để chọn thư mục muốn ánh xạ. Tiếp đó, bạn sẽ phải đặt một số cấu hình khởi đầu cho web site, những thông tin này có thể cấu hình lại tương tự như với Default Web Site ở trên

3. Tạo ứng dụng Web đầu tiên

3.1. Khởi động MS Visual Studio .Net

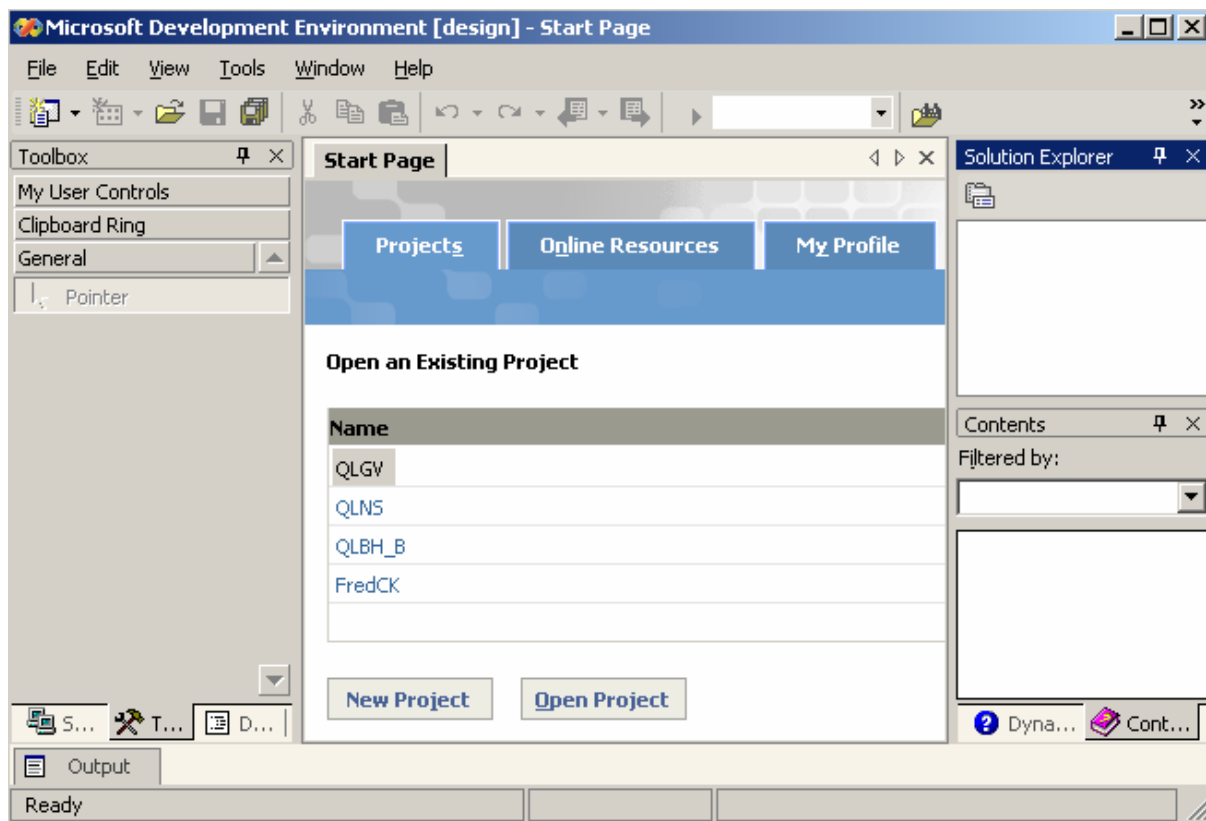
Chúng ta sẽ bắt đầu bằng việc làm quen với môi trường phát triển ứng dụng (IDE) của Visual Studio.NET. VS.NET có nhiều thay đổi so với VS 98.

Hình dưới là màn hình khởi đầu của VS.NET 2003. Vùng làm việc chính giữa đang hiển thị trang "Start page" với 3 mục chính: Projects, Online Resource và My Profile.

My Profile ghi nhớ thông tin về người sử dụng VS.NET. Các thông tin chủ yếu liên quan đến cách chúng ta sẽ sử dụng VS.NET như thế nào. Chẳng hạn như cách hiển thị các cửa sổ, các phím tắt, cách VS.NET hiển thị màn hình giúp đỡ,...

Online Resource cần một kết nối với Internet để download các thông tin từ website của Microsoft về máy tính của chúng ta.

Projects liệt kê các project mà chúng ta đã làm việc trong thời gian gần đây. Trên mục này, chúng ta cũng có thể tạo mới một project bằng cách nhấn vào nút New Project.



Màn hình Microsoft Visual Studio .Net

3.2. Tạo mới ứng dụng Web

1.2.1. Tạo ứng dụng web đầu tiên

Chúng ta có thể tạo ứng dụng Asp.Net sử dụng Visual Basic Project theo các bước sau:

Bước 1. Chọn từ thực đơn File | New | Project. Xuất hiện hộp thoại tạo mới Project.

Chọn loại Project là **Visual Basic Project** từ Project Types / Chọn **ASP.Net Web Application** từ vùng Template

Ứng dụng mới được tạo mặc định có tên là **WebApplicationXX** (XX là số thứ tự tự động). Chúng ta có thể thay đổi tên của Project tại điều khiển Location. Trong ví dụ này, chúng tôi thay đổi tên Project **WebApplication1** thành **MinhHoa**.

Project được tạo mặc định lưu tại thư mục: C:\inetpub\wwwroot

1.2.2. Bổ sung điều khiển và thi hành ứng dụng

Thiết lập thuộc tính pageLayout của trang mặc định (WebForm1.aspx) là FlowLayout (thực hiện thông qua cửa sổ thuộc tính)

Thêm 2 điều khiển Label có trên trang WebForms của thanh công cụ Toolbox.

Tên điều	Thuộc tính Text
lblChao	Chào bạn đến với lập trình web với
lblThoi_gian	[Chuỗi rỗng]

Viết lệnh cho sự kiện Page_Load

2. Trình tự thực hiện:

Bước 1. Chọn từ thực đơn File | New | Project.

Bước 2: Xuất hiện hộp thoại tạo mới Project.

Chọn loại Project là **Visual Basic Project** từ Project Types/Chọn **ASP.Net Web Application** từ vùng Template

Ứng dụng mới được tạo mặc định có tên là **WebApplicationXX** (XX là số thứ tự tự động). Chúng ta có thể thay đổi tên của Project tại điều khiển Location.

Bước 3: Trong ví dụ này, chúng tôi thay đổi tên Project **WebApplication1**

Bước 4: Project được tạo mặc định lưu tại thư mục: C:\Inetpub\wwwroot

Bài 2. WEB SERVER CONTROL

Mã bài: MD21-B02

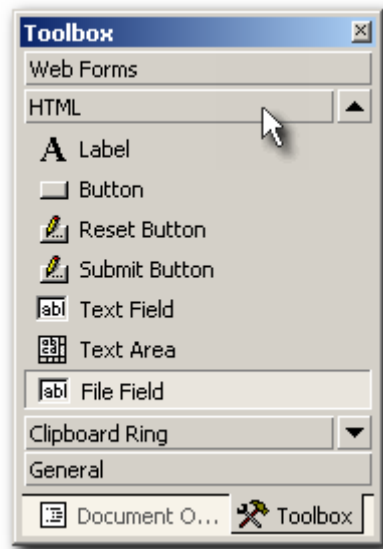
Mục tiêu

- Sử dụng cửa sổ phát triển ứng dụng web MS Visual studio.NET
- Sử dụng được tất cả các công cụ trong ToolBox
- Tổ chức và quản lý được ứng dụng Web
- Đảm bảo an toàn cho người và thiết bị

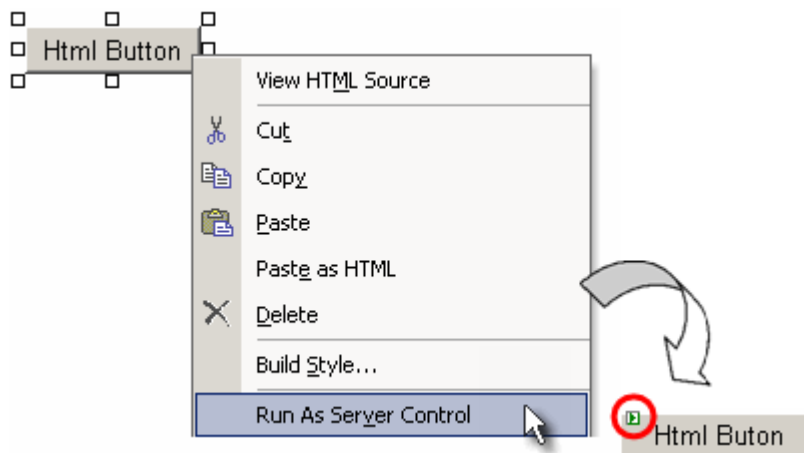
Nội dung chính:

1. HTML Control

Điều khiển HTML (tag HTML) trong trang ASP.Net có thể xem như những chuỗi văn bản bình thường. Để có thể được sử dụng lập trình ở phía Server, ta gán thuộc tính **runat="Server"** cho các điều khiển HTML đó. Những điều khiển HTML (tag HTML) có thuộc tính **runat="Server"** được gọi là HTML Server Control.



Các điều khiển HTML trên thanh công cụ



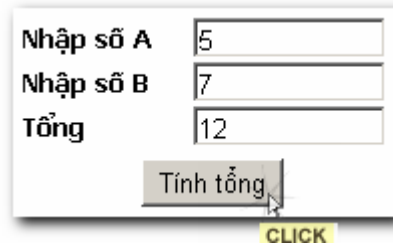
Để chuyển các điều khiển HTML thành điều khiển HTML Server, ta chọn Run As Server Control từ thực đơn ngữ cảnh.

Chuyển điều khiển HTML thành điều khiển HTML Server

Ví dụ: Các điều khiển HTML: Label, Textbox, Button

Xử lý sự kiện:

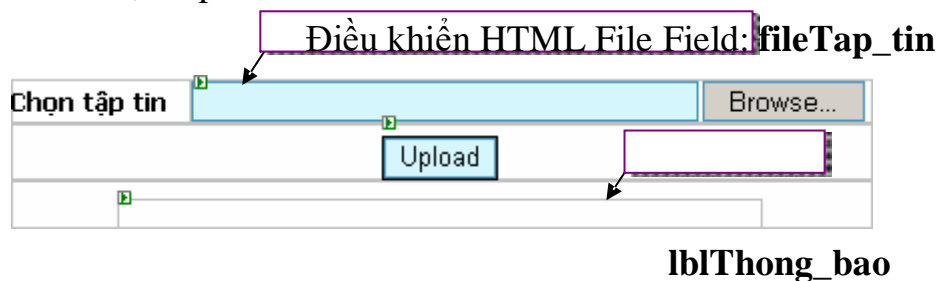
```
Private Sub butTong_ServerClick(...) ...
    txtTong.Value = Val(txtA.Value) +
Val(txtB.Value) End Sub
```



Khi thi hành ứng dụng Ví dụ: Upload file với điều khiển HTML File Field

Trong ví dụ sau, chúng ta sẽ thực hiện Upload tập tin lên server, cụ thể hơn, tập tin vừa Upload sẽ được lưu trong thư mục **Upload**.

