

Friday, January 23, 2004

Quyển 1

HƯỚNG DẪN LẬP TRÌNH DELPHI

Nhập môn

Phụ lục

	TỰA ĐỀ	Trang
	Giới thiệu thông tin	2
	Tóm lược môi trường lập trình Delphi	3
Bài 1	Chương trình Delphi đơn giản nhất	4
Bài 2	Sử dụng nút nhấn và thuộc tính	5
Bài 3	Xuất văn bản và sử dụng nhãn	7
Bài 4	Ra và vào của văn bản	9
Bài 5	Sử dụng lệnh IF	12
Bài 6	Adding Machine - Alter it to do [- * div] as well.	13
Bài 7	Thêm Items vào một List Box	14
Bài 8	Tạo menu - Một Text Editor - Files	15
Bài 9	Graphics, sự lặp lại sử dụng FOR và các biến	20
	Ngôn ngữ Pascal	

Lời giới thiệu

Delphi cho phép bạn tạo ra một chương trình bằng các công cụ rất mạnh. Chấn hạn như với 3 cái click chuột và không cần xử lý mã lệnh bạn có thể tạo và chạy một chương trình soạn thảo đơn giản. Công việc mà phải mất vài tuần lễ để làm được với các ngôn ngữ cũ.

Mỗi chương trình Delphi đều có 1 PROJECT (dự án), chứa đựng mọi thứ cần thiết cho chương trình. Đây là ngôn ngữ dựa trên nền tảng ngôn ngữ hướng đối tượng của Pascal (Object Pascal). Đặt một dự án trong 1 thư mục riêng biệt.

Tài liệu này yêu cầu bạn phải biết về những công cụ trong Window và làm sao để quản lý hồ sơ (File), thư mục (Folder) để bắt đầu công việc này!

Thư mục chương trình.

Đây là 1 ý tưởng tốt để các dự án Delphi của bạn vào trong thư mục mà bạn đã định (thư mục của mình). Bạn sẽ nhầm lẫn nếu có quá nhiều các dự án đặt trong các thư mục khác nhau. Việc đầu tiên là bạn hãy tạo một thư mục để cất giữ chương trình của bạn chấn hạn:

D:\Delphi\My Program

Có thể sử dụng MS DOS or Windows Explorer để tạo thư mục.

Tóm lược môi trường lập trình Delphi

Có nhiều cách để thực hiện các lệnh trong Windows. Những ví dụ đã cho bên dưới:

- 1) **Alt+F4** .Nghĩa là bạn phải nhấn giữ phím **Alt**, tiếp theo bấm **F4** và thả phím **Alt**.
- 2) **Alt | File | Save**. Bấm Alt tiếp theo F cuối cùng S.
Chữ bị gạch dưới biểu hiện là phím nóng. Bạn đừng nên giữ Alt sau khi bấm F.
- 3) Click chuột vào **File** sau đó click **Save**.
- 4) Click vào biểu tượng trên ToolBar như vậy nè:

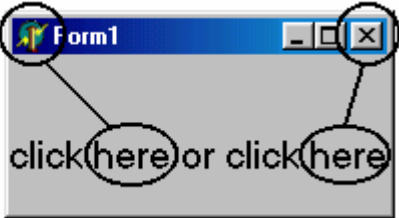


- 5) Học và sử dụng các phím tắt trong menu
- 6) Để File Save-All nó là Ctrl+V


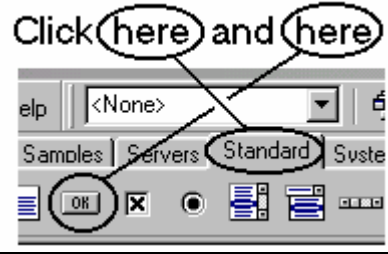
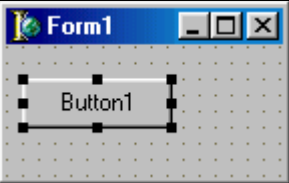
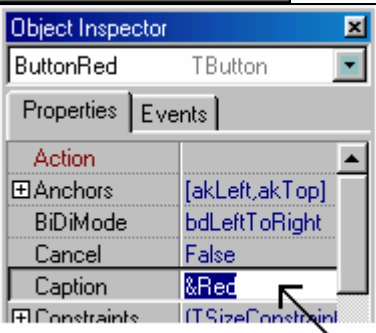
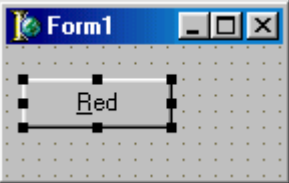
Các phím tắt cần thiết

Đóng cửa sổ hiện thời.	Alt+F4
Thoát	Alt File Exit
Chạy chương trình	F9
Ghi nhận Form thiết kế	F12
Ghi nhận thay đổi đối tượng	F11
Chuyển tới điều khiển tiếp theo	Tab
Chuyển đổi thuộc tính tính và các Tab trong đối tượng	Ctrl+Tab
Tạo xử lý được lựa chọn trong đối tượng	Ctrl+Enter
Thuộc tính có sẵn trong đối tượng	Ctrl+Enter
Thả list của Control trong đối tượng	Ctrl+Down-Arrow
Thả list của đối tượng	Alt+Down-Arrow

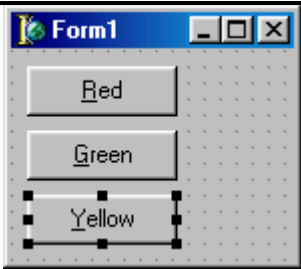
Học Bài 1 - Chương trình Delphi đơn giản nhất

Do This	Explanation
Nhiệm vụ Viết chương trình theo những bước bên dưới.	 <p>Double click here or click here</p>
Khởi động Delphi	
Tạo một dự án (project)	Alt File New Application
Bạn cần tạo thư mục này. Use MS DOS or File Manager.	Alt File Save All Hai file sẽ được lưu.
Lưu project của bạn tới đó. F:\Delphi\My Program	UNIT1.PAS PROJECT1.DPR
Chạy project. Bạn sẽ có một form trống, có thể đổi kích thước, di chuyển và đóng.	F9 Nhớ phím này.
Đóng chương trình bạn đang chạy. Nó rất quan trọng để nhớ làm nó. Delphi sẽ xử lý khác nếu bạn còn chương trình nào chạy và nó có thể “bối rối”.	Ba phương pháp có sẵn Alt SpaceBar Close Alt+F4 Click vào góc cạnh của form để đổi kích thước
Bấm F11 nhiều lần và ghi nhớ nó.	F11 F11 F11 Nhớ phím này.
Bấm F12 nhiều lần và ghi nhớ nó	F12 F12 F12 Nhớ phím này.
Lưu công việc của bạn.	Alt File Save All
Sau đó thoát từ Delphi.	Alt File Exit

Học bài 2 – Nút nhấn và thuộc tính.

