



Lập trình Visual Basic



Tổng số tiết: 45 tiết

Lý thuyết: 20 tiết - Thực hành: 15 tiết

Tự nghiên cứu: 10 tiết

Nội dung chương trình

PHẦN 1: TỔNG QUAN VỀ NNLT VISUAL BASIC

PHẦN 2: CÁC ĐỐI TƯỢNG KIỂU DỮ LIỆU, BIẾN, BIẾN HẲNG

PHẦN 3: CÁC ĐỐI TƯỢNG THƯỜNG DÙNG

PHẦN 4: MẢNG – MẢNG CÁC ĐIỀU KHIỂN



Bài giảng VB 1

Tài liệu tham khảo

- Tự học ngôn ngữ lập trình VB 6.0
- Kỹ năng lập trình Visual Basic 6.0
- Các tài liệu về NNLT VB trên internet
- Website:
 - www.vovisoft.com
 - www.quantrimang.com
 - www.edu.net.vn
- CD:
 - Souce code VB 6.0
 - MSDN For VB

Lập trình visual basic

5t

Phần 1

TỔNG QUAN VỀ NNLT VISUAL BASIC



Bài giảng VB 1

Nội dung chính

- Giới thiệu về ngôn ngữ lập trình VB 6.0
- Các thành phần trên màn hình
- Cấu trúc một chương trình viết bằng VB
- Tạo một project đầu tiên
- Tổ chức lưu trữ một project
- Thực thi chương trình
- Ví dụ minh họa

1. Giới thiệu về ngôn ngữ lập trình VB 6.0

- Tham khảo tài liệu

2. Các thành phần trên màn hình

1

2

3

4

5

Project1 - Microsoft Visual Basic [design] - [FrmMayTinhTay (Form)]

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

240, 960 615 x 495

General

FrmMayTinhTay

Project - Project1

Project1 (ProMayTinhTay.v

Forms

FrmMayTinhTay (FrmMa

Modules

MdlMayTinhTay (MdlMa

Properties - Command1(1)

Command1(1) CommandButton

Alphabetic Categorized

(Name)	Command1
Appearance	1 - 3D
BackColor	8H800000F
Cancel	False
Caption	7
CausesValidation	True
Default	False
DisabledPicture	(None)
DownPicture	(None)

Immediate

```
?date  
12/9/2005  
?now
```

Caption

Returns/sets the text displayed in an

Form Layout

start

Bai Giang VB 6.0

Chuong 1 - Mi...

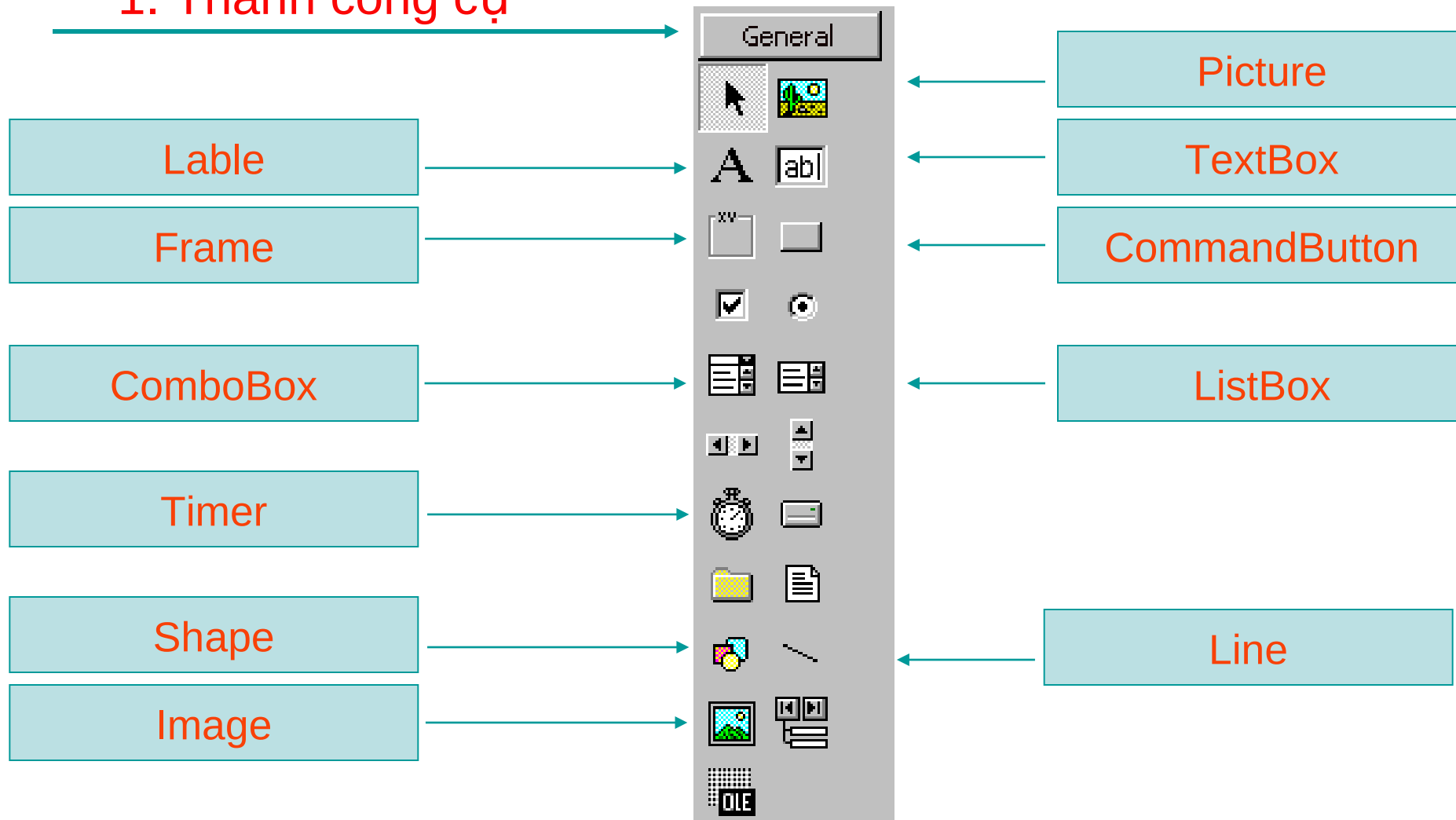
Project1 - Mic...

Desktop

7:49 AM

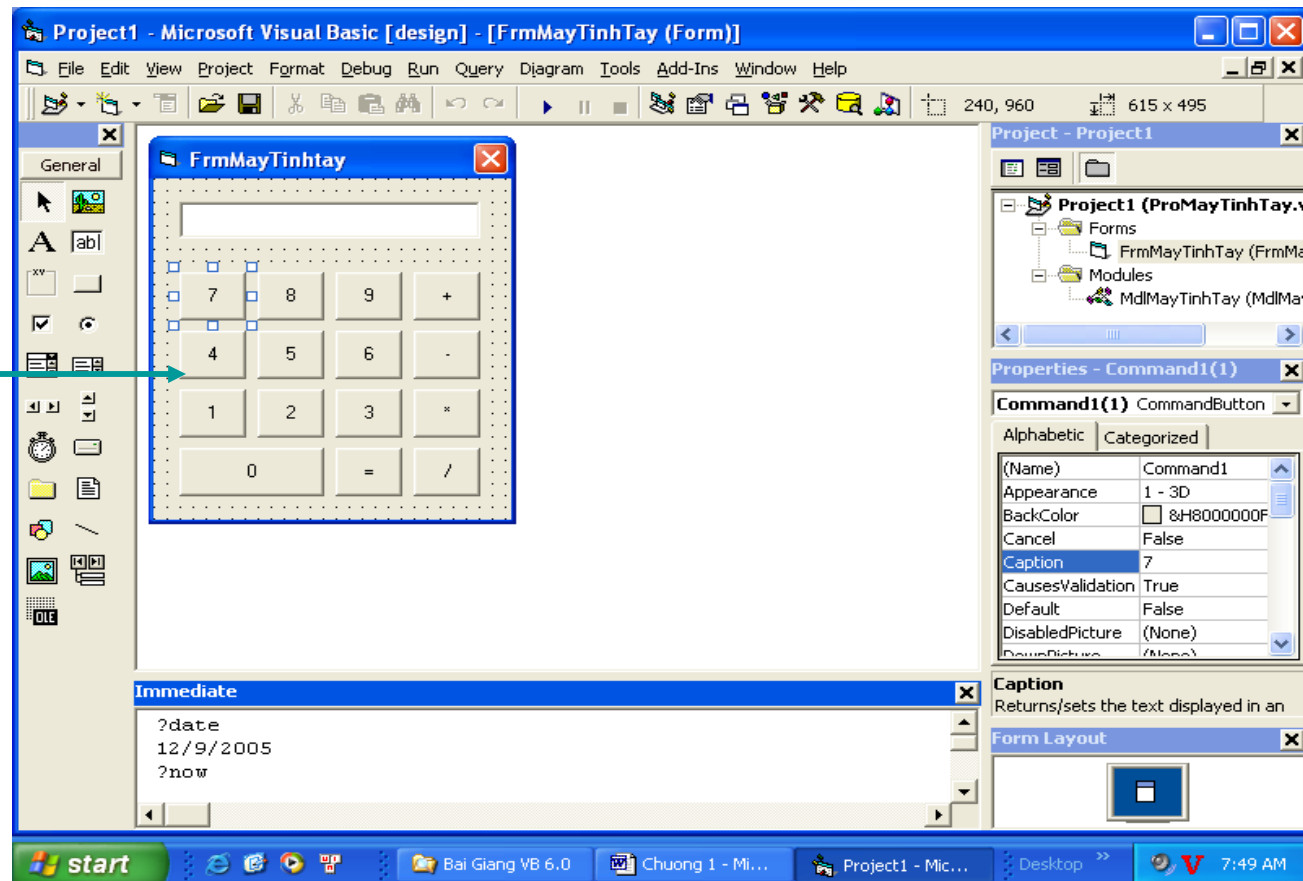
2. Các thành phần trên màn hình

1. Thanh công cụ



2. Các thành phần trên màn hình

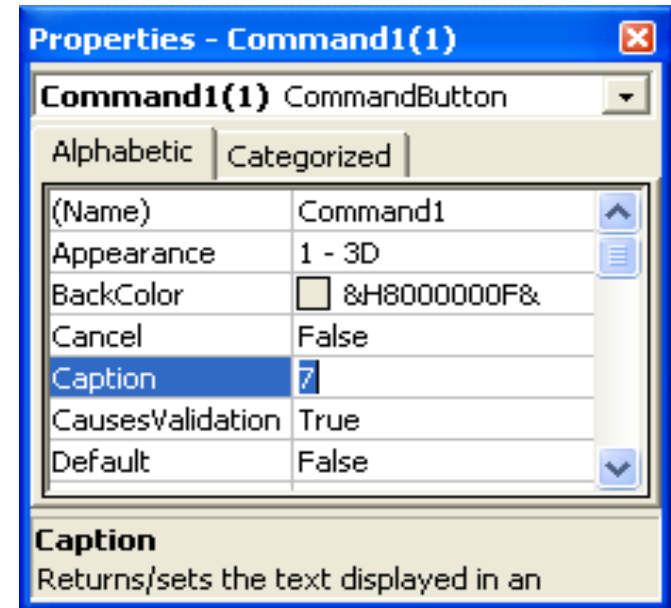
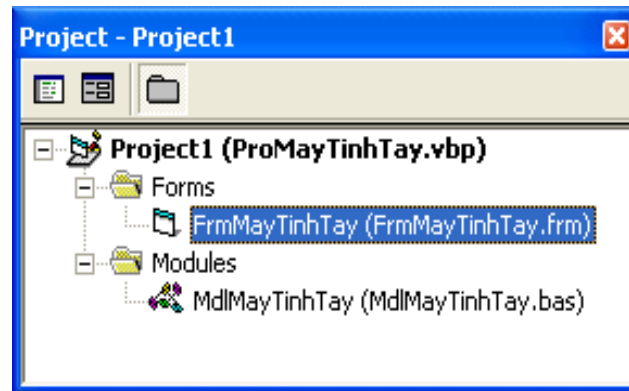
2. Form design



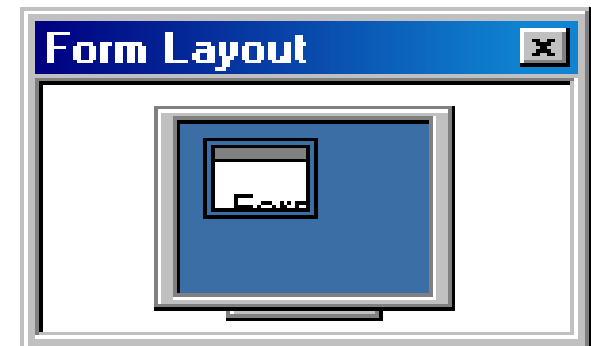
2. Các thành phần trên màn hình

4. Properties windows

3. Project explorer

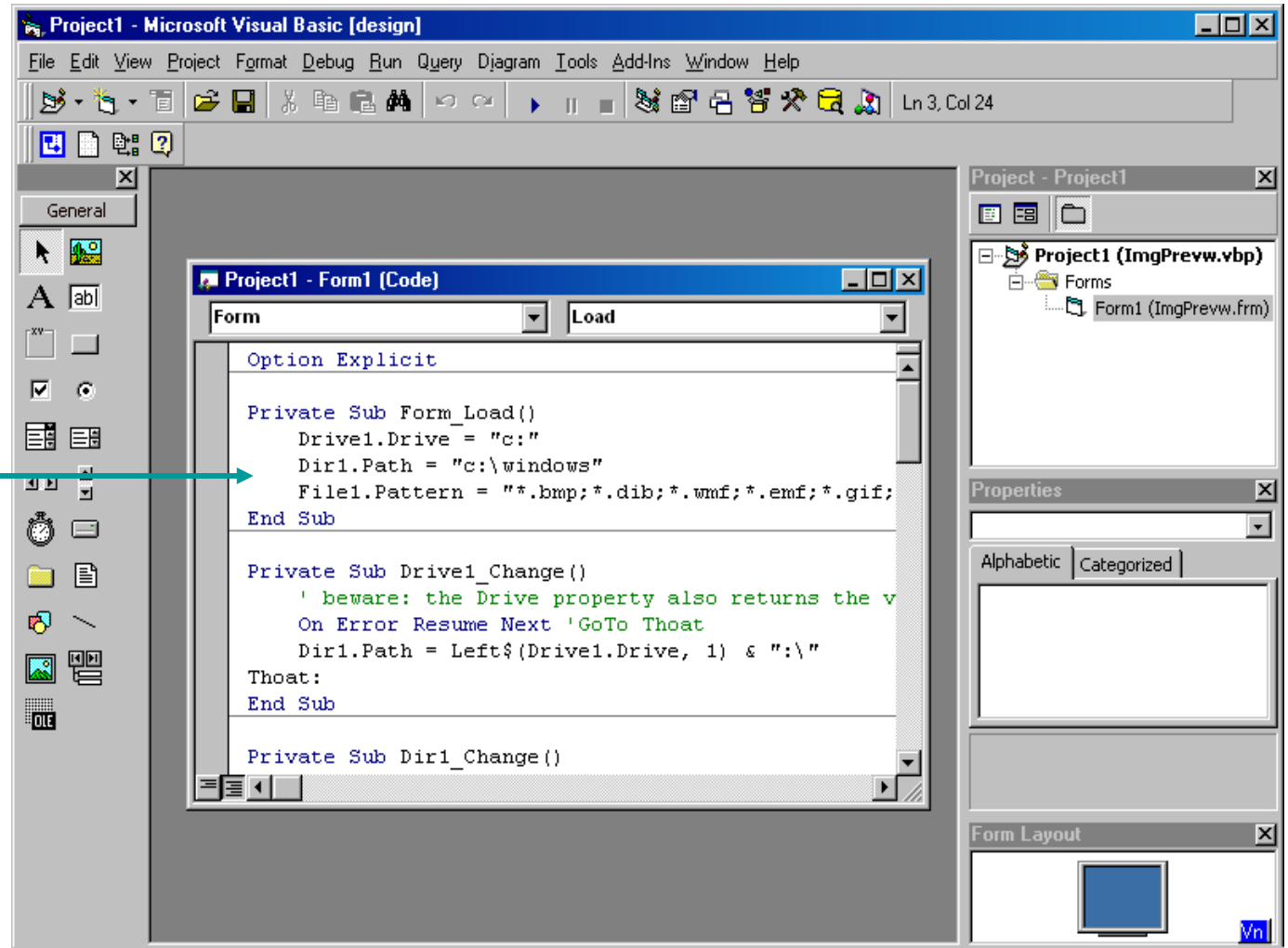


5. Form layout



2. Các thành phần trên màn hình

Cửa sổ viết code



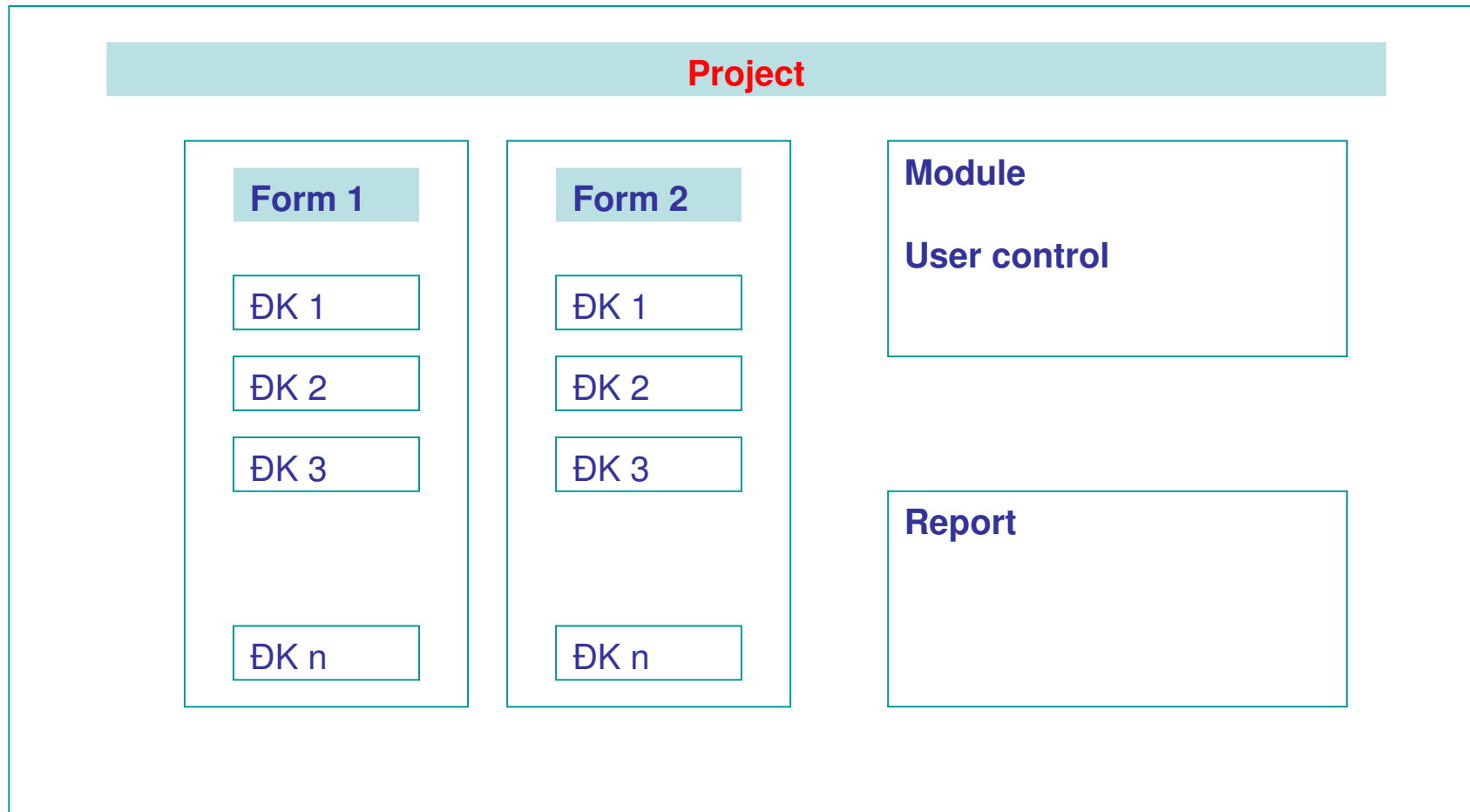
2.1 Các khái niệm cơ sở

- **Màn hình giao tiếp (Form):** Nơi thiết kế giao diện chương trình để giao tiếp với người sử dụng.
- **Điều khiển (Control):** Các thành phần sẽ được vẽ lên form. Ví dụ: Nhãn, ô nhập dữ liệu, nút lệnh...*(Điều khiển nằm trên hộp công cụ).*
- **Thuộc tính (Properties):** Tập hợp các thông tin liên quan đến đối tượng. Ví dụ: Điều khiển ô nhập dữ liệu được mô tả như sau: tên là txtThongtin, màu chữ là đỏ, màu nền là trắng... Những thông tin như: tên, màu chữ, màu nền chính là thuộc tính của đối tượng ô nhập dữ liệu.