Chú ý: Để chép được tập tin lên thư mục Upload, bạn cần phải cấp quyền cho phép ghi trên thư mục Upload



Màn hình ở chế độ thiết kế Xử lý sự kiện:

```
Private Sub butUpload_ServerClick(..., e ... ) ...
    Dim sTap_tin As
String Dim sTen_file As
String
    sTap_tin = fileTap_tin.PostedFile.FileName
    'Phân tích đường dẫn tập tin để lấy tên tập tin
    sTen_file = sTap_tin.Substring(sTap_tin.LastIndexOf("\") + 1,
sTap_tin.Length - sTap_tin.LastIndexOf("\") + 1))
```

"Thực hiện chép tập tin lên thư mục Upload fileTap_tin. PostedFile.
 SaveAs(Server.MapPath("Upload\") & sTen_file) lblThong_bao.InnerHtml =
 "Thông báo: Bạn đã upload file thành công" End Sub

2. ASP.Net Web Control

2.1. Điều khiển cơ bản

Dưới đây là các lý do bạn nên sử dụng ASP.Net Web Control:

Đơn giản, tương tự như các điều khiển trên Windows Form.

Đồng nhất: Các điều khiển Web server có các thuộc tính giống nhau để tìm hiểu và sử dụng.

Hiệu quả: Các điều khiển Web Server tự động phát sinh ra các tag HTML theo từng loại Browser.

Bảng liệt kê các thuộc tính chung của các Web control

Thuộc (ID)	Kiểu	Ý nghĩa
AccessKey	String	Qui định ký tự để di chuyển nhanh đến điều khiển - ký tự xử lý phím nóng.
Attributes	AttributeCollecti	Tập hợp các thuộc tính của điều khiển HTML.
BackColor	Color	Qui định màu nền của điều khiển.
BorderColor	Color	Qui định màu đường viền của điều khiển.
BorderStyle	BorderStyle	Qui định kiểu đường viền của điều khiển.
BorderWid	Unit	Qui định độ rộng của đường viền.
CssClass	String	Qui định hình thức hiển thị của điều khiển qua tên
Enabled	Boolean	Qui định điều khiển có được hiển thị hay không. Giá trị mặc định của thuộc tính này là True – được phép
Font	FontInfo	Qui định Font hiển thị cho điều khiển.
ForeColor	Color	Qui định màu chữ hiển thị trên điều khiển
Height	Unit	Qui định chiều cao của điều khiển.
ToolTip	String	Dòng chữ sẽ hiển thị khi rê chuột vào điều khiển.
Width	Unit	Qui định độ rộng của điều khiển.

1. Label

Label thường được sử dụng để hiển thị và trình bày nội dung trên trang web. Nội dung chính: được hiển thị trong label được xác định thông qua thuộc tính Text. Thuộc tính Text có thể nhận và hiển thị nội dung với các tag HTML.

Ví dụ:

lblA.Text = "Đây là chuỗi văn bản thường"

lblB.Text = "Còn đây là chuỗi văn bản được in đậm"

Đây là chuỗi văn bản thường
Còn đây là chuỗi văn bản được in đậm

2. HyperLink

Điều khiển này được sử dụng để tạo ra các liên kết siêu văn bản.

Các thuộc tính

ImageURL: Qui định hình hiển thị trên điều khiển.

Text: Chuỗi văn bản sẽ được hiển thị trên điều khiển. Trong trường hợp cả 2 thuộc tính ImageURL và Text được thiết lập, thuộc tính ImageURL được ưu tiên, thuộc tính Text sẽ được hiển thị như Tooltip.

NavigateUrl: Đường dẫn cần liên kết đến.

Target: Xác định cửa sổ sẽ hiển thị cho mỗi liên kết

_blank: Hiển thị trang liên kết ở một cửa sổ mới.

_self: Hiển thị trang liên kết tại chính cửa sổ chứa liên kết đó. □ □

_parent: Hiển thị trang liên kết ở frame cha.

Ví dụ:

```
hplASP_net.Text = "Trang chủ  
ASP.Net" hplASP_net.ImageUrl =  
"Hinh\Asp_net.jpg"  
hplASP_net.NavigateUrl =  
"http://www.asp.net" hplASP_net.Target =  
"_blank"
```



Kết quả hiển thị trên trang Web

3. TextBox

TextBox là điều khiển được dùng để nhập và hiển thị dữ liệu. TextBox thường được sử dụng nhiều với các ứng dụng trên windows form.

Các thuộc tính

Text: Nội dung chứa trong Textbox

TextMode: Qui định chức năng của Textbox, có các giá trị sau: □ □

SingleLine: Hiển thị và nhập liệu 1 dòng văn bản

MultiLine: Hiển thị và nhập liệu nhiều dòng văn bản

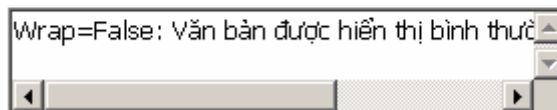
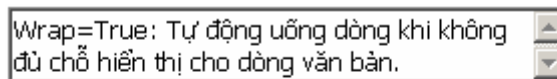
Password: Hiển thị dấu * thay cho các ký tự có trong Textbox.

Rows: Trong trường hợp thuộc tính TextMode = MultiLine, thuộc tính Rows sẽ qui định số dòng văn bản được hiển thị.

Maxlength: Qui định số ký tự tối đa được nhập vào cho TextBox

Wrap: Thuộc tính này qui định việc hiển thị của văn bản có được phép tự động xuống dòng khi kích thước ngang của của điều khiển không đủ để hiển thị dòng nội dung văn bản. Giá trị mặc định của thuộc tính này là True - tự động xuống dòng.

Ví dụ:



AutoPostBack: Thuộc tính này qui định điều khiển có được phép tự động PostBack về Server khi nội dung trong Textbox bị thay đổi hay không. Giá trị mặc định của thuộc tính này là False -không tự động Postback.

4. Image

Điều khiển này được dùng để hiển thị hình ảnh lên trang Web.

Thuộc tính;

ImageUrl: Đường dẫn đến tập tin hình ảnh cần hiển thị;

AlternateText: Chuỗi văn bản sẽ hiển thị khi tập tin được thiết lập trong thuộc tính;

ImageUrl không tồn tại.

ImageAlign: Vị trí hiển thị giữa hình và nội dung văn bản.

5. Button, ImageButton, LinkButton

Các điều khiển Button, ImageButton, LinkButton mặc định đều là các nút Submit Button, mỗi khi được nhấn vào sẽ PostBack về Server.

Khi chúng ta thiết lập giá trị thuộc tính CommandName cho các điều khiển này, chúng ta gọi tên chung cho các điều khiển này là Command Button.

Các thuộc tính chung của Button, ImageButton, LinkButton

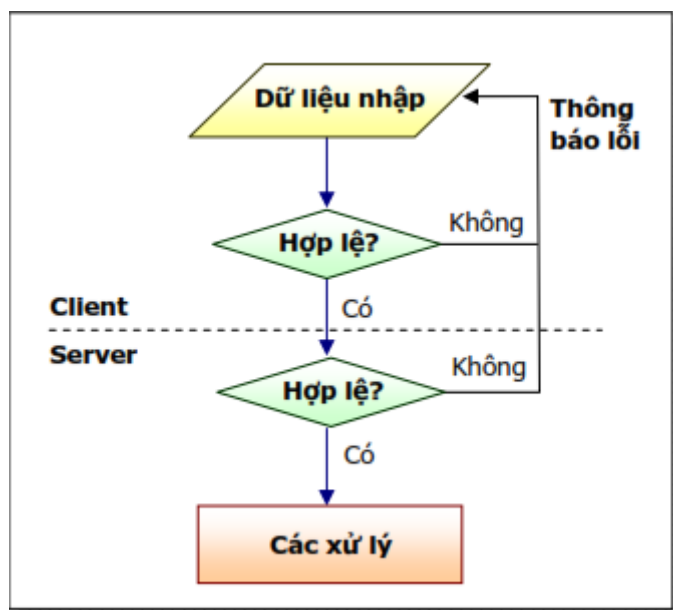
Thuộc tính

Ý nghĩa

Text	Chuỗi văn bản hiển thị trên điều khiển.
CommandName	Tên lệnh. Được sử dụng trong sự kiện Command.
CommandArgum	Thông tin bổ sung cho sự kiện Command.
CausesValidation	Trang web mặc định kiểm tra tính hợp lệ dữ liệu mỗi khi được PostBack. Các điều khiển Button, ImageButton, LinkButton luôn PostBack về Server mỗi khi được nhấn <input type="checkbox"/> luôn kiểm tra tính hợp

2.2. Điều khiển kiểm tra dữ liệu

Trong phần này chúng ta sẽ tìm hiểu về các điều khiển được dùng để kiểm tra dữ liệu.



Sơ đồ xử lý kiểm tra dữ liệu nhập tại Client và Server

Như các bạn đã biết, mỗi khi PostBack về Server, trang Web luôn kiểm tra tính hợp lệ dữ liệu (nếu có yêu cầu khi thiết kế). Nếu dữ liệu không hợp lệ (bỏ trống, vi phạm miền giá trị, mật khẩu nhập lại không đúng, ...), trang web sẽ không thể PostBack về Server.

Các thuộc tính chung của các điều khiển Validation Control

Thuộc tính	Ý nghĩa
ControlToValidate	Tên điều khiển cần kiểm tra. Đây là thuộc tính mà các bạn phải xác định khi sử dụng Validation Control.
Text	Chuỗi thông báo xuất hiện khi có lỗi.

ErrorMessage	Chuỗi thông báo xuất hiện trong điều khiển Validation Summary. Giá trị này sẽ được hiển thị tại vị trí của điều khiển nếu chúng ta không gán giá trị cho thuộc tính Text.
Display	Qui định hình thức hiển thị: <i>f</i> None: Không hiển thị thông báo lỗi (vẫn có kiểm tra dữ liệu) <i>f</i> Static: Trong trường hợp không có vi phạm dữ liệu, điều khiển không có hiển thị nhưng vẫn chiếm vị trí như trong lúc thiết kế.
EnableClientScript	Có cho phép thực hiện kiểm tra ở phía Client hay không. Giá trị mặc định là True - có kiểm tra.

Ví dụ: Minh họa thuộc tính Display: Tại ô nhập lại mật khẩu, ta có 2 điều khiển kiểm tra dữ liệu: một điều khiển kiểm tra không được phép rỗng (rfvNhap_lai), một điều khiển kiểm tra xem nhập lại mật khẩu có giống với mật khẩu đã nhập ở trên hay không.

rfvNhap_lai.Display = Static

The image shows two screenshots of a login form titled "Thông tin đăng nhập".