Do This	Explanation
<p>Nhiệm vụ</p> <p>Viết chương trình theo chỉ dẫn dưới đây</p>	
<p>Mở F:\Delphi\lkn\Project1.Dpr</p> <p>Chọn một button bằng cách click chuột trái vào biểu tượng như hình bên</p>	<p>Alt File Open Project</p> <p>Click here and here</p> 
<p>Bấm chuột trái lên form1 và rê chuột để vẽ nút nhấn như hình bên</p>	
<p>Thay đầu đề (Caption) của nút bên phần “Properties”. Thay vì Button1, bạn hãy thay bằng “&Red” (dấu “&” để chỉ thị phím tắt, ở đây là chữ “R”). Sau đó bấm F11.</p>	 <p>Type &Red into this space</p>
<p>Bấm đúp chuột trái vào button này. Khi đó bạn sẽ làm gì, Delphi sẽ viết vài dòng lệnh Pascal cho bạn</p>	
<p>Đây là đoạn mã lệnh Delphi tạo ra. Bạn không được thay đổi mà chỉ đánh thêm lệnh thực thi vào đó</p>	<pre> procedure TForm1.Button1Click(Sender: TObject); begin end; </pre>
<p>Đánh dòng lệnh này.</p>	

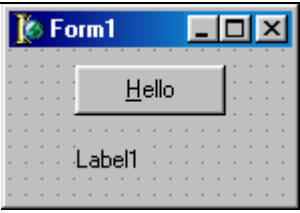
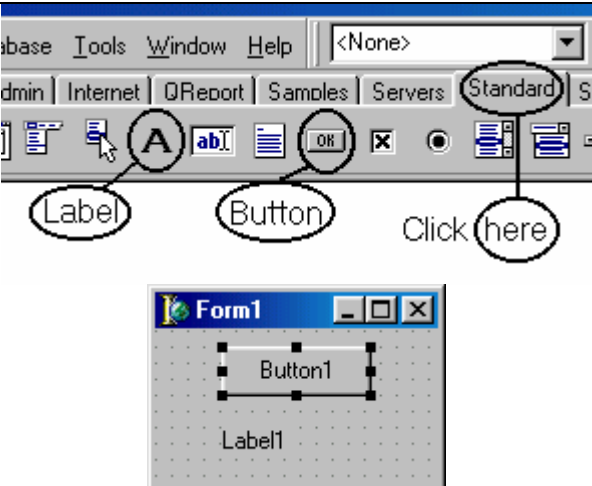
<p><code>Form1.Color := clRed;</code> cl nghĩa là màu. Nó là CLRED chứ không phải là C1RED.</p>	<p>procedure TForm1.Button1Click(Sender: TObject); begin <code>Form1.Color := clRed;</code> end;</p>
<p>Chạy chương trình và nhấn chuột trái vào nút có nhãn 'Red'. Form sẽ đổi thành màu đỏ</p>	<p>F9</p>
<p>Đóng chương trình mà bạn đang chạy</p>	<p>Alt+F4</p>

<p>Nhiệm vụ: Thêm nhiều nút lệnh theo dạng clYellow, clGreen như hình bên</p> <p>Xem các mã lệnh tạo bằng Delphi.</p> <p>NHỚ rằng bạn KHÔNG ĐƯỢC thay đổi or xóa mã lệnh tạo bằng Delphi. Bạn chỉ có thể xóa, thay đổi mã mà mình tạo. Sự khởi đầu cho việc tạo mã lệnh bằng Delphi.</p>	
<p>Lưu dự án (project) của bạn.</p>	<p>Alt File Save Project</p>
<p>Thoát khi bạn đã kết thúc</p>	<p>Alt File Exit</p>

Học Bài 3 – Nhấn và đầu ra văn bản

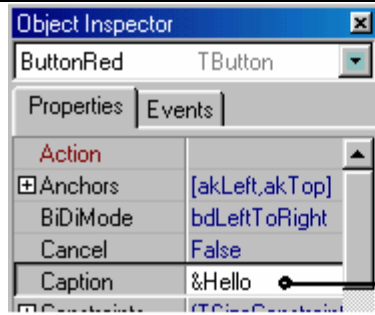
Trong bài học này, bạn sẽ viết một chương trình hiển thị văn bản (text) khi nhấn vào nút. Việc trình bày văn bản là kỹ năng cơ bản cần trong lập trình.


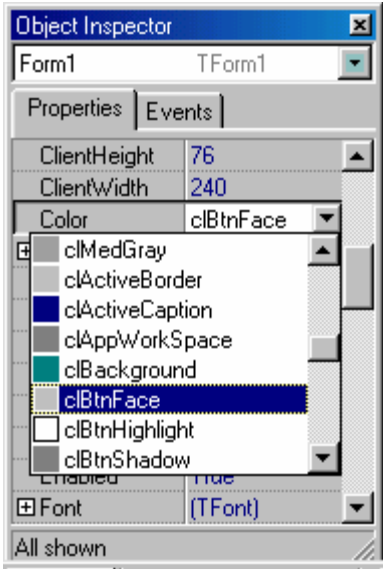
Trong bài học 2, bạn đã viết đoạn mã để trả lời khi nhấn nút. Mã này được gọi là xử lý sự kiện. Những sự kiện ở đây là những biến cố khi nhấn nút, chuột di chuyển hoặc bấm chuột. Trong bài học này, bạn sẽ kích hoạt được các sự kiện xử lý nhiều hơn.

Do This	Explanation
Nhiệm vụ Viết chương trình theo hướng dẫn dưới đây. Khi bạn nhấn nút, đoạn text 'Hello World' sẽ xuất hiện.	
Tạo thư mục cho dự án này. Use MS DOS or Windows File Manager.	Tạo thư mục này: C:\Delphi\Hello
Tạo dự án Delphi mới	Alt File New Application
Lưu dự án này tới C:\Delphi\Hello	Alt File Save All
Tạo một Button và một Label trên form này. Click vào label. Click vào form. Click vào button. Click vào form.	

Thay đổi Tiêu đề của Button1 là '&Hello' giống như bài học 2.

