Chương 1 THIẾT KẾ BIỂU MẪU DÙNG CÁC ĐIỀU KHIỂN

Mục tiêu:

Chương này gồm các bài tập nhằm rèn luyện cho sinh viên các thao tác cần thiết cho phép thiết kế các ứng dụng đơn giản trong môi trường lập trình Visual Basic cũng như một số kỹ năng lập trình cơ bản khi làm việc với Visual Basic.

Học xong chương này, sinh viên phải nắm bắt được các vấn đề sau:

- Sử dụng các điều khiển để thiết kế giao diện trong Visual Basic.
- Vận dụng các cấu trúc lập trình trong Visual Basic để viết mã lệnh.
- Sử dụng một số cấu trúc dữ liệu trong Visual Basic.

Kiến thức có liên quan:

- Giáo trình “Visual Basic”; Chương 1, 2, 3, 4, 5.

Tài liệu tham khảo:

- **Visual Basic 6 Certification Exam Guide** - Chapter 1, Page 1; Chapter 2, Page 41; Chapter 4, Page 89 - **Dan Mezick & Scot Hillier** - **McGraw-Hill - 1998**.

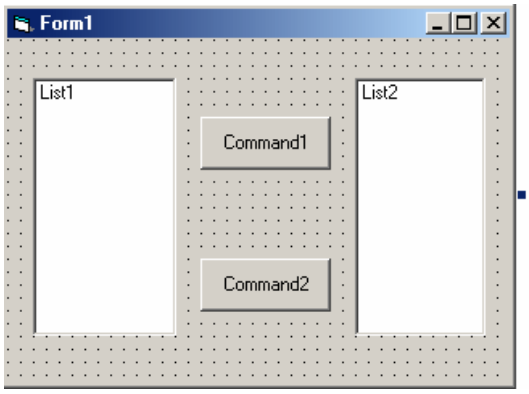
I. SỬ DỤNG MỘT SỐ ĐIỀU KHIỂN

I.1 Bài tập có hướng dẫn

Bài tập 1I-1
THAO TÁC TRÊN LISTBOX

Bước 1: Tạo thư mục Basic\Bt1-1. Tạo một dự án mới kiểu Standard EXE, lưu vào trong thư mục trên.
Bước 2: Thêm 2 List Box và một Button vào form (hình 1). Nhấn đúp lên form để mở ra cửa sổ Code, nhập các đoạn mã sau trong sự kiện Form_Load:

```
Form1.List1.AddItem "Thing 3"
Form1.List1.AddItem "Thing 2"
Form1.List1.AddItem "Thing 1"
```



Hình 1.1: Thao tác với List Box

Bước 3: Chạy ứng dụng bằng cách chọn Run/Start. List1 hiển thị 3 phần tử vừa thêm vào ở bước 2. Chấm dứt chương trình bằng cách chọn Run/End trên menu để trở về môi trường soạn thảo.
Bước 4: Nhấp đúp lên Button Command1 để hiển thị sự kiện Click của Command1.
Bước 5: Mục đích của Command1 là chuyển những phần tử được chọn từ List1 sang List2. Thêm đoạn mã sau vào thủ tục sự kiện Click của Command1:

```
' Kiem tra neu mot phan tu duoc chon
If Form1.List1.ListIndex = -1 Then Exit Sub
' Chep cac phan tu duoc chon tu List1 sang List2
Form1.List2.AddItem Form1.List1.List(Form1.List1.ListIndex)
```

Bước 6: Chạy ứng dụng. Nhấp phần tử thứ nhất của List1, sau đó nhấp Command1. Điều gì xảy ra? Phần tử được chọn của List1 phải được hiển thị bên List2. Chấm dứt ứng dụng và trở về môi trường soạn thảo.
Bước 7: Tìm trong phần trợ giúp các thuộc tính sau của ListBox:

- o ListCount
- o List

- ListIndex

Bước 8: Tìm trong phần trợ giúp các hàm sau của ListBox:
- AddItem
- RemoveItem
- Clear

