

Đề Cương Kỹ Thuật Lập Trình

I. Biến.

- **Khái niệm:** là ô nhớ hoặc vùng nhớ trên máy, dùng để địa chỉ hóa, đưa dữ liệu vào, tính toán.
- **Các loại biến, cách khai báo, phạm vi xác định và sử dụng:**

Khai báo chung: `[Dim] <tên biến> [As <kiểu dữ liệu>]`

Or `[dim] <tên biến> <nhận dạng biến>`

Kiểu dữ liệu	Mô tả
Boolean	Dùng để xác định đúng /sai. Gồm 2 giá trị: TRUE & FALSE.
Byte	Là số nguyên mã hóa =1 byte. Dải xác định: 0 – 255
Integer	Dải xác định: -32768 – 32767 ; không biểu diễn sau dấu chấm phẩy. Nhận dạng: % . số nguyên.
Long	Dải xác định: 2^{-31} đến $2^{31}-1$. Nhận dạng: &. Số nguyên dài.
Single	Dải xác định: 38 chữ số. Độ chính xác: 7 chữ số sau dấu thập phân. Nhận dạng: ! . số thực.
Double	Dải xác định: 308 chữ số. độ chính xác: 14 chữ số sau dấu thập phân. Nhận dạng: # . Số thực dài.
Currency	Dải xác định: 15 chữ số. độ chính xác: 4 chữ số thập phân. nhận dạng: @
String	Dải xác định: 2 tỉ kí tự ký tự hay ký tự số, thậm chí là các giá trị đặc biệt như ^%@. Giá trị kiểu chuỗi được đặt giữa 2 dấu ngoặc kép
Date	Dải: 0h ngày 1/1/100 đến 31/12/9999, giá trị được đặt giữa cặp dấu ##. Việc định dạng hiển thị tùy thuộc vào việc thiết lập trong Control Panel của windows. Dùng để lưu trữ thông tin về thời gian.
Variant	Chứa mọi giá trị của các kiểu dữ liệu khác, kể cả mảng. khi 1 biến k được khai báo, Vb mặc định đó là kiểu Variant.

II. Điều khiển dòng chương trình.

1. Vòng lặp xác định.

- Lệnh rẽ nhánh if.

Một dòng lệnh:

If <điều kiện> **Then** <dòng lệnh>

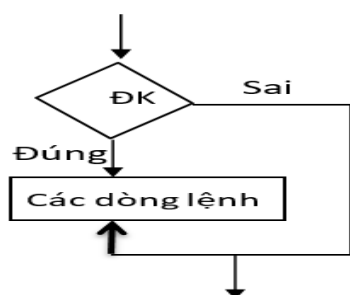
Nhiều dòng lệnh:

If <điều kiện> **Then**

Các dòng lệnh

End If

- Lưu đồ cú pháp:



+ Trong đó, <điều kiện>: biểu thức mà kết quả trả về kiểu Boolean.

+ Ý nghĩa câu lệnh: *Các dòng lệnh* hay *dòng lệnh* sẽ được thi hành nếu như điều kiện là đúng. Còn nếu như điều kiện là sai thì câu lệnh tiếp theo sau cấu trúc If ... Then được thi hành.

Dạng đầy đủ: **If ... Then ... Else**

If <điều kiện 1> **Then**

[Khối lệnh 1]

ElseIf <điều kiện 2> **Then**

[Khối lệnh 2]...

[Else

[Khối lệnh n]]

End If

VB sẽ kiểm tra các điều kiện, nếu điều kiện nào đúng thì khối lệnh tương ứng sẽ được thi hành. Ngược lại nếu không có điều kiện nào đúng thì khối lệnh sau từ khóa Else sẽ được thi hành.

- Vòng lặp select case.

với cấu trúc **Select...Case**, biểu thức điều kiện sẽ được tính toán một lần vào đầu cấu trúc, sau đó VB sẽ so sánh kết quả với từng trường hợp (**Case**). Nếu bằng nó thì hành khối lệnh trong trường hợp (**Case**) đó.

Select Case <biểu thức kiểm tra>

Case <Danh sách kết quả biểu thức 1>

[Khối lệnh 1]

Case <Danh sách kết quả biểu thức 2>

[Khối lệnh 2]

.

.

.