Các khái niệm cơ sở

- **Phương thức (Method):** Phương thức của đối tượng là tập hợp các hành động của đối tượng. Ví dụ: Khi cần di chuyển nút lệnh từ vị trí này đến vị trí khác ta dùng phương thức move.
- **Biến cố (Event):** Những gì phát sinh với đối tượng khi ứng dụng đang thực thi.

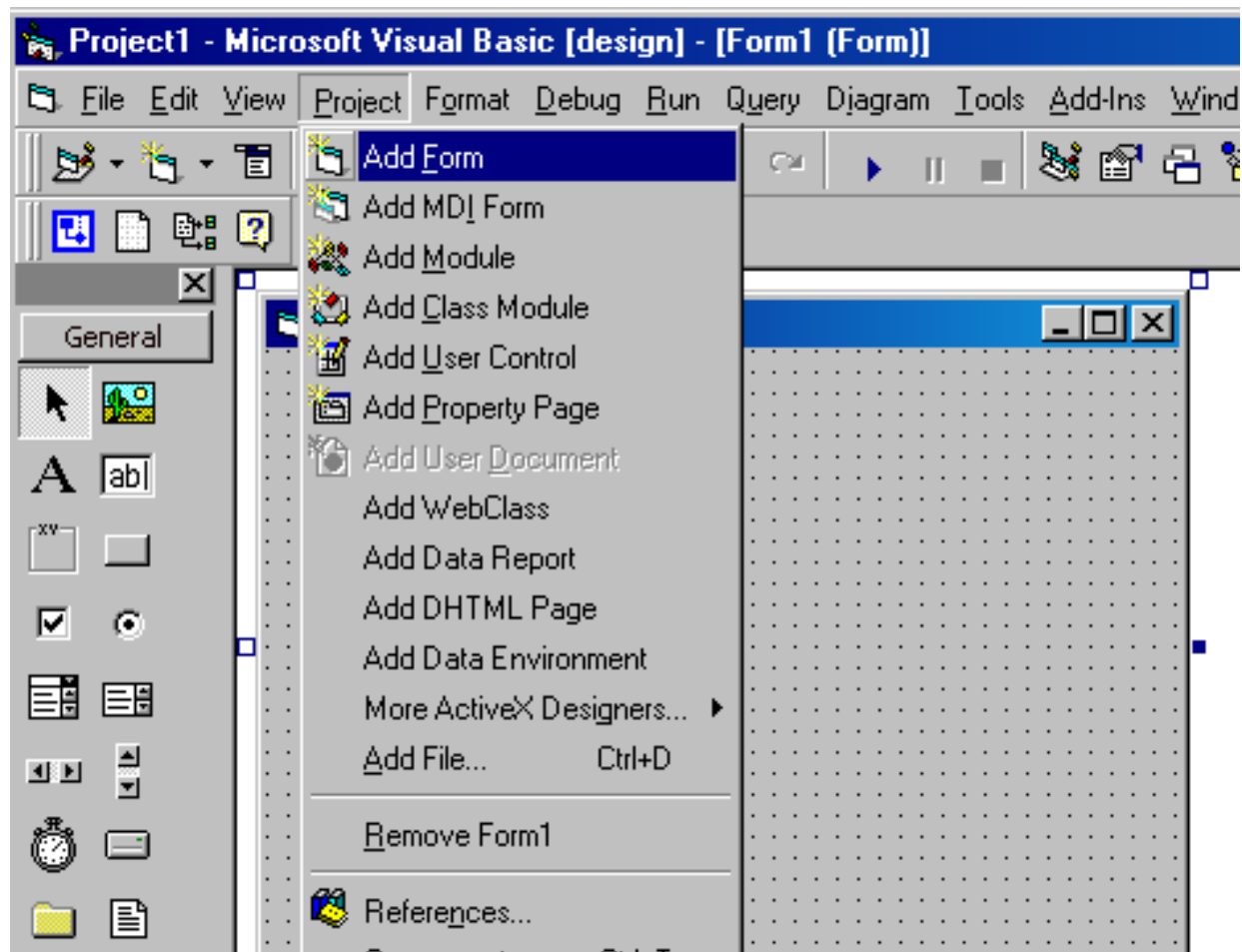
3. Cấu trúc chương trình VB



4. Tạo một project đầu tiên

- Tạo project
- Tạo Form
- Tạo module
- Tạo report
- Tạo user control

4.1 Tạo Project

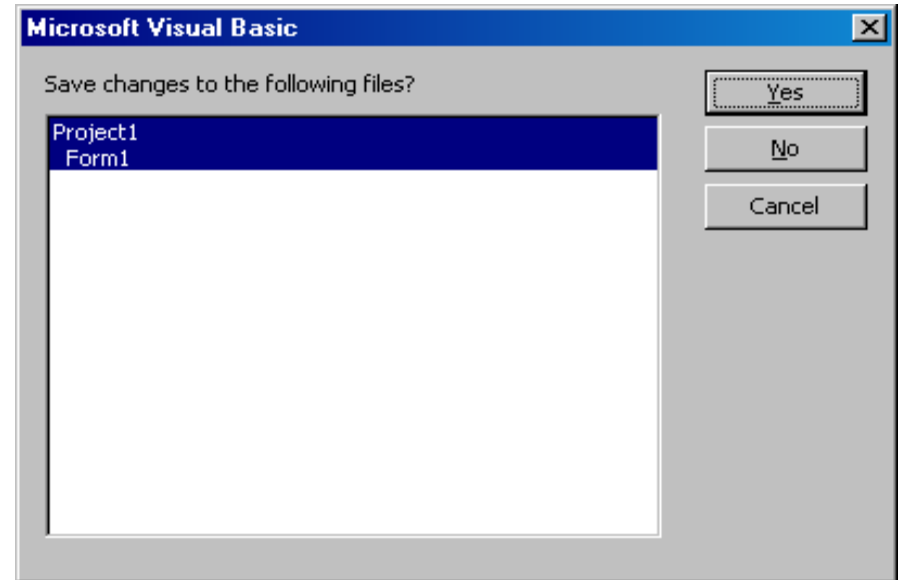


Bài giảng VB 1

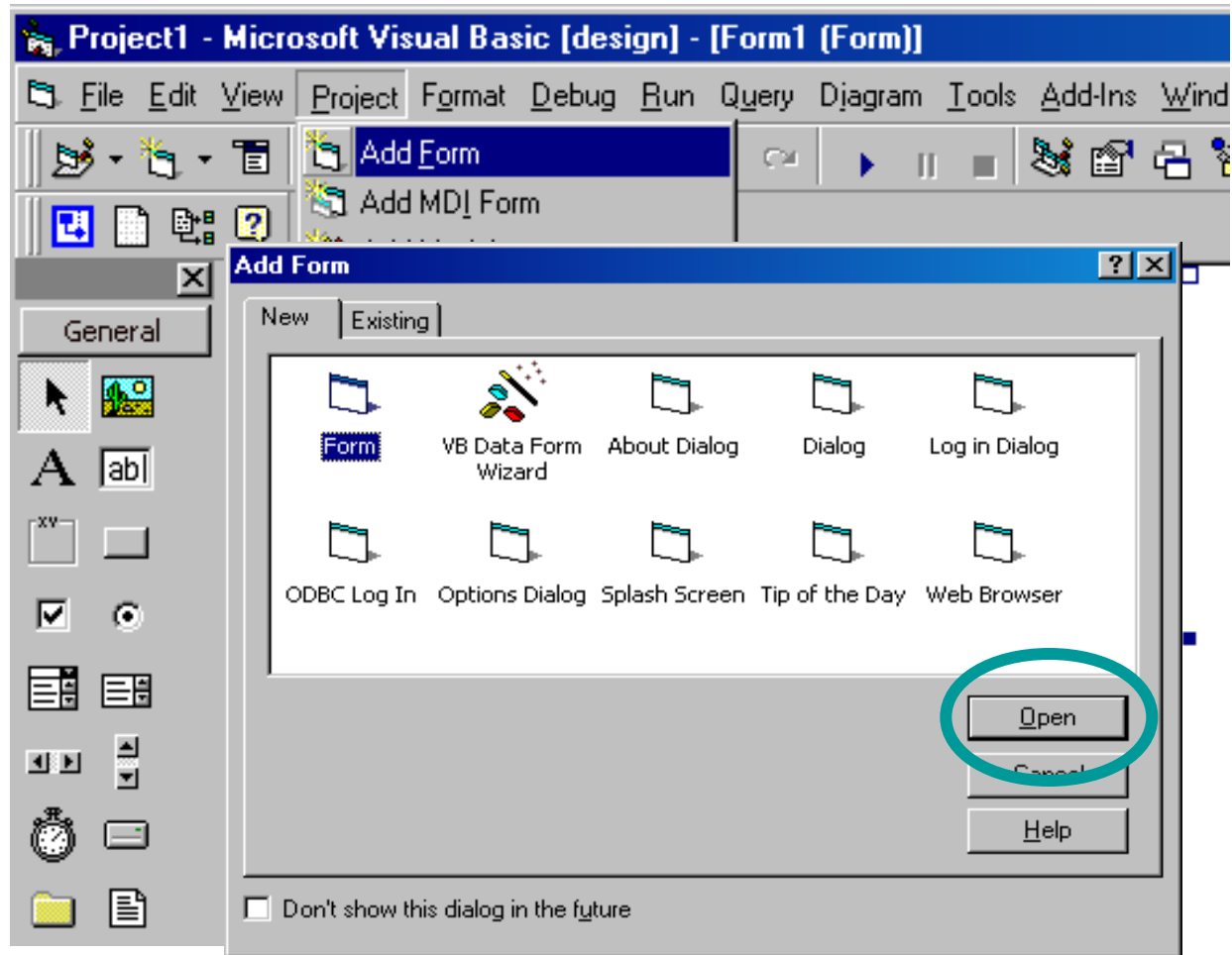
4.1.1 Lưu ý khi tạo Project

Khi tạo project mới, chương trình sẽ hỏi bạn có lưu dự án đang thực hiện không?

- Đồng ý lưu chọn **Yes**
- Ngược lại chọn **No**

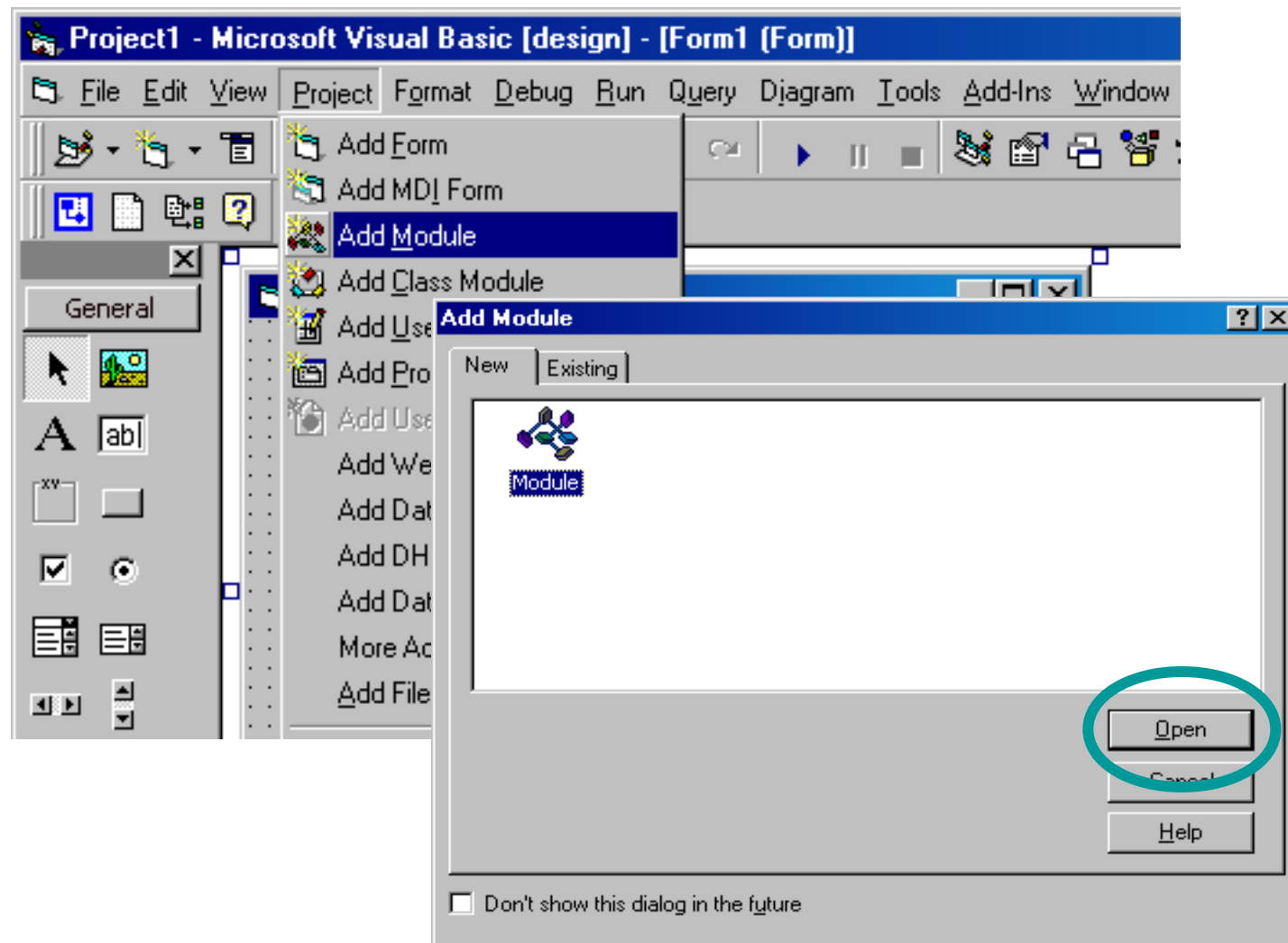


4.2 Tạo Form



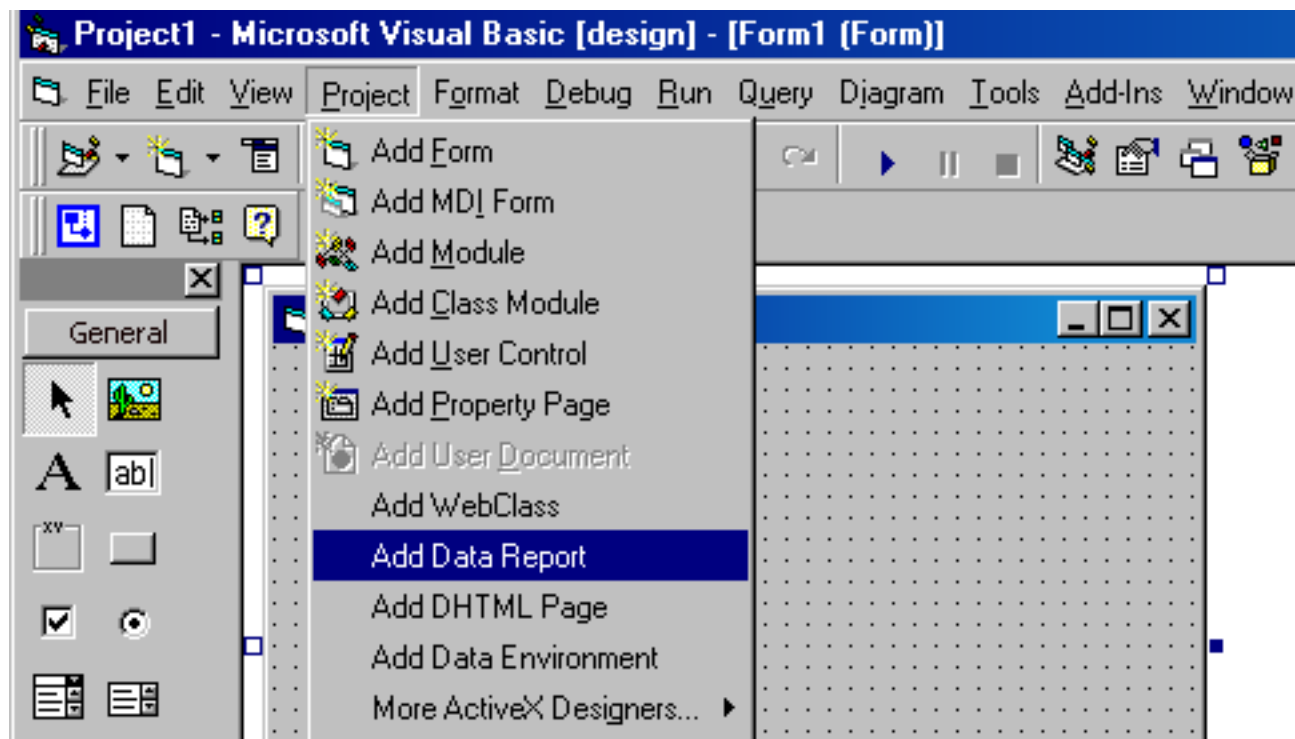
Bài giảng VB 1

4.3 Tạo Module

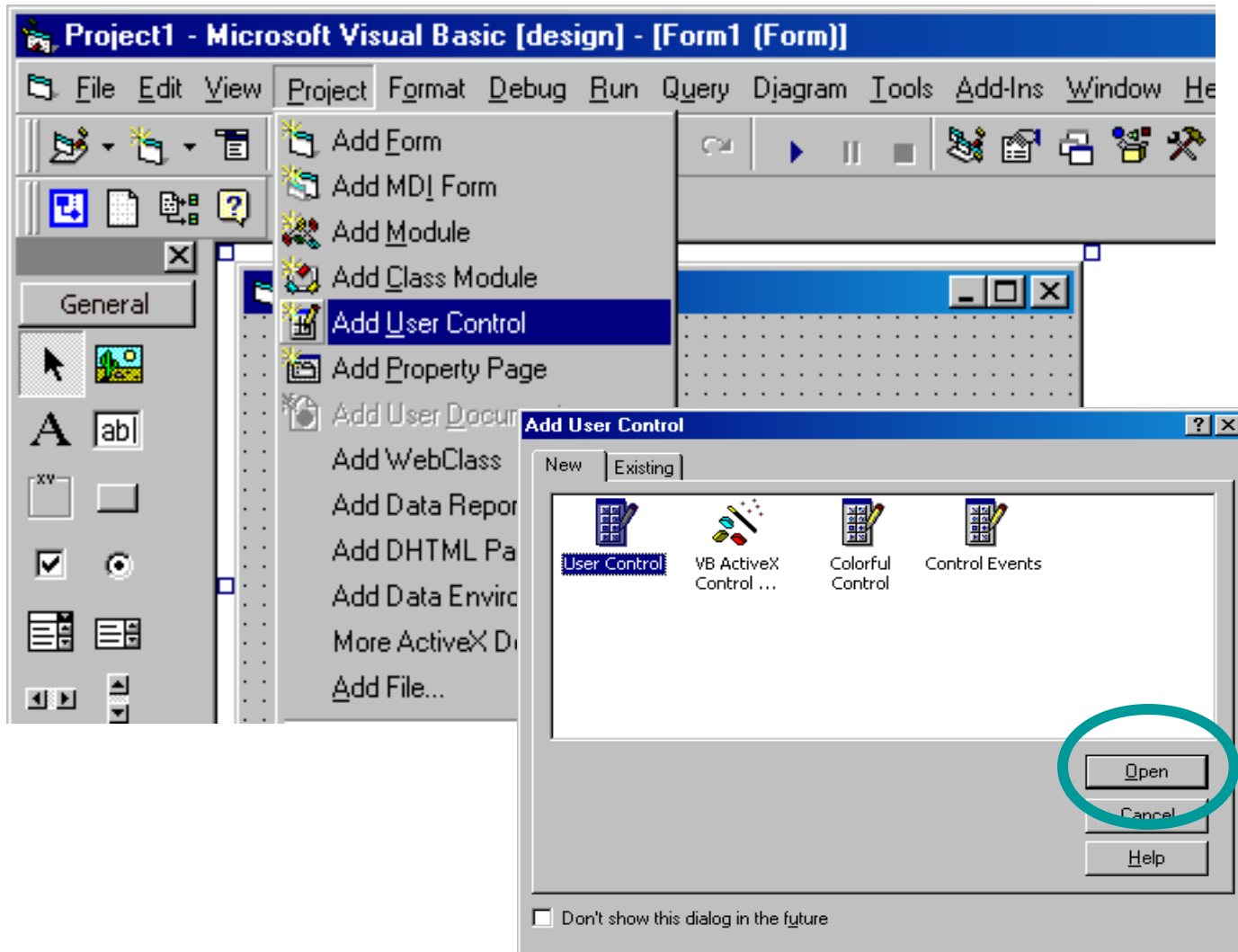


Bài giảng VB 1

4.4 Tạo Report



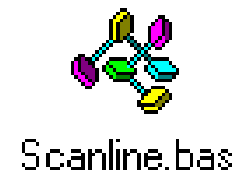
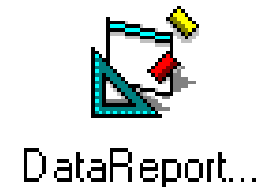
4.5 Tạo user control