Thay đổi Caption của Label1 là để trống.




Bấm đúp vào nút ' &Hello ' để phát sinh xử lý sự kiện	Ghi chú Mã lệnh bên dưới gọi là Bộ xử lý sự kiện EVENT HANDLER
Đây là mã lệnh Delphi tạo. Bạn Không Thể thay đổi nhưng có thể thêm vào nó	procedure TForm1.Button1Click(Sender: TObject); begin end;
Thêm vào bộ xử lý sự kiện	procedure TForm1.Button1Click(Sender: TObject); begin Label1.Caption := 'Hello World!'; end;
Chạy chương trình. Click vào button. Đoạn text Hello World sẽ xuất hiện.	F9
Nhiệm vụ Thêm 2 button và label vào form. Làm đoạn text xuất hiện tương ứng khi nhấn nút. Thực hành nó cho đến khi bạn thấy tốt hoặc quá chán!	
Nhiệm vụ Hãy thử nghiệm các thuộc tính khác của đối tượng. Chặn hạn thay đổi màu của form tới clLime. Bạn hãy click vào mũi tên để tìm các màu trong danh sách. Phải chắn là bạn có thể thay màu, font chữ. Tất cả các nhiệm vụ này thực hiện chỉ bằng cái click chuột.	

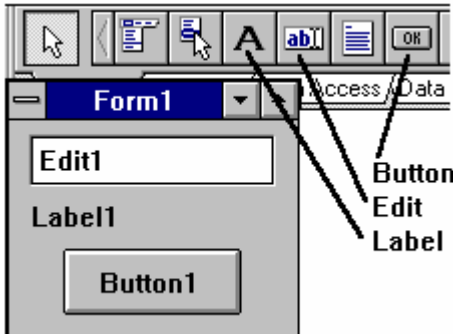
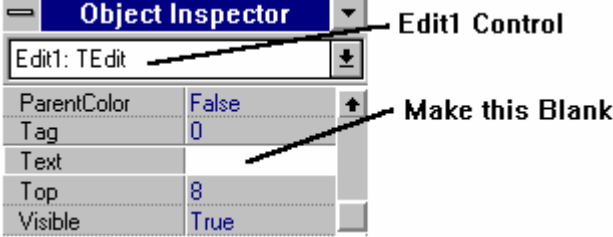
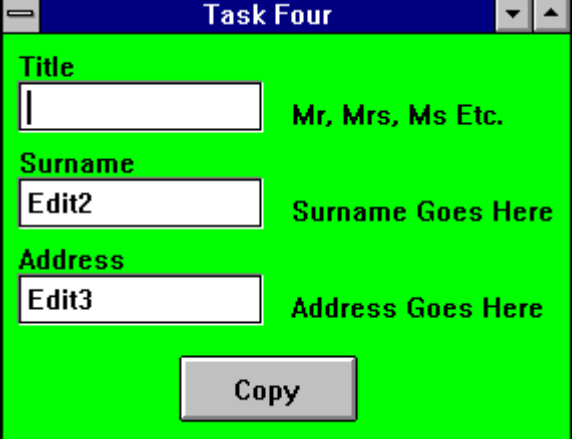
Học Bài 4 – Ra, vào Văn Bản.

Trong bài học này, bạn sẽ viết chương trình nhận văn bản từ bàn phím và in ra màn hình khi bạn nhấn nút. Đây là kỹ năng cơ bản cần trong lập trình.

Bạn cũng học làm sao để chuyển dịch những đối tượng từ form.

Bạn sẽ có thể nhầm lẫn khi thực hiện nhiệm vụ này trừ khi bạn nhớ tên các Labels, Buttons và Edit Controls của bạn

Do This	Explanation
<p>Nhiệm vụ</p> <p>Viết chương trình theo chỉ dẫn dưới đây</p> <p>Đánh text vào Edit Control (top). Khi bạn bấm nút Copy, text sẽ hiện ra trong label ở cửa sổ (hình bên)</p>	
Mở dự án đã làm lúc trước.	Alt File Open Project
Xóa tất cả Button, Label ở trong form đó.	Bấm chuột vào các đối tượng và bấm Del . Nếu các Label có thể khó thấy, bạn có thể bấm Tab để chuyển qua Label đó.
<p>Xóa các mã xử lý sự kiện cũ.</p> <p>Đây là một khó khăn, bạn hãy xóa các lệnh mà bạn đánh.</p> <p>Delphi sẽ tự động xóa các mã này nếu không dùng.</p> <p>Nếu việc xóa có sai lệch thì bạn có thể tạo dự án mới cho chắc!</p>	<p>Bấm F12 để hiện cửa sổ mã lệnh và xóa chúng.</p> <pre> procedure TForm1.Button1Click(Sender: TObject); begin Label1.Caption := 'Hello World'; end; </pre> <p>Xóa dòng xử lý sự kiện các chữ in đậm (Chữ in đậm Bold)</p> <p>Lặp lại bước này với tất cả các mã lệnh mà bạn đã tạo để xử lý sự kiện.</p> <p>Sử dụng Alt File Save Project để Delphi dọn dẹp các mã lệnh mà nó tạo ra.</p>

<p>Bây giờ chúng ta bắt đầu xây dựng chương trình.</p> <p>Tạo một Edit, 1 Label và 1 Button trên form chương trình.</p>	
<p>Bấm F11 để kiểm tra đối tượng (Object).</p> <p>Thuộc tính Text của Edit1 để trống</p> <p>Thuộc tính Caption của Label1 để trống.</p> <p>Thuộc tính Caption của Button1 là “Copy”.</p>	
<p>Bấm đúp vào nút Copy để tạo sự kiện tương ứng.</p> <p>Bạn phải nhập hàng một. Delphi sẽ tạo ra cái khác.</p>	<p>procedure TForm1.Button1Click(Sender: TObject); begin Label1.Caption := Edit1.Text; end;</p> <p>Khi bạn bấm vào nút nhấn, thì Label1.Caption sẽ thể hiện nội dung của Edit1.Text.</p>
<p>Nhiệm vụ</p> <p>Viết một chương trình với nhiều điều khiển hơn. Chương trình này phải copy Name và Address khi bạn bấm nút Copy . Xem thử nó có giống như ví dụ trên không?</p>	

Sử dụng Object Inspector để đổi tên Labels, Buttons và Edit controls như ở đây. Bạn phải kết luận với

ButtonCopy
EditAddress
EditSurname
EditTitle
LabelAddress
LabelAddressOutput
LabelSurname
LabelSurnameOutput
LabelTitle
LabelTitleOutput



Edit2 renamed to
EditSurname

Tạo sự kiện điều khiển

```
procedure TForm1.ButtonCopyClick(Sender:
TObject);
begin
    LabelTitleOutput.Caption := EditTitle.Text;
    LabelSurnameOutput.Caption :=
EditSurname.Text;
    LabelAddressOutput.Caption :=
EditAddress.Text;
end;
```

Học bài 5 - Quyết định sử dụng lệnh IF

Một kỹ năng cơ bản mà người lập trình phải làm. Đây là nhiệm vụ của bài 2 nhưng chúng ta phải thay đổi màu từ form, nó chọn 1 màu mới thay thế màu cũ.