[**Case Else**

[Khối lệnh n]]

End Select

Mỗi danh sách kết quả biểu thức sẽ chứa một hoặc nhiều giá trị. Trong trường hợp có nhiều giá trị thì mỗi giá trị cách nhau bởi dấu phẩy (,). Nếu có nhiều Case cùng thỏa điều kiện thì khối lệnh của Case đầu tiên sẽ được thực hiện.

- Vòng lặp for...next.

Đây là cấu trúc biết trước số lần lặp, ta dùng biến đếm tăng dần hoặc giảm dần để xác định số lần lặp.

For <biến đếm> = <điểm đầu> **To** <điểm cuối> [**Step** <bước nhảy>] [khối lệnh]

Next

Biến đếm, điểm đầu, điểm cuối, bước nhảy là những giá trị số (Integer, Single,...). Bước nhảy có thể là âm hoặc dương. Nếu bước nhảy là số âm thì điểm đầu phải lớn hơn điểm cuối, nếu không khối lệnh sẽ không được thi hành.

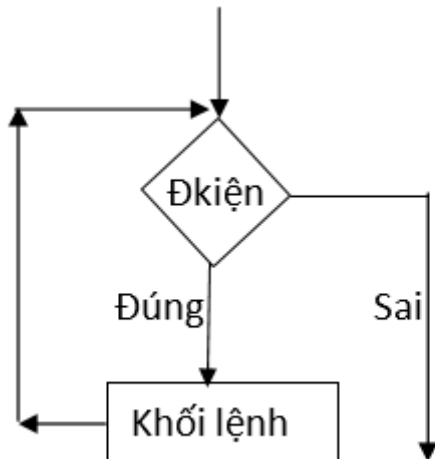
Khi Step không được chỉ ra, VB sẽ dùng bước nhảy mặc định là một.

2. Lặp không xác định

Do ... Loop: Đây là cấu trúc lặp không xác định trước số lần lặp, trong đó, số lần lặp sẽ được quyết định bởi một biểu thức điều kiện. Biểu thức điều kiện phải có kết quả là True hoặc False. Cấu trúc này có 4 kiểu:

Kiểu 1:**Do While** <điều kiện>

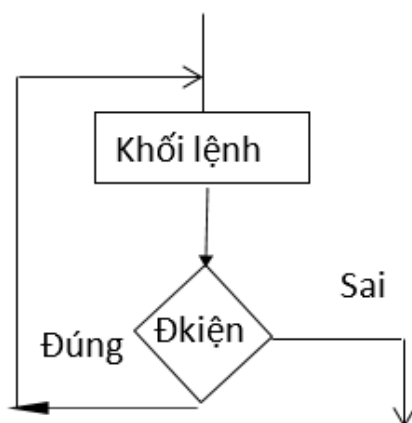
<khối lệnh>

Loop

Khối lệnh sẽ được thi hành đến khi nào điều kiện không còn đúng nữa. Do biểu thức điều kiện được kiểm tra trước khi thi hành khối lệnh, do đó có thể khối lệnh sẽ không được thực hiện một lần nào cả.

Kiểu 2:**Do**

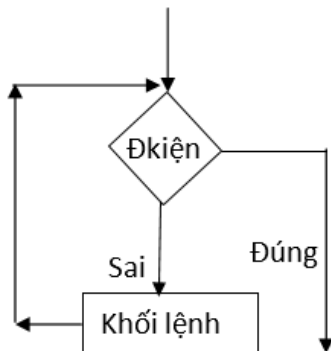
<khối lệnh>

Loop While <điều kiện>

Khối lệnh sẽ được thực hiện, sau đó biểu thức điều kiện được kiểm tra, nếu điều kiện còn đúng thì, khối lệnh sẽ được thực hiện tiếp tục. Do biểu thức điều kiện được kiểm tra sau, do đó khối lệnh sẽ được thực hiện ít nhất một lần.

Kiểu 3:**Do Until** <điều kiện>

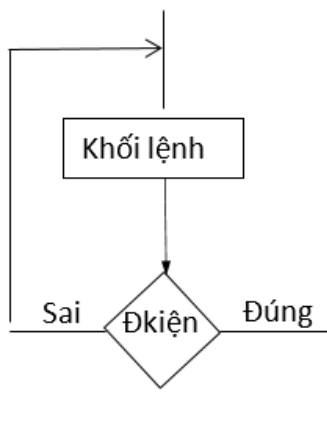
<Khối lệnh>

Loop

Cũng tương tự như cấu trúc Do While ... Loop nhưng khác biệt ở chỗ là khối lệnh sẽ được thi hành khi điều kiện còn sai.