The top screenshot shows the form with three input fields: "Tên đăng nhập", "Mật khẩu", and "Nhập lại mật khẩu". The "Mật khẩu" field has a validation control labeled "rfvNhap_lai" associated with it. The "Nhập lại mật khẩu" field has a validation control labeled "(*)Không trùng".

The bottom screenshot shows the form with data entered: "Tên đăng nhập" is "dlthien", "Mật khẩu" is masked with dots, and "Nhập lại mật khẩu" is also masked with dots. A red arrow points to the "Nhập lại mật khẩu" field, and a callout box says: "Khi không có lỗi, điều khiển Validation vẫn chiếm dụng vị trí trên trang web".

Lựa chọn hình thức hiển thị

rfvNhap_lai.Display = Dynamic

Lựa chọn hình thức hiển thị

1. Điều khiển Required Field Validator

Điều khiển này được dùng để kiểm tra giá trị trong điều khiển phải được nhập.

Sử dụng điều khiển này để kiểm tra ràng buộc dữ liệu khác rỗng (bắt buộc nhập).

Thuộc tính

- **InitialValue**: Giá trị khởi động. Giá trị bạn nhập vào phải khác với giá trị của thuộc tính này. Giá trị mặc định của thuộc tính này là chuỗi rỗng.

2. Điều khiển Compare Validator

Điều khiển này được dùng để so sánh giá trị của một điều khiển với giá trị của một điều khiển khác hoặc một giá trị được xác định trước.

Thông qua thuộc tính Operator, chúng ta có thể thực hiện các phép so sánh như: =, <>, >, >=, <, <= hoặc dùng để kiểm tra kiểu dữ liệu (DataTypeCheck).

Sử dụng điều khiển này để kiểm tra ràng buộc miền giá trị, kiểu dữ liệu, liên thuộc tính.

Lưu ý: Trong trường hợp không nhập dữ liệu, điều khiển sẽ không thực hiện kiểm tra vi phạm.

Các thuộc tính

- **ControlToCompare**: Tên điều khiển cần so sánh giá trị. Nếu bạn chọn giá trị của thuộc tính Operator = DataTypeCheck thì không cần phải xác định giá trị cho thuộc tính này.

- **Operator**: Qui định phép so sánh, kiểm tra kiểu dữ liệu f Equal: = (Đây là giá trị mặc định)

- f GreaterThan: >

f

GreaterThanEqu

al: >= f

LessThan: <

f

LessThanEqu

al: <= f

NotEqual: <>

f DataTypeCheck: Kiểm tra kiểu dữ liệu

– Type: Qui định kiểu dữ liệu để kiểm tra hoặc so sánh.

f String

f Integer

f Double

f Date

f Currency

– ValueToCompare: Giá trị cần so sánh. Trong trường hợp bạn xác định giá trị của cả 2 thuộc tính ControlToCompare và ValueToCompare

2.3. Một số điều khiển khác

1.1. Điều khiển Literal

Tương tự như điều khiển Label, điều khiển Literal cũng được sử dụng để hiển thị chuỗi văn bản trên trang Web.

Nếu muốn hiển thị một chuỗi văn bản trên trang Web, chúng ta có thể đánh nội dung trực tiếp vào trang Web mà không cần phải sử dụng điều khiển. Chỉ sử dụng các điều khiển như Label, Literal để hiển thị khi cần thay đổi nội dung hiển thị ở phía server.

Điểm khác biệt chính giữa Label và Literal là khi hiển thị nội dung lên trang web (lúc thi hành), điều khiển Literal không tạo thêm một tag Html nào cả, còn Label sẽ tạo ra một tag span (được sử dụng để lập trình ở phía client).

Ví dụ:

<pre>[blLabel] [Literal "ltrLiteral"]</pre>	<p>Đây là chuỗi trong Label <i>Còn đây là chuỗi trong Literal</i></p>
--	---

Khi thiết kế

Khi thi hành

Lệnh xử lý:


```

Private Sub Page_Load(...) Handles MyBase.Load lblLabel.Text =
"<B>Đây là chuỗi trong Label</B>"
ltrlLiteral.Text = "<I>Còn đây là chuỗi trong Literal</I>" End Sub

```

Chọn chức năng từ thực đơn View | Source trên Browser:

```

<span id="lblLabel"><B>Đây là chuỗi trong Label</B></span><br>
<I>Còn đây là chuỗi văn bản trong Literal</I>

```

1.2. Điều khiển Panel và Placeholder

Hai điều khiển Panel và Placeholder được sử dụng để chứa các điều khiển khác. Thuộc tính thường dùng của 2 điều khiển này là Visible. Nếu giá trị của thuộc tính này = True, các điều khiển chứa bên trong sẽ được hiển thị, ngược lại (Visible = False), không có điều khiển nào chứa bên trong được hiển thị.

Tuy nhiên, điều khiển Panel cho phép chúng ta kéo những điều khiển vào bên trong nó lúc thiết kế, còn điều khiển Placeholder thì không.

Để thêm những điều khiển vào bên trong lúc thi hành, chúng ta phải thực hiện thông qua tập hợp Controls của điều khiển:

Ví dụ:

```
Dim txtSo_A As New TextBox pnl.Controls.Add(txtSo_A)
```

1.3. Điều khiển Table

Điều khiển Table thường được sử dụng để hiển thị dữ liệu theo các dòng lệnh đã được cài đặt. Điều khiển này được tạo thành từ tập hợp các dòng (thông qua thuộc tính **Rows**) - TableRow, mỗi dòng được tạo thành từ tập hợp các ô (thông qua thuộc tính **Cells**) – TableCell.

Mỗi ô - cell (TableCel) trong Table có thể là một điều khiển chứa các điều khiển khác. Ta có thể thao tác với các điều khiển trong ô thông qua tập hợp **Controls** của ô đó.

Màn hình khi thiết kế Xử lý sự kiện:

```

Private Sub Page_Load(..., e ... ) Handles MyBase.Load Ve_bang(3, 3)
End Sub

Public Sub Ve_bang(ByVal pSo_dong As Integer, ByVal pSo_cot As Integer) Dim r%, c%
Dim tr As TableRow, td As TableCell 'Tiến hành tạo bảng dữ liệu
For r = 0 To pSo_dong - 1 tr = New TableRow tr.Height = New Unit(20)
For c = 0 To pSo_cot - 1

```

```

td = New TableCell If r = c Then
    'Xử lý thêm các Textbox
    Dim txtTextbox As New TextBox txtTextbox.Text = "Dòng " & r & " cột
" & c txtTextbox.BackColor = Color.Yellow txtTextbox.Width = New Unit(90)
td.Controls.Add(txtTextbox)
Else
    td.Text = "Dòng " & r & " cột " & c End If
tr.Cells.Add(td) Next c tblBang.Rows.Add(tr)
Next r End Sub

```

1.4. Điều khiển AdRotator

Điều khiển AdRotator được dùng để tạo ra các banner quảng cáo cho trang web, nó tự động thay đổi các hình ảnh (đã được thiết lập trước) mỗi khi có yêu cầu,PostBack về server.

a. Thuộc tính

– AdvertisementFile: Tên tập tin dữ liệu (dưới dạng xml) cho điều khiển.

Dưới đây là cú pháp của tập tin Advertisement (*.xml)

```

<Advertisem
ents>
<Ad>
    <ImageUrl>
        Đường dẫn đến tập tin hình ảnh </ImageUrl>
    <NavigateUrl>
        Đường dẫn đến liên kết </NavigateUrl> <AlternateText>
        Chuỗi văn bản được hiển thị như Tooltip </AlternateText>
    <Keyword>
        Từ khóa dùng để lọc hình ảnh </Keyword>
    <Impressions>
        Tần suất hiển thị của hình ảnh </Impressions>
</Ad> </Advertisements>

```

Lưu ý: Phải nhập đúng các giá trị trong tag như mẫu trên. Các giá trị trong tag có phân biệt chữ Hoa chữ thường.

Trong đó

- ImageUrl: Đường dẫn đến một tập tin hình ảnh
- NavigateUrl: Đường dẫn đến trang web sẽ được liên kết đến khi người dùng nhấn vào hình ảnh đang hiển thị.

– AlternateText: Giá trị này sẽ được hiển thị nếu như đường dẫn đến tập tin hình ảnh (qua thuộc tính NavigateUrl) không tồn tại. Đối với một số trình duyệt, tham số này được hiển thị như ToolTip của hình quảng cáo.

– Keyword: Được dùng để phân loại các quảng cáo. Thông qua giá trị này, ta có thể lọc các quảng cáo theo một điều kiện nào đó.

– Impressions: Tham số này quyết định tần suất hiển thị của hình ảnh. Giá trị này càng lớn, khả năng hiển thị càng nhiều.

– KeywordFilter: Được dùng để chọn lọc và hiển thị những hình quảng cáo có giá trị của tham số Keyword = giá trị của tham số này.

Giá trị của tham số này mặc định không được thiết lập. **Hiện thị** tất cả những hình có trong tập tin XML. Trong trường hợp nếu không có hình nào có giá trị Keyword bằng giá trị của thuộc tính này, sẽ không có hình nào được hiển thị.

– Target: Qui định cửa sổ hiển thị trang liên kết

f _blank: Trang liên kết sẽ được mở ở một cửa sổ mới.

f _self: Trang liên kết sẽ được mở ở chính cửa sổ chứa điều khiển. *f*

_parent: Trang liên kết sẽ được mở ở cửa sổ cha.

b. Sự kiện

– AdCreated: Xảy ra khi điều khiển tạo ra các quảng cáo. **Ví dụ:** Tạo Quảng cáo sử dụng điều khiển AdRotator

Bước 1. Thiết kế giao diện

Bước 2. Tạo tập tin dữ liệu: Hinh.xml

f Sử dụng chức năng Add New Item... từ thực đơn ngữ cảnh *f* Chọn XML File trong hộp thoại Add New Item

f Nhập vào cú pháp qui định cho tập tin Hinh.xml (theo cú pháp của tập tin Advertisement)

```
<?xml version="1.0" encoding="utf-8" ?> <Advertisements>
```

```
  <Ad> <ImageUrl>Hinh\Asp_net.jpg</ImageUrl>
```

```
  <NavigateUrl>http://www.asp.net</NavigateUrl>
```

```
  <AlternateText>Trang chủ Asp.net</AlternateText>
```

```
  <Keyword>Hoc_tap</Keyword> <Impressions>10</Impressions>
```

```
  </Ad>
```

```
<Advertisem
```

```
ents>
```

f Chuyển màn hình qua trang Data, nhập liệu trực tiếp trên màn hình

này.

