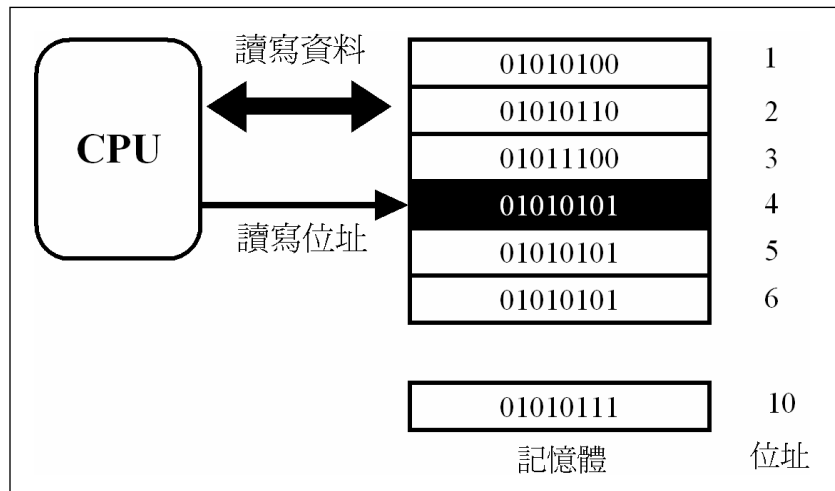


變數與運算子

資科系
林偉川

電腦記憶體的基本



變數的命名與宣告

- 以高階程式語言來說，程式是由資料和指令所組成，如下：
 - 資料部分：指變數和資料型別。
 - 指令部分：主要是指運算子、流程控制和副程序。

3

VB.NET的命名規則

- 在程式碼除了程式語言的關鍵字，大部分都屬於程式設計者自訂的元素名稱，稱為識別字，例如：變數、程序、函數和物件名稱等。

4

VB.NET的命名規則

- VB.NET的命名原則，如下所示：
 - 識別字不可使用VB.NET的**關鍵字**或系統的物件名稱。
 - **名稱**必需是**英文字母**或**底線"_"**開頭，如果以**底線開頭**，至少需要一個其它的英文字母或數字。
 - 名稱長度不可超過**16383個字元**，不區分英文字母的大小寫，**abc**和**ABC**代表同一個名稱。
 - 名稱中間不能有句點"."、**運算子**（例如：**+*/^**等）或空白，只能是英文字母、數字和底線。
 - 在宣告的有效範圍內需唯一，有效範圍請參閱程序和函數。

5

VB.NET的命名範例

- 一些VB.NET元素名稱的範例，如下所示：
 - abc, ab_123, size1, _123, _abc** ' 合法名稱
 - Car, count, s1, s2** ' 合法名稱
- 一些不合法VB.NET元素名稱的範例：
 - _, 123abc** ' 不合法名稱，只有底線或數字開頭

6

VB.NET慣用的命名法

- **匈牙利命名法**是將變數的前三個字元視為資料型別，依照不同的資料型別加上字首字串，如下表所示：

資料類型	字首	範例
Booleam	bln	blnDoit
Char	chr	chrA
Integer	int	intCounter
Object	obj	objCar
String	str	strName

7

VB.NET的慣用的命名法

- **CamelCasing**命名法是**第一個英文字小寫**之後為**大寫**，**變數、程序/函數**的命名也可以使用不同英文字母大小寫的組合，如下表所示：

識別字種類	習慣命名原則	範例
常數	使用英文大寫字母和底線“_”符號	MAX_SIZE、PI
變數	使用英文小寫字母開頭，如果是兩個英文字所組成，第2個之後的英文字以大寫開頭	size、userName
程序/函數	使用英文小寫字母開頭，如果是2個英文字組成，其它英文字使用大寫開頭	pressButton、scrollScreen