Bước 9: Tìm trợ giúp cho lệnh VB:

Exit Sub

Bước 10: Đoạn mã trong thủ tục Command1_Click thực hiện thao tác chép phần tử từ một ListBox sang một ListBox khác. Bây giờ ta làm ngược lại: loại bỏ phần tử trong List1. Để làm điều này ta nhấp đúp lên Command1 và thêm dòng code sau vào cuối thủ tục:

```
' Xoa  phan tu duoc chon trong List1
Form1.List1.RemoveItem Form1.List1.ListIndex
```

Bước 11: Chạy chương trình và chọn phần tử thứ nhất trong List1. Điều gì xảy ra?

Bước 12: Nếu không chọn phần tử nào trong List1, nhấp Command1. Điều gì xảy ra? Tại sao?

Bước 13: Ta đã có một button dùng để chuyển các phần tử được lựa chọn từ trái sang phải (List1 sang List2), với button còn lại ta sẽ dùng để chuyển các phần tử được chọn từ phải sang trái (List2 sang List1).

Bước 14: Với Command2 ta sẽ copy đoạn mã từ Command1 với 1 vài thay đổi nhỏ.

Bước 15: Command2 thực hiện các thao tác giống với Command1, nhưng có nhiệm vụ di chuyển phần tử được lựa chọn từ List2 sang List1. Đoạn mã trong Command1 sẽ được sử dụng lại với một vài thay đổi nhỏ. Nhấp đúp lên Command1, chọn các mã lệnh đã thêm vào ở các bước trước. Chọn Edit/Copy trên menu.

Bước 16: Đóng cửa sổ Code và nhấp đúp lên Command2. Sự kiện Command2_Click sẽ hiển thị trong cửa sổ Code. Nhấp bất kỳ bên trong thủ tục sự kiện và chọn Edit/Paste trên menu. Như vậy ta đã chép đoạn mã từ Command1 sang Command2.

Bước 17: Sửa lại các mã lệnh vừa được chép. Thay đổi các chú thích cho thích hợp; đổi List1 thành List2 và ngược lại. Những sửa đổi này giúp Command2 có thể thực hiện thao tác chuyển các phần tử được chọn từ List2 sang List1.

Lưu các công việc đã thực hiện bằng cách chọn File/Save Project.

Bước 18: Chạy chương trình. Chọn phần tử thứ nhất trong List1 và chọn Command1 để chuyển nó sang List2. Bây giờ chọn phần tử thứ nhất trong List2, và nhấp Command2. Nếu Command2 không thực thi, trở lại môi trường soạn thảo. Kiểm tra lại đoạn mã lệnh trong thủ tục Command2_Click ta vừa chép ở bước trên.

Bước 19: Lưu ý rằng các phần tử ở cả 2 ListBox không được sắp thứ tự; nếu muốn sắp thứ tự, ta nhấp List1 và đổi thuộc tính Sorted thành True, tương tự đối với List2.

Bước 20: Lưu dự án lại và chạy chương trình. Tất cả các phần tử phải được hiển thị theo thứ tự trong cả 2 ListBox, bất chấp thứ tự chúng được thêm vào trong ListBox.

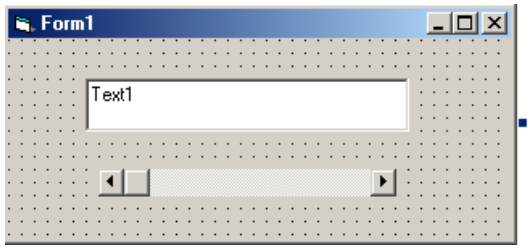
Bài tập 1I-2

THAO TÁC VỚI SPINCONTROL

Một SpinControl là sự kết hợp của TextBox và Slider. Slider tạo một miền giá trị số được hiển thị trong TextBox. Các giá trị này có thể được thay đổi bằng cách nhập trực tiếp vào trong TextBox.