Kiểu 4:**Do**

<Khối lệnh>

Loop Until <điều kiện>

Khối lệnh được thi hành trong khi điều kiện còn sai và có ít nhất là một lần lặp.

III. Biến cấu trúc (mảng).

Mảng là tập hợp các phần tử có cùng một kiểu.

Dùng mảng sẽ làm cho chương trình đơn giản và gọn hơn vì ta có thể sử dụng vòng lặp. Mảng sẽ có biên trên và biên dưới, trong đó các thành phần của mảng là liên tiếp trong khoảng giữa hai biên này.

Có hai loại biến mảng: mảng có chiều dài cố định và mảng có chiều dài thay đổi lúc thi hành.

Mảng có chiều dài cố định:

Dim <Tên biến mảng>(<Kích thước>) [**As** <Kiểu>]

Lúc này phần tử đầu tiên có chỉ số là 0 & phần tử cuối cùng có chỉ số là <Kích thước>.

Dim <Tên biến mảng>(<Chỉ số đầu> **To** <Chỉ số cuối>) [**As** <Kiểu>]

- Hàm **UBound** trả về biên trên của một mảng.
- Hàm **LBound** trả về biên dưới của một mảng.

✓ *Lưu ý:* ta có thể khai báo một mảng nhiều chiều như sau

Dim Multi3D (3, 1 To 10, 9) **As** **Double**

Khai báo này tạo ra một mảng 3 chiều với kích thước 4 x 10 x 10.

Mảng động:

- Đây là mảng có kích thước thay đổi, đó là một trong những ưu điểm của mảng động vì nó giúp ta tiết kiệm tài nguyên hệ thống. Ta có thể sử dụng một mảng có kích thước lớn trong một thời gian nào đó rồi xoá bỏ để trả lại vùng nhớ cho hệ thống.
- Khai báo một mảng động bằng cách cho nó một danh sách không theo chiều nào cả. Cú pháp: **Dim** <Tên mảng> () [**As** <Kiểu>]

Ví dụ: **Dim** DynArray() **As** **Integer**

Sau đó ta có thể cấp phát số phần tử thật sự bằng lệnh **ReDim**.

ReDim <Tên mảng>(N)

trong đó N là biểu thức kiểu integer.

Redim dùng để xác định hay thay đổi kích thước của mảng động.

Mỗi lần gọi Redim tất cả các giá trị chứa trong mảng sẽ mất.

IV. DLL and Tập tin.

DLL học 4 cái DLL chính windows. (58/giáo trình)

a/ Tạo đối tượng FileSystemObject

Có hai cách, khai báo một biến kiểu FileSystemObject hoặc dùng phương thức CreateObject của lập trình hướng đối tượng.

Cách 1:

```
Dim fso As New FileSystemObject
```

Cách 2:

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

Trong đó Scripting là tên thư viện và FileSystemObject là tên đối tượng.