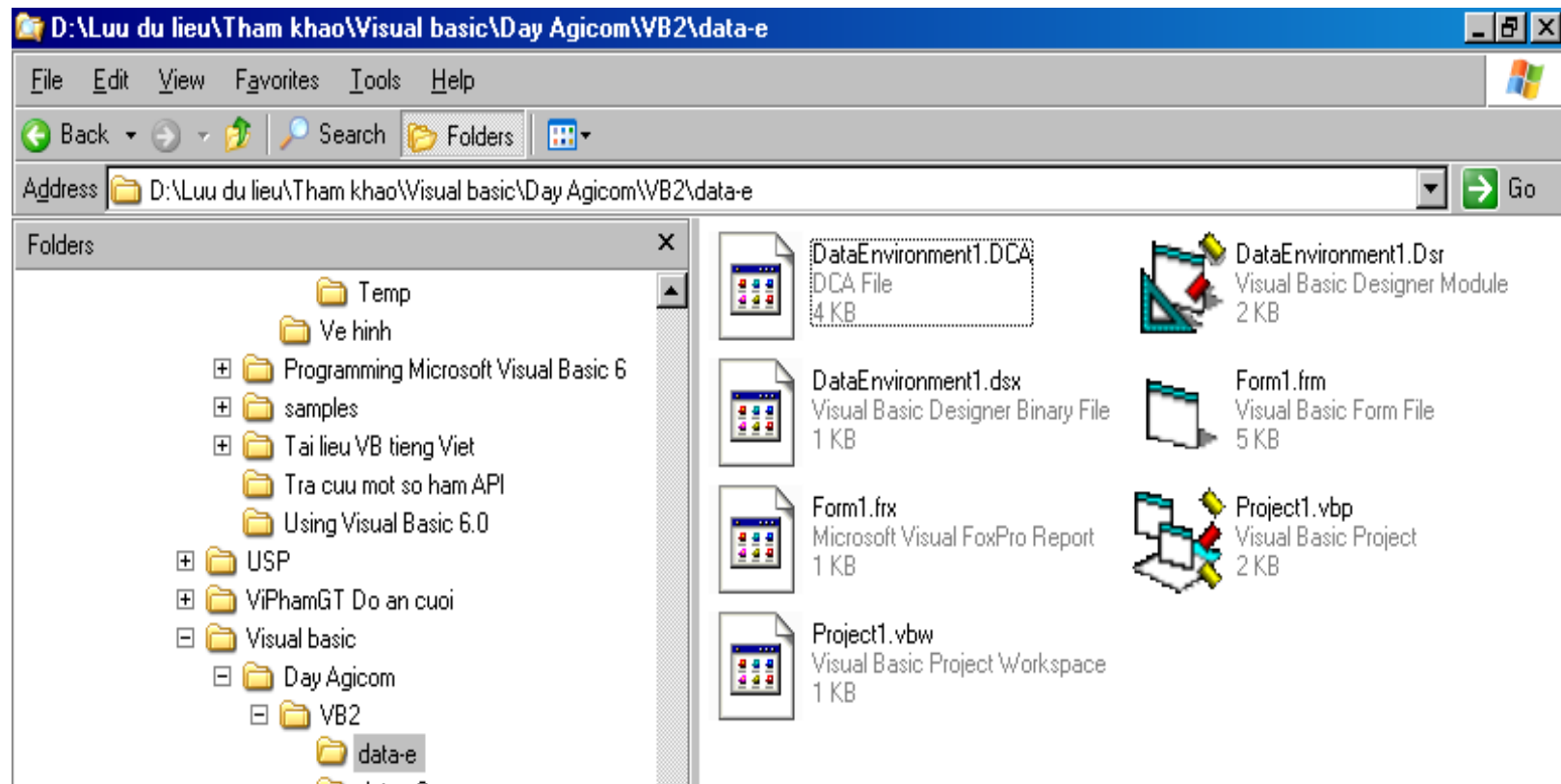
Bài giảng VB 1

5. Tổ chức lưu trữ project

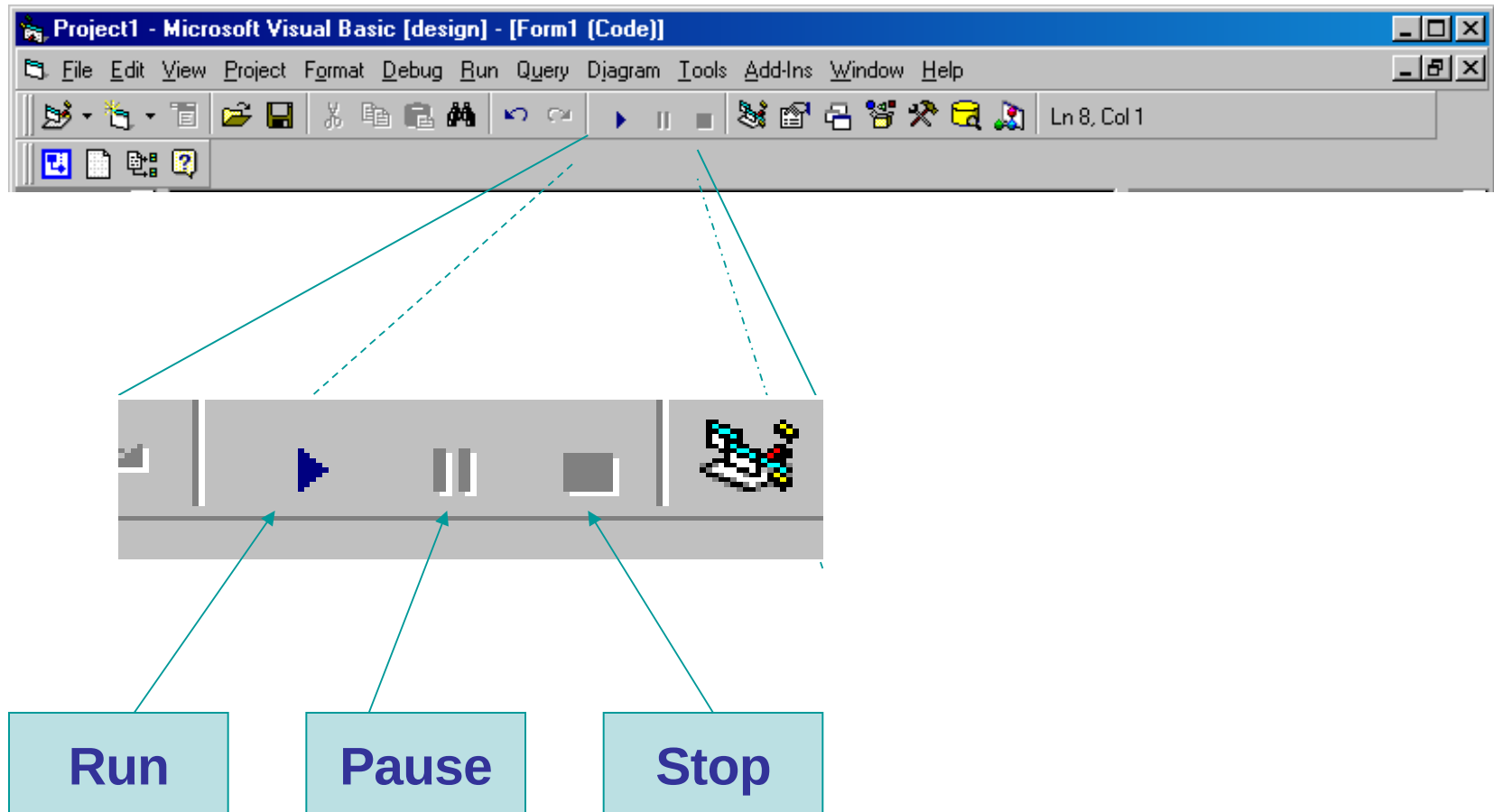
- Mỗi dự án (project) phải lưu cùng một thư mục.
- Một dự án gồm nhiều thành phần như: form, module, report.... Mỗi đối tượng trên phải lưu trong cùng một thư mục với các tên khác nhau.
- Biểu tượng của các đối tượng trên khi lưu (*xem hình bên*)



5.1 Dự án đã được lưu



6. Thực thi chương trình



Bài giảng VB 1

7. Form và các điều khiển cơ bản

TT	Thuộc tính	Ý nghĩa
1	Name	Đặt tên cho form
2	Caption	Nội dung thanh tiêu đề của Form
3	BackColor	Đặt màu nền của form
4	BorderStyle	0: Form không có khung viền xung quanh, không có tiêu đề. 1: Form không thể thay đổi kích thước. 2: Form có thể thay đổi kích thước.
5	Height, Width	Chiều cao và rộng form
6	Picture	Tranh nền của form
7	MaxButton	True: Hiển thị nút max trên form False: Không hiển thị nút max trên form

Bài giảng VB 1

7.Form (tt)

TT	Thuộc tính	Ý nghĩa
8	MinButton	True: Hiển thị nút min trên form False: Không hiển thị nút min trên form
9	WindowState	Kích thước khi bắt đầu thực thi. 0: Bình thường 1: Nhỏ nhất (hiển thị trên thanh Taskbar) 2: Lớn nhất (giáp màn hình)

Chú ý: 01 chương trình VB có nhiều form. Do đó để chọn thực thi trước tiên ta làm như sau:

Vào menu project – Chọn project properties – Trong mục StartUp Object chọn tên form cần thực thi trước tiên.

7. Form (Các biến cố của Form)

STT	Biến cố	Phát sinh khi
1	Click	Ấn chuột lên form
2	DblClick	Ấn đôi chuột lên form
3	Load	Khi form được nạp
4	Unload	Khi form đóng
5	Resize	Khi thay đổi kích thước Form
6	Activate	Form được Activate
7	Deactivate	Form được Deactivate
8	QueryUnload	Khi form đóng
9	Paint	Khi form nạp

Đóng và mở Form

- Đóng form
 - Cú pháp: <Unload> <tên form cần đóng>
 - Ví dụ: Unload FormMain
- Mở form
 - Cú pháp: <Tên form>.Show
 - Ví dụ: FormMain.Show
- Ẩn form
 - Cú pháp: <Tên form>.Hide
 - Ví dụ: FormMain.Hide
- Kết thúc chương trình
 - Dừng lệnh: End
 - *Chú ý: Khi sử dụng lệnh End thì tất cả các form đang thực thi sẽ kết thúc.*

Các điều khiển cơ bản

Label

TT	Thuộc tính	Ý nghĩa
1	Name	Tên cho lable
2	Caption	Nội dung hiển thị
3	Alignment	0: Nội dung canh trái 1: Nội dung canh phải 2: Nội dung canh giữa
4	AutoSize	True: Tự động thay đổi kích thước của lable theo nội dung. False: ngược lại
5	ForeColor	Chọn màu chữ
6	BackColor	Chọn màu nền
7	Font	Định dạng font chữ

Textbox

TT	Thuộc tính	Ý nghĩa
1	Name	Đặt tên cho textbox
2	Alignment	0: Nội dung canh trái 1: Nội dung canh phải 2: Nội dung canh giữa
3	Enable	False: Không cho phép nhập dữ liệu vào textBox True: Cho phép nhập dữ liệu vào textBox
4	Visible	True: Hiện textbox . False: Ẩn textBox .
5	Font	Định dạng font chữ
6	Locked	True: Không nhập dữ liệu vào textbox . False: Cho phép nhập dữ liệu vào textbox .

TextBox (tt)

TT	Thuộc tính	Ý nghĩa
7	MultiLine	True: Nhập được nhiều dòng False: Chỉ nhập trên cùng 1 dòng.
8	BackColor	Màu nền
9	ForeColor	Màu chữ
10	ScrollBars	0: Không xuất hiện thanh cuộn. 1: XH Thanh cuộn ngang. 2: XH thanh đứng. 3: XH thanh cả 2
11	Text	Giá trị ô text đang chứa

CommandButton

TT	Thuộc tính	Ý nghĩa
1	Name	Đặt tên cho nút lệnh
2	Caption	Nội dung hiển thị trên nút lệnh
3	Enable	False: Nút lệnh không thể tác động. True: Được phép thao tác trên nút lệnh.
4	Visible	True: Hiện nút lệnh; False: Ẩn nút lệnh.
5	Font	Định dạng font chữ
6	Cancel	True: Tác dụng khi ấn phím “ESC”. Trên form chỉ có di nhất một nút lệnh được thiết lập thuộc tính cancel=true.
7	Default	True: Khi ấn phím “ENTER” nút sẽ được chọn. Trên form chỉ có di nhất một nút lệnh được thiết lập thuộc tính Default=True.
8	ToolTipText	Văn bản hiển thị chú thích cho nút lệnh

8. Ví dụ: (Tạo dự án)

Tạo dự án với tên project là “**ProPheptinh**” *lưu vào thư mục d:\Pheptinh*. Các thành phần của chương trình.

Form1	Form2
<ul style="list-style-type: none">- Name: FrmMain- Caption: Phep tinh- BoderStyle=1- WindowsStart=0	<ul style="list-style-type: none">- Name: FrmTacGia- Caption: Gioi thieu tac gia- BoderStyle=2- WindowsStart=2

Module tên **mdlMain**

8. Ví dụ



Yêu cầu: Khi ấn vào nút lệnh “+” thì textbox kết quả = textbox hệ số 1 + textbox hệ số 2.

Hướng dẫn thực hiện:

1. Thiết lập thuộc tính cho các đối tượng.

Textbox he so 1:

`.name=txtHeso1`

`.text=""`

Chú ý: “” là giá trị rỗng

Bài giảng VB 1

8. Ví dụ

- textbox he so 2:
- .name=txtHeso2
- .text="" Chú ý: "" là giá trị rỗng
- textbox kết quả
- .name=txtKetQua
- .text="" Chú ý: "" là giá trị rỗng

8.Ví dụ

- ComandButton:
- `.name=CmdCong`
- `.caption=+`
- Đoạn mã của chương trình.
- **Private Sub CmdCong_Click()**
- `Me.TxtKetQua.Text = Val(Me.TxtHeSo1.Text) + Val(Me.TxtHeSo2.Text)`
- **End Sub**

5t

Phần 2

CÁC ĐỐI TƯỢNG VỀ KIỂU DỮ LIỆU, BIẾN, BIẾN HẲNG



Bài giảng VB 1

Nội dung chính

- Kiểu dữ liệu cơ sở, biến, hằng
- Kiểu dữ liệu do người dùng định nghĩa
- Các phép toán cơ sở
- Chương trình con
- Các hàm xử lý số, chuỗi, ngày giờ
- Hàm kiểm tra dữ liệu, chuyển kiểu
- Hàm InputBox, MsgBox
- Cấu trúc điều khiển
 - Cấu trúc lặp
 - Cấu trúc rẽ nhánh

1. Kiểu dữ liệu cơ sở

TT	Tên kiểu	Byte	Vùng giá trị
1	Byte	1	Số nguyên (0..255) Ví dụ: Dim n as byte ' Dòng khai báo n=-1 ' Dòng lệnh này bị lỗi n=5 ' Gán n bằng 5
2	Integer	2	Số nguyên (-32.768..+32.767) Ví dụ: Dim n as integer ' Dòng khai báo n=-100000 ' Dòng lệnh này bị lỗi n=-500 ' Gán n bằng -500
3	Long	4	Số nguyên (-2.147.483.648..+2.147.483.647) Ví dụ: Dim n as Long ' Dòng khai báo N=54 ' Gán n bằng 54

1. Kiểu dữ liệu cơ sở

TT	Tên kiểu	Byte	Vùng giá trị
4	Boolean	2	True(-1) False(0) Ví dụ: Dim n as Boolean 'Dòng khai báo $n=-1 \Leftrightarrow n=True$ $n=0 \Leftrightarrow n=False$
5	Single	4	Số thực Ví dụ: (-3.4E38..-1.4E-45) (1.4E-45..3.4E38) Dim n as Single 'Dòng khai báo N=-1
6	Double	8	Số thực (-1.7E308..-4.9E-324) (4.9E-324..1.7E308) Dim n as Double 'Dòng khai báo $n=5E+3 \Leftrightarrow n=5000$ $n=5E-3 \Leftrightarrow n=0.001$

1. Kiểu dữ liệu cơ sở

TT	Tên kiểu	Byte	Vùng giá trị
7	String	2	Kiểu chuỗi Tối đa 2 tỷ ký tự
8	Date	8	Ngày, giờ (1/1/100..31/12/9999) Dim d as date 'Dòng khai báo d=#1/1/2005#' d="1:15:0 AM" Chú ý: Chuỗi ngày phải để trong cặp dấu #
9	Variant	16	Biến Variant có thể lưu bất kỳ giá trị nào?

2. Kiểu dữ liệu cho người dùng định nghĩa

- Khai báo:
 - Cú pháp:
 - **<Type> tênkiểudữliệu**
 - Tên thuộc tính **<as>** kiểu dữ liệu
 - ...
 - **<End type>**
 - Ví dụ:
 - **Type ConNguoi**
 - Ten as String
 - NgaySinh as Date
 - GioiTinh as Boolean
 - **End Type**

2. Kiểu dữ liệu do người dùng định nghĩa

- Khai báo biến
 - Cú pháp: Dim <Tên biến> as <Tên kiểu>
 - Ví dụ: Dim N as ConNguoi
- Truy xuất thành phần thuộc tính của kiểu dữ liệu do người dùng định nghĩa:
 - Cú pháp: <Tênbiến>.<Tênthuộctính>
 - Ví dụ:
 - N.Ten="Nguyen Van Tam"
 - N.NgaySinh="15/02/1943"
 - N.GioiTinh=-1

2. Kiểu dữ liệu do người dùng định nghĩa

- Truy xuất thành phần thuộc tính của kiểu dữ liệu do người dùng định nghĩa:
 - Có thể sử dụng mệnh đề **With**
 - Ví dụ:
 - With N
 - .Ten="Nguyen Van Tam"
 - .NgaySinh="15/02/1943"
 - .GioiTinh=-1
 - End With

Các phép toán cơ sở

- Phép toán số: +, -, *, /, mod, \, ^
- Phép toán logic: AND, OR, NOT
- Phép gán:
 - $\langle \text{biến 1} \rangle = \langle \text{giá trị} \rangle$
 - $\langle \text{biến 1} \rangle = \langle \text{biến 2} \rangle$
 - $\langle \text{biến 1} \rangle = \langle \text{biến 2} \rangle = \langle \dots \rangle = \langle \text{giá trị} \rangle$
 - Ví dụ: $n=10$ ‘ Gán $n=10$
- Toán tử kết nối chuỗi: &
 - **Ví dụ:** $s = \text{“ABC”} \ \& \ \text{“DEF”}$
 - Lúc này giá trị của biến s là **ABCDEF**.

3. Biến

- ĐN: Vùng nhớ chứa dữ liệu, giá trị của biến có thể thay đổi trong quá trình chương trình thực thi.
- Cú pháp: Dim <tên biến> as <kiểu dữ liệu>
- Ví dụ:
 - Dim n as Integer
 - Dim n as String
- Một số hình thức khai báo biến
 - Biến toàn cục (Global Variables):
 - ĐN: Được khai báo đầu module bắt đầu với từ khóa Public.
 - Ví dụ: Public Tong as Long

3. Biến

- **Biến cấp module (Module-Level Variables):**
 - **ĐN:** Được khai báo ở đầu module hoặc form, bắt đầu với từ khóa **Private** hoặc **Dim**.
 - **Ví dụ:**
 - Dim N as Long
 - Private a as String
 - Private D as Date
- **Biến cục bộ (Local Variable):**
 - **ĐN:** Biến được khai báo trong thủ tục.

4. Hằng

- ĐN: Là một biến dùng để lưu giá trị, giá trị mà nó lưu sẽ không đổi trong quá trình chạy chương trình.
- Khai báo: `Const <tên hằng> = <giá trị>`
- Ví dụ: `Const n=500`

5. Chương trình con

- CTC là gì?
- Thủ tục, hàm
- Truyền tham số cho chương trình con
- Bẫy lỗi trong chương trình con
- Tham số có tùy chọn Optional

5. Chương trình con

- CTC là một đoạn lệnh do người dùng viết nhằm để thực hiện một công việc mong muốn.
- CTC bao gồm 02 dạng: **Hàm** và **thủ tục**.
- Thêm CTC: Menu Tools\Add Procedure

The diagram illustrates the relationship between the 'Add Procedure' dialog box and the Vietnamese terms for procedure types. Three boxes on the left, labeled 'Tên CTC', 'Thủ tục', and 'Hàm', have arrows pointing to the 'Name', 'Type', and 'Scope' fields of the dialog box, respectively. A dashed arrow points from the 'Scope' field to a text box at the bottom.