Bước 3. Thiết lập thuộc tính cho điều khiển adQuang_cao

f AdvertisementFile: Hinh.xml

f Target: _blank Khi nhấn vào sẽ hiển thị liên kết ở cửa sổ mới.

f KeywordFilter: Không thiết lập Khi hiển thị tất cả các hình ảnh

Bước 4. Thi hành ứng dụng

1.5. Điều khiển Calendar

Một điều chắc chắn rằng điều khiển Calendar đã quá quen thuộc với các bạn lập trình ứng dụng trên windows, nó có giao diện trực quan, vì vậy, người dùng có thể chọn ngày dễ dàng.

a. Thuộc tính

– DayHeaderStyle: Qui định hình thức hiển thị tiêu đề của các ngày trong tuần – DayStyle: Qui định hình thức hiển thị của các ngày trong điều khiển.

– NextPrevStyle: Qui định hình thức hiển thị của tháng trước/sau của tháng đang được chọn. – SelectedDayStyle: Qui định hình thức hiển thị của ngày đang được chọn.

– SelectedDate: Giá trị ngày được chọn trên điều khiển

– TitleStyle: Qui định hình thức hiển thị dòng tiêu đề của tháng được chọn – TodayDayStyle: Qui định hình thức hiển thị của ngày hiện hành (trên server). – WeekendDayStyle: Qui định hình thức hiển thị của các ngày cuối tuần (thứ 7, chủ nhật)

– OtherMonthDayStyle: Qui định hình thức hiển thị của các ngày không nằm trong tháng hiện hành.

b. Sự kiện

– SelectionChanged: Xự kiện này xảy ra khi bạn chọn một ngày khác với giá trị ngày đang được chọn hiện hành

– VisibleMonthChanged: Xự kiện này xảy ra khi bạn chọn tháng khác với tháng hiện hành

2. Trình tự thực hiện

Bước 1. Thiết kế giao diện

Bước 2. Tạo tập tin dữ liệu: Hinh.xml

f Sử dụng chức năng Add New Item... từ thực đơn ngữ cảnh f Chọn XML File trong hộp thoại Add New Item

f Nhập vào cú pháp qui định cho tập tin Hinh.xml (theo cú pháp của tập tin Advertisement)

```
<?xml version="1.0" encoding="utf-8" ?> <Advertisements>
  <Ad> <ImageUrl>Hinh\Asp_net.jpg</ImageUrl>
  <NavigateUrl>http://www.asp.net</NavigateUrl>
<AlternateText>Trang          chủ          Asp.net</AlternateText>
<Keyword>Hoc_tap</Keyword> <Impressions>10</Impressions>
</Ad>
```

<Advertiseents>

f Chuyển màn hình qua trang Data, nhập liệu trực tiếp trên màn hình này

Bước 3. Thiết lập thuộc tính cho điều khiển adQuang_cao

f AdvertisementFile: Hinh.xml

f Target: _blank Khi nhấn vào sẽ hiển thị liên kết ở cửa sổ mới.

f KeywordFilter: Không thiết lập Khi hiển thị tất cả các hình ảnh

Bước 4. Thi hành ứng dụng

Bài 3. CƠ BẢN VỀ LẬP TRÌNH ASP.NET

Mã bài: MD21-B03

Mục tiêu

- Sử dụng được ngôn ngữ lập trình ASP.NET
- Quản lý được chương trình bởi không gian tên.
- Khai báo được lớp đối tượng
- Khai báo được các lớp kế thừa với các kiểu dữ liệu.

Nội dung chính:

1. Khai báo biến

Các câu lệnh lặp, bao gồm các câu lệnh lặp **for**, **while** và **do... while**. Ngoài ra ngôn ngữ C# còn bổ sung thêm một câu lệnh lặp **foreach**, lệnh này mới đối với người lập trình C/C++ nhưng khá thân thiện với người lập trình VB. Cuối cùng là các câu lệnh nhảy như **goto**, **break**, **continue**, và **return**.

+ *Câu lệnh nhảy goto*

Lệnh nhảy **goto** là một lệnh nhảy đơn giản, cho phép chương trình nhảy vô điều kiện tới một vị trí trong chương trình thông qua tên nhãn. Hầu hết các người lập trình có kinh nghiệm đều tránh dùng lệnh **goto**. Sau đây là cách sử dụng lệnh nhảy **goto**:

+ Tạo một nhãn

+ **goto** đến nhãn

Nhãn là một định danh theo sau bởi dấu hai chấm (:). Thường thường một lệnh **goto** gắn với một điều kiện nào đó

2. Mảng

* Mảng

Cú pháp:

```
type[ ] array-name;
```

thí dụ:

```
int[] myIntegers; // mảng kiểu số nguyên
```

```
string[] myString ; // mảng kiểu chuỗi chữ
```

Bạn khai báo mảng có chiều dài xác định với từ khoá **new** như sau:

```
// Create a new array of 32 ints
```

```
int[] myIntegers = new int[32];
```

```
integers[0] = 35; // phần tử đầu tiên có giá trị 35
```

```
integers[31] = 432; // phần tử 32 có giá trị 432
```

Bạn cũng có thể khai báo như sau:

```
int[] integers;
```

```
integers = new int[32];
```

```
string[] myArray = {"first element", "second element", "third element"};
```

3. Làm việc với mảng (Working with Arrays)

Ta có thể tìm được chiều dài của mảng sau nhờ vào thuộc tính Length thí dụ sau:

```
int arrayLength = integers.Length
```

```
Array.Sort(myArray);
```

Cuối cùng chúng ta có thể đảo ngược mảng đã có nhờ vào the static Reverse() method:

```
Array.Reverse(myArray);
```

```
string[] artists = {"Leonardo", "Monet", "Van Gogh", "Klee"};
```

```
Array.Sort(artists);
```

```
Array.Reverse(artists);
```

```
foreach (string name in artists)
```

```
{
```

```
    Console.WriteLine(name);
```

```
}
```

Bài 4. SQL SERVER

Mã bài: MD21-B04

Mục tiêu

- Sử dụng được phần mềm quản trị cơ sở dữ liệu MS SQL Server;
- Tạo cơ sở dữ liệu trong MS SQL Server;
- Thực hiện được các lệnh Select, Insert, Update và Delete có điều kiện lọc;
- Sử dụng được các hàm gộp khi sử dụng mệnh đề Group By.

Nội dung chính:

1. Tổng quan về SQL

1.1. Một cơ sở dữ liệu - CSDL(DataBase):

Là một kho dữ liệu được tổ chức theo một nguyên tắc nào đó. Đó là một tập hợp các tập tin có liên quan với nhau, được thiết kế nhằm làm giảm thiểu sự dư thừa dữ liệu, đảm bảo tính tin cậy khi truy xuất dữ liệu. Các tập tin này chứa các thông tin biểu diễn các đối tượng trong một ứng dụng thế giới thực.

CSDL lưu giữ thông tin của một trường đại học như : khoa, giảng viên, sinh viên, khóa học,...

Thông thường, một cơ sở dữ liệu sẽ bao trùm tất cả các thông tin của một ứng dụng, không nên đặt hai cơ sở dữ liệu vào một ứng dụng.

1.2. Hệ quản trị cơ sở dữ liệu DBMS(DataBaseManagement System):

Là một hệ thống gồm một CSDL và các thao tác trên CSDL. Đó là hệ thống chương trình, công cụ cho phép quản lý và tương tác với CSDL. Trên đó người dùng có thể định nghĩa, thao tác, và xử lý dữ liệu trong một CSDL để xuất ra những thông tin có nghĩa.

Ví dụ 1-5 : một DBMS có thể quản trị cơ sở dữ liệu của một trường đại học cũng như những cơ sở dữ liệu có ý nghĩa khác như : cơ sở dữ liệu phục vụ tổng thu nhập quốc gia, một cơ sở dữ liệu liên hợp quốc về dữ liệu địa lý thế giới, v.v...

Một hệ cơ sở dữ liệu (HCSDL/ DBS: DataBase System) là một phần mềm cho phép xây dựng một HQTCSDL.

1.2.1. Các vấn đề cần xử lý của hệ cơ sở dữ liệu

Một số điểm bất lợi chính của việc lưu giữ *thông tin có tổ chức* trong hệ thống xử lý file thông thường mà hệ HCSDL cần lưu ý:

* **Dư thừa dữ liệu và tính không nhất quán** (Data redundancy and inconsistency): Do các file và các trình ứng dụng được tạo ra bởi các người lập trình khác nhau, nên các file có định dạng khác nhau, các chương trình được viết trong các ngôn ngữ lập trình khác nhau, cùng một thông tin có thể được lưu giữ trong các file khác nhau. Tính không thống nhất và dư thừa này sẽ làm *tăng chi phí truy xuất và lưu trữ*, hơn nữa, nó sẽ dẫn đến tính không nhất quán của dữ liệu: *các bản sao của cùng một dữ liệu có thể không nhất quán*.

* **Khó khăn trong việc truy xuất dữ liệu**: Môi trường của hệ thống xử lý file thông thường không cung cấp các công cụ cho phép truy xuất thông tin một cách hiệu quả và thuận lợi.

* **Sự cô lập dữ liệu** (Data isolation): Các giá trị dữ liệu được lưu trữ trong cơ sở dữ liệu phải thoả mãn một số các ràng buộc về tính nhất quán của dữ liệu (ràng buộc nhất quán / consistency constraints).

Trong hệ thống xử lý file thông thường, rất khó khăn trong việc thay đổi các chương trình để thoả mãn các yêu cầu thay đổi ràng buộc. Vấn đề trở nên khó khăn hơn khi các ràng buộc liên quan đến các hạng mục dữ liệu nằm trong các file khác nhau.

* **Các vấn đề về tính nguyên tử** (Atomicity problems): Tính nguyên tử của một hoạt động (giao dịch) là: *hoặc nó được hoàn tất trọn vẹn hoặc không có gì cả*. Điều này có nghĩa là một hoạt động (giao dịch) *chỉ làm thay đổi* các dữ liệu bền vững khi nó đã hoàn tất (kết thúc thành công) nếu không, giao dịch không để lại một dấu vết nào trên CSDL. Trong hệ thống xử lý file thông thường khó đảm bảo được tính chất này.

* **Tính bất thường trong truy xuất cạnh tranh**: Một hệ thống cho phép nhiều người sử dụng cập nhật dữ liệu đồng thời, có thể dẫn đến kết quả là dữ liệu không nhất quán. Điều này đòi hỏi một sự giám sát. Hệ thống xử lý file thông thường không cung cấp chức năng này.

* **Vấn đề an toàn** (Security problems): một người sử dụng hệ cơ sở dữ liệu không cần thiết và cũng không có quyền truy xuất tất cả các dữ liệu. Vấn đề này đòi hỏi hệ thống phải đảm bảo được tính phân quyền, chống truy xuất trái phép... Các bất lợi nêu trên đã gợi mở sự phát triển các DBMS. Phần sau của giáo trình sẽ đề cập đến các quan niệm và các thuật toán được sử dụng để phát triển một hệ cơ sở dữ liệu nhằm *giải quyết các vấn đề nêu trên*.