Bước 1: Tạo thư mục Basic\Bt1-2. Tạo dự án mới trong thư mục trên.

Bước 2: Trong Form1, thêm một TextBox và Horizontal Scroll Bar như hình 2. Thiết lập các thuộc tính sau cho mỗi Control:



Hình I.2: Spin Control

Item1: TextBox
Name: Text1
Text: <blank>
Item2: Horizontal Scroll Bar
Name: Hscroll1
LargeChange: 10
Max: 100

Bước 3: Nhấp đúp lên scrollbar để nhập mã lệnh, đây là sự kiện Change của Scroll Bar gọi là hàm HScroll1_Change. Thêm đoạn mã sau để hiển thị giá trị hiện thời của scroll bar trong TextBox.

```
Text1.Text = HScroll1.Value
```

Bước 4: Chạy ứng dụng bằng cách chọn Run/Start trên menu. Bây giờ nhấp các mũi tên trái và phải của scroll bar. Giá trị trong TextBox phải thay đổi.

Bước 5: Bây giờ thêm mã để thay đổi giá trị bằng cách nhập trực tiếp giá trị trong TextBox. Nhấp đúp vào TextBox và thêm đoạn mã sau để thiết lập giá trị cho scroll bar khi TextBox thay đổi:

```
HScroll1.Value = Text1.Text
```

Bước 6: Chạy chương trình và nhập 50 vào TextBox. Vạch của scroll bar thay đổi theo. Thay đổi vạch của scroll bar, giá trị trong TextBox cũng thay đổi.

Bước 7: Trong khi chạy chương trình, nhập ký tự A vào TextBox. Điều gì xảy ra? Nguyên nhân vì scroll bar chỉ nhận các giá trị là số chứ không phải ký tự.

Bước 8: Để ngăn chặn những ký tự không mong muốn được nhập vào TextBox, ta sử dụng sự kiện KeyPress. Sự kiện này xảy ra khi có một phím trên bàn phím được nhấn, nhưng trước khi giá trị thực sự được hiển thị trên TextBox. Sự kiện này nhận một giá trị số nguyên của phím được nhấn, gọi là ASCII. Mỗi ký tự trên bàn phím được đại diện bằng một mã ASCII duy nhất. Do đó ta có thể kiểm tra phím nào được nhấn và bỏ qua nó nếu ta thấy không cần thiết.

Bước 9: Thêm đoạn mã sau vào sự kiện Text1_KeyPress để ngăn chặn các giá trị không phải là số.

```
' Loai bo ky tu khong can thiet
```

```
If KeyAscii = vbKeyBack Then Exit Sub
If KeyAscii < vbKey0 Or KeyAscii > vbKey9 Then
        KeyAscii = 0
End If
```

Bước 10: Lưu dự án lại và chạy chương trình.

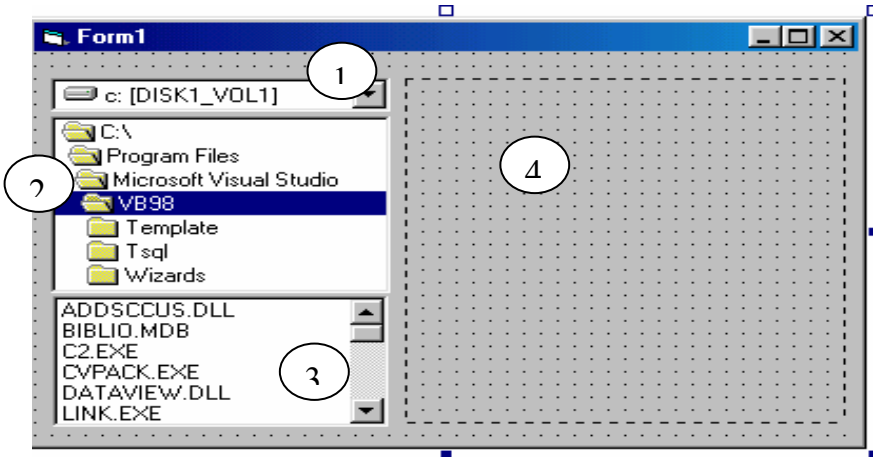
Bài tập 1I-3

THAO TÁC VỚI DRIVELISTBOX, DIRLISTBOX, FILELISTBOX

Trong ví dụ này ta phải tạo 5 đối tượng, trong đó có 4 điều khiển:

- Một Form.
- Một điều khiển DriveListBox
- Một điều khiển DirListBox
- Một điều khiển FileListBox
- Một điều khiển ImageBox

Bước 1: Tạo giao diện người dùng. Ta chỉ cần nhấp và vẽ đúng vị trí từng điều khiển trên Form.



Hình I.3: Giao diện lựa chọn tập tin hình ảnh để hiển thị

1: **DriveListBox**
Name: drvSource

2: **DirListBox**
Name: dirSource

3: **FileListBox**
Name: filSource
Pattern: *.bmp;*.wmf;*.ico;*.jpg

4: **ImageBox**
Name: ImgSource
Stretch: TRUE

Bước 2: Viết mã trao đổi thông tin giữa các đối tượng:

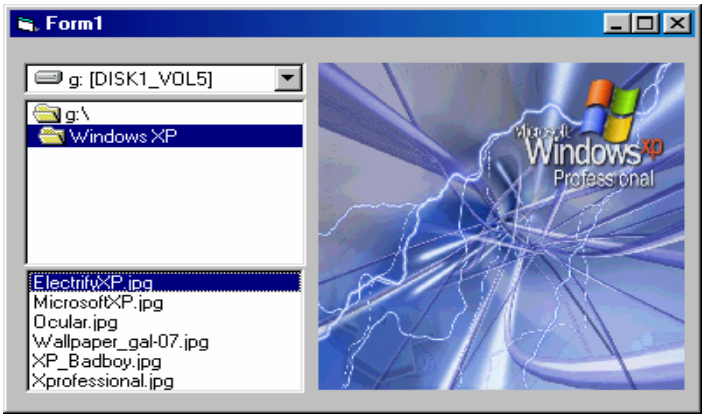
Trong cửa sổ thiết kế Form, nhấp đúp vào DriveListBox, cửa sổ Code hiện ra, xử lý sự kiện sau:

```
Private Sub drvSource_Change()
   dirSource.Path = drvSource.Drive
End Sub
```

Tương tự cho DirListBox & FileListBox

```
Private Sub dirSource_Change()
   filSource.Path = dirSource.Path
End Sub
Private Sub filSource_Click()
        imgSource.Picture = LoadPicture(filSource.Path & "\" & filSource.FileName)
End Sub
```