Public: Có thể được gọi ở một module khác.
Private: Chỉ được gọi ở trong module mà nó được khai báo.

6. Thủ tục và hàm

6.1 Thủ tục

Public [Private] Sub <tên thủ tục>(Danh sách đối số)

‘Danh sách các lệnh

Exit sub

End Sub

Exit sub: Thoát khỏi thủ tục

Gọi thủ tục:

- <Tên thủ tục> tham số 1, tham số 2, ...
- **Call** <Tên thủ tục>(tham số 1, tham số 2, ...)

6.1 Thủ tục – ví dụ

```
Public Sub InDS()  
    Dim i As Byte  
    For i = 0 To 10  
        Debug.Print i  
        If i > 0 Then Exit Sub  
    Next i  
End Sub
```

Exit Sub: Lệnh dùng để thoát khỏi thủ tục

Call InDS

6.2 Hàm

```
Public [Private] Function <tên hàm>(Danh sách đối số) as  
Kiểu trả về  
    'các lệnh  
    tên hàm = giá trị|biểu thức  
End Function
```

Exit Function: Lệnh thoát khỏi hàm

Gọi hàm: Giá trị của hàm dùng trong biểu thức

6.2 Hàm và ví dụ

```
Public Function ChinhPhuong(ByVal so as Long) as Boolean
    ChinhPhuong=(SQR(so)= round(SQR(so)))
End Sub
```

Ví dụ:

```
Dim n as Long
```

```
n=4
```

```
IF ChinhPhuong(n) then
```

```
Debug.print “Chinh phuong”
```

```
Else
```

```
Debug.print “Khong chinh phuong”
```

```
End If
```

7. Truyền biến dạng tham trị và tham chiếu

7.1 Truyền biến dạng tham trị

- Giá trị tham số trị không bị thay đổi khi khi tham số hình thức thay đổi.
 - Từ khóa **Byval** đứng trước đối số CTC.
 - Ví dụ:

```
Public Sub InDS( byval ten as String)
```

```
    Debug.Print ten
```

```
    ten=""
```

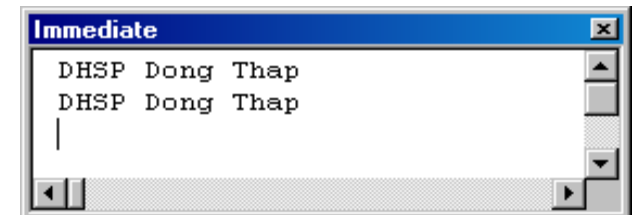
```
End Sub
```

```
Dim str as string
```

```
str="DHSP Dong Thap"
```

```
Call InDS(str)
```

```
Debug.Print str
```

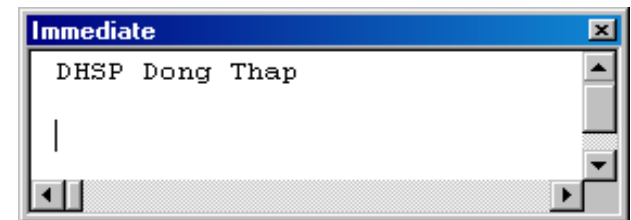


7.2 Truyền biến dạng tham chiếu

- Giá trị **tham số thực sự** bị thay đổi khi khi tham số hình thức thay đổi.
 - Từ khóa **Byref** đứng trước đối số CTC.
 - Ví dụ:

```
Public Sub InDS( byref ten as String)
    Debug.Print ten
    ten=""
End Sub
```

```
Dim str as string
str="DHSP Dong Thap"
Call InDS(str)
Debug.Print str
```



8. Bẫy lỗi trong chương trình con

- Khi chạy chương trình có thể sẽ phát sinh lỗi ngoài ý muốn (lỗi trong quá trình thực thi). Nếu các lỗi này chưa được xử lý thì VB sẽ thông báo lỗi và chương trình sẽ bị kết thúc.
- Việc kết thúc ngang chương trình sẽ làm người dùng khó chịu, một trong các giải pháp khắc phục các lỗi trên là bẫy lỗi cho chương trình.
- Ưu điểm: Chương trình sẽ không bị kết thúc khi có lỗi phát sinh trong quá trình chương trình thực thi.

8. Bẫy lỗi trong chương trình con

- **Cú pháp:**

- **On Error Goto <Tên nhãn>**
- ‘Danh sách các lệnh trong CTC
- Exit Sub [Exit Function]
- **<Tên nhãn><:>**
- <Thông báo lỗi>
- [Resume][Resume Next]

- **Giải thích:**

- **Resume:** Thực hiện lại lệnh đã gây ra lỗi.
- **Resume Next:** Bỏ qua lệnh gây ra lỗi và thực hiện lệnh kế tiếp.

Tham số các tùy chọn Optional

- Trong CTC, ta có thể khai báo tham số hình thức là tùy chọn hoặc bắt buộc.
- Khi tham số hình thức là bắt buộc thì người dùng phải truyền tham số thực sự vào khi gọi CTC.
- Nếu tham số hình thức là tùy chọn thì người dùng cũng có thể không gửi tham số thực sự khi gọi CTC.
- Để thiết lập tham số là tùy chọn ta dùng từ khóa **Optional** trước đối số trong CTC.
- Cú pháp:
 - Public Sub [Function] <Tên CTC>
(*Optional* TênBiến as Kiểu dữ liệu)

Tham số các tùy chọn Optional

- Ví dụ:
 - Public Sub InDS(**Optional** Ten as String)
- Khi gọi thủ tục ta có 2 cách gọi:
 - Call InDS ‘ **Không truyền giá trị cho đối số.**
 - Call InDS(“DHSP Dong Thap”) ‘ **Có truyền**
- Khi không truyền giá trị thì tham số tùy chọn sẽ có giá trị mặc nhiên là giá trị của kiểu dữ liệu được khai báo.

Kiểu dữ liệu	Giá trị mặc nhiên
Number	0
String, Variant	Null
Boolean	False (0)

Tham số các tùy chọn Optional

- Chúng ta có thể khởi tạo giá trị mặc nhiên cho tham số tùy chọn. Nếu tham số này không được truyền giá trị thì nó sẽ lấy giá trị mặc nhiên.
- Cú pháp:
 - Public Sub Tên thủ tục(**Optional** TênBiến as **Kiểu dữ liệu = Giá trị**)
 - **Giá trị** phải phù hợp với **kiểu dữ liệu** đã khai báo.
 - **Ví dụ:**
 - Public Sub InDS(Optional So As **Long** = "aa")
 - Public Sub InDS(Optional So As **Long** = 10)

9. Các hàm xử lý số, chuỗi, ngày giờ

Tên hàm	Ý nghĩa	Ví dụ
Int(n)	Lấy phần nguyên của 1 số n	Int(125.456) =125
Round(n,m)	Làm tròn số n lấy m số hàng thập phân	Round(1254.365,2)=1254.36
Len(s)	Trả về chiều dài chuỗi S	Len(DHDT) = 4
Left(s, n)	Cắt bên trái chuỗi S n ký tự	Left("ABCDEF") = "AB"
Right(s,n)	Cắt bên phải chuỗi S n ký tự	Right("ABCDEF")="EF"
Mid(s,n,m)	Cắt lấy chuỗi con trong chuỗi S tại vị trí n , m ký tự	Mid("ABCD-EFG",2,3)="CD-
LTrim(s)	Cắt bỏ khoảng trắng thừa bên trái	LTrim(" ABDC ") = "ABDC "
Rtrim(s)	Cắt bỏ khoảng trắng thừa bên phải	Rtrim(" ABCD ") = " ABCD"
Trim(s)	Cắt bỏ khoảng trắng thừa 2 bên chuỗi S	Trim (" GHRFOP ") = "GHRFOP"

9. Các hàm xử lý số, chuỗi, ngày giờ

Tên hàm	Ý nghĩa	Ví dụ
Replace(s, s1, s2)	Tìm và thay thế chuỗi s1 bằng chuỗi s2 trong chuỗi s	S=Replace("ABCDAEF", "A","HH") Lúc này S=" HHBCDHHEF"
InStr(vt, s, s1, tc)	Tìm vị trí xuất hiện đầu tiên của chuỗi s1 trong chuỗi s. Vị trí bắt đầu tìm là vt. Mặc định vt=1 Tc=1 Tìm không phân biệt hoa thường. Tc=0 Tìm phân biệt hoa thường	
Day(d)	Trả về ngày của d	d="#12/1/2005#" Day(d)=12
Month(d)	Trả về tháng của d	d="#12/1/2005#" Month(d)=1

9. Các hàm xử lý số, chuỗi, ngày giờ

Tên hàm	Ý nghĩa	Ví dụ
Year(d)	Trả về năm của d	d="#12/1/2005#" Year(d)=2005
Date	Trả về ngày tháng hiện tại	
Now	Trả về ngày, giờ hiện tại	
Time	Trả về giờ hiện tại	
Minute(T)	Trả về phút của thời gian T	d="#12/1/2005#" Year(d)=2005
Hour(T)	Trả về giờ của thời gian T	Hour(11:00:48 AM)=11
Second(T)	Trả về giây của thời gian T	Second(11:00:48 AM)=48

9. Các hàm xử lý số, chuỗi, ngày giờ

Tên hàm	Ý nghĩa	Ví dụ
DateAdd(dv, n, d)	Cộng giá trị n vào d theo đơn vị tính dv dv="d": ngày dv="m": tháng dv="yyyy": năm	d="#12/1/2005#" DateAdd("m",1, d) Lúc này d= "12/2/2005"

9.1 Hàm kiểm tra kiểu dữ liệu, chuyển kiểu

Tên hàm	Ý nghĩa	Ví dụ
IsDate(s)	Kiểm tra s có phải chuỗi ngày không	IsDate("1b/25/2005")=False IsDate("1/25/2005")=True
IsEmpty(s)	Kiểm tra s có rỗng không	S="" IsDate(s)=True

9.1 Hàm kiểm tra dữ liệu, chuyển kiểu

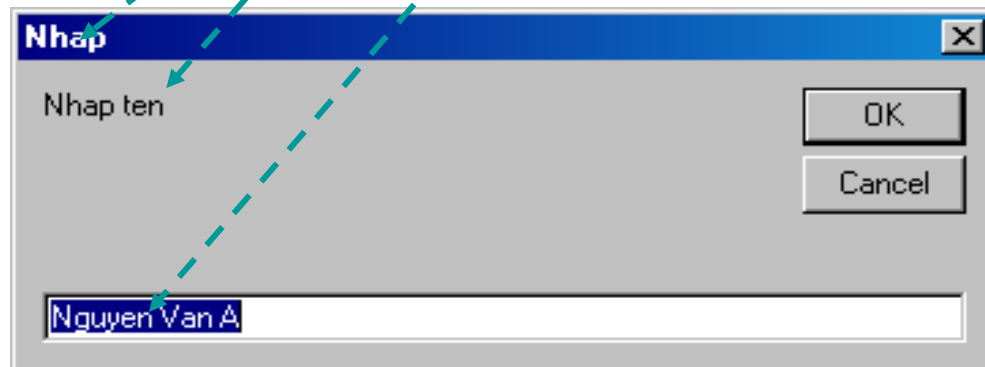
Tên hàm	Ý nghĩa	Ví dụ
IsNumeric(n)	Kiểm tra n có phải số không	IsNumeric("nh")=False IsNumeric(1)=True
CLng(n)	Chuyển số n sang kiểu Long	
CStr(s)	Chuyển s chuỗi	CStr(125)="125"
CByte(n)	Chuyển số n sang kiểu Byte	
CDate(s)	Chuyển chuỗi ngày s sang giá trị ngày	
Val(s)	Chuyển chuỗi số s sang số s	Val("125")=125
CSng(n)	Chuyển số n sang kiểu Single	
CDbl(n)	Chuyển số n sang kiểu Double	

10. Hàm InputBox, MsgBox

- Hàm InputBox
- Lệnh MsgBox
- Hàm MsgBox

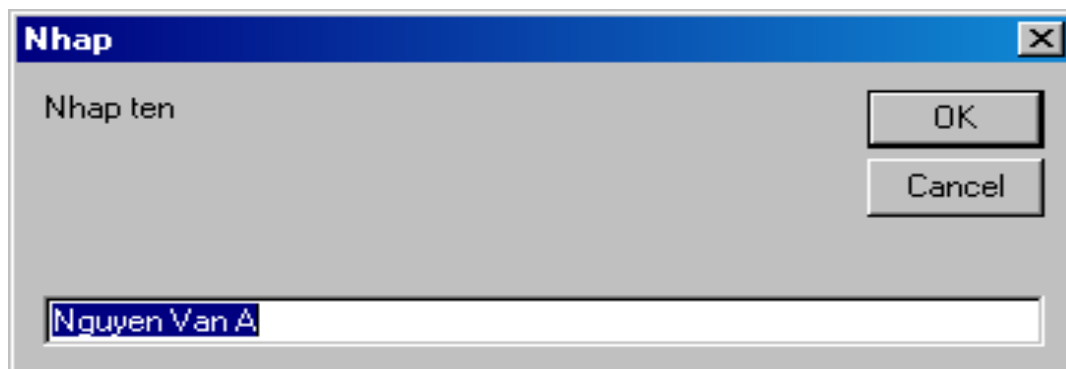
10.1 Hàm InputBox

- Cú pháp hàm: `input (s, t, d)`
 - s: Là chuỗi thông báo
 - t: Chuỗi tiêu đề
 - d: giá trị mặc định sẽ hiện trong hộp nhập



10.1 Hàm InputBox

- **Công dụng:** Nhận dữ liệu từ người dùng
- Ví dụ:
 - Dim s as String
 - s = InputBox("Nhập ten", "Nhập", "Nguyen Van A")
- Đoạn chương trình sẽ hiển thị hộp nhập như hình sau
- Khi người dùng ấn nút OK thì giá trị mà người dùng nhập sẽ được lưu trong biến s



Bài giảng VB 1

10.2 Lệnh MsgBox

- Cú pháp:

Msgbox “Chuỗi thông báo”, ,”Tiêu đề”

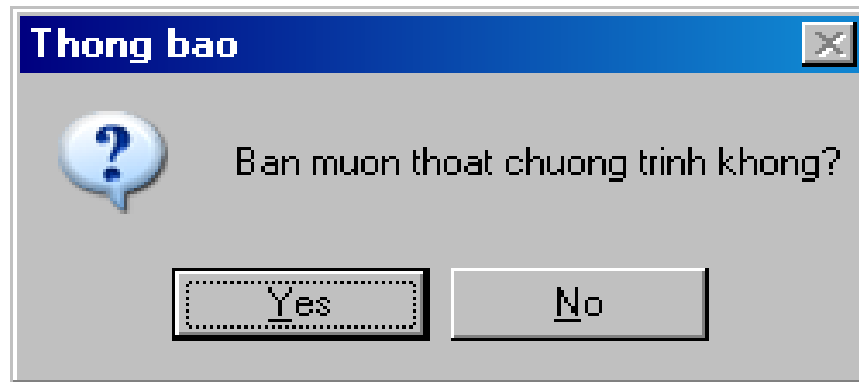
- Ví dụ:

Lệnh **MsgBox** "Chao cac ban!", , "Thong bao" sẽ hiển thị thông báo như hình:



10.3 Hàm MsgBox

- Dùng để hỏi ý người dùng.
- Ví dụ khi muốn thoát khỏi chương trình ta muốn hỏi xem người dùng có đồng ý thoát hay không. Xem hình!



Cú pháp hàm MsgBox

<tl>=Msgbox (s, h+n, t)

Trong đó:

- **s**: Là chuỗi thông báo. Trong trường hợp trên s là “Ban muon thoat chuong trinh khong?”
- **h**: Là biểu tượng xuất hiện trên form.
- *Danh sách các biểu tượng:*
 - vbQuestion: Hình ?
 - vbCritical: Hình x.
 - vbInformation: Hình I.
 - vbExclamation: Hình !.
 - Trong trường hợp trên **h** là: vbQuestion.

Cú pháp hàm MsgBox

- **n**: Các nút lệnh xuất hiện.
- *Danh sách các nút lệnh*:
 - vbYesNo: Nút yes+no
 - vbYesNoCancel: Nút Yes+No+Cancel
 - vbOKCancel: Nút OK+Cancel
 - vbOKOnly: Nút OK.
- **t**: Tiêu đề của trên form. Trong trường hợp trên t là “Thông bao”
- **tl**: Là biên kiểu số nguyên để lưu kết quả người dùng ấn vào nút lệnh nào?
 - **tl**=vbOK nếu người dùng ấn vào nút OK.
 - **tl**=vbYes nếu người dùng ấn vào nút Yes.
 - **tl**=vbNo nếu người dùng ấn vào nút No.
 - **tl**=vbCancel nếu người dùng ấn vào nút Cancel.

11. Cấu trúc điều khiển (CT lặp và CT rẽ nhánh)

11.1 Cấu trúc lặp For.....Next

- Cú pháp:

```
FOR <tênbiến = giátrịđầu> TO <giátrịcuối> [Step n]  
    Tập công việc  
NEXT tên biến
```

- Giải thích:

n: Giá trị tênbiến sẽ được cộng **n** sau mỗi lần lặp.
*Mặc định (nếu không để Step **n**) thì $n=1$.*

Cấu trúc lặp For....Next

- Ví dụ 1:

```
-----Sub-----
```

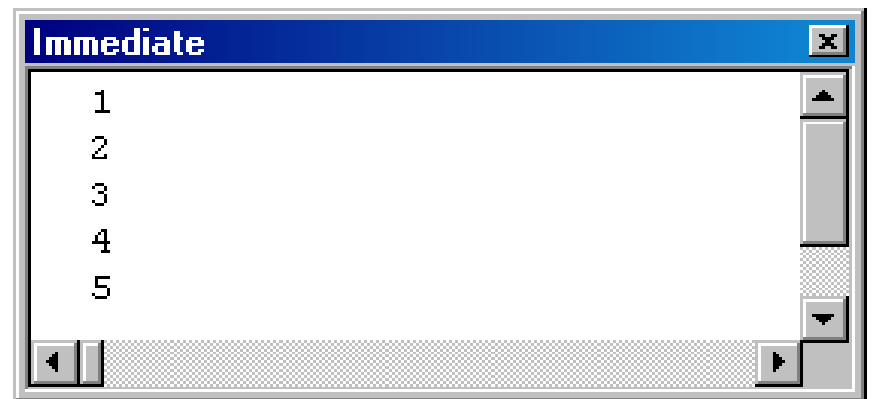
```
Dim i as byte
```

```
FOR i = 1 TO 5 Step 1
```

```
    Debug.Print i 'Hiện nội dung ra cửa sổ Debug.
```

```
NEXT i
```

```
-----End Sub-----
```



Cấu trúc lặp For....Next

- Ví dụ 2:

```
-----Sub-----
```

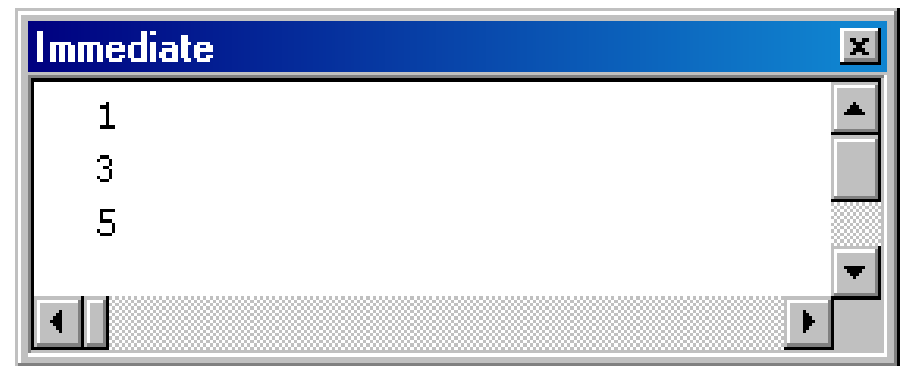
```
Dim i as byte
```

```
FOR i = 1 TO 5 Step 2
```

```
    Debug.Print i 'Hiện nội dung ra cửa sổ Debug.
```

```
NEXT i
```

```
-----End Sub-----
```



11.2 Cấu trúc lặp Do...Loop

- **Cú pháp 1:** Trong khi điều kiện đúng thì thực hiện công việc

DO WHILE <Điều kiện>

Tập hợp công việc

[Lệnh Exit Do]

LOOP

Giải thích:

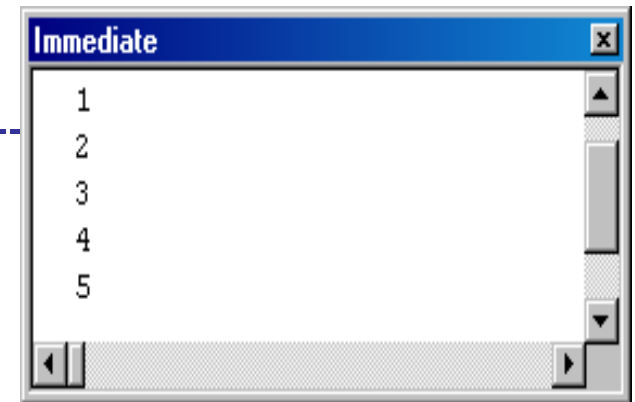
Tập hợp công việc sẽ thực hiện trong khi điều kiện đúng.