1.2.2. Hầu hết các hệ quản trị CSDL đều thực hiện các chức năng sau :

Lưu trữ dữ liệu

- Tạo ra và duy trì CSDL
- Cho phép nhiều người dùng truy xuất đồng thời
- Hỗ trợ tính bảo mật và riêng tư
- Cho phép xem và xử lý dữ liệu lưu trữ
- Cho phép cập nhật và lưu trữ dữ liệu sau khi cập nhật
- Cung cấp một cơ chế chỉ mục (index) hiệu quả để truy cập nhanh các dữ liệu lựa chọn
- Cung cấp tính nhất quán giữa các bản ghi khác nhau
- Bảo vệ dữ liệu khỏi mất mát bằng các quá trình sao lưu (backup) và phục hồi (recovery).

1.2.3. Mô hình dữ liệu quan hệ

Mô hình dữ liệu quan hệ được Codd đề xuất năm 1970 và đến nay trở thành mô hình được sử dụng phổ biến trong các hệ quản trị cơ sở dữ liệu thương mại. Nói một cách đơn giản, một cơ sở dữ liệu quan hệ là một cơ sở dữ liệu trong đó tất cả dữ liệu được tổ chức trong các bảng có mối quan hệ với nhau. Mỗi một bảng bao gồm các dòng và các cột: mỗi một dòng được gọi là một bản ghi (bộ) và mỗi một cột là một trường (thuộc tính). Hai hay nhiều bảng có thể có liên kết nếu chúng có một hay nhiều trường chung)

Hình 1.1 minh họa cho ta thấy được 3 bảng trong một cơ sở dữ liệu

MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823462
DHT04	Khoa Hoá học	
...	...	

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy	2000	5	DHT01
C24102	Tin K24	24	Chính quy	2000	8	DHT02
C24103	Lý K24	24	Chính quy	2000	7	DHT03
C24301	Sinh K24	24	Chính quy	2000	5	DHT05

MASV	HODEM	TEN	NGAYSINH	GIOTINH	NOISINH	MALOP
0241010001	Ngô Thị Nhật	Anh	Nov 27 1982	0	Quảng Ninh, Quảng Bình	C24101
0241010002	Nguyễn Thị Ngọc	Anh	Mar 21 1983	0	Tân Kỳ, Nghệ An	C24101
0241010003	Ngô Việt	Bắc	May 11 1982	1	Yên Khánh, Ninh Bình	C24101
0241010004	Nguyễn Đình	Bình	Oct 6 1982	1	Huế	C24101
0241010005	Hồ Đăng	Chiến	Jan 20 1982	1	Phong Điền, TTHuế	C24101
0241020001	Nguyễn Tuấn	Anh	Jul 15 1979	1	Do Linh, Quảng Trị	C24102
0241020002	Trần Thị Kim	Anh	Nov 4 1982	0	Phong Điền, TTHuế	C24102
0241020003	Võ Đức	Ấn	May 24 1982	1	Huế	C24102
0241020004	Nguyễn Công	Bình	Jun 6 1979	1	Thăng Bình, Quảng Nam	C24102
0241020005	Nguyễn Thanh	Bình	Apr 24 1982	1	Huế	C24102
...

Hình 1.1. Các bảng trong một cơ sở dữ liệu

1.2.4. Bảng (Table)

Như đã nói ở trên, trong cơ sở dữ liệu quan hệ, bảng là đối tượng được sử dụng để tổ chức và lưu trữ dữ liệu. Một cơ sở dữ liệu bao gồm nhiều bảng và mỗi bảng được xác định duy nhất bởi tên bảng. Một bảng bao gồm một tập các dòng và các cột: mỗi một dòng trong bảng biểu diễn cho một thực thể (trong hình 1.1, mỗi một dòng trong bảng SINHVIEN tương ứng với một sinh viên), và mỗi một cột biểu diễn cho một tính chất của thực thể (chẳng hạn cột NGAYSINH trong bảng SINHVIEN biểu diễn cho ngày sinh của các sinh viên được lưu trữ trong bảng).

Như vậy, liên quan đến mỗi một bảng bao gồm các yếu tố sau:

* Tên của bảng: được sử dụng để xác định duy nhất mỗi bảngn trong cơ sở dữ liệu.

* Cấu trúc của bảng: Tập các cột trong bảng. Mỗi mộtcột trong bảng được xác định bởi một tên cột và phải có một kiểu dữ liệu nào đó (chẳng hạn cột NGAYSINH trong bảng SINHVIEN ở hình 1.1 có kiểu là DATETIME). Kiểu dữ liệu của mỗi cột qui định giá trị dữ liệu có thể được chấp nhận trên cột đó.

* Dữ liệu của bảng: Tập các dòng (bản ghi) hiện có trong bảng.

1.2.5. Khoá của bảng

Trong một cơ sở dữ liệu được thiết kế tốt, mỗi một bảng phải có một hoặc một tập các cột mà giá trị dữ liệu của nó xác định duy nhất một dòng trong một tập các dòng của bảng.

Tập một hoặc nhiều cột có tính chất này được gọi là khoá của bảng.

Việc chọn khoá của bảng có vai trò quan trọng trong việc thiết kế và cài đặt các cơ sở dữ liệu quan hệ. Các dòng dữ liệu trong một bảng phải có giá trị khác nhau trên khoá. Bảng MONHOC trong hình dưới đây có khoá là cột MAMONHOC

MAMONHOC	TENMONHOC	SODVHT
HO-001	Hoá đại cương	3
TI-001	Tin học đại cương	4
TI-002	Ngôn ngữ C	5
TI-003	Lý thuyết hệ điều hành	4
TI-004	Cấu trúc dữ liệu và giải thuật	4
TO-001	Đại số tuyến tính	4
TO-002	Giải tích 1	4
TO-003	Bài tập Đại số	2
TO-004	Bài tập Giải tích 1	2
VL-001	Vật lý đại cương	3

Hình 1.2. Bảng MONHOC với khoá chính là MAMONHOC

Một bảng có thể có nhiều tập các cột khác nhau có tính chất của khoá (tức là giá trị của nó xác định duy nhất một dòng dữ liệu trong bảng). Trong trường hợp này, khoá được chọn cho bảng được gọi là khoá chính (primary key) và những khoá còn lại được gọi là khoá phụ hay là khoá dự tuyển (candidate key/unique key).

1.2.6. Mỗi quan hệ và khoá ngoài

Các bảng trong một cơ sở dữ liệu không tồn tại độc lập mà có mối quan hệ mật thiết với nhau về mặt dữ liệu. Mối quan hệ này được thể hiện thông qua ràng buộc giá trị dữ liệu xuất hiện ở bảng này phải có xuất hiện trước trong một bảng khác. Mối quan hệ giữa các bảng trong cơ sở dữ liệu nhằm đảm bảo được tính đúng đắn và hợp lệ của dữ liệu trong cơ sở dữ liệu.

Trong hình 1.3, hai bảng LOP và KHOA có mối quan hệ với nhau. Mối quan hệ này đòi hỏi giá trị cột MAKHOA của một dòng (tức là một lớp) trong bảng LOP phải được xác định từ cột MAKHOA của bảng KHOA.

MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823462
...

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy	2000	5	DHT01
C25101	Toán K25	25	Chính quy	2001	5	DHT01
C25102	Tin K25	25	Chính quy	2001	6	DHT02
C24102	Tin K24	24	Chính quy	2000	8	DHT02
...

Hình 1.3. Mối quan hệ giữa hai bảng LOP và KHOA trong cơ sở dữ liệu

Mối quan hệ giữa các bảng trong một cơ sở dữ liệu thể hiện đúng mối quan hệ giữa các thực thể trong thế giới thực. Trong hình 1.3, mối quan hệ giữa hai bảng LOP và KHOA không cho phép một lớp nào đó tồn tại mà lại thuộc vào một khoa không có thật.

Khái niệm khoá ngoài (Foreign Key) trong cơ sở dữ liệu quan hệ được sử dụng để biểu diễn mối quan hệ giữa các bảng dữ liệu. Một hay một tập các cột trong một bảng mà giá trị của nó được xác định từ khóa chính của một bảng khác được gọi là khoá ngoài.

Trong hình 1.3, cột MAKHOA của bảng LOP được gọi là khoá ngoài của bảng này, khoá ngoài này tham chiếu đến khoá chính của bảng KHOA là cột MAKHOA.

2. Tổng quan về CSDL quan hệ

2.1. Lý thuyết liên quan:

Trong định nghĩa cơ sở dữ liệu chúng ta có nói đến các chức năng chính của hệ quản trị cơ sở dữ liệu. Mục này nhằm thảo luận các chức năng đó một cách chi tiết hơn. Vì hầu hết các chức năng của hệ quản trị cơ sở dữ liệu là trong suốt đối với người dùng, định nghĩa sau được phát biểu dưới góc độ người dùng. Định nghĩa 1.4 Hệ quản trị cơ sở dữ liệu là một phần mềm quản lý cấu trúc của cơ sở dữ liệu và kiểm soát việc truy xuất dữ liệu, bao gồm các chức năng: 1. Tạo và bảo toàn cơ sở dữ liệu. 2. Cho phép truy xuất cơ sở dữ liệu theo thẩm quyền. 3. Phục hồi dữ liệu theo yêu cầu. 4. Cập nhật dữ liệu. Như vậy nhiệm vụ chính của hệ quản trị cơ sở dữ liệu là quản lý dữ liệu hiệu quả cho nên một hệ quản trị cơ sở dữ liệu phải thực hiện các chức năng quan trọng nhằm đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong cơ sở dữ liệu. Nó có hai khả năng chính là khả năng quản lý bền vững dữ liệu và khả năng truy xuất một số lượng lớn dữ liệu một cách hiệu quả