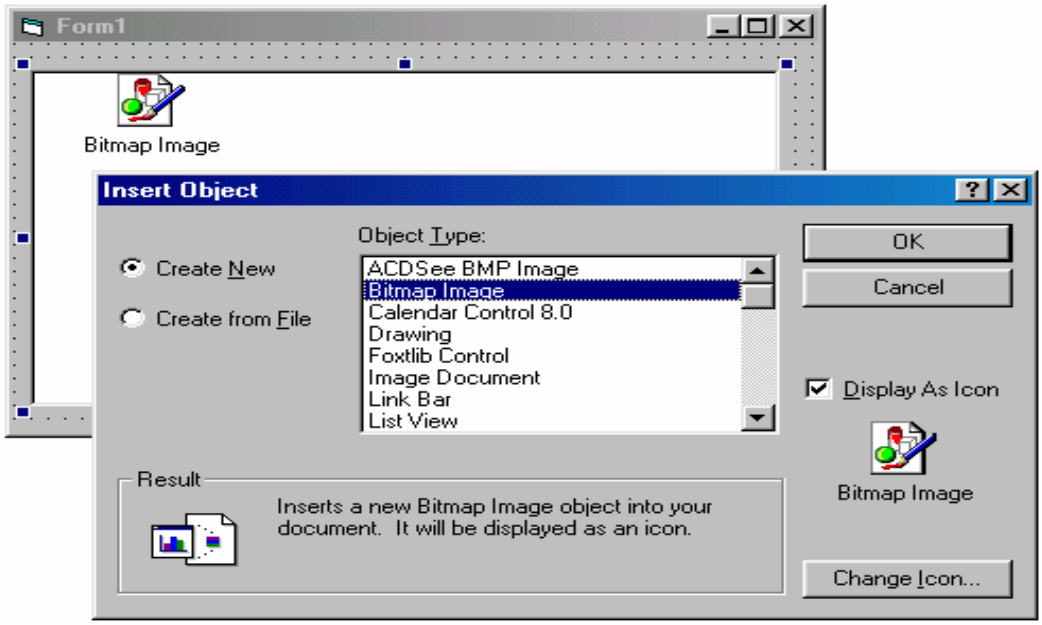
Bước 3: Lưu dự án lại vào thư mục Basic\Bt1-3. Chạy chương trình nhờ phím F5.



Hình I.4: Kết quả thực thi

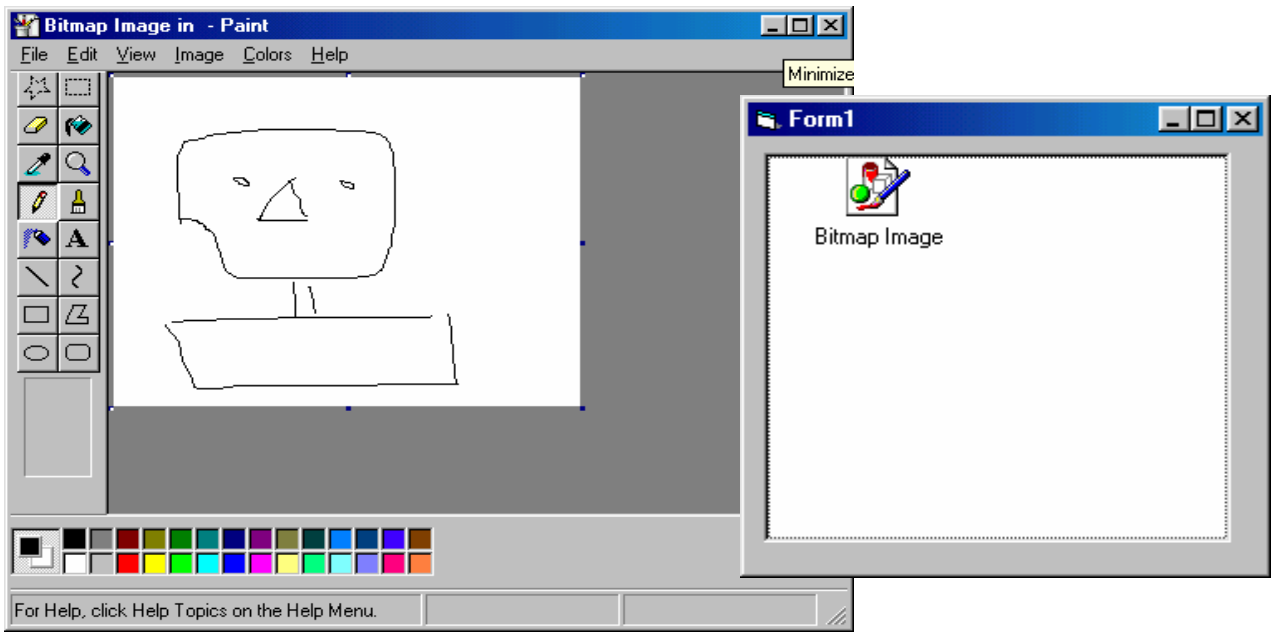
Bước 1: Tạo dự án mới, trong đó ta có sử dụng OLE.