Xây dựng lại chương trình trong bài học 2 làm cho form màu đỏ. Cho chương trình làm việc. Thấy rằng bạn có thể làm nó mà không cần sử dụng hướng dẫn này. rồi thay đổi các sự kiện như sau. Việc này tiêu biểu để sử dụng **If Statement** (lệnh IF).

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    messageBeep(0); { Tiếng bíp báo hiệu khi bạn nhấn nút}

    if form1.color = clBtnFace then { Nếu form có màu của nút}
    begin                               {khi đó form thành }
        form1.color := clRed           {màu đỏ}
    end
    else if form1.color = clRed then {nếu form không phải màu đỏ}
    begin
        form1.color := clYellow;      {chuyển thành màu vàng}
    end
    else
    begin
        form1.color := clBtnFace;
    end
end;
end;
```

Nhiệm vụ Thêm nhiều màu hơn vào sự chọn lựa đã thực hiện ở trên.

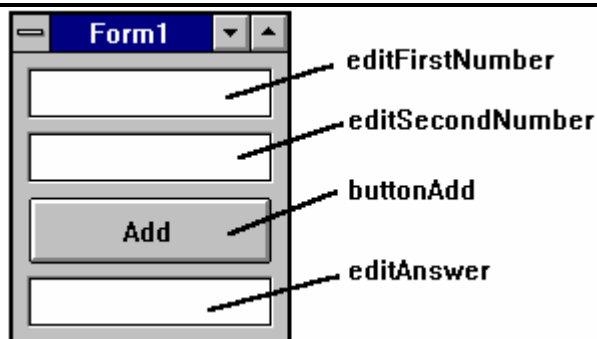
Thêm vào button thứ hai vào form để khi qua bước khác qua những màu nối tiếp với nhau.

Thay đổi lệnh “messageBeep(0)” để chương trình kêu tiếng bíp lên khi form trở lại màu đỏ.

Học bài 6 – Thêm phép tính

Nhiệm vụ

Viết chương trình chứa đựng những thành phần sau và có tên như hình bên.



Tạo sự kiện xử lý của nút nhấn như sau.

```
procedure TForm1.ButtonAddClick(Sender: TObject);  
begin  
    editAnswer.text := intToStr(    strToInt(editFirstNumber.Text) +  
                                   strToInt(editSecondNumber.Text));  
end;
```

Ví dụ này là một khó khăn chung với người lập trình. Hầu hết các ngôn ngữ đều tốt về inputting và outputting văn bản. Việc làm này cũng thế nhưng đối với những số thì không phải dễ như vậy đâu. Chương trình Pascal cũ phải chẳng đã bỏ quên văn bản được nhập khi một số được chờ đợi. Duy nhất Delphi là chấp nhận văn bản này. Có nghĩa là văn bản phải được chuyển đổi tới số tương ứng của nó. Sử dụng hàm sau đây để thực hiện việc này.

strToInt('Text Goes Here')

Hàm này có tên là **String To Integer** nhưng được rút ngắn. **String** là kiểu dữ liệu Pascal để cất giữ văn bản. Bạn có thể tìm hiểu Pascal để biết thêm chi tiết. **Integer** là kiểu dữ liệu Pascal để cất giữ những số nguyên, không bao gồm các số sau dấu phẩy của số thập phân.

Sự chuyển đổi đảo ngược được sử dụng **intToStr(Number goes Here)**. Hàm này có tên là **Integer To String** nhưng tên được rút gọn.

Nhiệm vụ Thêm ba nút lên form của bạn có nhãn là Subtract, Multiply and Divide. Những ký hiệu toán học cho Delphi như sau:

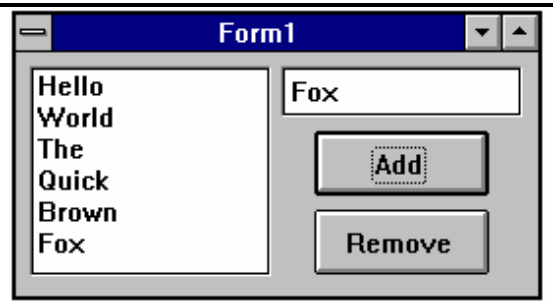
Thao tác	Ký hiệu use cho số nguyên	Ký hiệu sử dụng cho số thực
Cộng	+	+
Trừ	-	-
Nhân	*	*

Chia lấy thương	DIV	/
----------------------------	------------	----------

Học Bài 7 - Thêm Items tới một List Box

Nhiệm vụ

Thêm 1 ListBox, 1 Edit và 2 Buttons vào form. Gắn nhãn đúng cho nút. Tạo xử lý sự kiện như bên dưới. Học cách sử dụng `ItemIndex`.



Những xử lý sự kiện

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    listBox1.Items.Add(Edit1.text);
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
    listBox1.Items.Delete(ListBox1.ItemIndex);
```

```
end;
```

Nhiệm vụ: Sử dụng **Delphi Help System** để biết thông tin trên topic như **ItemIndex**.
Chèn con trỏ vào đó và bấm F1.

```
Delete(ListBox1.ItemIndex);
```



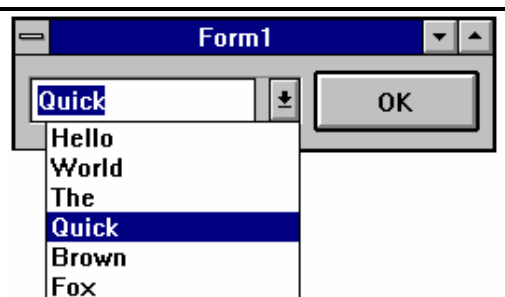
Place text cursor here and press the F1 key

Hệ thống giúp đỡ này thì khó với những người mới nhưng ta phải sử dụng thường xuyên. nếu bạn xem những chủ đề trong hướng dẫn, Bạn sẽ biết được làm cách nào để biết cách sử dụng các thành phần mà bạn chưa từng biết. Nếu bạn đã biết quá rõ những vấn đề chính thì bạn có thể lờ đi.