8

變數的宣告-屬性

屬性名稱	說明
名稱 (Name)	變數名稱，這是一個標籤，用來在程式中識別出它是一個變數
位址 (Address)	在記憶體中儲存此變數的記憶體位址
尺寸 (Size)	變數所佔用的記憶體尺寸，以位元組為單位
型別 (Type)	變數儲存資料的資料型別
值 (Value)	變數值，也就是在記憶體位址中儲存的資料
壽命 (Lifetime)	在執行程式時，變數存在的期間，有些變數是在整個執行過程中都存在，有些變數是在執行期間自動或由程式碼建立
範圍 (Scope)	在程式碼的那些程式敘述可以存取此變數

9

變數的宣告

- VB.NET的變數是使用**Dim指令**進行宣告，如下所示：
Dim size '變數size的資料型別是Object
- As指令宣告變數的資料型別，如下所示：
Dim size1,size2 As Integer '都是Integer資料型別
- 同一列的兩個變數屬於不同的資料型別，可以分別宣告其資料型別，如下所示：**Dim name As String, size3 As Integer**

10

變數的初值

- 在宣告變數同時指定其初值，如下所示：
`Dim name As String, size3 As Integer = 100 ' 變數初值`
`Dim title As String = "VB.NET程式"`
- 上述程式碼宣告字串變數`name`和`title`，同時指定整數變數`size3`的初值為`100`和字串變數`title`的初值。

11

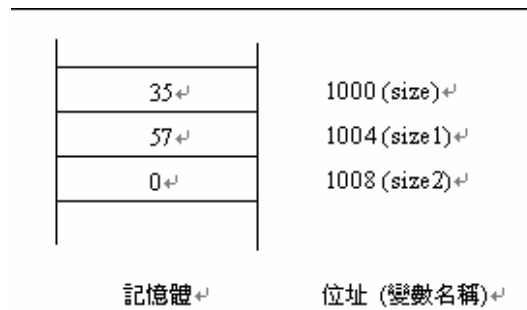
指定敘述的使用

- 在VB.NET宣告好變數後，可以使用指定敘述指定變數值，也就是`=`等號運算子，如下所示：
`Dim size`
`Dim size1, size2 As Integer`
`Dim name, title As String`
`size = 35`
`size1 = 57`
`name = "林偉川"`

12

指定敘述

- 指定敘述"="等號左邊的變數稱為**左值**，指的是變數的位址屬性，等號的右邊稱為「右值」，這是變數的**值屬性**。目前變數的記憶體圖例，如下圖所示：



13

Option指令敘述

- VB.NET的Option指令敘述共有三種，如下所示：
 - Option Explicit**：設定程式的變數是否一定需要宣告，Off表示不用宣告。
 - Option Strict**：指定程式是否使用嚴格的型別檢查，例如：型別轉換和變數的型態宣告，預設值Off表示不嚴格檢查。
 - Option Compare**：指定字串比較的方式是使用**二進位**或**文字比較**，可以是**Binary**或**Text**。

14

常數的宣告與使用

- 常數在程式碼中只是使用一個名稱來取代固定的數字或字串，與其將常數視為變數，不如說它是名稱的轉換，將一些數值的值使用有意義的名稱來取代。
- VB.NET本身擁有內建常數，例如：`vbTab`、`vbNewLine`等。
- 使用者可以使用**Const指令**建立常數，常數在宣告時同時需要指定其值，如下所示：

```
Const MAX_SIZE = 100
```

```
Const PI = 3.1415926
```

15

列舉資料型別Enumeration

- 列舉常數是使用符號名稱代表一組整數資料型別的值。如同常數使用名稱取代固定數值，列舉常數是使用一組不同名稱來取代一組數值，其宣告是使用**Enum/End Enum**程式敘述，如下所示：

```
Enum Color As Integer
```

```
White = 1
```

```
Red
```

```
Blue = 5
```

```
Green
```

```
Black = Green
```

```
End Enum
```