Hộp thoại **Insert Object** hiện ra để ta lựa chọn, ở đây chọn kích hoạt **Create New**, **Object Type** là **Bitmap Image**; đánh dấu chọn **Display as Icon**.



Hình I.5: Sử dụng OLE Control

Bước 2: Nhấp OK, VB sẽ gọi trình ứng dụng Paint & ta vẽ hình trên cửa sổ Paint. Sau đó chọn **Exit & Return** trong cửa sổ Form, ta được:

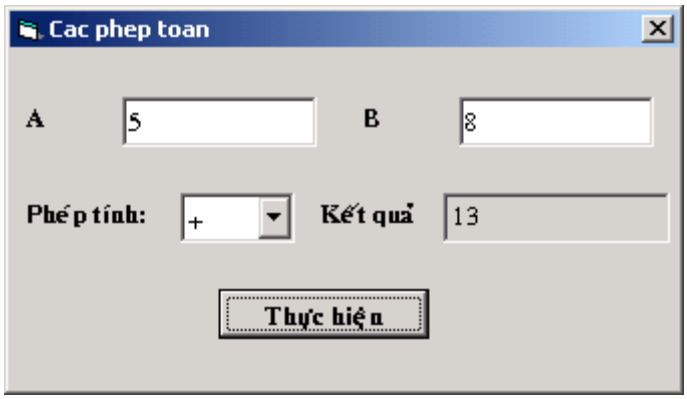


Hình I.6: Kết quả thực thi ứng dụng

Bước 3: Lưu dự án vào thư mục Basic\Bt1I-4 và chạy chương trình; nhấp đúp vào biểu tượng Bitmap Image,VB sẽ khởi động Paint để ta hiệu chỉnh hình vẽ đầu.

I.2 Bài tập tự làm

1) Thiết kế chương trình như sau:



Hình I.7 Các phép tính cơ bản

Nhập vào 2 giá trị A, B; sau đó chọn một phép toán (+,-,*,/). Nhấp chọn nút nhấn **Thực hiện**, kết quả sẽ hiển thị trong điều khiển nhãn **Kết quả**.

2) Thiết kế chương trình để nhập vào tọa độ của hai điểm (x1,y1); (x2,y2) và cho phép:

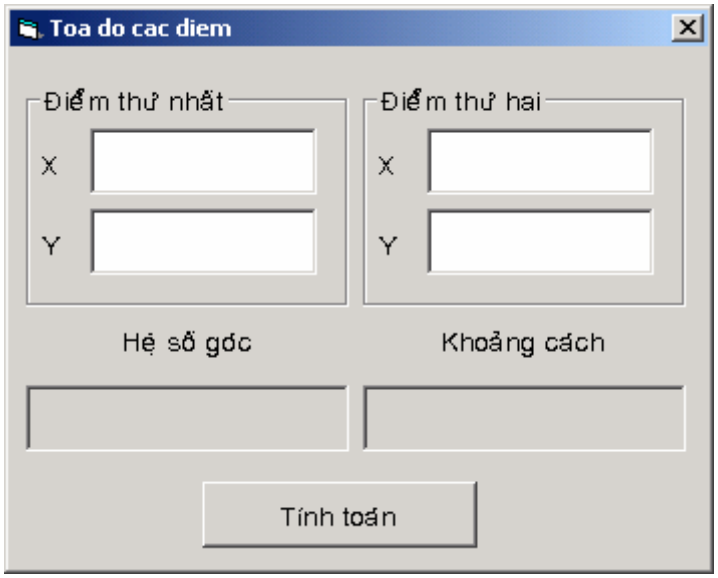
a) Tính hệ số góc của đường thẳng đi qua hai điểm đó theo công thức:

Hệ số góc = (y2 - y1) /(x2 - x1)

b) Tính khoảng cách giữa hai điểm theo công thức:

khoảng cách = $\sqrt{(x2-x1)^2+(y2-y1)^2}$

Giao diện chương trình có thể như sau:



Hình I.8: Tọa độ các điểm

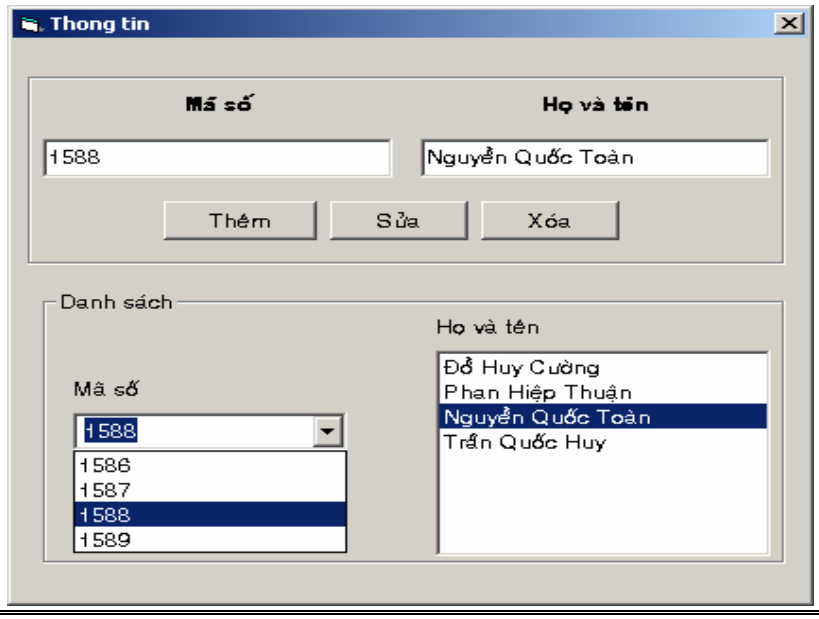
3) Viết chương trình cho phép nhập vào bán kính r của một hình tròn. Tính chu vi và diện tích của hình tròn theo công thức :

Chu vi CV = 2*Pi*r

Diện tích Dt = Pi*r*r

Hiển thị các kết quả lên màn hình.

4) Thiết kế chương trình có giao diện như hình dưới và thực hiện các chức năng sau:



Hình I.9: Lựa chọn tên

- Mỗi khi người sử dụng chương trình nhập thông tin vào 2 ô TextBox, sau đó nhấp chọn nút **Thêm**, giá trị của ô **Mã số** được đưa vào ComboBox, còn giá trị của ô **Họ và tên** được đưa vào ListBox.
- Mỗi khi họ chọn một **mã số** nào đó trong ComboBox, giá trị **họ và tên** tương ứng cũng sẽ được chọn trong ListBox; đồng thời chúng sẽ được hiển thị lên trên các điều khiển TextBox tương ứng (như hình). (*Xử lý sự kiện Combo1_Click & List1_Click*)
- Đối với **mã số** và **họ tên** của một người, ta có thể sửa đổi giá trị của chúng trong các ô nhập TextBox, sau đó chọn nút **Sửa**, giá trị của chúng trong ComboBox & ListBox cũng sửa đổi theo.
- Khi người dùng *chọn một mã số* (hay họ tên) trên ComboBox (hoặc ListBox), sau đó họ chọn **Xóa**, các thông tin này được xóa ra khỏi ComboBox & ListBox.