b/ Truy cập ổ đĩa, thư mục, tập tin

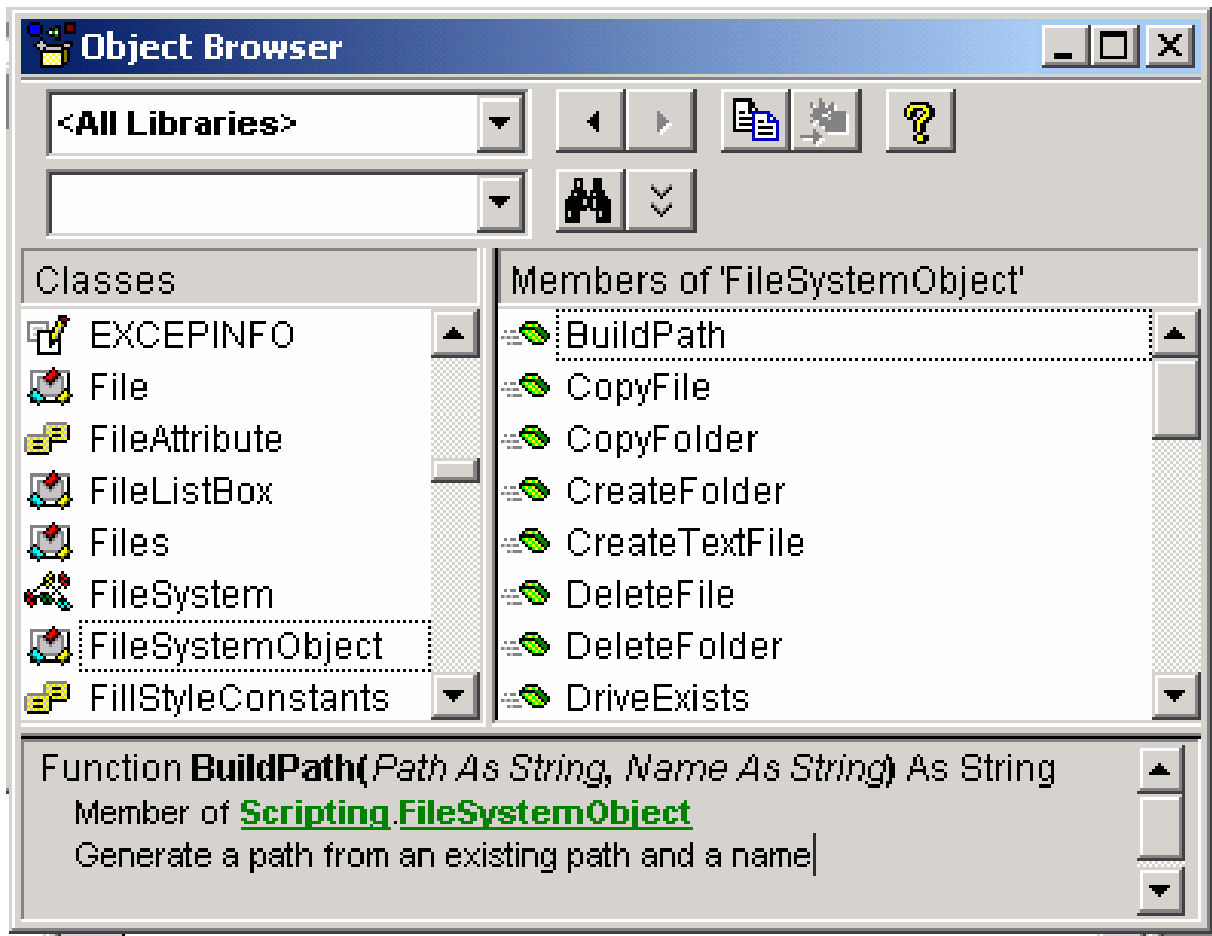
Dùng các phương thức GetDrive, GetFolder, GetFile. Ví dụ để tạo một handle trỏ đến tập tin "d:\tqding\text.txt" ta dùng các dòng lệnh sau:

```
Dim fso As New FileSystemObject, f As File
```

```
Set f = fso.GetFile("d:\tqding\text.txt")
```

Hoặc ta có thể tạo mới thư mục, tập tin thông qua các phương thức CreateFolder, CreateTextFile. Ngoài ra ta có thể xóa một thư mục hoặc một tập tin thông qua DeleteFolder, DeleteFile.

Đối tượng FileSystemObject còn có rất nhiều phương thức, ta có thể xem qua cửa sổ ObjectBrowser.



Hình VII.1 Cửa sổ ObjectBrowser với đối tượng FileSystemObject

c/ **Làm việc với thư mục**

Đây là các phương thức có cách sử dụng rất đơn giản, vì thế ta chỉ xét qua phương thức nào ứng với tác vụ gì (công việc gì) chứ không đi sâu phân tích từng phương thức.

Tác vụ	Phương thức
Tạo thư mục	FileSystemObject.CreateFolder
Xóa thư mục	FileSystemObject.DeleteFolder hay Folder.Delete
Di chuyển thư mục	FileSystemObject.MoveFolder hay Folder.Move
Sao chép thư mục	FileSystemObject.CopyFolder hay Folder.Copy
Lấy tên thư mục	Folder.Name
Kiểm tra thư mục có tồn tại trên ổ đĩa hay không	FileSystemObject.FolderExists
Trả về đối tượng Folder	FileSystemObject.GetFolder
Lấy tên của thư mục cha	FileSystemObject.GetParentFolderName
Lấy tên của thư mục hệ thống	FileSystemObject.GetSpecialFolder

d/ **Làm việc với tập tin**

Mở tập tin để ghi dữ liệu

- o Tạo tập tin mới: sử dụng phương thức CreateTextFile.

```
Dim fso As New FileSystemObject
fso.CreateTextFile("d:\home\lhbao\test.txt")
)
```

- o Mở tập tin để ghi với cờ ForWriting, lúc này ta sử dụng phương thức OpenAsTextStream của đối tượng File cùng với đối tượng TextStream để thao tác.