Nhiệm vụ

Viết chương trình tương tự như phần trên nhưng sử dụng 1 **ComboBox** thay cho 1 **ListBox**.

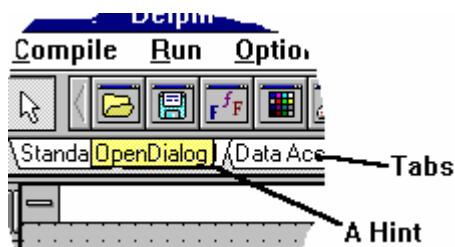
Use **Delphi Help** để làm bài này. Nếu bạn bấm OK, 'Quick' phải được thêm vào danh sách. Sau đó ComboBox phân loại danh sách. (Chỉ nhấp đúp chuột 1 lần là đủ để làm nó)



Một ứng dụng đầy đủ

Bài học này tiên tiến hơn và sử dụng nhiều đặc tính mới trong Delphi.

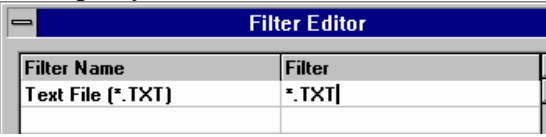
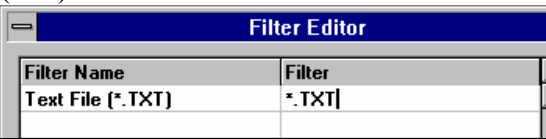
Bài này cũng sử dụng rất nhiều điều khiển. Bạn sẽ cần để tìm chúng. Nhìn vào ảnh ở dưới. Nếu giả sử chuột lướt qua nút nhận, một tip màu vàng hiện ra cho bạn biết chức năng của nút này. Nếu bạn sử dụng TABS, bạn sẽ chuyển qua điều khiển khác.



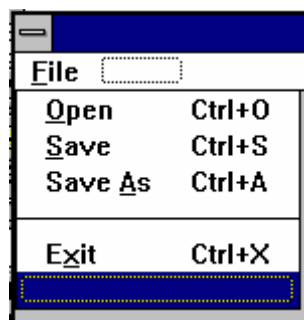
- 1 Tạo thư mục cho công việc của bạn chẵn hạn **F:\Delphi\Editor**
- 2 Khởi động Delphi
- 3 Tạo 1 dự án mới : **File | New | Project**
- 4 Lưu dự án tới **F:\Delphi\Editor**
- 5 Thêm các thành phần sau vào form của bạn và đặt các thuộc tính tương ứng.
Đổi tên các thành phần để khớp với chức năng khi đánh mã.

Ghi chú : Chạy chương trình sau mỗi bước thực hiện để phát hiện lỗi nhanh và dễ dàng sửa chữa.

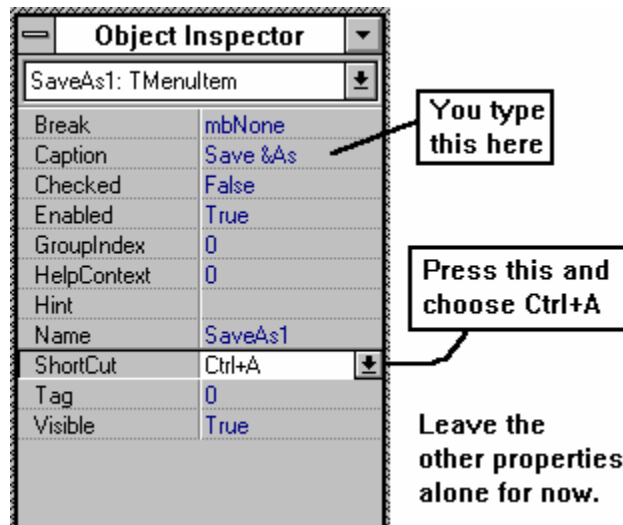
Thành phần	T. tính	Giá trị	Giải thích
Memo1	Align	alClient	
	Lines	Blank	Chọn TStrings [...] và xóa Memo1
	WantTabs	True	(Nhấp đúp chuột để làm điều này.)
MainMenu1			Không thuộc tính nào cần thay đổi.

OpenDialog1	DefaultExt	TXT	
	FileName	*.TXT	
	Filter	Text File *.TXT *.TXT	Click vào [...] để làm. Bạn sẽ điền vào cái hộp này và nhấn OK. 
	InitialDir	.	
	Title	Open A File	
SaveDialog1	DefaultExt	TXT	
	FileName	*.TXT	
	Filter	Text File *.TXT *.TXT	Click vào [...] để làm. Bạn điền vào hộp (box) và click OK. 
	InitialDir		
	Title	Save File With A New Name	
Form1	Caption	Untitled	

- 6 Click vào icon MainMenu1 để tạo menu chứa các submenu .
Đánh **Captions** và **ShortCuts (phím tắt)** như hình dưới đây.



Sử dụng '&' để xác định ký tự phím nóng khi bấm Alt và ký tự sau dấu "&". Ví dụ: **&File E&xit**
 Điền các thuộc tính của menu như hình bên.



- 7 Tạo sự kiện cho File | Exit bằng cách nhấn đúp chuột vào icon MainMenu1 và nhấn đúp vào tùy chọn menu File | Exit .

Delphi sẽ tạo các mã xử lý sự kiện sau:.

```
procedure TForm1.Exit1Click(Sender: TObject);
begin

end;
```

Bạn đánh vào đó

Close;

Nó sẽ như thế này: (thực hiện đóng chương trình)

```
procedure TForm1.Exit1Click(Sender: TObject);
begin
    Close;
end;
```

- 8 Tạo sự kiện xử lý File | Open tương tự như trên và nhấn đúp vào tùy chọn menu File | Open .

Delphi sẽ tạo mã lệnh xử lý sự kiện sau:

```
procedure TForm1.Open1Click(Sender: TObject);  
begin  
  
end;
```

- 9 Sử dụng Delphi Help System để hoàn tất bước thứ 8
Đọc và theo hướng dẫn trong đó nha bạn.