Khi cần kết thúc vòng lặp ta dùng lệnh Exit Do

Cấu trúc lặp Do...Loop

- Ví dụ:

```
-----Sub-----  
Dim i as byte  
i=0  
DO WHILE (i<10)  
    Debug.Print i  
    i=i+1  
    IF i>5 THEN EXIT DO  
LOOP  
-----End Sub-----
```



Cấu trúc lặp Do...Loop

- **Cú pháp 2:** Thực hiện công việc trong khi điều kiện đúng.

DO

Tập hợp công việc

[Lệnh Exit Do]

LOOP WHILE <Điều kiện>

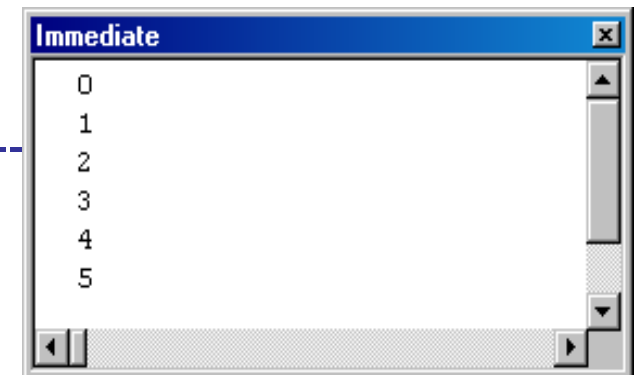
- **Giải thích:**

Exit Do: Thoát khỏi vòng lặp

Cấu trúc lặp Do.....Loop

- Ví dụ:

```
-----Sub-----  
Dim i as byte  
i=0  
DO  
    Debug.Print i  
    i=i+1  
    IF i>5 THEN EXIT DO  
LOOP WHILE i<10  
-----End Sub-----
```



11.3 Cấu trúc rẽ nhánh If.....Then

- **Cú pháp:**

```
IF <điều kiện> THEN
    Tập hợp công việc 1
ELSE
    Tập hợp công việc 2
END IF
```

- **Giải thích:** Tập hợp công việc 1 chỉ được thực hiện khi <**điều kiện**> là đúng. Ngược lại nếu điều kiện là sai thì tập hợp công việc 2 sẽ được thực hiện.

- **Ví dụ:**

```
IF a<0 THEN
    MsgBox “a<0”
ELSE
    MsgBox “a>=0”
END IF
```

Bài giảng VB 1

Cấu trúc rẽ nhánh Lệnh If...Then

- Cú pháp:
 IF <điều kiện> THEN
 Tập hợp công việc
 END IF
- Giải thích: **Tập hợp công việc** chỉ được thực hiện khi **<điều kiện>** là đúng.
- Ví dụ:
 IF a<b THEN
 c=a: a=b: b=c ‘Multicommand/line
 END IF
- Chú ý: Nếu ta chỉ cần thực hiện một công việc thì có thể ghi như sau:
 IF <điều kiện> THEN Công việc
- Ví dụ:
 IF a<>500 THEN b=a

Lệnh If.....Then

- Cú pháp:
IF <điều kiện 1> THEN
 Tập hợp công việc 1
ELSEIF <điều kiện 2> THEN
 Tập hợp công việc 2

...
ELSE
 Tập hợp công việc 3
END IF
- Ví dụ:
IF a=0 then
 Msgbox “a=0”
ELSEIF a<0 then
 Msgbox “a<0”
ELSE
 Msgbox “a>0”
END IF

Lệnh Select...Case

- **Cú pháp:**

SELECT CASE <Biểu thức>

CASE <Danh sách giá trị 1>

Các công việc cần thực hiện 1

CASE <Danh sách giá trị 2>

Các công việc cần thực hiện 2

.....

CASE ELSE

Các công việc cần thực hiện 3

END SELECT

Lệnh Select....case

- Ví dụ 1:

```
SELECT CASE (Ten)
CASE "Hoa"
    MsgBox "Chao ban Hoa!"
CASE "Tim"
    MsgBox "Chao ban Tim!"
CASE ELSE
    MsgBox "Chao ban!"
END SELECT
```

Lệnh Select....Case

- Ví dụ 2:

```
SELECT CASE Thang
CASE 1, 2, 3, 4
    MsgBox “Thang 1 2 3 4”
CASE 5 to 8
    MsgBox “Thang 5 6 7 8”
CASE ELSE
    MsgBox “Thang 9 10 11 12”
END SELECT
```

Ngoài ra còn một số vòng lặp khác

- **Thực hiện đến khi điều kiện đúng (KT điều kiện sau)**
DO
Tập hợp công việc
[Lệnh Exit Do]
LOOP UNTIL <Điều kiện>
- **Thực hiện đến khi điều kiện đúng (KT điều kiện trước)**
DO UNTIL <Điều kiện>
Tập hợp công việc
[Lệnh Exit Do]
LOOP
- **Thực hiện trong khi điều kiện đúng**
WHILE <Điều kiện>
Tập hợp công việc
WEND

Lập trình visual basic

5t

Phần 3

CÁC ĐỐI TƯỢNG THƯỜNG DÙNG

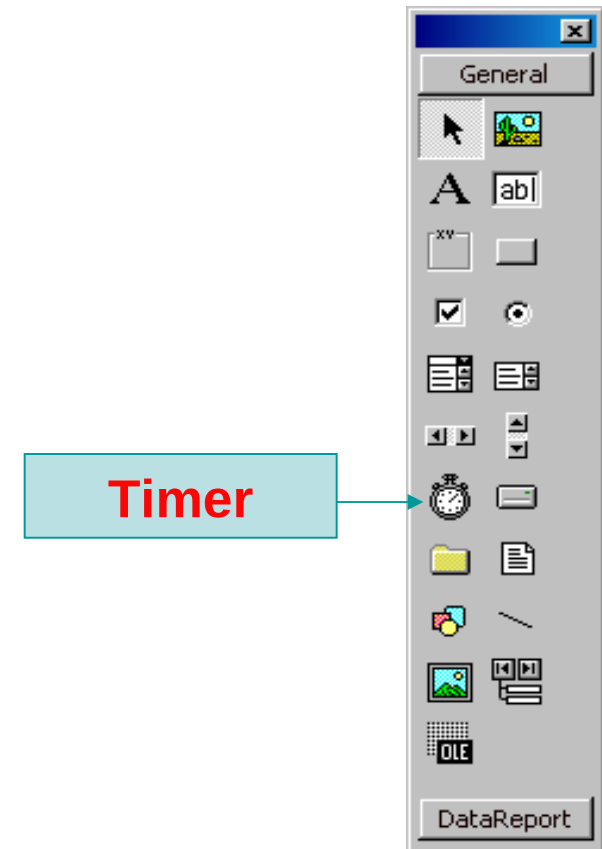


Bài giảng VB 1

1. Đối tượng Timer

Thuộc tính

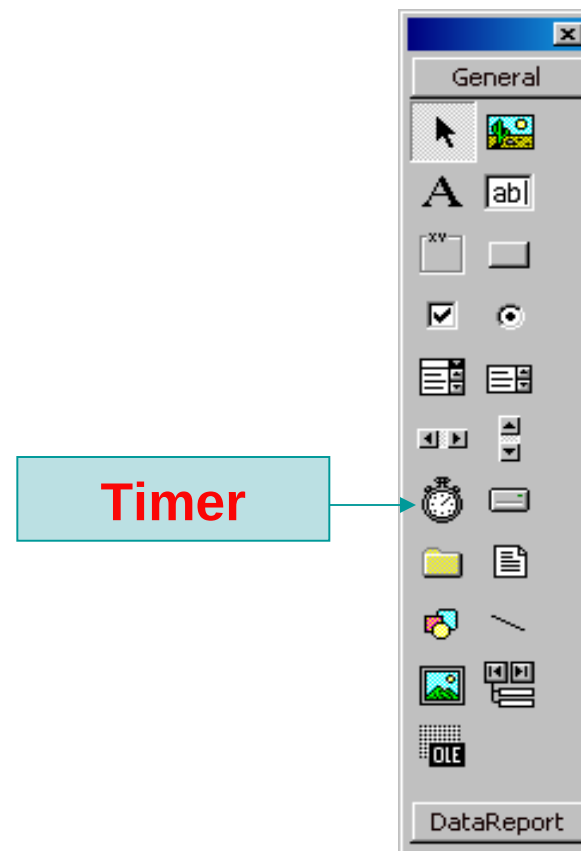
- **Enabled:** Có 2 giá trị True và False. Nếu được thiết lập = True thì cho phép đối tượng hoạt động ngược lại thiết lập = False.
- **Interval:** Qui định Khoảng thời gian (tính bằng mili giây) để sự kiện Timer của đối tượng có tác dụng.
- **Chú ý:**
 - Khi chương trình thực thi đối tượng Timer sẽ **không hiển thị** lên giao diện.
 - Nếu thuộc tính Interval=0 thì sự kiện Timer sẽ không phát sinh.



1. Đối tượng Timer (tt)

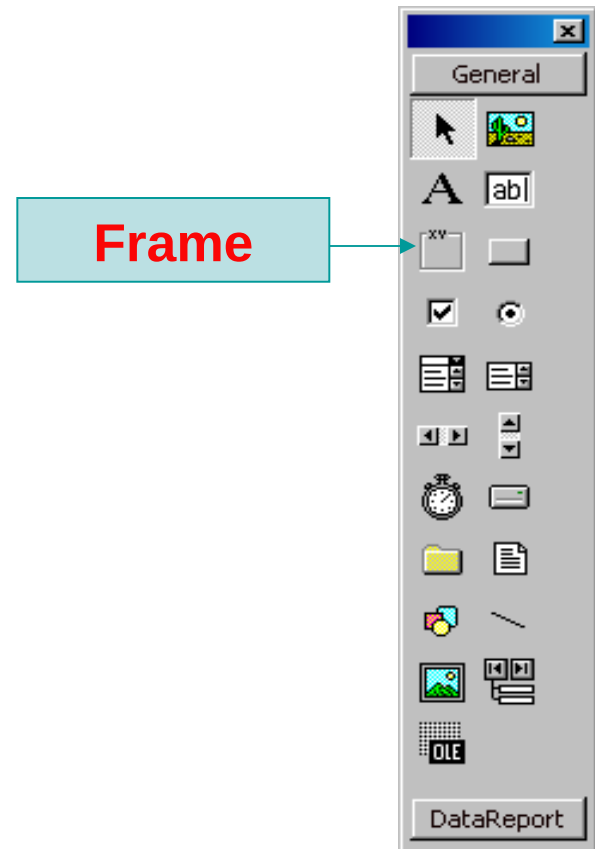
Ví dụ: Để hiển thị thời gian hệ thống lên Caption của Form ta làm như sau:

- Vẽ đối tượng Timer lên giao diện.
- Thiết thuộc tính Enable=True
- Interval = 1000 (1 giây).
- Vào sự kiện Timer của đối tượng ghi đoạn lệnh:
`Me.Caption=Time`

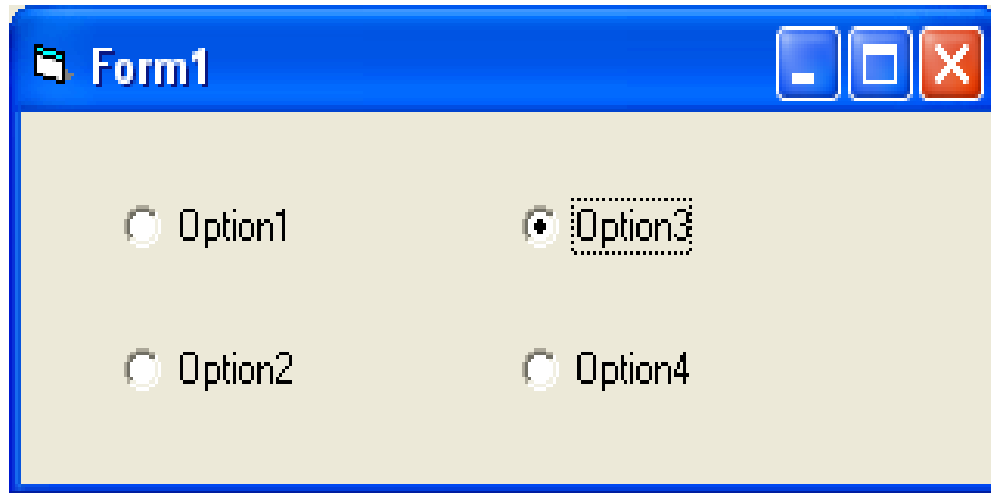


2. Đối tượng Frame

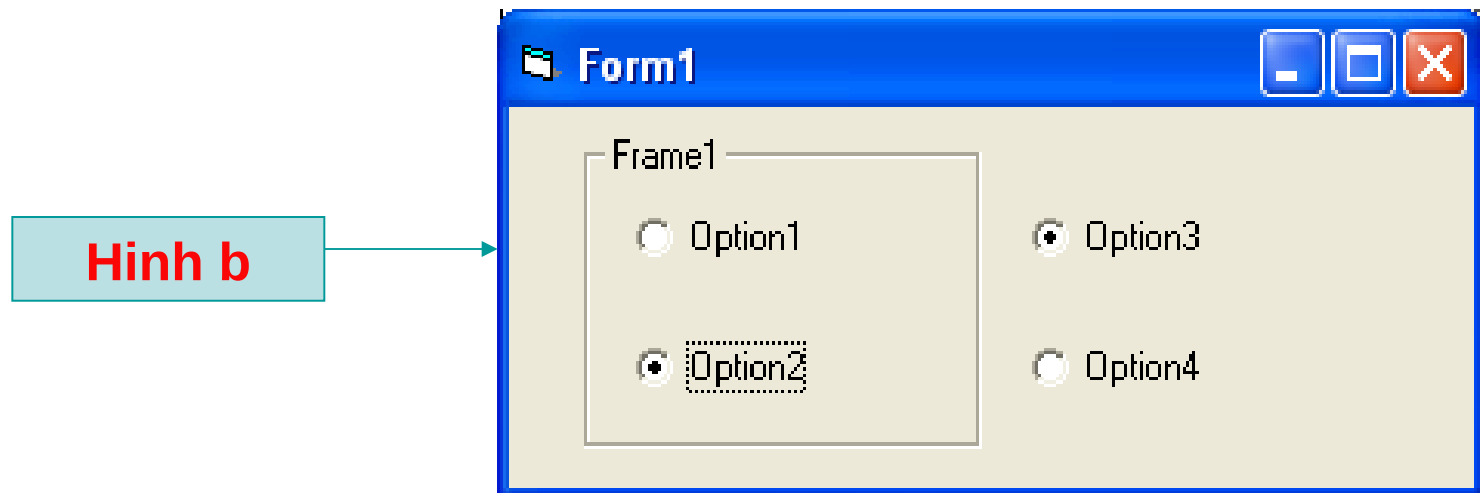
- **Chức năng:** Dùng trang trí trên giao diện. Ngoài ra nó dùng kết hợp với Option Button để nhóm các Option Button này vào một vùng riêng biệt.
- **Ví dụ:** 4 option ở hình a chỉ được phép chọn 1 trong 4. Như vậy nếu ta muốn chọn cùng lúc 2 Option thì ta làm sao? Nhóm các Option này vào 2 vùng riêng biệt. Xem hình b



2. Đối tượng Frame (tt)



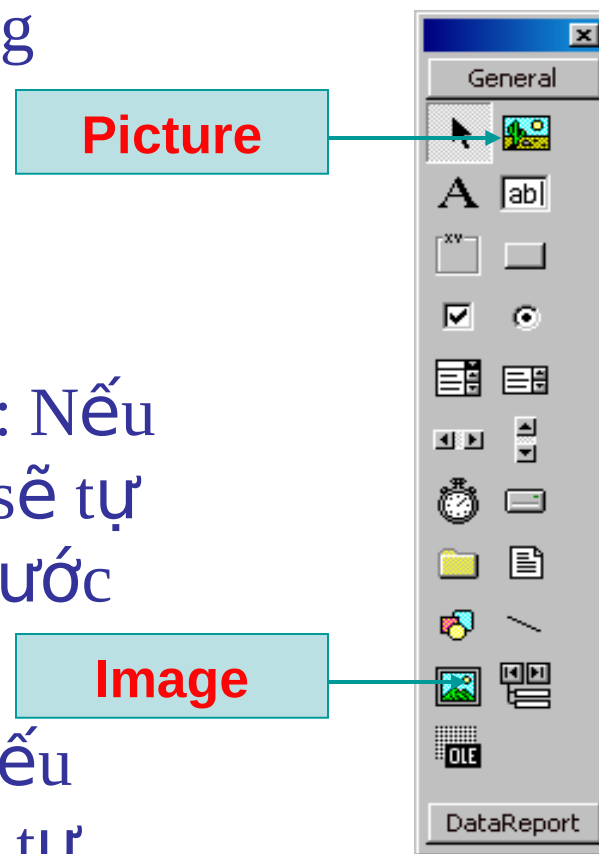
Hình a



Hình b

3. Đối tượng Picture – Image

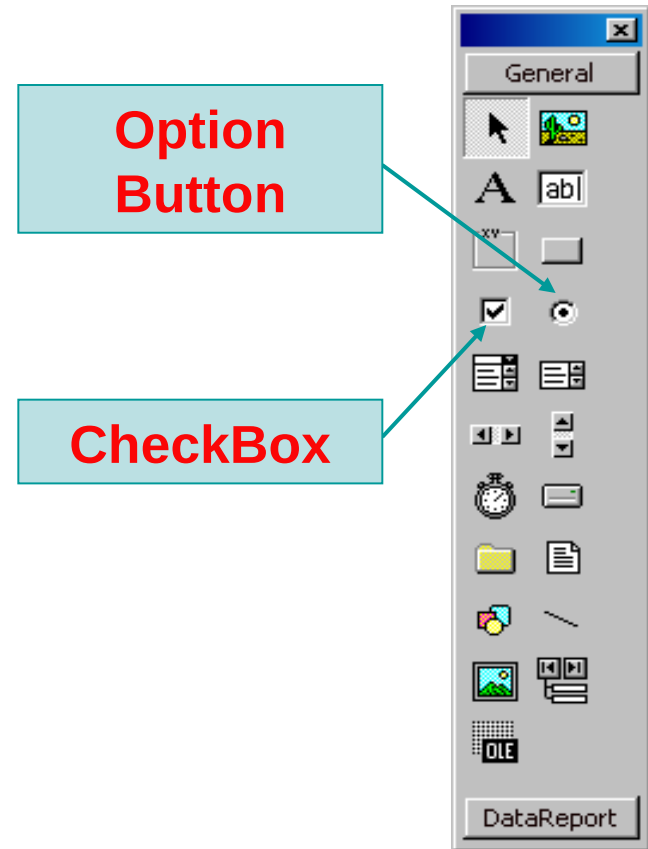
- Chức năng: Hiển thị hình ảnh lên chương trình.
- Thuộc tính:
 - Picture: Chọn hình ảnh hiển thị.
 - AutoSize(Chỉ có ở đối tượng Picture): Nếu được thiết lập = True thì khung hình sẽ tự động thay đổi kích thước theo kích thước của hình.
 - Stretch(Chỉ có ở đối tượng Image): Nếu được thiết lập = True thì hình ảnh sẽ tự động co giãn theo kích thước của khung hình.



4. Đối tượng CheckBox, OptionButton

- Chức năng

- Dùng thiết kế giao diện cho phép người dùng chọn giá trị.
- Điểm khác nhau giữa 02 loại điều khiển này là OptionButton chỉ cho phép chọn một giá trị (*Trong cùng một vùng*), Checkbox được chọn nhiều giá trị.



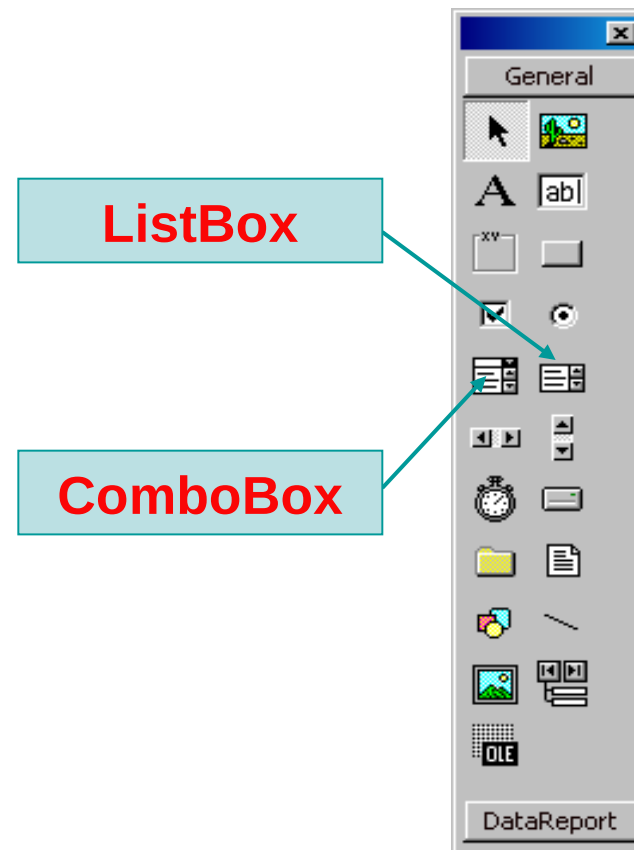
4. Đối tượng CheckBox, OptionButton (tt)

- Thuộc tính:
 - Caption: Nhãn của đối tượng.
 - Value: Lấy giá trị của đối tượng.