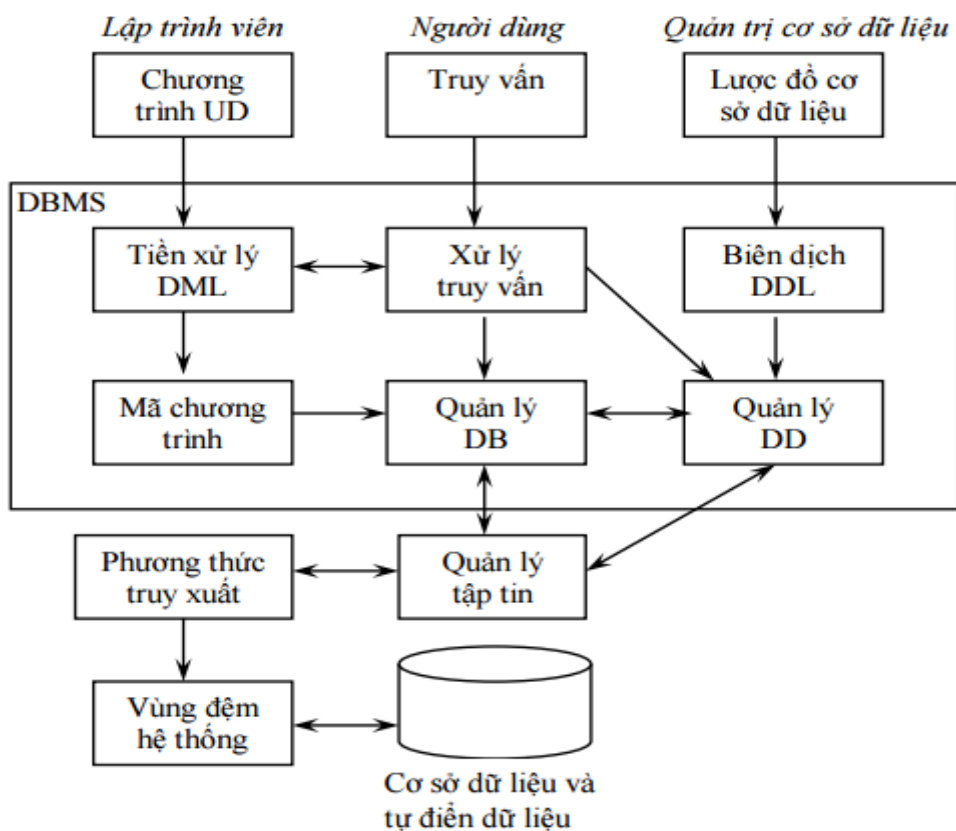
2.2. Chức năng

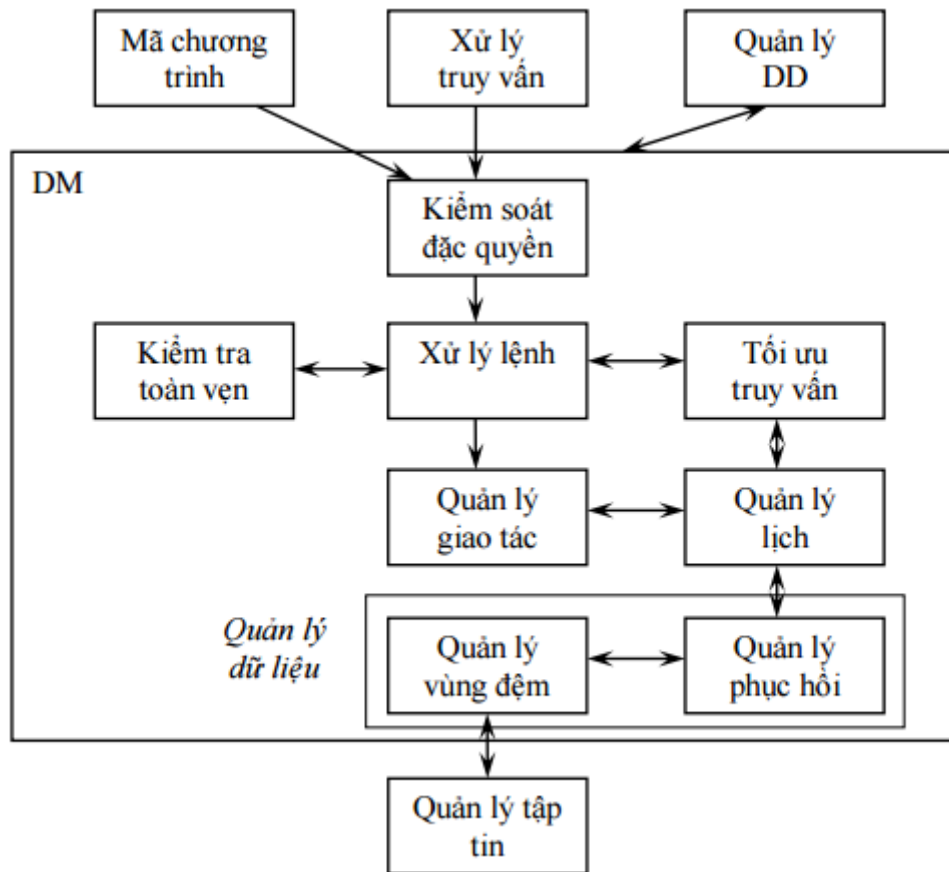
Hệ quản trị cơ sở dữ liệu có các chức năng sau: 17: Quản lý tự điển dữ liệu. Hệ quản trị quản lý tất cả các dữ liệu dùng để mô tả dữ liệu¹⁸. Nó lưu các định nghĩa về cấu trúc và quan hệ trong tự điển dữ liệu. Cho phép thực hiện trừu tượng hoá dữ liệu nhằm đảm bảo tính độc lập dữ liệu và độc lập cấu trúc. Quản lý lưu trữ. Hệ quản trị tạo ra các cấu trúc phức tạp để lưu trữ từ những dữ liệu cơ bản đơn giản đến các dữ liệu phức tạp, từ các định nghĩa về biểu mẫu, biểu báo đến các ràng buộc dữ liệu và các đoạn mã chương trình. Quản lý nhập xuất. Hệ quản trị biến đổi dữ liệu vào phù hợp với cấu trúc lưu trữ, nhờ đó ta không cần quan tâm đến định dạng vật lý của dữ liệu, và kết xuất dữ liệu ra theo các yêu cầu logic của người dùng. Quản lý bảo mật. Hệ quản trị tạo ra một hệ thống bảo mật kiểm soát quyền sử dụng của người dùng. Điều khiển truy xuất nhiều người dùng. Hệ quản trị tạo ra các cấu trúc phức tạp dùng các giải thuật tinh vi nhằm bảo đảm cơ sở dữ liệu luôn luôn nhất quán Quản lý sao lưu và phục hồi cơ sở dữ liệu. Hệ quản trị cung cấp các thủ tục sao lưu và phục hồi dữ liệu đảm bảo dữ liệu an toàn và toàn vẹn khi xảy ra sự cố hoặc khi có yêu cầu từ người dùng.

Quản lý sự toàn vẹn dữ liệu. Hệ quản trị cung cấp cơ chế nhằm giảm dư thừa và tăng sự nhất quán, đặc biệt trong cơ sở dữ liệu nhiều người dùng.

Ngôn ngữ truy xuất cơ sở dữ liệu và giao diện (interface) lập trình ứng dụng. Hệ quản trị cung cấp cơ chế truy xuất dữ liệu qua ngôn ngữ truy xuất cơ sở dữ liệu là ngôn ngữ hỏi gồm: ngôn ngữ định nghĩa dữ liệu DDL và ngôn ngữ thao tác dữ liệu DML. Nó cũng cung cấp các giao diện cho các ngôn ngữ lập trình. Giao diện giao tiếp với cơ sở dữ liệu. Hệ quản trị cung cấp các thủ tục giao tiếp đặc biệt được thiết kế cho phép cơ sở dữ liệu đáp ứng các yêu cầu của người dùng cuối trong môi trường mạng.

3. Thành phần





4. Trình tự thực hiện:

- Bước 1. Tiền xử lý ngôn ngữ thao tác dữ liệu;
- Bước 2. Xử lý truy vấn;
- Bước 3. Biên dịch ngôn ngữ định nghĩa dữ liệu;
- Bước 4. Mã đối tượng chương trình;
- Bước 5. Quản lý cơ sở dữ liệu;
- Bước 6. Quản lý từ điển dữ liệu;
- Bước 7. Quản lý tập tin.

4. Câu lệnh truy vấn

Câu lệnh SELECT (mang nghĩa lấy dữ liệu hay câu lệnh truy vấn) trong SQL trả về một tập kết quả các bản ghi từ một hoặc nhiều bảng.

Nó được sử dụng để lấy dữ liệu từ một hoặc nhiều bảng trong cơ sở dữ liệu, SELECT là lệnh thường dùng nhất của ngôn ngữ sửa đổi dữ liệu (tiếng Anh: Data Manipulation Language - DML). Trong việc tạo ra câu truy vấn SELECT, người sử dụng phải đưa ra mô tả cho những dữ liệu mình muốn lấy ra chứ không chỉ ra những hành động vật lý nào bắt buộc phải thực hiện để lấy ra kết quả đó. Hệ thống cơ sở dữ liệu, hay chính xác hơn là bộ tối ưu hoá câu

truy vấn (tiếng Anh: query optimizer) sẽ dịch từ câu truy vấn sang kế hoạch truy vấn tối ưu.

Những từ khóa liên quan tới SELECT bao gồm:

FROM dùng để chỉ định dữ liệu sẽ được lấy ra từ những bảng nào, và các bảng đó quan hệ với nhau như thế nào.

WHERE điều kiện để liên kết các bảng hoặc điều kiện để lấy những thuộc tính từ bảng from.

GROUP BY dùng để kết hợp các bản ghi có những giá trị liên quan với nhau thành các phần tử của một tập hợp nhỏ hơn các bản ghi.

HAVING dùng để xác định những bản ghi nào, là kết quả từ từ khóa **GROUP BY**, sẽ được lấy ra.

ORDER BY dùng để xác định dữ liệu lấy ra sẽ được sắp xếp theo những cột nào.

Bài 5. TRUY CẬP VÀ XỬ LÝ CƠ SỞ DỮ LIỆU VỚI .NET

Mã bài: MD21-B05

Mục tiêu

Sử dụng các đối tượng ADO để thực hiện các thao tác tìm kiếm, thêm, sửa, xóa các bản ghi trong cơ sở dữ liệu.

Nội dung chính:

1. Tổng quan về ADO.NET

Giống như hầu hết các thành phần của .NET Framework, ADO.NET không chỉ là vỏ bọc của một vài API sẵn có.

Nó chỉ giống ADO ở cái tên - các lớp và phương thức truy xuất dữ liệu đều khác hoàn toàn.

ADO (Microsoft's ActiveX Data Objects) là một thư viện của các thành phần COM đã từng được ca ngợi trong một vài năm trở lại đây. Phiên bản hiện tại là 2.7, các thành phần chủ yếu của ADO là Connection, Command, Recordset, và các Field object. Một connection có thể mở cơ sở dữ liệu, một vài dữ liệu được chọn vào một recordset, bao gồm các trường, dữ liệu này sau đó có thể thao tác, cập nhập lên server, và connection cần phải được đóng lại. ADO cũng giới thiệu một disconnected recordset, cái được dùng khi không muốn giữ kết nối trong một thời gian dài.

Có một vài vấn đề với ADO đó là sự không hài lòng về địa chỉ, sự cồng kềnh của một disconnected recordset. Hỗ trợ này không cần thiết với sự tiến hoá của tin học "web-centric", vì vậy nó cần được loại bỏ. Có một số giống nhau giữa lập trình ADO.NET và ADO (không phải ở cái tên), vì thế việc chuyển từ ADO không qua khó khăn. Hơn thế nữa, nếu bạn dùng SQL Server, có một bộ các quản mới rất tuyệt cho việc thao tác bên ngoài cơ sở dữ liệu. Chừng đó lí do cũng đủ để các bạn quan tâm đến ADO.NET.