Click vào đối tượng **OpenDialog1**.
Bấm **F1** để giúp đỡ.
Click **Methods** để thấy phương pháp làm.
Click vào **Execute** là phương pháp sẵn có duy nhất.
Click vào **Example** để xem ví dụ làm sao
sử dụng phương pháp này.
Sử dụng **Edit | Copy** để sao chép những hàng
ở giữa **Begin** và **End**. Bạn cần nó giúp đỡ.
Bôi đen những dòng bạn cần và hãy bấm
Copy button. Chép những dòng sau:

```
if OpenDialog1.Execute then  
begin  
    Memo1.Lines.LoadFromFile(OpenDialog1.FileName);  
    SaveDialog1.FileName := OpenDialog1.FileName;  
    Caption := OpenDialog1.FileName;  
end;
```

Paste nó vào giữa **Begin** và **End** trong
sự kiện bạn tạo ở bài 8.

Bạn nên kết thúc lên trên với hàng này. Như vậy nè:

```
procedure TForm1.Open1Click(Sender: TObject);  
begin  
    if OpenDialog1.Execute then  
    begin  
        Memo1.Lines.LoadFromFile(OpenDialog1.FileName);  
        SaveDialog1.FileName := OpenDialog1.FileName;  
        Caption := OpenDialog1.FileName;  
    end;  
end;
```

- 10 Tạo sự kiện xử lý cho **File | Save** tương tự
như các phần trên để xử lý tùy chọn menu
File | Save.

Bạn nên kết thúc các sự kiện như thế này.
Bạn phải đánh vào giữa **Begin** and **End**.
Delphi sẽ tạo phần còn lại.

Học bài 9 - Đồ họa

Sự lặp lại sử dụng FOR và các Biến

Các Biến

Variables có tên như “pigeon holes” <ai dịch hộ với>. Đây là chương trình sử dụng biến số nguyên nơi số được cất giữ. Số này sử dụng từ 1 tới 10. Để tạo một biến bạn có thể đánh như bên dưới. Tên biến ở đây là ‘i’ nhưng bạn có thể đặt tên nào mà bạn thích và dễ nhớ cũng được.

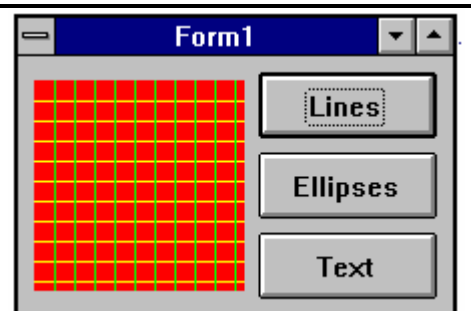
Var i : Integer;

Vòng lặp FOR

Cái này được sử dụng khi bạn muốn chương trình lặp lại một nhiệm vụ với số lượng và thời gian định trước. Ví dụ bên dưới là vòng lặp cần lặp lại 10 lần

```
For i := 1 To 10 Do           { Lặp lại 10 lần }  
Begin  
    These  
    Steps  
    Are  
    Repeated  
    Ten  
    Times  
End;
```

Nh.Vụ Tạo 1 Image và 3 Buttons lên form. Cho mỗi nút mỗi sự kiện thích hợp để xử lý, bạn hãy chép các dòng lệnh ở dưới. Sau đó hãy nâng cao lên, tạo các xử lý và các sự kiện cho riêng bạn. Sử dụng Delphi help để tìm những thủ tục đồ họa khác hơn.



Các xử lý sự kiện

procedure TForm1.Button1Click(Sender: TObject);

Var i : Integer;

begin

Image1.Canvas.Pen.Color := clRed; { Set Pen Colour }
Image1.Canvas.Brush.Color := clRed; { Set Brush Colour }

 { Colour image Red }
Image1.Canvas.Rectangle(0, 0, Image1.Width, Image1.Height);

Image1.Canvas.Pen.Color := clYellow; { Set Pen Colour }
Image1.Canvas.Brush.Color := clYellow; { Set Brush Colour }

{ Do horizontal lines }
For i := 1 **To** 10 **Do** { Repeat 10 times }

Begin
 Image1.Canvas.moveTo(0, 10 * I); { Start of line }
 Image1.Canvas.lineTo(Image1.width, 10 * I); { End of line }
End;

Image1.Canvas.Pen.Color := clLime;
Image1.Canvas.Brush.Color := clLime;

{ Do vertical lines }
For i := 1 **To** 10 **Do**
Begin
 Image1.Canvas.moveTo(10 * i, 0);
 Image1.Canvas.lineTo(10 * i, Image1.height);
End;

end;

```

procedure TForm1.Button2Click(Sender: TObject);
begin
    Image1.Canvas.Pen.Color := clBlue;           { Set Pen Colour  }
    Image1.Canvas.Brush.Color := clBlue;         { Set Brush Colour }

                                                    { Colour image Red }
    Image1.Canvas.Rectangle(0, 0, Image1.Width, Image1.Height);

    Image1.Canvas.Pen.Color := clYellow;         { Set Pen Colour  }
    Image1.Canvas.Brush.Color := clYellow;       { Set Brush Colour }

    Image1.Canvas.Ellipse(20, 20, 80, 80);

    Image1.Canvas.Pen.Color := clred;            { Set Pen Colour  }
    Image1.Canvas.Brush.Color := clRed;          { Set Brush Colour }

    Image1.Canvas.Ellipse(30, 30, 70, 70);
end;

```

```

procedure TForm1.Button3Click(Sender: TObject);
begin
    Image1.Canvas.Pen.Color := clYellow;         { Set Pen Colour  }
    Image1.Canvas.Brush.Color := clYellow;       { Set Brush Colour }

                                                    { Colour image Red }
    Image1.Canvas.Rectangle(0, 0, Image1.Width, Image1.Height);

    Image1.Canvas.Pen.Color := clred;            { Set Pen Colour  }
    Image1.Canvas.TextOut(10, 40, 'Hello World');
end;

```

Ngôn ngữ Pascal

Khu vực này giới thiệu những kiến thức cơ bản của Pascal để lập trình Delphi (vì Delphi cũng do Borland sản xuất nên các lệnh cũng tương tự như Pascal)

Các kiểu dữ liệu.

Những chương trình được sử dụng để xử lý dữ liệu. Sau đây là những kiểu dữ liệu để xây dựng thường dùng trong Pascal. Có một số thường rất ít sử dụng. Khi cần dữ liệu quan trọng để lựa chọn đúng những dữ liệu đó.