Checkbox		Optionbox	
Val	Ý nghĩa	Val	Ý nghĩa
0	Không được chọn	True	Được chọn
1	Được chọn	False	Không được chọn
2	Chọn một phần		

5. Đối tượng ListBox, ComboBox

- Tập hợp các phần tử có trong ListBox hoặc ComboBox được xem như là một mảng các chuỗi giá trị. Bắt đầu từ vị trí 0 và kết thúc ở vị trí $n - 1$ (n là số lượng phần tử).



5. Đối tượng ListBox, ComboBox

- Thuộc tính giống nhau
 - Text
 - ListCount
 - List
 - ListIndex
- Thuộc tính Khác nhau
 - Style
 - Selected của ListBox
 - MultiSelect của ListBox

5.1 Thuộc tính Text

- Thuộc tính Text: Lấy giá trị phần tử đang được chọn.
- Ví dụ:

```
Dim str1, str2 as string
```

```
Str1= TenListBox.Text
```

```
Str2= TenComboBox.Text
```

Phương thức SelStart, SelLength

- SelStart: Chọn nội dung từ vị trí.
- SelLength: Chọn bao nhiêu ký tự.

Ví dụ chọn hết nội dung văn bản trong textBox khi textbox nhận Focus

```
Private Sub Text1_GotFocus()  
Me.Text1.SelStart = 0  
Me.Text1.SelLength = Len(Me.Text1.Text)  
End Sub
```

Một số sự kiện chính

- Change: Khi có sự thay đổi nội dung của đối tượng hiện tại.
- Click: Khi click lên đối tượng.
- DblClick: Khi DblClick lên đối tượng.
- KeyPress: Khi có phím ấn xuống.
- KeyDown: Khi đè phím xuống.
- KeyUp: Khi buông phím lên.
- GotFocus: Khi đối tượng nhận được Focus.
- LostFocus: Khi Focus rời khỏi đối tượng.
- MouseDown: Khi ấn chuột xuống
- MouseMove: Khi di chuyển chuột lên đối tượng
- MouseUp: Khi buông chuột ra

5.2 Thuộc tính ListCount

- Thuộc tính ListCount: Cho biết số lượng phần tử
- Giải thuật duyệt qua các phần tử trong danh sách

‘-----

For i=0 to TenList.ListCount – 1

 ‘Các công việc cần thực hiện

Next i

‘-----

5.3 Thuộc tính List

- Thuộc tính List: Lấy giá trị của phần tử thứ k trong danh sách. ($k \geq 0$ và $k \leq \text{số lượng phần tử} - 1$)
- Ví dụ: Để lấy phần tử thứ 3 ta làm như sau
Dim str1, str2 as string
Str1= TenListBox.List 3
Str2= TenComboBox.List 3

5.4 Thuộc tính ListIndex

- Thuộc tính ListIndex: Chỉ vị trí của phần tử được chọn trong danh sách.

ListIndex=0 'Phần tử đầu tiên

ListIndex=1 'Phần tử thứ 2

...

ListIndex=ListCount-1 'Phần tử cuối cùng

5.5 Thuộc tính ListIndex

- Như vậy ngoài phương pháp dùng thuộc tính Text để lấy giá trị phần tử đang được chọn trong danh sách ta có thể dùng kết hợp thuộc tính List và ListIndex để lấy giá trị phần tử đang được chọn.

Xem ví dụ sau

```
Dim str1, str2 as string
```

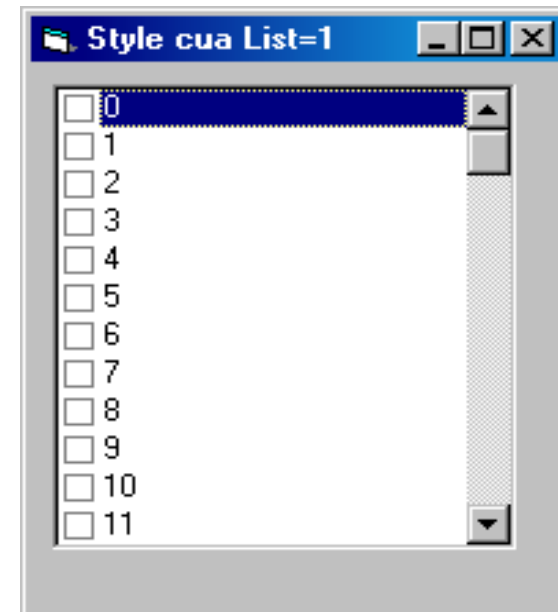
```
Str1= <TenListBox>.Text
```

```
Str2= <TenListBox>.List <TenListBox>.ListIndex
```

‘Lúc này str1=str2

5.6 Thuộc tính Style

- Thuộc tính Style của ComboBox
 - Style=0: Cho phép nhập và chọn dữ liệu.
 - Style=1: Không hiển thị hộp chọn, cho phép nhập dữ liệu.
 - Style=2: Chỉ chọn dữ liệu.
- Thuộc tính Style của ListBox
 - Style=0: Dạng chuẩn.
 - Style=1: Hiển thị ô check bên trái mỗi phần tử.



5.7 Thuộc tính Selected của ListBox

- Thuộc tính Selected của ListBox dùng để kiểm tra xem phần tử có được chọn hay không.
 - Selected(i)=True: Phần tử tại vị trí thứ i được chọn
 - Selected(i)=False: Phần tử tại vị trí thứ i không được chọn

5.8 Thuộc tính MultiSelect của ListBox

- MultiSelect =0: Chọn từng hàng
- MultiSelect =1: Được chọn cùng lúc nhiều hàng. (Đè phím Ctrl để chọn liên tục)
- MultiSelect=2: Được chọn cùng lúc nhiều hàng.(Đè phím Shift hoặc phím Shift để chọn liên tục)

Thuộc tính này được thiết lập khi thuộc tính Style=0)

5.8 Thuộc tính MultiSelect của ListBox (tt)

Giải thuật duyệt qua các phần tử được chọn

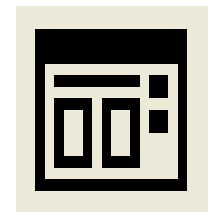
```
‘-----  
dim i as Integer  
For i=0 to <Tên ListBox>.ListCount-1  
    If <Tên ListBox>.Selected=True then  
        ‘Công việc cần thực hiện trên phần tử được chọn  
    else  
        ‘Công việc cần thực hiện trên phần tử không chọn  
    End If  
Next i
```

6. Điều khiển nâng cao

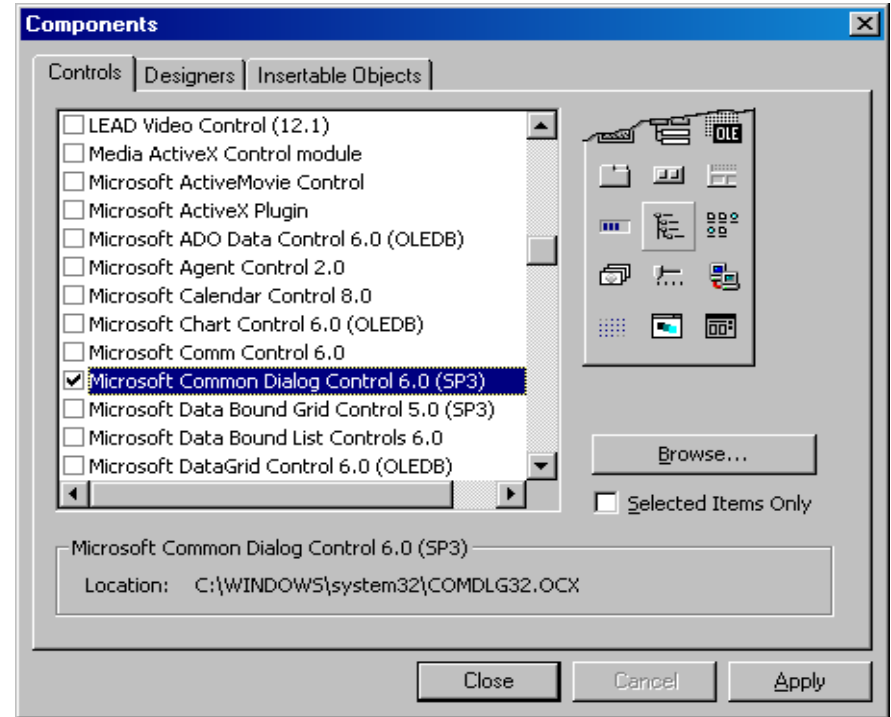
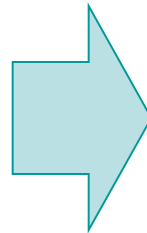
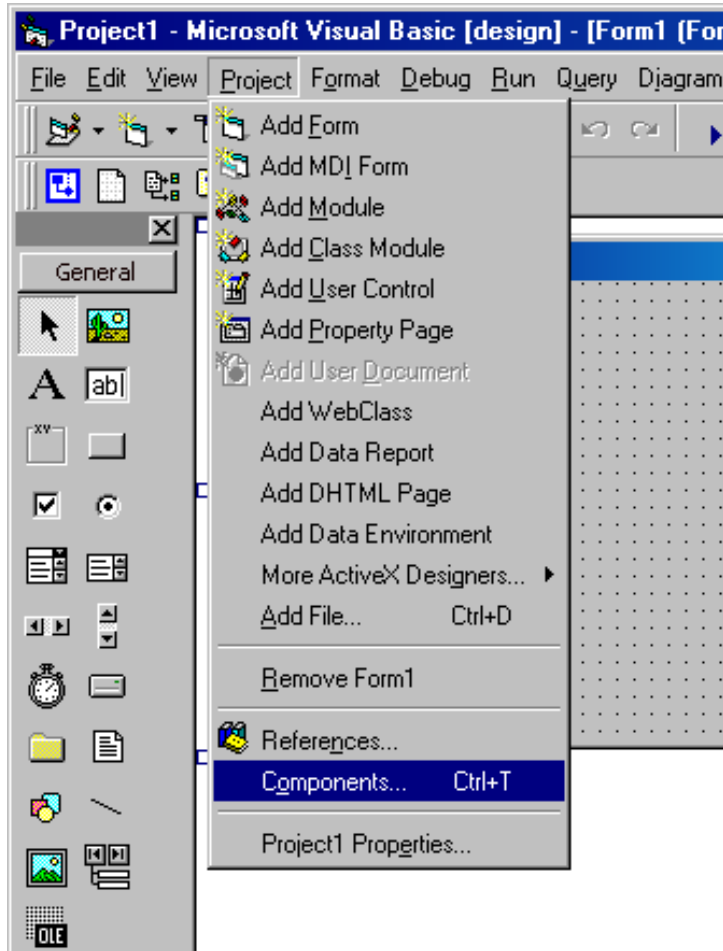
- CommonDailog
- RichTextBox
- FileListBox, DriveListBox, DirListBox
- MediaPlayer, WindowsMediaPlayer
- ImageList, Toolbar

6.1 CommonDialog

- Công dụng: Cho phép người dùng hiển thị các hộp thoại chọn đường dẫn khi lưu và mở file, chọn màu, chọn Font chữ, chọn máy in...
- Thuộc tính DialogResult: Đặt caption cho hộp thoại khi hiển thị.
- Thêm đối tượng CommonDialog: Vào menu Project – Components – Check vào Microsoft Common Dialog Control 6.0 – Nhấp OK. Biểu tượng Common Dialog:



6.1 Add CommonDialog (tt)



6.1 Các phương thức

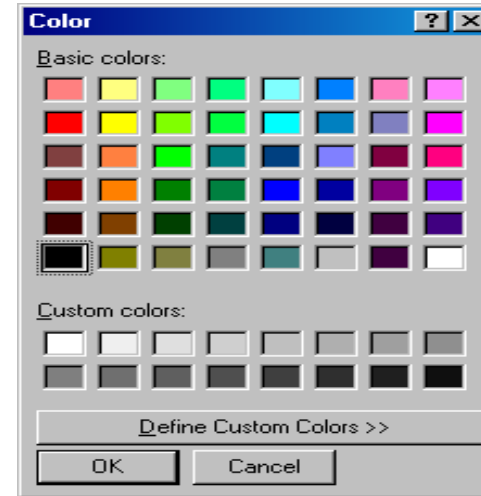
- ShowColor
- ShowFont
- ShowOpen
- ShowSave

6.1 ShowColor

- **Công dụng:** Hiện hộp thoại chọn màu
- Mặc định khi sử dụng phương thức ShowColor sẽ hiển thị hộp thoại chọn màu ở hình 1. Để hiển thị được hộp thoại chọn màu ở hình 2 ta cần thiết lập thuộc tính:

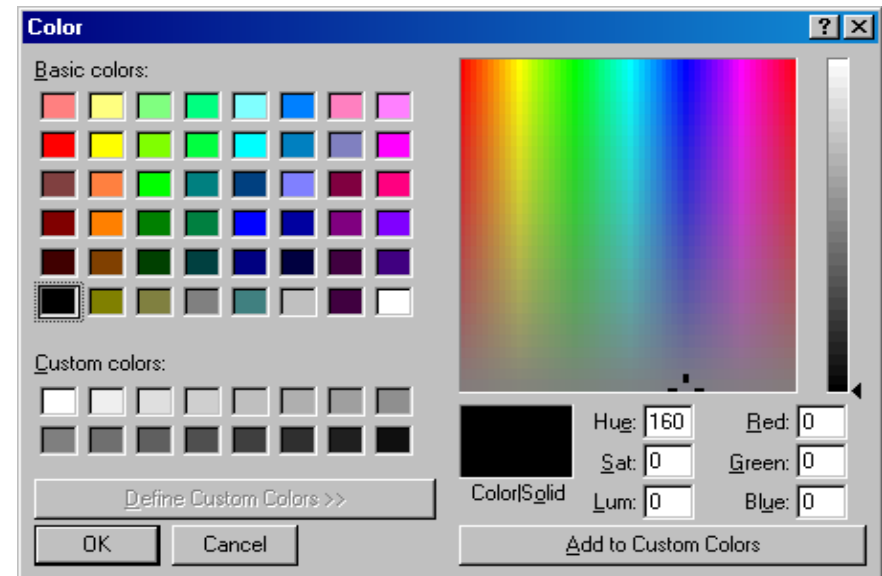
Flags = cdlCCFullOpen trước khi gọi phương thức **ShowColor**

- Muốn lấy giá trị màu mà người dùng đã nhập ta dùng thuộc tính **Color**



Hình 1

Hình 2



6.1 ShowColor (tt)

Ví dụ: Khi chạy chương trình sẽ hiển thị hộp thoại chọn màu. Giá trị màu sẽ hiển thị sau khi người dùng chọn màu.

Private Sub Form_Load()

With Me.CommonDialog1

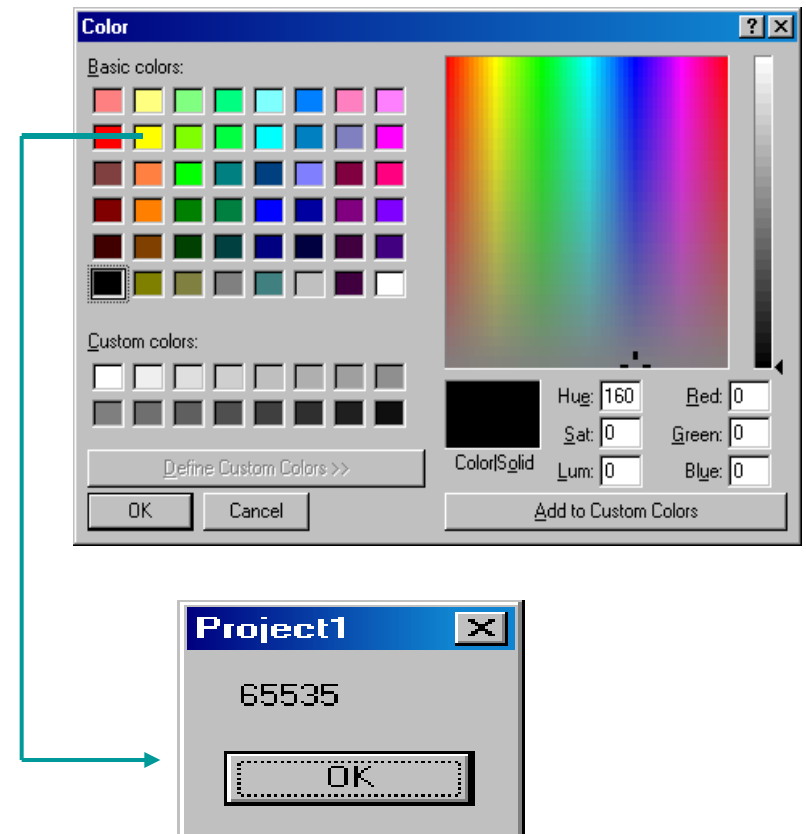
.Flags = cdlCCFullOpen

.ShowColor

MsgBox .Color

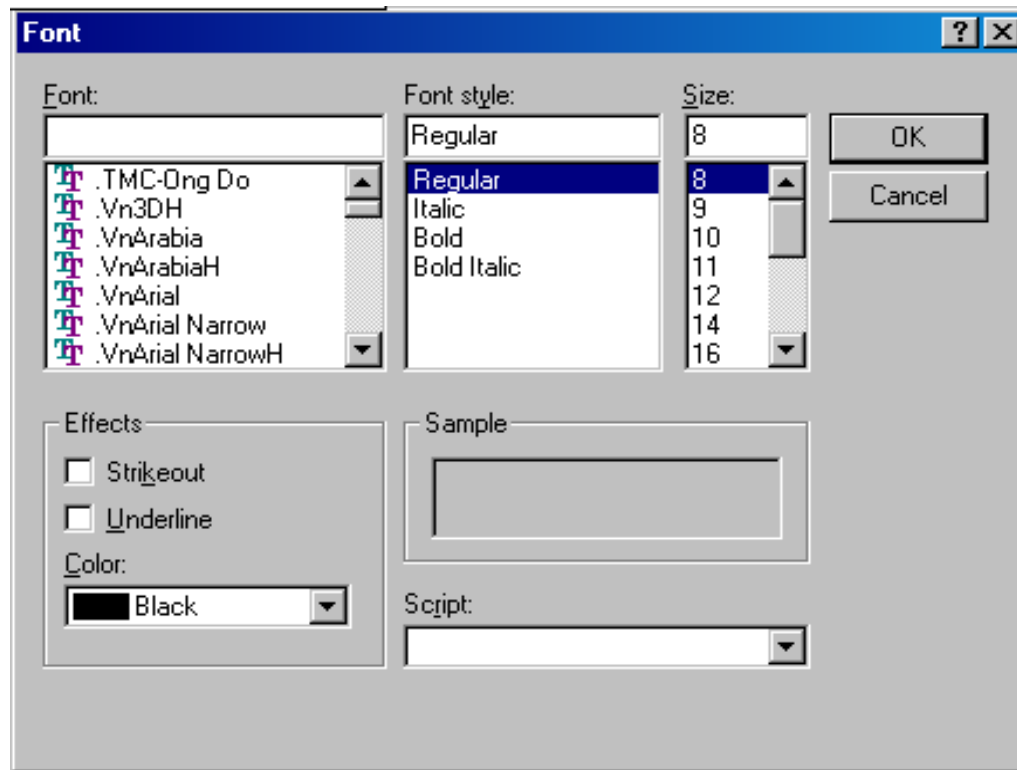
End With

End Sub



6.2 ShowFont

- Cú pháp: <Tên đối tượng>.ShowFont
Hiển thị hộp thoại chọn Font.



6.2 ShowFont

- Để hiển thị hộp thoại Font ta làm như sau:

With <Tên đối tượng>

.Flags = cdlCFEffects Or cdlCFBoth

.ShowFont

End With

Thông tin có thể lấy sau khi hiển thị hộp thoại Font

– **Color:** Màu chữ.

– **FontName:** tên Font.

– **FontSize:** Kích thước chữ.

– **FontBold:** Chữ in đậm.