ADO.NET chứa hai không gian tên cơ sở dữ liệu - một cho SQL Server, và một cái khác cho các cơ sở dữ liệu được trình bày thông qua một giao diện OLE DB. Nếu cơ sở dữ liệu của bạn chọn là một bộ phận của OLE DB, bạn có thể dễ dàng kết nối với nó từ .NET - chỉ cần dùng các lớp OLE DB và kết nối thông qua các driver cơ sở dữ liệu hiện hành của bạn.

Các Namespace

Tất cả các ví dụ trong chương này truy xuất dữ liệu trong một vài cách. Các không gian tên sau chỉ ra các lớp và các giao diện được dùng cho việc truy xuất dữ liệu trong .NET:

- System.Data - Các lớp truy xuất dữ liệu chung;
- System.Data.Common - Các lớp dùng chung bởi các data provider khác nhau;

- System.Data.OleDb - Các lớp của OLE DB provider;
- System.Data.SqlClient - Các lớp của SQL Server provider;
- System.Data.SqlTypes - Các kiểu của SQL Server;
- Các lớp chính trong ADO.NET được liệt kê dưới đây:
 - + Các lớp dùng chung;
 - + ADO.NET chứa một số lớp được dùng không quan tâm là bạn đang dùng các lớp của SQL Server hay là các lớp của OLE DB.

Các lớp trong không gian tên System.Data được liệt kê sau đây:

- + DataSet - Đối tượng này chứa một bộ các DataTable, có thể bao gồm quan hệ giữa các bảng, và nó được thiết kế cho truy xuất dữ liệu không kết nối.

- + DataTable - Một kho chứa dữ liệu. Một DataTable bao gồm một hoặc nhiều DataColumn, và khi được tạo ra nó sẽ có một hoặc nhiều DataRow chứa dữ liệu;

- + DataRow - Một bộ giá trị, có liên quan với một dòng trong bảng cơ sở dữ liệu, hoặc một dòng của bảng tính;

- + DataColumn - Chứa cá định nghĩa của một cột, chẳng hạn như tên và kiểu dữ liệu;

- + DataRelation - Một liên kết giữa hai DataTable trong một DataSet. Sử dụng cho khóa ngoại và các mối quan hệ chủ tớ;

- + Constraint - Định nghĩa một qui tắc cho một DataColumn (hoặc một bộ các cột dữ liệu), như các giá trị là độc nhất;

Sau đây là hai lớp được tìm thấy trong không gian tên System.Data.Common:

- DataColumnMapping - Ánh xạ tên của một cột từ cơ sở dữ liệu vào tên của một cột trong một DataTable.

- DataTableMapping - Ánh xạ tên của một bảng từ cơ sở dữ liệu vào một bảng trong một DataSet.

2. Phương thức của đối tượng Connection

2.1. .NET Data Providers

.NET Data Providers là một tập các đối tượng phục vụ cho việc trao đổi dữ liệu giữa Data Source (dữ liệu nguồn) và đối tượng DataSet. Nó chia ra gồm 2 loại tập đối tượng: 1 tập các đối tượng chịu trách nhiệm quản lý các kết nối

(connections) tới DataSource (dữ liệu nguồn) và 1 tập các đối tượng còn lại chịu trách nhiệm xử lý dữ liệu.

ADO.NET cung cấp 2 loại .NET Data Providers đó là SQL Server .NET Data Provider và OLE DB .NET Data Provider. SQL Server .NET Data Provider sử dụng truy xuất Database SQL Server 7.0 trở về sau. Còn OLE DB .NET Data Provider sử dụng cho phiên bản 6.5 trở về trước và các loại Database khác (vd như Access, Oracle...). SQL Server .NET Data Provider hoạt động hiệu quả hơn bởi vì nó thao tác trực tiếp với SQL Server mà không cần phải thông qua OLE DB Provider.

Để sử dụng 2 loại .NET Data Provider này ta phải import 2 Namespace: System.Data.SqlClient cho SQL Server .NET Data Provider và System.Data.OleDb cho OLE DB Data Provider.

2) Đối tượng Connection

Đối tượng Connection là đối tượng chịu trách nhiệm quản lý kết nối tới nguồn dữ liệu (DataSource). Có 2 dạng Connection tương ứng với 2 kiểu dữ liệu SQL Server và OLE DB đó là: SqlConnection và OleDbConnection. Cả 2 đối tượng này đều implement từ interface IDbConnection. Bằng cách sử dụng Interface IDbConnection, các nhà cung cấp dịch vụ Database khác nhau có thể tạo ra các cài đặt phù hợp cho Database riêng của họ.

Đối tượng Connection của ADO.NET chỉ nhận một tham số đầu vào là chuỗi kết nối (connection string). Trong chuỗi kết nối, các thông số được cách nhau bằng dấu “;”, connection string có các thông số sau :

- Provider: Tên nhà cung cấp Database, đối với OLEDB cần khai báo là SQLOLEDB. Đối với SQL Server thì không thuộc tính này.
- DataSource (hoặc Server): tên/địa chỉ database server cần kết nối tới.
- Initial catalog (hoặc Database): tên của Database cần truy xuất.
- Uid: username để đăng nhập vào Database Server.
- Pwd: password để đăng nhập vào Database Server.

Sau đây là vd về 1 chuỗi kết nối đối với Database dạng OLE DB: Provider=SQLOLEDB.1;Data Source=MySQL; Initial Catalog=NorthWind; uid=sa; pwd=sa;

Còn đây là vd về 1 chuỗi kết nối đối với Database dạng SQL Server: Server=CSC; database=Northwind; uid=sa;pwd=sa;

Sau đây là các thuộc tính, phương thức, sự kiện thông dụng của cả SqlConnection và OleDbConnection

3. Đối tượng SqlCommand

Sau khi chúng ta đã kết nối vào nguồn dữ liệu, chúng ta cần phải thực hiện thao tác các dữ liệu. Để thao tác được với các dữ liệu chúng ta phải dùng đối tượng Command. Đối tượng Command là đối tượng rất “đa năng”, nó vừa xử lý được sqlserver stored procedures vừa xử lý được các giao tác (transaction). Tương tự như đối tượng Connection, đối tượng Command cũng chia ra làm 2 loại tùy theo nguồn dữ liệu: SqlCommand (cho SQL Server) và OleDbCommand (cho OLE DB).

Một số thuộc tính và phương thức thông dụng của đối tượng Command:

Thuộc tính:

- CommandText: Thiết lập/ Lấy lệnh thao tác với dữ liệu.
- CommandTimeout : Thiết lập/ Lấy thời gian chờ thực hiện lệnh. Sau khi chờ 1 khoảng thời gian nếu vượt quá sẽ báo lỗi.
- CommandType : Thiết lập/ Lấy kiểu của đối tượng lệnh (lệnh trực tiếp, stored procedure...)
- Parameters: Các tham số truyền vào cho đối tượng command (ở kiểu OleDbParameterCollection/ SqlParameterCollection).
- Connection : Thiết lập / lấy kết nối đang được đối tượng Command sử dụng.
- Transaction: Thiết lập / lấy giao tác mà đối tượng Command thực thi.

3.1. Phương thức:

- ExecuteReader: Thực thi câu lệnh CommandText của đối tượng Command và trả về kiểu DataReader (OleDbDataReader / SqlDataReader).
- ExecuteNonQuery: Thực thi câu lệnh CommandText của đối tượng Command, đây là dạng câu lệnh cập nhật cơ sở dữ liệu (xoá/sửa) nên chỉ trả về số dòng bị ảnh hưởng mà không trả về dòng dữ liệu nào.
- ExecuteScalar: Thực thi câu truy vấn của đối tượng Command và chỉ trả về cột đầu tiên của dòng đầu tiên của kết quả. Các kết quả còn lại bị bỏ qua.
- Các hàm khởi tạo của đối tượng SqlCommand (Tương tự cho đối tượng OleDbCommand):
 - o New() : không có tham số nào.
 - o New(cmdText as String) ,trong đó :
cmdText là câu lệnh truyền vào cho đối tượng Command.
 - o New(cmdText as String, connectin as SqlConnection), trong đó:

cmdText như trên.

connection là đối tượng kết nối truyền vào cho đối tượng Command.

o New(cmdText as String, connection as SqlConnection, transaction as SqlTransaction), trong đó :

cmdText, connection như trên.

Transaction: là giao tác truyền cho đối tượng Command. ♣

Các ví dụ:

- Sử dụng câu lệnh ExecuteNonQuery để update dữ liệu: \

```
Dim connStr As String = "server=CSC;database=Northwind;uid=sa;pwd=sa"
```

```
Dim cmdStr As String = "Update Customers set CompanyName='CSC' where CustomerID='Seves'"
```

```
Dim conn As New SqlConnection(connStr)
conn.Open()
Dim cmd As SqlCommand = New SqlCommand(cmdStr, conn)
cmd.ExecuteNonQuery()
```

```
conn.Close()
```

- Sử dụng câu lệnh ExecuteReader để lấy dữ liệu:

```
Dim conn As New SqlConnection("server=CSC;database=Northwind;uid=sa;pwd=sa")
```

```
Dim cmd As New SqlCommand("Select * From Customers", conn)
```

```
conn.Open()
Dim rdr As SqlDataReader = cmd.ExecuteReader()
```

```
While rdr.Read()
Response.Write(rdr.GetString(0) + "  
")
```

```
End While
```

```
rdr.Close()
```

- Sử dụng câu lệnh ExecuteScalar để lấy dữ liệu:

```
Dim conn As New SqlConnection("server=CSC;database=Northwind2;uid=sa;pwd=sa")
```

```
conn.Open()
```

```
Dim cmdStr As String = "Select * From Customers"
```

```
Dim cmd As SqlCommand = New SqlCommand(cmdStr, conn)
```

```
Response.Write(cmd.ExecuteScalar())
conn.Close()
```

3.2. Về khái niệm Connection Pooling

Trong ứng dụng web ASP.NET, Connection Pooling là khái niệm nhiều người dùng cùng truy xuất vào cùng 1 database để lấy về cùng một loại dữ liệu. ADO.NET cho phép quản lý Connection Pooling theo nhiều cách khác nhau. Nếu database đang sử dụng là dạng OLEDB thì connection pooling do chính Provider của Database đó quản lý, vì vậy chúng ta không cần phải quản lý. Trong trường hợp sử dụng database SQL Server thì Connection Pooling sẽ được quản lý một cách không tường minh nhưng cũng cho phép thiết lập những tùy chọn để chúng ta tự quản lý. Các thuộc tính để quản lý Pooling (thiết lập trong Connection String) là : Max Pool Size (số connection tối đa trong pool, Min Pool Size : số connection tối thiểu trong pool).

Khi 1 connection được “open” có 2 tình huống : nếu đã open rồi thì connection đó đã nằm trong pool và được “lấy ra”, nếu chưa có trong pool thì sẽ được thêm vào pool. Khi đóng connection thì tất là nó sẽ được trả về pool.