Integer (số nguyên)	Là những số như 12 or -22 -32768 tới 32767 là phạm vi sử dụng.
Real (số thực)	Những số thập phân như 22.7 or -0.0032
Char (kí tự)	Những đặc tính của các kí tự.
String[20] (chuỗi)	Cất giữ những số hay văn bản. Trong ví dụ này lên tới 20 kí tự có thể cất giữ.
Boolean (logic)	Giá trị True or False (Đúng or Sai).

Các Biến.

Variables are like named pigeon holes where data can be stored. Biến có những kiểu đặc biệt như trên. Bạn phải cất giữ đúng kiểu trong mỗi biến.

Sử dụng tên bất kì để đại diện. Điều này giúp cho chương trình dễ dàng hơn.

Good examples	VAT_Rate, dayOfWeek, TAX_Band
Bad Examples	X, L, Lemming.

Các lệnh gán giá trị

Đây là làm sao những giá trị được đặt vào trong những biển(pigeon holes). Trong ví dụ dưới, ở đây có các biển là X, sname và ok. Kí tự “:=” là đọc, trong tiếng anh, là **becomes (trở thành)**.

X := 3; X trở thành (là) 3.
Cất giữ 3 vào trong biến X.

sname := 'Bloggs'; sname là Bloggs.
Cất giữ Bloggs vào trong biến sname.

ok := True	ok là True.
	Cất True vào ok.

Những đối tượng.

Delphi là xây dựng đối tượng chương trình. Đối tượng này có thể sở hữu or chứa đựng đối tượng khác. Đối tượng có thuộc tính của nó. Trong ví dụ dưới đây

```
Form1.Label1.Caption := 'Hello World!';
```

Form1 là 1 đối tượng. Form1 chứa đối tượng khác gọi là Label1. Label1 có 1 thuộc tính gọi là Caption. Đoạn text 'Hello World!' được cất vào thuộc tính Caption của Label1. Caption là một biến để cất giữ chuỗi text. Bạn không thể cất giữ các số nguyên vào trong 1 Caption trừ khi nó được chuyển đổi vào một chuỗi trước đó.

Những thủ tục.

Những thủ tục như những chương trình nhỏ bên trong một chương trình. Những thủ tục được sử dụng để phá vỡ chương trình lớn và cho nó vào chương trình nó trong các bit. những thủ tục cho phép bạn thiết kế những chương trình bên trong được **Modular**. Điều này dễ dàng kiểm tra lỗi, kiểm tra chương trình, thiết kế, ghi chương trình nào đó. Đây là các trình bày các thủ tục

```
Procedure Lemming;  
Begin      Several lines go in here.  
End;
```

Những sự kiện.

Sự kiện là những hành động như bấm nút, click chuột.... Delphi được thiết kế để trả lời những sự kiện này. Để làm điều này bạn phải viết những bộ xử lý sự kiện.

Bộ xử lý sự kiện.

Đây là những thủ tục để đáp lại những sự kiện tương ứng. Chẵn hạn, thủ tục bên dưới trả lời tới Form1 khi Button1 được click. Trong ví dụ này, khi sự kiện xuất hiện, dạng khác gọi là tlightForm được hiện ra.

```
procedure Form1.Button1Click(Sender: TObject);  
begin  
    tlightForm.Show  
end;
```

Bình luận.

Pascal programs are usually heavily annotated with text based comments that act as reminders to the programmer of what is going on within the program. This is done so the programmer can still understand the program after not looking at it for a year or two. It is also done so a new programmer can take over from the original one when he or she moves to a new job.

BAD COMMENTS	simply repeat the code.
GOOD COMMENTS	explain what is going on and why.

X := 3;	{ X Becomes 3 (This is a bad comment) }
---------	---

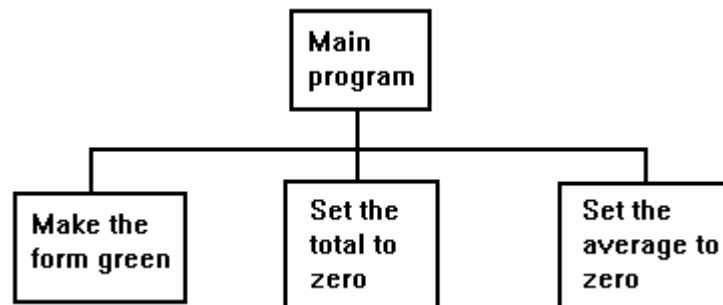
dayOfWeek := 3;	{ Store the third day (Wed) into dayOfWeek }
	{ This is a good comment }

Sự nối tiếp.

Đây là cách thực hiện biến đổi chương trình nối tiếp nhau theo một trình tự định sẵn và không lặp lại. Một ví dụ để làm rõ.

```
Form1.Color := clGreen;    { Nền của form màu xanh      }  
Total := 0;                { Tổng cộng được cất ở đây    }  
Average := 0; { Trung bình cất ở đây. }
```

Bên dưới là sơ đồ cấu trúc Jackson thể hiện sự nối tiếp đó



Sự chọn lọc.

Đây là sự quyết định bên trong chương trình. Trong Pascal, những quyết định sử dụng lệnh **If**. Chẩn hạn mã bên dưới đọc như sau trong tiếng Anh. Nếu màu của Form1 là màu đỏ thì làm cho màu của Form1 đổi thành màu xanh lá cây.

```
If Form1.Color = clRed then  
Begin  
    Form1.Color := clGreen;  
End;
```

If có thể sử dụng phức tạp hơn – ví dụ

```
If Form1.Color = clRed then  
Begin  
    Form1.Color := clGreen;  
End  
Else  
Begin  
    Form1.Color := clRed;  
End;
```

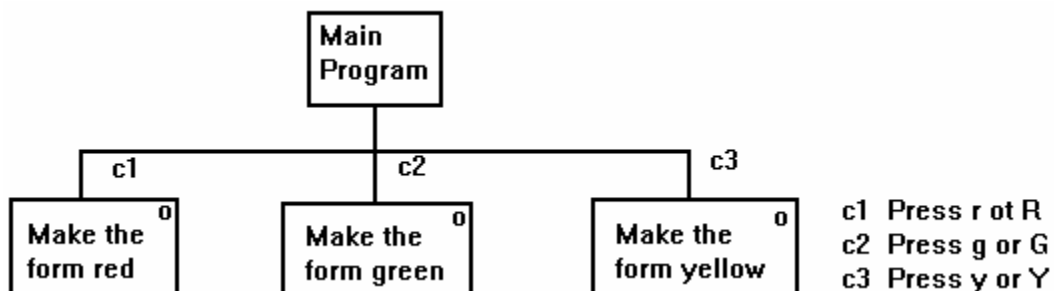
Nếu Form1 là màu đỏ thì đổi thành xanh lá cây. Nếu nó là màu khác tức không phải màu đỏ thì cho nó thành đỏ. Lệnh **If** có thể kết hợp một cách phức tạp hơn.

```
If Form1.Color = clRed then
Begin
    Form1.Color := clGreen;
End
Else If Form1.color = clGreen Then
Begin
    Form1.color := clBlue;
End
Else
Begin
    Form1.Color := clRed;
End;
```