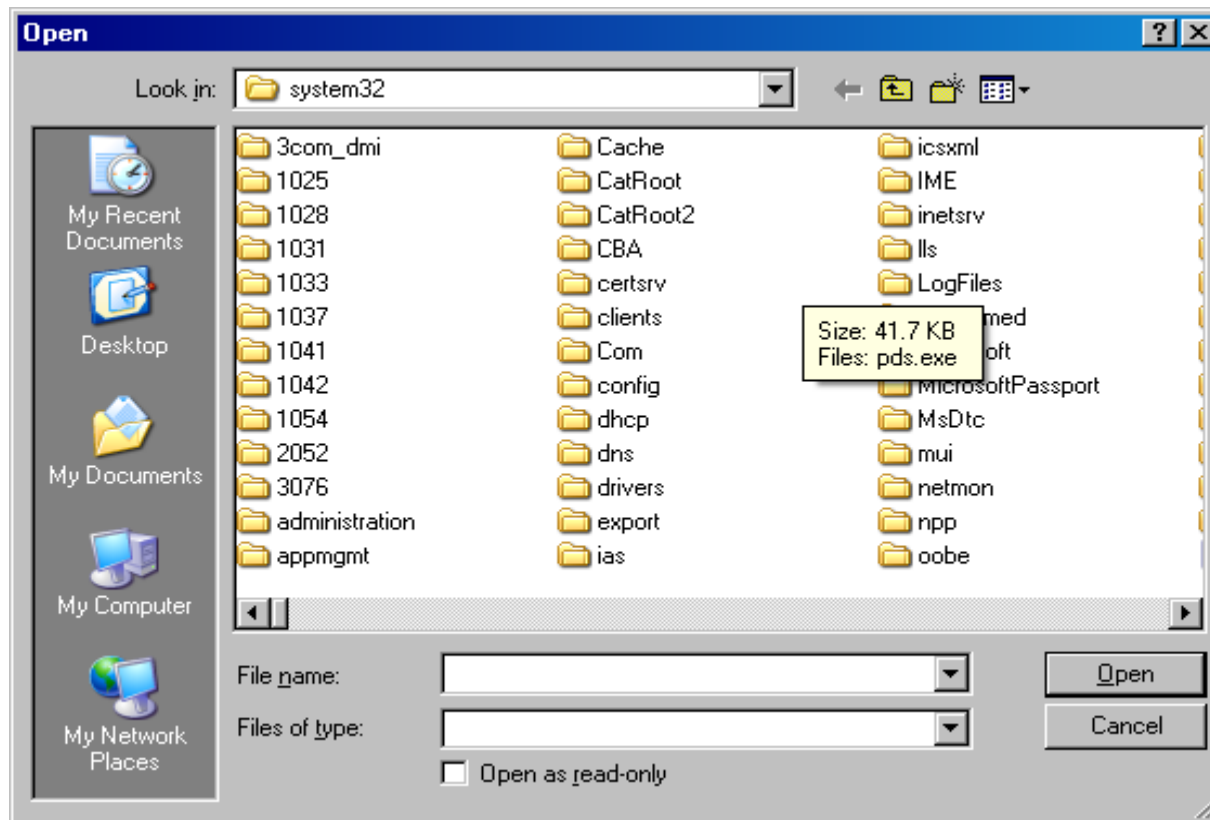
– **FontItalic:** Chữ in nghiêng.

– **FontUnderline:** Chữ gạch chân.

– **FontStrikethru:** Đường gạch ngang chữ.

6.3 ShowOpen

- Cú pháp: <Tên đối tượng>.ShowOpen
Hiển thị hộp thoại mở file.



6.3 ShowOpen (tt)

Một số thuộc tính cần thiết lập trước khi hiển thị:

Filter: Loại file sẽ hiển thị trong hộp thoại.

- **Ví dụ:** Để chỉ chọn các tập tin có phần mở rộng là .txt ta cần thiết lập như sau:

<Tên đối tượng>.Filter = "Van ban|*.txt"

- **Ví dụ:** Để chỉ chọn các tập tin có phần mở rộng là .txt và .bmp ta cần thiết lập như sau:

<Tên đối tượng>.Filter = "Hinh anh|*.bmp|Van ban|*.txt"

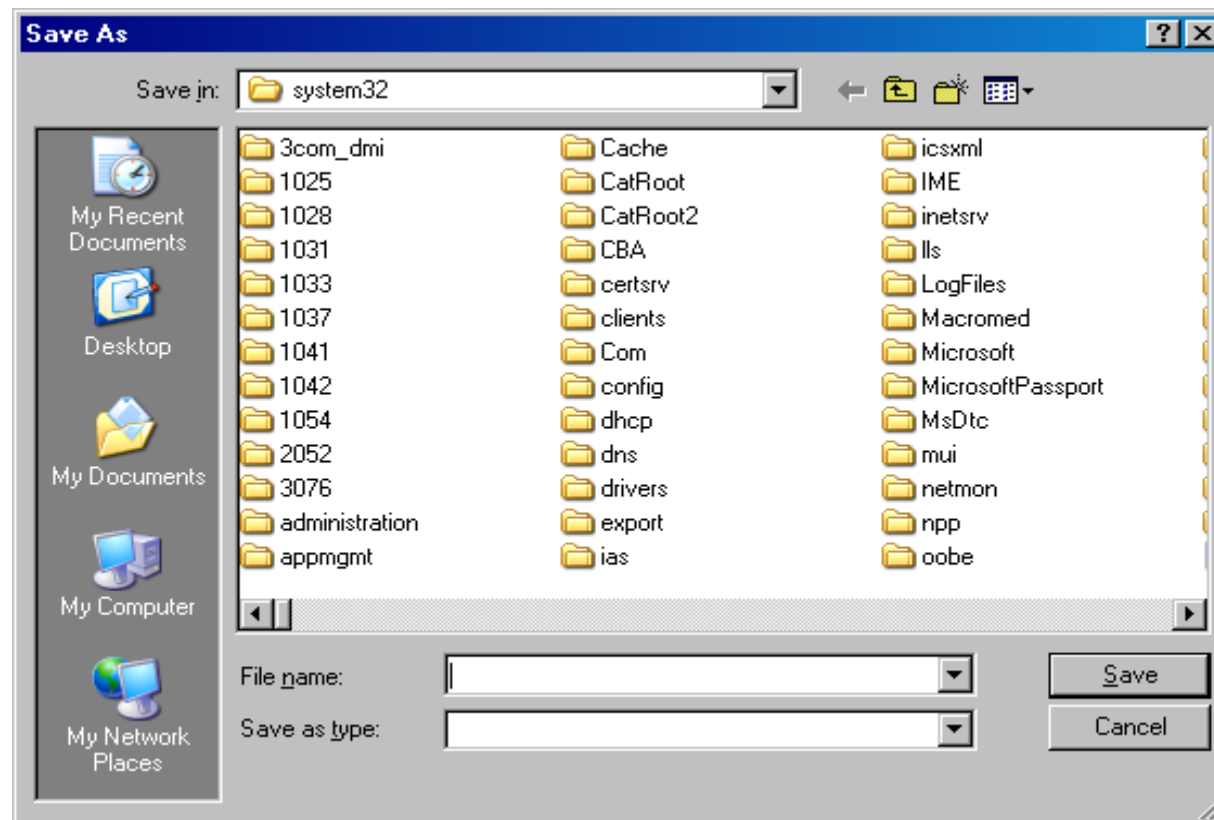
- **Ví dụ:** Để hiển thị tất cả các tập tin trong hộp thoại ta thiết lập như sau:

<Tên đối tượng>.Filter = "Tat ca tap tin|*.*"

FileName: *Đường dẫn đến tập tin sau khi người dùng chọn Open.*

6.4 ShowSave

- Cú pháp: <Tên đối tượng>.ShowSave
Hiển thị hộp thoại lưu file.



Bài giảng VB 1

6.4 ShowSave (tt)

- Một số thuộc tính cần thiết lập trước khi hiển thị

Filter: (Tương tự hộp thoại Open).

FileName: Đường dẫn đến tập tin sau khi người dùng chọn Save.

7. RichTextBox

- **Thêm đối tượng:** Vào menu Project - Chọn Components – Microsoft Rich Textbox Control 6.0 - Chọn Apply - Chọn OK.
- **Các thuộc tính thường dùng:**
 - MultiLine: =True cho phép nội dung hiển thị trên nhiều dòng, ngược lại =False.
 - ScrollBars: =0 không xuất hiện thanh cuộn. =1 Chỉ hiện thanh cuộn ngang. =2 Chỉ hiện thanh cuộn đứng. =3 Xuất hiện cả 2 thanh cuộn.
 - Locked: =True người dùng không thể thay đổi nội dung văn bản, ngược lại =False.
 - SelBold: =True vùng văn bản được chọn sẽ in đậm, ngược lại =False.

7. RichTextBox (tt)

- **SelColor:** Màu chữ cho vùng văn bản được chọn.
- **SelFontName:** Font chữ cho vùng văn bản được chọn.
- **SelFontSize:** Kích thước chữ cho vùng văn bản được chọn.
- **SelItalic:** =True vùng văn bản được chọn in nghiêng ngược lại = False.
- **SelAlignment:** Canh dữ liệu cho vùng văn bản được chọn. (0: trái, 1 phải, 2 giữa).
- **SelText:** Lấy nội dung văn bản được chọn.
- **SelUnderline:** =True gạch chân vùng văn bản được chọn, ngược lại = False.
- **SelStrikeThru:** =True gạch chân nội dung văn bản được chọn, ngược lại =False.

7. RichTextBox (tt)

Các phương thức thường dùng:

- SetFocus: Đưa con trỏ vào RichTextBox.
- SaveFile <spath>: Lưu nội dung văn bản vào tập tin có đường dẫn spath.

Cú pháp: <tên đối tượng>.SaveFile <Đường dẫn đến file cần save>

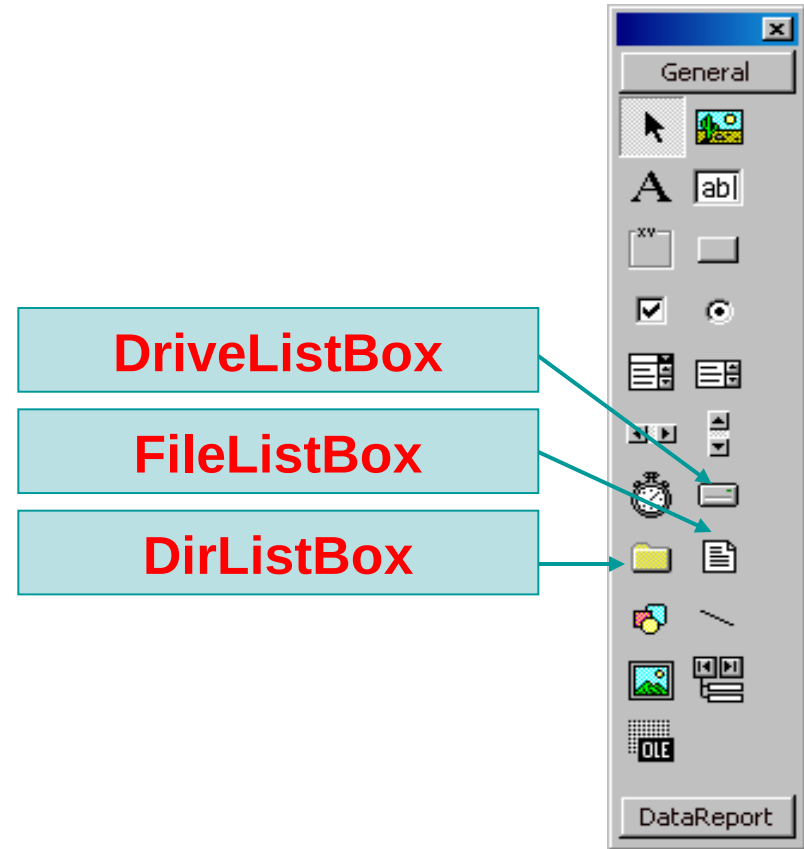
- LoadFile <spath>: Mở nội dung văn bản hiển thị trong RichTextBox, tập tin văn bản được lưu tại đường dẫn spath.

Cú pháp: <tên đối tượng>.LoadFile <Đường dẫn đến file cần open>

8. FileListBox, DriveListBox, DirListBox

- DriveListBox
 - Chọn ổ đĩa
- DirListBox
 - Chọn thư mục
- FileListBox
 - Chọn tập tin
 - Thuộc tính: Pattern
 - Qui định file sẽ hiển thị

Kết hợp 3 điều khiển trên để tạo công cụ cho người dùng chọn File



8. FileListBox, DriveListBox, DirListBox

```
Private Sub Dir1_Change()
```

```
    Me.File1.Path = Me.Dir1.Path
```

```
End Sub
```

```
Private Sub Drive1_Change()
```

```
    Me.Dir1.Path = Me.Drive1.Drive
```

```
End Sub
```

```
Private Sub File1_Click()
```

```
    Me.Text1.Text = Me.File1.FileName
```

```
    Me.Text2.Text = Me.File1.Path
```

```
    If Right(Me.File1.Path, 1) = "\" Then
```

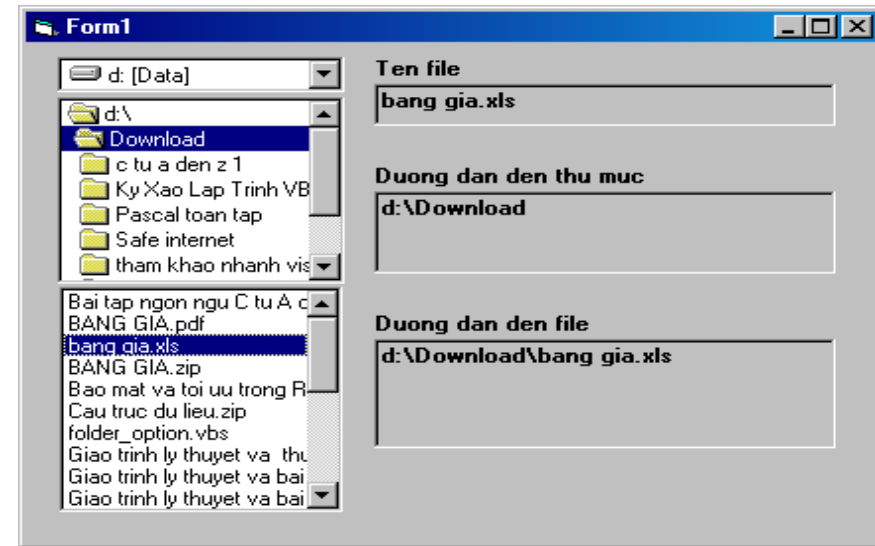
```
        Me.Text3.Text = Me.File1.Path & Me.File1.FileName
```

```
    Else
```

```
        Me.Text3.Text = Me.File1.Path & "\" & Me.File1.FileName
```

```
    End If
```

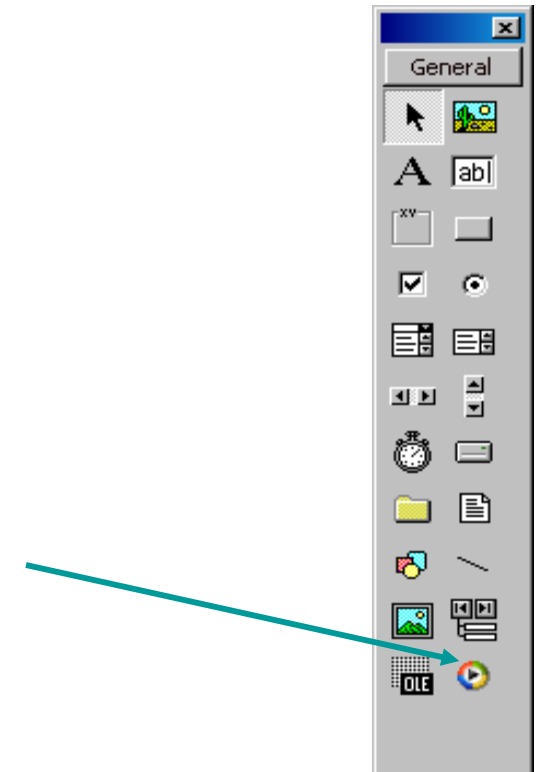
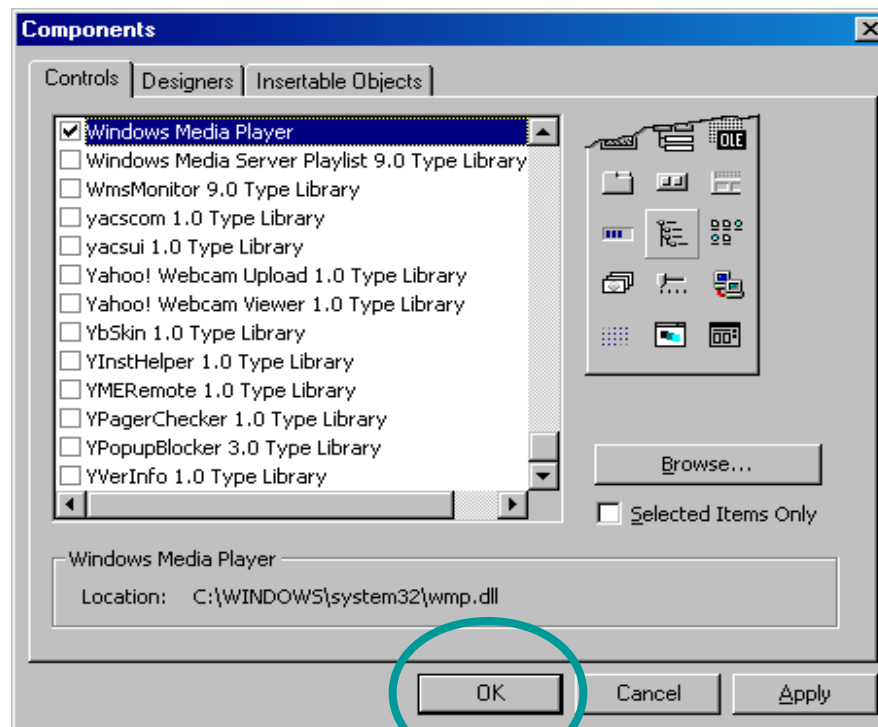
```
End Sub
```



9. WindowsMediaPlayer

- Thêm đối tượng

Vào menu Project – Components – Check vào Window Media Player - OK



Bài giảng VB 1

9. WindowsMediaPlayer (tt)

- Thuộc tính thường dùng
 - Controls.Next
 - Controls.Play
 - Controls.Stop
 - Controls.Previous
 - URL: Đường dẫn đến file cần hát

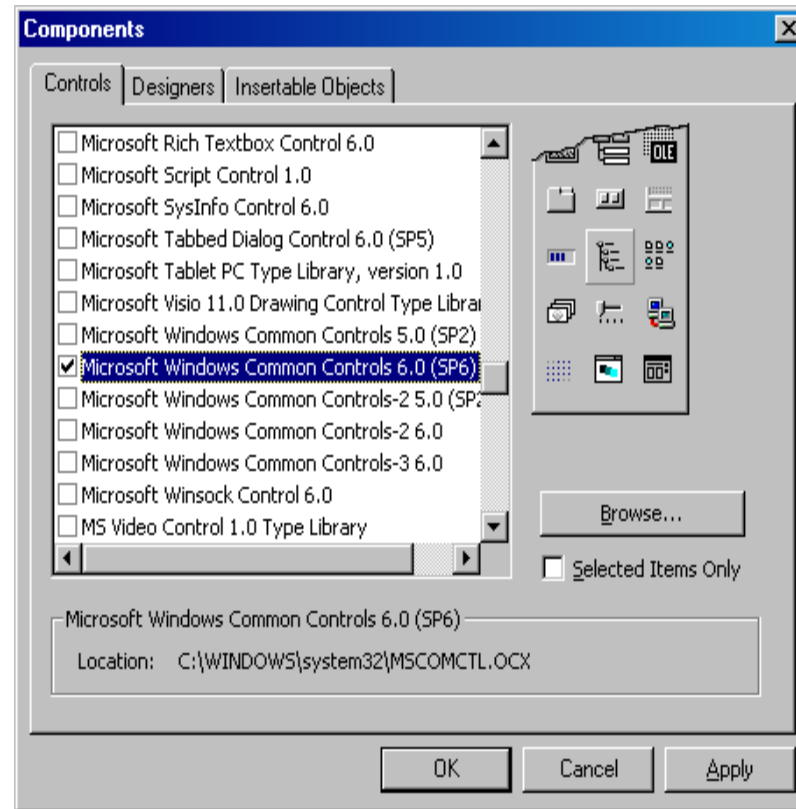
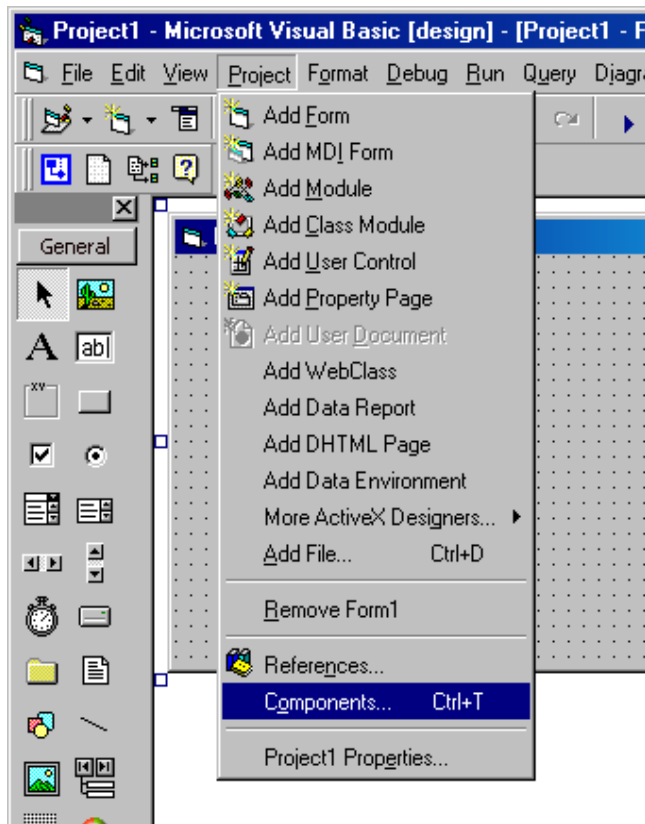
10. MediaPlayer