Ví dụ:

```
Dim fso As New FileSystemObject, f As File
Dim ts As TextStream
fso.CreateTextfile("d:\home\lhbao\test.txt")
Set f =
fso.GetFile("d:\home\lhbao\test.txt") Set ts
= f.OpenAsTextStream(ForWriting)
```

- o Ghi dữ liệu lên tập tin: ta có thể ghi dữ liệu vào tập tin đang mở bằng phương thức Write hay WriteLine của đối tượng TextStream. Sự khác biệt giữa hai phương thức này là sẽ có thêm ký tự xuống dòng nếu như sử dụng WriteLine. Nếu muốn ghi một dòng trắng vào tập tin đang mở, ta sử

dụng phương thức WriteBlankLines.

Cú pháp: ts là đối tượng TextStream

ts.Write(s) ‘ Ghi chuỗi s lên tập tin

ts.WriteLine(s) ‘ Ghi chuỗi s lên tập in

ts.WriteBlankLines(N) ‘ Ghi N dòng trắng lên tập tin

Mở tập tin để đọc dữ liệu

○ Mở tập tin để đọc với cờ ForReading, lúc này ta sử dụng phương thức OpenAsTextStream của đối tượng File cùng với đối tượng TextStream để thao tác.

Ví dụ:

```
Dim fso As New FileSystemObject, f As File
```

```
Dim ts As TextStream
```

```
Set f = fso.GetFile("D:\Home\lhbao\Test.txt")
```

```
Set ts = f.OpenAsTextStream(ForReading)
```

○ Đọc dữ liệu từ tập tin: Ta có ba phương thức để đọc dữ liệu từ một tập tin văn bản, đó là Read, ReadLine và ReadAll. Ba phương thức này cho phép đọc một số ký tự, một dòng của văn bản và toàn bộ văn bản.

Trong khi đọc nội dung của tập tin, ta có thể sử dụng phương thức Skip, SkipLine để nhảy đến phần tử dữ liệu mới.

Cú pháp: ts là đối tượng TextStream

ts.Read(N) As String: Đọc N ký tự từ tập tin

ts.ReadLine As

String ts.ReadAll As

String

Đóng tập tin: Sử dụng phương thức Close của đối tượng TextStream.

Di chuyển, sao chép và xóa tập tin

Sự khác biệt giữa di chuyển và sao chép một tập tin đó là tập tin nguồn có còn tồn tại ở thư mục nguồn hay không. Ứng với một thao tác, ta cũng có hai phương thức để thực hiện, đó là các phương thức thuộc đối tượng FileSystemObject và đối tượng File.

Tác vụ	Phương thức
Di chuyển một tập tin	FileSystemObject.MoveFile hoặc File.Move
Sao chép một tập tin	FileSystemObject.CopyFile hoặc File.Copy
Xóa một tập tin	FileSystemObject.DeleteFile hoặc File.Delete

Các phương thức ứng với thao tác di chuyển và sao chép tập tin cần có đối số là đường dẫn đến nơi chứa tập tin đích.

ABOUT

Hỗ trợ ôn tập là một dự án phi lợi nhuận hướng tới cộng đồng.

Với mục đích đem đến kiến thức miễn phí cho tất cả mọi người, chúng tôi sẽ hỗ trợ các bạn tốt nhất trong lĩnh vực giáo dục bằng cách cung cấp cho các bạn tài liệu ôn tập miễn phí, đề cương ôn tập miễn phí.

Các bạn sẽ không cần phải lo về đề cương, về tài liệu, về sách,... Các bạn chỉ việc theo dõi và để lại yêu cầu cho đội nhóm chúng tôi, còn việc tìm kiếm và biên soạn tài liệu đã có chúng tôi lo!!!!

Hiện giờ, chúng tôi đang hỗ trợ về

1. Tài liệu ôn tập tiếng anh FREE.
2. Tài liệu ôn thi đại học FREE
3. Tài liệu ôn thi cấp 3 FREE
4. Đề cương ôn thi chương trình Đại học FREE.
5. Một số tài liệu khác.

Liên hệ và kết nối với chúng tôi:

- ✓ Facebook: facebook.com/HoTroOnTap
- ✓ Fanpage: facebook.com/HoTroOnTapPage
- ✓ Group: facebook.com/groups/HoTroOnTapGroup
- ✓ Website: hotroontap.com