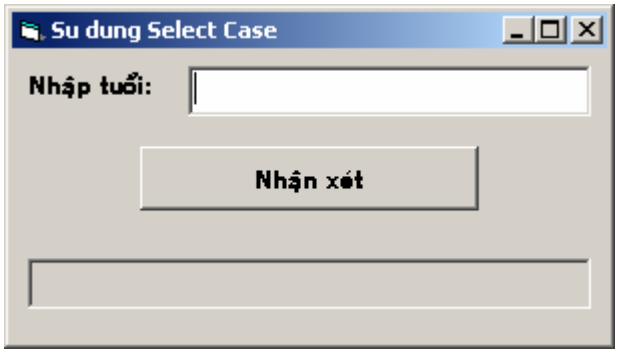
II. CÁC CẤU TRÚC LẬP TRÌNH TRONG VB

II.1 Bài tập có hướng dẫn

Bài tập 1II-1

SỬ DỤNG SELECT CASE

Tạo thư mục Basic\Bt1II-1. Thiết kế chương trình có giao diện & lưu trong thư mục trên:



Hình I.10: Select Case

Ở đây, người sử dụng chương trình nhập vào một tuổi nào đó trong ô nhập tuổi, sau đó họ nhấp nút **Nhận xét**, một nhận xét sẽ xuất hiện ứng với tuổi mà họ nhập từ bàn phím.

Lúc này ta sử dụng toán tử so sánh (=, <, <=, >, >=, <>) cùng với các từ khóa **Is** và **To** trong biểu thức.

Is: so sánh biến với biểu thức được liệt kê sau từ khóa **Is**.

To: định nghĩa phạm vi của giá trị.

Sự kiện Command1_Click():

```
Dim Age As Integer
Age = Val(Text1.Text)
```

```
     Select Case Age
          Case Is < 18
             Label2.Caption = "Ban con thieu nien, ban phai hoc
thoi!"
          Case 18 To 30
             Label2.Caption = "Ban da truong thanh, lap gia dinh
thoi!"
          Case 31 To 60
             Label2.Caption = "Lua tuoi trung nien roi!"
          Case Else
             Label2.Caption = "Ban co con chau day dan roi
nhe!"
     End Select
```

Bài tập 1II-2

BIẾN VÀ CẤU TRÚC

Bước 1: Tạo thư mục Basic\Bt1II-2. Tạo dự án mới (VB Standard EXE) trong thư mục trên; thêm một modul vào dự án, trong modul nay thêm vào đoạn mã sau:

```
Public Const tieude As String = "Quan ly hanh chinh"
Public Const sohieu As String = "1.0"
```

Thêm đoạn mã sau vào hàm xử lý sự kiện Form_Load của Form1:

```
Form1.Caption = tieude & " phien ban " & sohieu
```

Chạy ứng dụng, ta thấy tiêu đề của Form: "Quan ly hanh chinh phien ban 1.0".

Bây giờ, mở Modul1 và thay Public bằng Private. Chạy chương trình. Điều gì xảy ra?

Bước 2: Đổi các khai báo trên thành Public, thêm dòng sau đây vào đầu thủ tục Form_Load:

```
tieude = "Loi xuat hien"  & "Hang so khong the thay doi duoc."
```

Chạy chương trình, điều gì xảy ra?

Bước 3: Thêm dòng sau trong hàm xử lý sự kiện Form_Resize:

```
MsgBox "FORM RESIZE"
```

Bước 4: Chạy chương trình, khi Form bắt đầu được hiển thị (sự kiện Form_Load), sự kiện Resize của Form được thực hiện. Chỉ có hàm xử lý sự kiện Resize mới cho biết chắc rằng hàm Form_Load được thực thi. Để kiểm chứng ta tạo một biến trên form và trong hàm Form_Load ta thiết lập giá trị của nó. Sau đó, hàm Form_Resize có thể kiểm tra biến và xử lý trên biến này.

Bước 5: Khai báo một biến Private trong Form1 tên *sukienLoad*:

```
Private sukienLoad As Boolean
```

Trong hàm Form_Load, đặt giá trị True cho biến trên:

```
sukienLoad = True
```

Bây giờ ta kiểm tra giá trị của biến trong hàm Form_Resize. Thêm vào đoạn mã sau trong hàm Form_Resize:

```
If sukienLoad = True Then
      SukienLoad =False
      Exit Sub
End If
MsgBox "Form Resize"
```