- Thêm đối tượng
 - Vào menu Project – Components – Check vào Window Media Player - OK
- Thuộc tính thường dùng
 - Next
 - Play
 - Stop
 - Previous
 - FileName: Đường dẫn đến file cần hát
- Icon:



10. ImageList, Toolbar

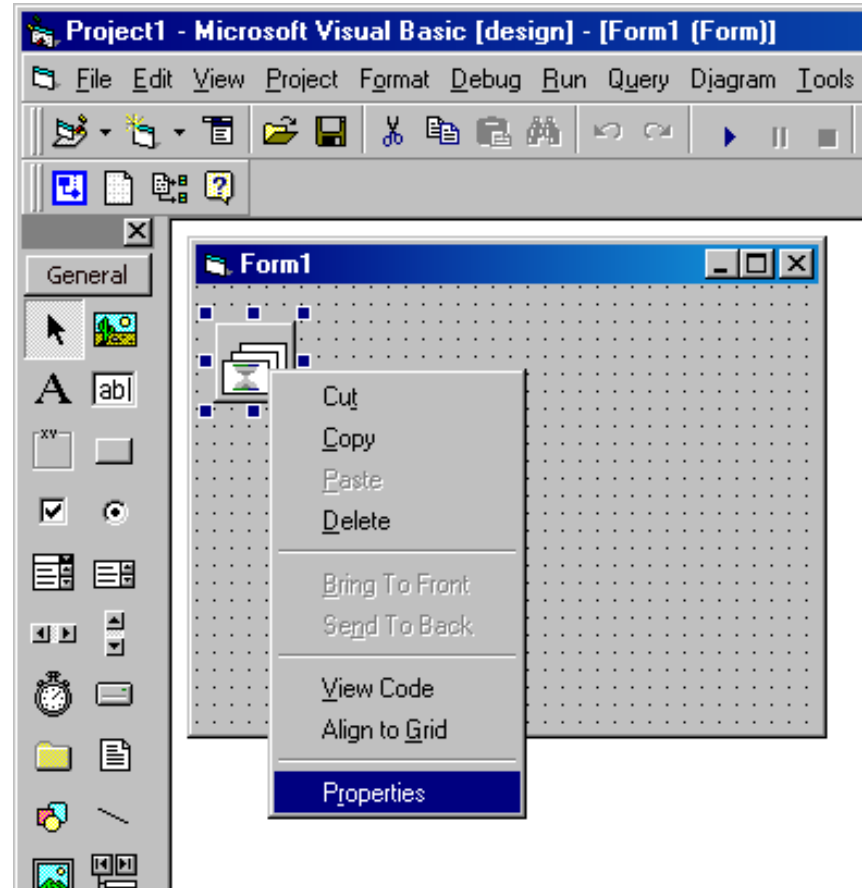
Thêm đối tượng: Project – Components – Microsoft common controls 6.0 – Apply.



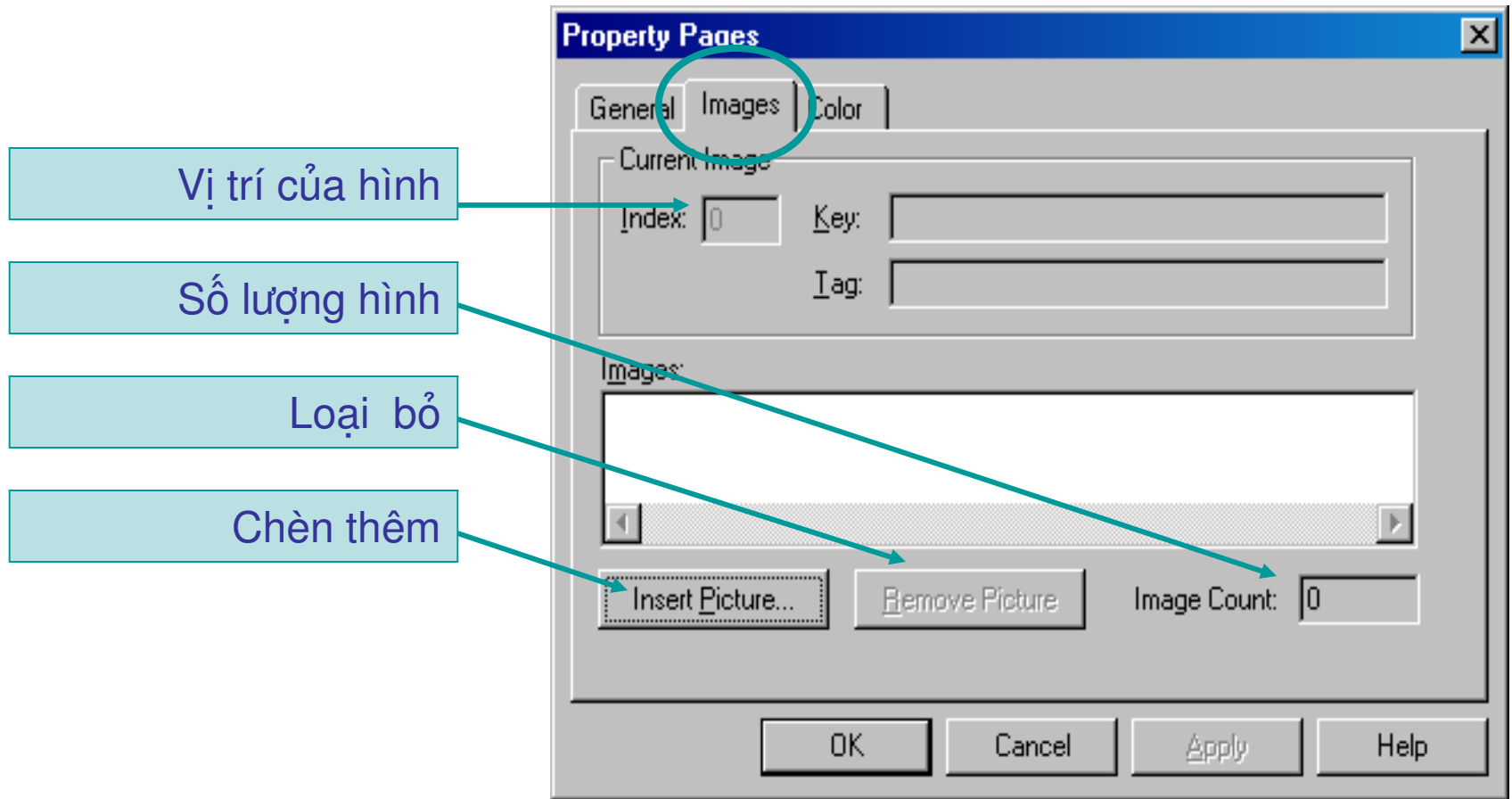
Bài giảng VB 1

10.1 ImageList

- Chức năng: Dùng để lưu trữ hình ảnh.
- Thêm hình ảnh:
 - Thêm lúc thiết kế
 - Vẽ đối tượng lên giao diện.
 - Right Click lên đối tượng chọn Property
 - Chọn Images



10.1 ImageList (Thêm lúc thiết kế)



10.1 ImageList (Thêm lúc chạy chương trình)

- Cú pháp:

`<Tên ImageList>.ListImages.Add [index], [key] ,
LoadPicture(Đường dẫn đến hình cần thêm)`

- Ví dụ:

`Me.ImageList1.ListImages.Add , ,
LoadPicture("c:\hinh.bmp")`

10.1 ImageList (Xoá lúc chạy chương trình)

- Cú pháp:
`<Tên ImageList>.ListImages.Remove <index>`
- Ví dụ: Xoá hình thứ nhất trong ImageList
`Me.ImageList1.ListImages.Remove 1`

10.1 ImageList

- Lấy hình ảnh trong ImageList

– Cú pháp:

Me.ImageList1.ListImages(Vị trí index).Picture

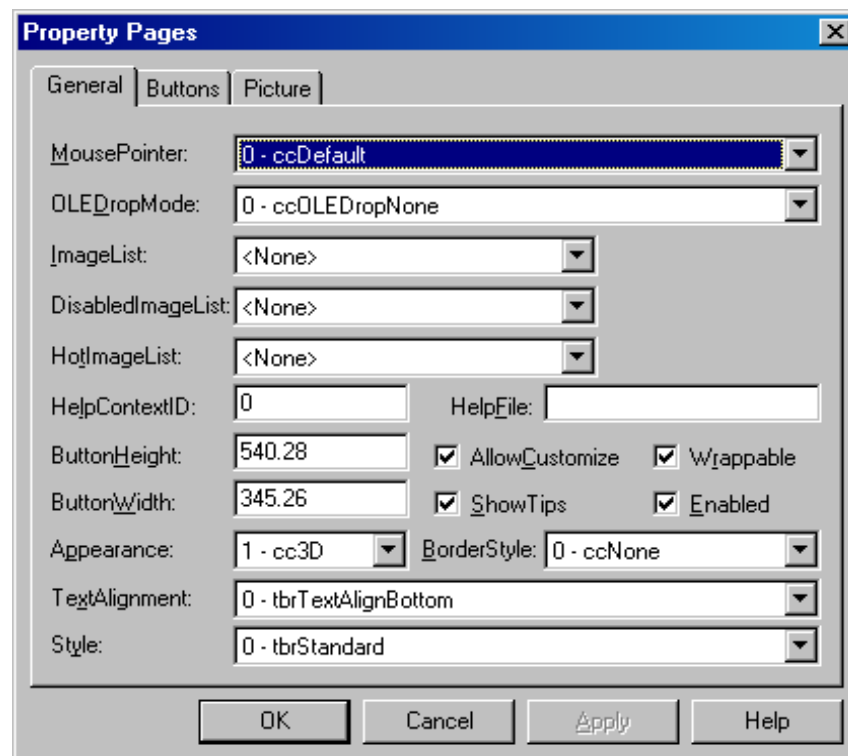
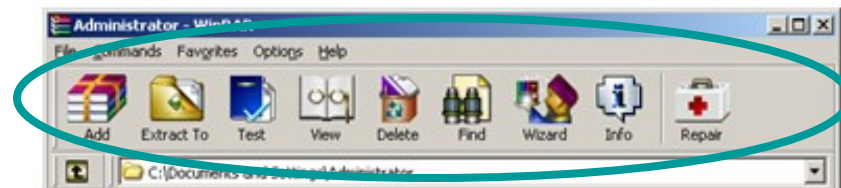
– Ví dụ: Lấy hình thứ nhất trong ImageList hiển thị lên form, ta viết lệnh như sau:

Me.Picture =

Me.ImageList1.ListImages.Item(1).Picture

10.2 Toolbar

- Công dụng: Tạo thanh công cụ cho chương trình.
- Thiết lập:
 - Vẽ ImageList
 - Add hình vào Imagelist
 - Vẽ ToolBar
 - Right click vào Toolbar- Chọn Properties
 - Xuất hiện hình bên

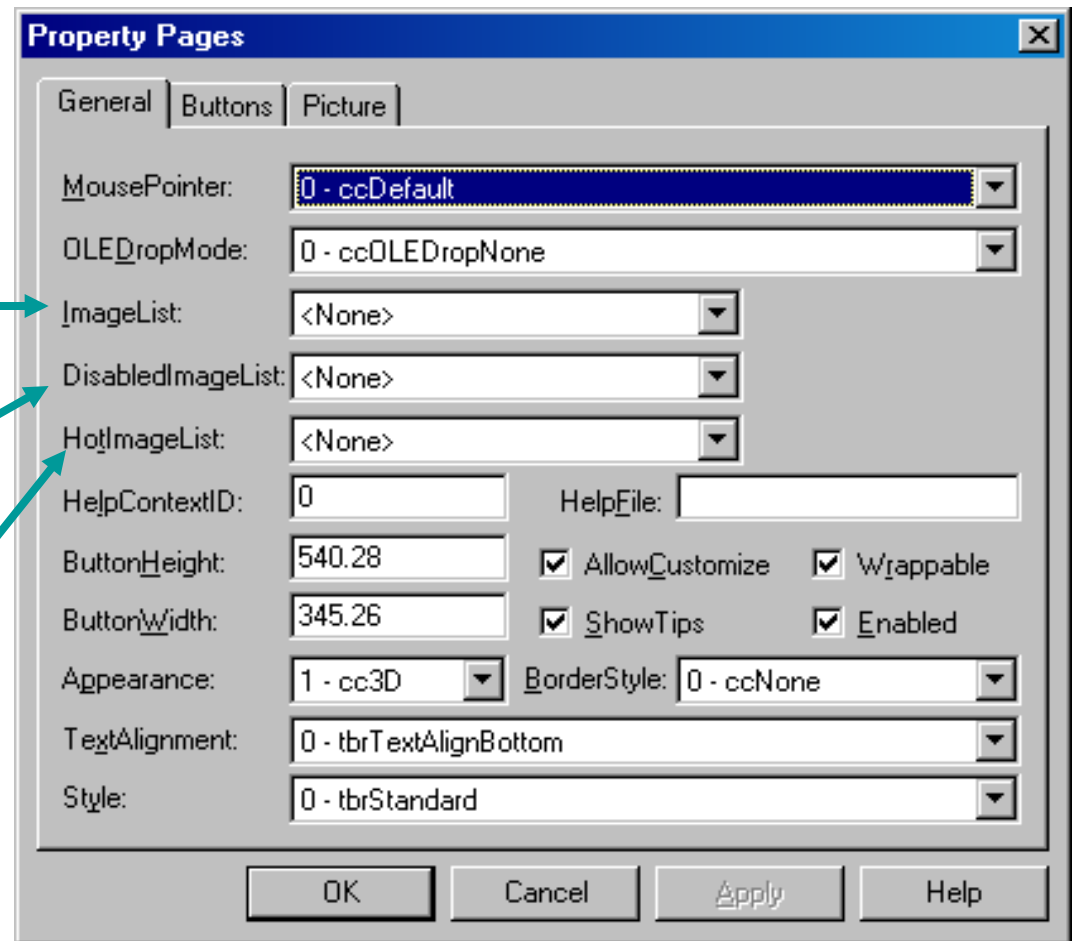


10.2 Toolbar (Chi tiết thiết lập)

ImageList chứa hình
cho Toolbar

ImageList chứa hình
cho Toolbar khi bị
Disabled

ImageList chứa hình
cho Toolbar khi di
chuyển chuột lên
Toolbar



Lập trình visual basic

5t

Chuyên đề 4

MẢNG – MẢNG CÁC ĐIỀU KHIỂN



Bài giảng VB 1

Nội dung chính

- Mạng tĩnh - mạng động
- Các hàm về mạng
- Mạng các điều khiển

1. Mảng tĩnh - mảng động

- Mảng tĩnh (Static array)

- Khai báo:

Dim <tên mảng>(phần tử đầu to phần tử cuối) as
<kiểu dữ liệu>

Dim <tên mảng>(phần tử đầu to phần tử cuối ,
phần tử đầu to phần tử cuối) as <kiểu dữ liệu>

- Chú ý: Nếu không để phần tử đầu thì phần tử đầu = 0

- Truy xuất phần tử trên mảng:

Tênmảng(vị trí phần tử trong mảng)

1. Mảng tĩnh - mảng động (tt)

– Ví dụ:

Dim Number(1 to 100) **as** integer.

Dim Number(100) **as** integer

‘Lấy giá trị phần tử thứ 5: **Number(5)**

1. Mảng tĩnh - mảng động (tt)

- Mảng động (Static Array)

- Khai báo:

Dim <tên mảng>() as <kiểu dữ liệu>

- Chú ý: Khi khai báo mảng động ta không cho biết số phần tử của mảng. Lúc này mảng chưa sử dụng được, khi bạn muốn sử dụng phải khai báo lại. Cú pháp khai báo lại như sau:

ReDim <tên mảng>(số phần tử)

- Chú ý: Tên mảng phải trùng tên mảng đã khai báo ban đầu.

1. Mảng tĩnh - mảng động (tt)

- Ví dụ 1:
Dim M() as String
M(1)="A" 'Dòng lệnh này sẽ bị lỗi
'Khai báo lại
ReDim M(6)
'Mảng tạo có 7 phần tử bắt đầu là 0
- Ví dụ 2:
Dim M() as Long
'Khai báo lại
ReDim M(4 to 6)
'Mảng tạo được có 3 phần tử
M(2)=3 'Dòng lệnh này bị lỗi.
M(6)=45

1. Mảng tĩnh - mảng động (tt)

- Từ khóa **Preserve**: Khi khai báo lại số lượng phần tử của một mảng động (khai báo lần 2) thì giá trị của các phần tử trong mảng sẽ được khởi tạo lại giá trị mặc định. Muốn giá trị các phần không khởi tạo lại ta dùng từ khóa Preserve khi khai báo.

Ví dụ 1:

Dim M() as byte

ReDim M(3) 'Khai báo lại lần 1

M(1)=5

M(2)=3

'Mảng M có 02 phần tử M(1)=5 và M(2)=3

ReDim M(4) 'Khai báo lại lần 2

'Mảng M có 02 phần tử M(1)=0 và M(2)=0

1. Mảng tĩnh - mảng động (tt)

- Ví dụ 2:

Dim M() as byte

ReDim M(3) 'Khai báo lại lần 1

M(1)=5

M(2)=3

'Mảng M có 02 phần tử M(1)=5 và M(2)=3

ReDim Preserve PM(4) 'Khai báo lại lần 2

'Lúc này mảng M có 02 phần tử M(1)=5 và M(2)=3

2. Một số hàm dùng trong mảng

- **Hàm Lbound:** Vị trí phần tử đầu tiên của mảng.
- **Hàm Ubound:** Vị trí phần tử cuối cùng của mảng.

- **Ví dụ 1:**

Dim M(2 to 6) as Byte

Dim n, m as byte

n=Lbound(m)

m=Ubound(m)

‘Lúc này n=2 và m=6

- **Ví dụ 2:**

Dim M(6) as Byte

Dim n, m as byte

n=Lbound(m)

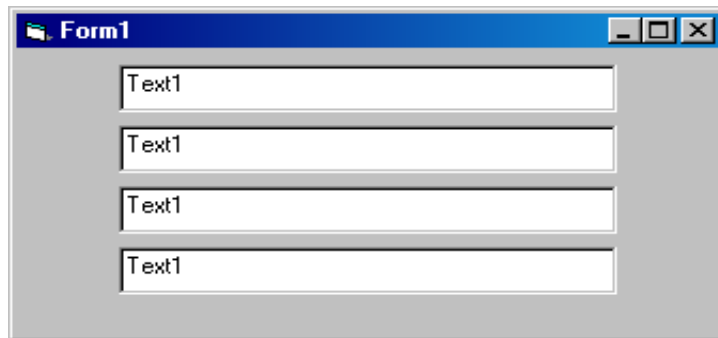
m=Ubound(m)

‘Lúc này n=0 và m=6

Bài giảng VB 1

3. Mảng các điều khiển

- ĐN: Mảng điều khiển là tập hợp các điều khiển cùng tên được phân biệt thông qua thuộc tính index của điều khiển.
- Chú ý: Trên 1 form không tồn tại 2 điều khiển cùng tên và chỉ số index.



3. Mảng các điều khiển

- **Ví dụ:** Trên giao diện có 4 Textbox. Để người dùng không thể thay đổi nội dung của các textbox trên ta viết lệnh như sau:

```
Text1.Locked=True
```

```
Text2.Locked=True
```

```
Text3.Locked=True
```

```
Text4.Locked=True
```

- **Nhận xét:** Nếu ta có trên 10 textbox thì sao? Ta phải viết tập hợp 10 dòng lệnh.
- **Nếu ta dùng mảng điều khiển thì sao?**

```
Dim i as Integer
```

```
For i=0 to 4
```

```
    Text(i).Locked=True
```

```
Next i
```