Lệnh **If** này có thể lồng vào trong lệnh **If** khác. Trong ví dụ bên dưới, bạn ngăn ngừa chia cắt bằng zero.

```
If Op = 'Divide' then
Begin
    If N2 = 0 then
        Begin
            messageDlg(
                'You can not divide by zero.',
                mtWarning, [mbOK], 0);
        End
    Else
        Begin
            Answer := N1 div N2;
        End
    End;
```

Sơ đồ cấu trúc Jackson thể hiện sự chọn lọc này.



Vòng lặp.

Có ba cách làm cho chương trình lặp lại.

WHILE	Repeat Zero or More times.
REPEAT	Repeat One or More times.
FOR	Lặp lại 1 số lượng thời gian biết trước.

Ví dụ:

FOR

```
For I := 1 To 100 Do      { Repeat 100 times. }
Begin
  Label1.Caption := IntToStr(I);
  Label1.Update
End;
```

WHILE

```
While NOT allSorted Do   { Keep going until it is finished }
Begin
  SortTheStuff;
End;
```

DO

```
Do                                { Keep going until Finished }
  allThisStuff;                  { Unusual because Begin      }
  andAllThis;                    { and End are not needed    }
Until Finished;
```

FOR

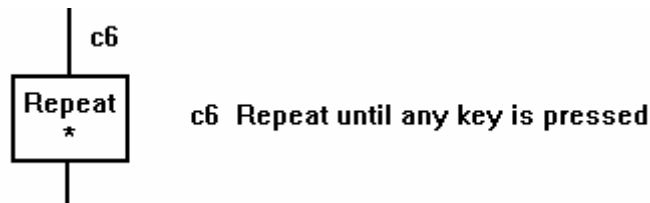
```
For Ch := 'A' To Z Do
Begin
  Label1.Caption := CharToStr(I);
  Label1.Update
End;
```

FOR Backwards

```
For I := 100 DownTo 1 Do  
  Begin  
    Label1.Caption := IntToStr(I);  
    Label1.Update  
  End;
```

```
For Ch := 'Z' DownTo 'A' Do  
  Begin  
    Label1.Caption := CharToStr(Ch);  
    Label1.Update  
  End;
```

Sơ đồ cấu trúc Jackson cho vòng lặp



Chuỗi văn bản

Từ 'Hello' là chuỗi text. Cần sáu byte để hiển thị. Chiều dài của văn bản được cất giữ từ vị trí số 0. những thuộc tính được cất giữ trong những byte kế tiếp từ 1- 5.

0	1	2	3	4	5	6	7	8	9
Length									
5	H	e	l	l	o	Wasted Space			

Để cất giữ 'Fred Bloggs' cần 12 Bytes.

11	F	r	e	d		B	l	o	g	g	s
----	---	---	---	---	--	---	---	---	---	---	---

Hàm:

Thường dùng để tính toán hoặc sản sinh ra dữ liệu chứa trong một kết quả riêng biệt. Ở đây là ví dụ về các hàm.

Sum := Add(N1, N2); { Một cách để thêm 2 số # nhau }

Text := Join(FirstBit, SecondBit);

Bigger := Max(N1, N2); { Trở lại nếu có bất cứ giá trị nào lớn hơn N1 or N2 }

Đây là ví dụ về hàm dùng để chuyển đổi một đặc tính đơn vào trong chuỗi văn bản. Nó được viết và bình luận trong thái độ phê chuẩn.

```
{
  Definition   String := CharToStr(Char)

  Input        Hàm này dùng kí tự đơn cho đầu vào (Ch).

  Process      Đặc tính nhập vào xử lý bằng cách cất giữ
               tại một vị trí trong chuỗi (S). Tại vị trí Zero
               chuỗi (S) có chiều dài (1) được cất giữ

  Output       Đầu ra chuỗi kí tự đơn chứa Ch.
}
```

Function TForm1.CharToStr(Ch : Char) : **String**;

Var S : String;

Begin

 S[0] := Chr(1); { Cất giữ độ dài xâu tại posn 0 }

 S[1] := Ch; { Cất giữ Ch tại posn 1 }

 CharToStr := S { Trả lại kết quả }

End;

Logic

AND, OR, NOT, XOR (VÀ, HOẶC, KHÔNG, XOR)

Những cái này dùng để sử dụng với **If**, **While** và **Do**. Rõ ràng trong tiếng Anh chúng có nghĩa cho nhau phải không bạn? Chúng sẽ dễ học nếu thể hiện theo kiểu bảng.

IF A AND B Then Nếu cả A và B đúng, khi đó.

IF A OR B Then Một trong A hoặc B phải đúng.

While NOT Done Do Nếu nó đúng thì làm cho nó sai
và nếu sai thì làm cho nó đúng (^_^).

If A XOR B Then Nó đúng hoặc cái khác đúng nhưng không phải cả 2
và không phải là không đúng. (???Bạn hiểu không???)

Câu tiếng Anh của nó là vậy, có ai dịch giúp với:

One or other must be true but not both and not neither

Bảng giá trị

0 là Sai (False).

1 là đúng (True).

AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

XOR		
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

NOT	
A	Output
0	1
1	0

Ví dụ:

```
If (Button = mbLeft) AND (X = 20) Then  
Begin  
    doThis;  
End;
```

```
If (day = 'Mon') OR (day = 'Tue') Then  
Begin  
    doShopping  
End;
```

```
While NOT Finished Do  
Begin  
    doThis;  
End;
```

MỜI CÁC BẠN CHUYỂN QUA QUYỂN 2 ĐỂ HỌC

BIÊN DỊCH: Lê Khắc Như

Cảm ơn các bạn đã sử dụng
Nếu bạn thấy có chỗ nào không đúng thì nhớ báo cho mình sửa lại nha
Email: laptrinh04@yahoo.ca
Website: <http://www.lkn.b4.to>