16

VB.NET的資料型別

- VB.NET的**資料型別**是指在變數的記憶體位址儲存的是什麼樣的資料，VB.NET的資料型別可以分為**數值型別**和**參考型別**兩種資料型別，如下所示：
 - **數值型別**：變數值儲存的是真正的數值，也就是記憶體儲存的内容。VB.NET提供**Byte**、**Short**、**Integer**、**Long**、**Single**、**Double**、**Char**和**Boolean**等基本資料型別。
 - **參考型別**：變數值並不是記憶体的内容，而是記憶体的位址，例如：**Module**、**Class**、**String**和**Array**陣列等。

17

VB.NET的基本資料型別

資料型別	說明	位元組	範圍
Boolean	布林值	2	True 或 False
Byte	正整數	1	0~255
Char	字元	2	0~65535
Short	短整數	2	-32,768~32,767
Integer	整數	4	-2,147,483,648~2,147,483,647
Long	長整數	8	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807

18

VB.NET的基本資料型別

Single	單精度的 浮點數	4	負值範圍為-3.4028235E+38~-1.401298E-45， 正值的範圍為 1.401298E-45~3.4028235E+38
Double	雙精度的 浮點數	8	負值範圍為-1.79769313486231570E+308 ~- 4.94065645841246544E-324， 正值範圍為 4.94065645841246544E-324 ~ 1.79769313486231570E+308
Decimal	數值	16	0~+/-79,228,162,514,264,337,593,543,950,335 沒有小數，0~+/- 7.9228162514264337593543950335 帶 28 位小數， 最小的非零值為 +/-0.000000000000000000000001(+/-1E-28)

19

VB.NET的基本資料型別

Date	日期	8	0001 年 1 月 1 日~9999 年 12 月 31 日
String	字串	依平台	0~2 百萬 Unicode 字元
Object	物件	4	物件型態變數可以儲存各種資料型別的值

20

整數資料型別

- **整數資料型別**是指變數的資料為整數沒有小數點，依照整數資料長度的不同(即佔用的**記憶體位元數**)，分為**Byte**、**Short**、**Integer**和**Long**四種整數資料型別，其中**Byte**是正整數，其它可以是正整數或負整數，程式設計者可以依照整數的範圍決定宣告的變數型別。

21

整數資料範例

整數值	十進位值	說明
19	19	十進位整數
&O377	255	八進位整數
&Hff	255	十六進位整數
&H3E7	999	十六進位整數

22

浮點數資料型別

- **浮點數資料型別**是指整數加上小數，例如：3.14、100.567等，依照長度的不同(即佔用的記憶體位元數)，分為**Single**、**Double**和**Decimal**三種浮點數的資料型別。

23

浮點數資料範例

浮點數值	十進位值	說明
0.0005	0.0005	浮點數
.0005	0.0005	浮點數
5E-4	0.0005	使用 e 指數的浮點數

24

布林資料型別

- **布林資料型別**只能有兩個值true和false，對應「真」或「偽」的狀態。

25

字串資料型別

- **字串資料型別**的資料是字串，字串是0或多個依序的Unicode字元使用ASCII字碼的雙引號所括起的文字內容，這就是「字串值」，如下所示：

```
Dim str1 As String = "VB.NET程式設計"
```

```
Dim str2 As String = "Hello World!"
```

```
Dim str3 As String = "\\localhost\download\joe"
```

26

字元資料型別

- 字元資料型別是單一的Unicode字元，如果是數值，可以儲存無符號16位元整數。
- 如果在VB.NET程式直接使用字元值，就是使用雙引號括起的單一字元字串，如下所示：

`Dim a As Char = "A" C` 'C表示這是字元，不是字串

27

日期/時間資料型別

- 「日期/時間資料型別」是用來儲存日期/時間資料，「日期/時間值」的日期時間資料需要使用“#”括起，如下所示：

`Dim tday As Date = #9/22/2002 12:00:00 AM#`

28

VB.NET的運算式與運算子

- VB.NET指定敘述的等號右邊除了是文字值、變數外，也可以是**運算式**，「運算式」是由**運算子和運算元**組成。
- VB.NET擁有完整的算術、關係、連結和邏輯運算子。