3.3 Xử lý các ngoại lệ ADO.NET

Để bắt lỗi trong 1 chương .NET thông thường thì ta dùng cú pháp try...catch thông thường . Đối với chương trình ADO.NET có lớp chịu trách nhiệm xử lý các ngoại lệ là SqlException (cho CSDL SQL Server) và OleDbException (cho CSDL OleDb).

Ví dụ về xử lý ngoại lệ trong SQL Server

```
Dim connString As String =
"server=CSC;database=Northwind;uid=sa;pwd=sa"
Dim conn As SqlConnection = New SqlConnection(connString)
Try
conn.Open()
Catch ex As SqlException
Response.Write(ex.ToString())
End Try
```

Bài 6. CÁC ĐIỀU KHIỂN LIÊN KẾT DỮ LIỆU

Mã bài: MD21-B06

Mục tiêu

- Kết nối cơ sở dữ liệu
- Sử dụng các điều khiển liên kết dữ liệu để hiển thị dữ liệu lên trình duyệt
- Sử dụng các đối tượng connection để thực hiện các thao tác tìm kiếm, thêm, sửa, xóa các bản ghi trong cơ sở dữ liệu.

Nội dung chính:

1. Điều khiển DataGrid

DataGrid là một điều khiển khá linh hoạt và hiệu quả trong việc hiển thị, định dạng và thao tác với dữ liệu. Bên cạnh đó, chúng ta có thể thực hiện sắp xếp dữ liệu, thực hiện phân trang với sự hỗ trợ khá tốt của VS .Net trong quá trình thiết kế.

1.1. Các thao tác định dạng lưới

Để thực hiện các thao tác định dạng, chúng ta chọn chức năng **Property Builder...** từ thực đơn ngữ cảnh.

1.2. Trang General

Trong trang này, có các mục chọn sau:

Show header: Qui định dòng tiêu đề trên có được phép hiển thị hay không. (mặc định là có hiển thị dòng tiêu đề)

Show footer: Qui định dòng tiêu đề dưới có được phép hiển thị hay không. (mặc định là không hiển thị dòng tiêu đề dưới)

Allow sorting: Có cho phép sắp xếp dữ liệu hay không. (mặc định là không cho phép sắp xếp)



Các mục chọn trong Tab General

1.3. Trang Columns (Quản lý thông tin các cột)

Trang Columns quản lý thông tin các cột sẽ hiển thị trên lưới.

Create columns automatically at runtime: Khi chọn chức năng này, DataGrid sẽ tự động phát sinh đầy đủ các cột có trong nguồn dữ liệu. Nếu chúng ta muốn qui định các cột cần hiển thị, chúng ta không chọn chức năng này.

Column list: Qui định các cột được hiển thị trong lưới.

Bound Column: Cột có liên kết với nguồn dữ liệu.

Button Column: Cột dạng nút lệnh đã được thiết kế sẵn. Điều khiển DataGrid cung cấp cho chúng ta 3 loại cột dạng này:

Select: Nút lệnh chọn dòng dữ liệu

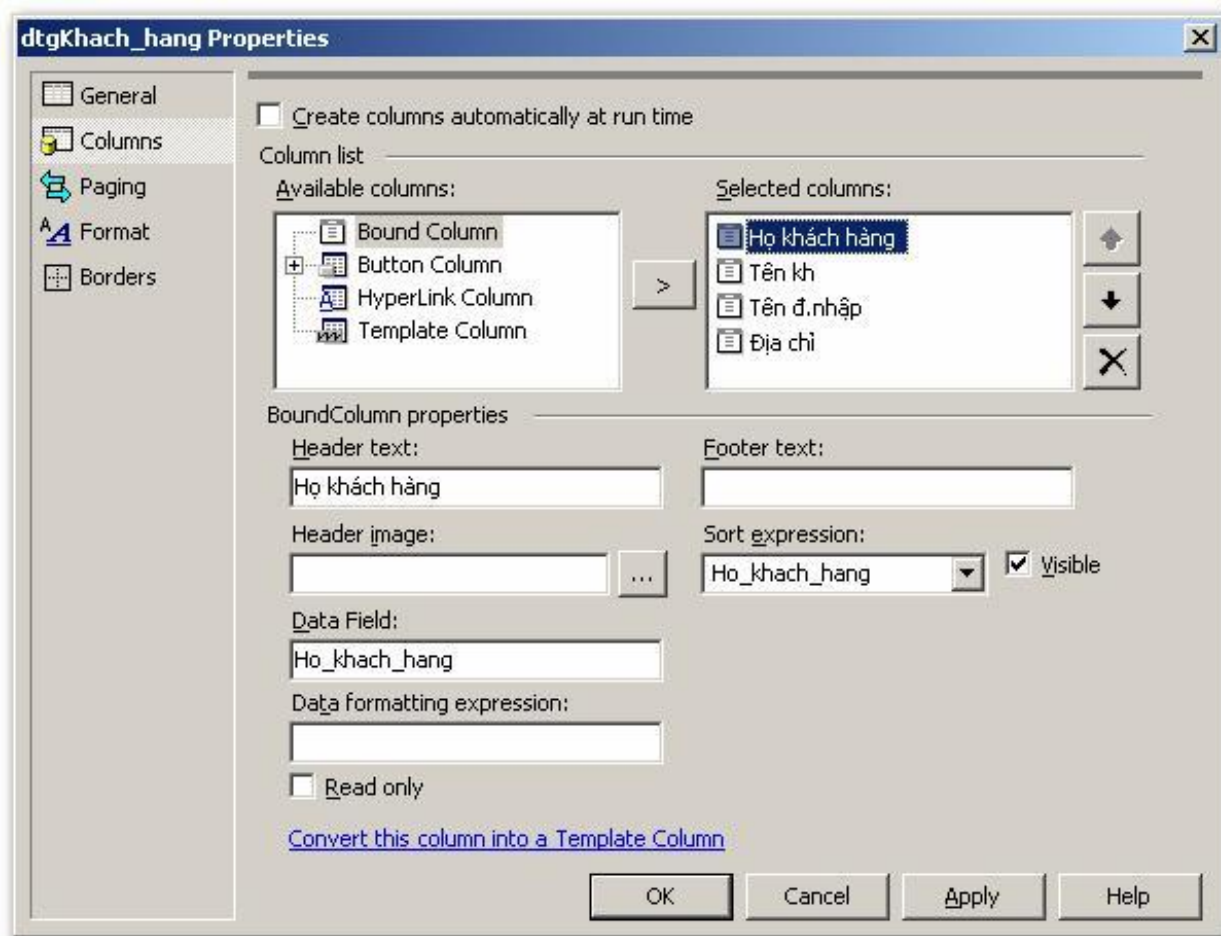
Edit, Cancel, Update: Các nút lệnh hỗ trợ chức năng cập nhật dữ liệu trực tiếp trên lưới.

Delete: Nút lệnh xóa dòng dữ liệu

Chúng ta sẽ có dịp tìm hiểu kỹ hơn về các nút lệnh này trong phần Cập nhật dữ liệu trực tiếp trên lưới.

Hyperlink Column: Cột có liên kết dữ liệu dạng liên kết.

Template Column: Cột do người dùng tự thiết kế. Đây là loại cột có khả năng làm việc khá linh hoạt.



2. Điều khiển DataList

2.1. Sử dụng DataList để hiển thị dữ liệu

Như điều khiển DataGrid, điều khiển DataList được sử dụng để hiển thị dữ liệu. Tuy nhiên, đối với DataList, chúng ta phải tự thiết kế hình thức hiển thị dữ liệu (giống như Template Column của DataGrid).

Huy Cận Về Tác Giả Và Tác Phẩm
NXB: Giáo dục



Cuốn **Huy Cận Về Tác Giả Và Tác Phẩm** tập hợp những bài nghiên cứu, phê bình của các nhà văn, nhà thơ, các cán bộ giảng dạy, các nhà nghiên cứu phê bình văn học, các nhà nghiên cứu văn hóa nước ngoài đã được công bố trên sách, báo, tạp chí. Các bài viết này được sắp xếp theo thứ tự thời gian và chủ đề, để bạn đọc có thể hình ...

Giá: 45,500.00 VND
[\[Đặt hàng\]](#) [\[Xem Tiếp\]](#)

Địa Chất Công Trình (Giáo Trình Dùng Cho Sinh Viên Ngành Xây Dựng Cầu Đường)
NXB: Giao thông vận tải



Địa chất công trình là một môn được đưa vào chương trình đào tạo kỹ sư ngành Xây dựng cầu đường của trường Đại học GTVT từ lâu. Những hiểu biết về địa chất công trình sẽ giúp ích nhiều cho kỹ sư cầu đường trong khảo sát, thiết kế và thi công các công trình giao thông ...

Giá: 14,000.00 VND
[\[Đặt hàng\]](#) [\[Xem Tiếp\]](#)

Một số thuộc tính cần chú ý của DataList: Qui định hướng hiển thị dữ liệu Horizontal: Hiển thị dữ liệu theo chiều ngang



RepeatDirection = Horizontal

f Vertical (mặc định): Hiển thị dữ liệu theo chiều đứng



2.2. Cập nhật dữ liệu với DataList

Ngoài việc hiển thị dữ liệu, DataList cũng hỗ trợ các thao tác cập nhật dữ liệu. Để thực hiện chức năng cập nhật dữ liệu với DataList, chúng ta cần phải thiết kế thêm vùng EditItemTemplate cho DataList. (xem hình)

3. Điều khiển Repeater

Như 2 điều khiển DataList & DataGrid, điều khiển Repeater cũng được dùng để hiển thị dữ liệu. Tuy nhiên, để hiển thị dữ liệu, chúng ta phải tự thiết kế hình thức hiển thị thông qua các tag HTML. Điều khiển Repeater có các tag sau:

`<HeaderTemplate></HeaderTemplate>` (tùy chọn)

Qui định hình thức hiển thị cho tiêu đề. (Chỉ xuất hiện 1 lần, phía trên của điều khiển)

`<ItemTemplate></ItemTemplate>` (Bắt buộc phải có)

Qui định hình thức hiển thị cho các mục dữ liệu trong điều khiển.

`<AlternatingItemTemplate></AlternatingItemTemplate>` (tùy chọn)

Qui định hình thức hiển thị cho các mục dữ liệu trong điều khiển. Nội dung được qui định trong cặp tag này sẽ hiển thị xen kẽ với các nội dung trong cặp tag `<ItemTemplate> </ItemTemplate>`

`<SeparatorTemplate></SeparatorTemplate>` (tùy chọn) Qui định hình thức hiển thị giữa các dòng dữ liệu

`<FooterTemplate></FooterTemplate>` (tùy chọn)

Qui định hình thức hiển thị cho tiêu đề dưới. (Chỉ xuất hiện 1 lần, phía dưới của điều khiển)