29

運算式的基礎

- 運算式是由運算元和運算子所組成，在指定敘述中的運算式一定在等號的右邊，如下所示：

$C = A + B - 1$

$C = (A \geq B)$

$C = A \& B$

$C = (A < B \text{ And } A > 1)$

- 上述運算式的變數A、B和數值1是運算元，+、-、<、>=和>為運算子，變數C是左值，也就是運算式將計算結果存入變數C。

30

運算子的優先順序

- 優先順序的規則，如下所示：
 - "("括號內比括號外先執行，括號的目的是為了推翻現有的優先順序，在括號內依照正常的優先順序。
 - 當運算式擁有多種運算子時，算術運算子最先，接著是關係運算子，最後才是邏輯運算子，如下所示：
算術運算子 > 關係運算子 > 邏輯運算子
 - 關係運算子擁有相同的優先順序。
 - 算術運算子的加和減法優先順序相同，乘和除法擁有相同的優先順序，不過乘除高於加減，如下所示：
^指數 > (-)負號 > */乘除 > \整數除法 > Mod > +-加減 > &
 - 邏輯運算子的優先順序，如下所示：
Not > And, AndAlso > Or, OrElse > Xor

31

算術運算子

運算子	說明	運算式範例
^	指數	$5 \wedge 2 = 25$
-	負號	-7
+	正號	+7
*	乘法	$5 * 6 = 30$
/	除法	$7 / 2 = 3.5$
\	整數除法	$7 \setminus 2 = 3$
Mod	餘數	$7 \text{ Mod } 2 = 1$
+	加法	$4 + 3 = 7$
-	減法	$4 - 3 = 1$

32

字串連結與Like運算子

運算子	說明	運算式範例
&	字串連結	"abc" & "de" = "abcde"
Like	字串比較	使用範本的高用字元進行字串的比較，如果符合傳回布林值 True，否則為 False

33

指定運算子

運算子	範例	相當的運算式	說明
=	x = y	N/A	指定敘述
^=	x ^= y	x = x ^ y	指數
+=	x += y	x = x + y	數字相加
-=	x -= y	x = x - y	減法
*=	x *= y	x = x * y	乘法
/=	x /= y	x = x / y	除法
\=	x \= y	x = x \ y	整數除法
&=	x &= y	x = x & y	字串連結

34

資料型別的轉換

- **資料型別轉換**在程式中會經常的出現，例如：將**整數轉換成浮點數**，**字串轉換成整數**進行運算，或將浮點數值轉換成字串以方便顯示。
- 資料型別轉換是指轉換變數儲存的資料，而不是變數本身，因為不同型別佔用的位元組數不同，在進行資料型別轉換時，例如：**Double轉換成Single**，變數資料會損失一些精確度。

35

隱含型別轉換

- 「**隱含型別轉換**」並不需要特別語法，在**運算式**或**指定敘述的兩端**，如果擁有**不同型別的變數**，將會將儲存的資料自動轉換成相同的資料型別，一些運算式的範例，如下表所示：

運算式	結果	說明
"250" + 125	375	算術運算子加號的字串會轉換成數值
"250" & 125	"250125"	字串連結運算子的數值會轉換成字串
"250" + "125"	"250125"	因為運算元都是字串，所以加號變成字串連結運算子

36

明顯型別轉換

- 「明顯型別轉換」是在進行運算前，自行使用函數將它轉換成相同的資料型別，如下表所示：

轉換函數	說明
CInt(Value)	將參數的值轉換成整數
Cdbl(Value)	將參數的值轉換成浮點數 Double，在第 4 章的範例程式會使用此轉換函數
CStr(Value)	將參數的值轉換成字串