

迴圈結構

資訊科技系
林偉川

計數迴圈

- VB.NET的計數迴圈可以自行維護計數器變數，只需指定範圍和增量，就可以重複執行程式區塊的程式碼。

For To/Next迴圈

- For To/Next迴圈可以執行固定次數的迴圈，預設每次增加1，如果到達結束條件，就結束迴圈的執行，其語法如下所示：

For 變數 = 起始值 To 終止值 step 遞增值

程式區塊

Next 變數

- 上述迴圈的「變數」是計數器變數，其範圍是從「起始值」到「終止值」，每次增加1，執行迴圈的次數是：「終止值」-「起始值」+1次，其中Next指令後的「變數」可以省略。

3

For To/Next迴圈

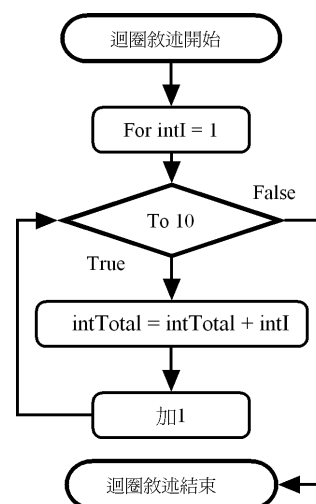
- 計算1加到10總和的迴圈，如下所示：

Dim intI, intTotal As Integer

For intI = 1 To 10 step 1

intTotal = intTotal + intI

Next intI



For Step/Next迴圈

- 如果For/Next迴圈的增量不是1，可以使用Step指令指定每次增加或減少的量，如果使用負數值，表示每次是將計數器變數減掉Step值，此時的For To Step/Next迴圈，如下所示：

```
For intI = 100 To 1 Step -1  
    intTotal = intTotal + intI  
Next intI
```

5

條件迴圈種類

- 條件測試的位置分成兩種，如下所示：
 - **前測式重複結構**：在迴圈開始使用While或Until指令測試迴圈條件，其語法如下所示：

```
Do [While | Until] 條件  
    程式區塊
```

Loop

- **後測式重複結構**：在迴圈結尾使用While或Until指令測試迴圈條件，因為是在結尾測試條件，此迴圈**至少會執行一次**，其語法如下所示：

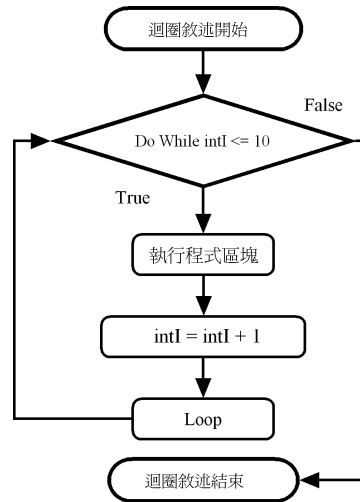
```
Do  
    程式區塊  
Loop [While | Until] 條件
```

6

條件迴圈-Do While/Loop

- 當While指令的測試條件成立才能進入迴圈，如下所示：

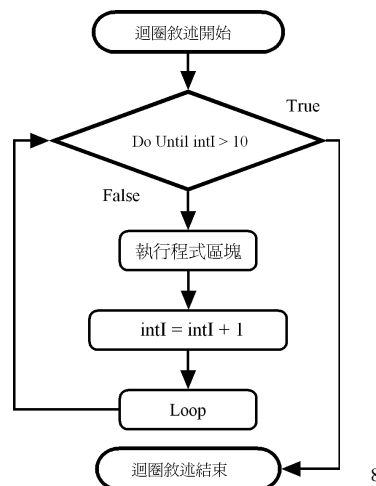
```
intI = 1
intTotal = 0
Do While intI <= 10
    intTotal = intTotal + intI
    intI = intI + 1
Loop
```



條件迴圈-Do Until/Loop

- Until條件在迴圈開始檢查，迴圈是執行到條件成立為止，如果條件不成立就進入迴圈，如下所示：

```
intI = 1
intTotal = 0
Do Until intI > 10
    intTotal = intTotal + intI
    intI = intI + 1
Loop
```

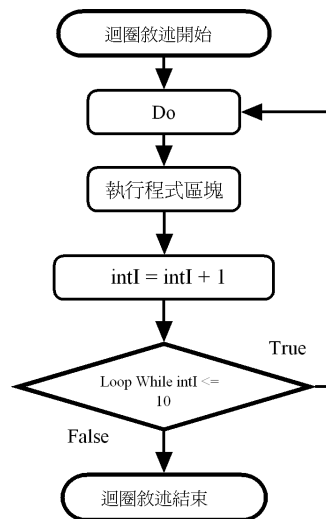


8

條件迴圈-Do/Loop While

- Do/Loop迴圈可以使用 While條件在迴圈結尾檢查，這個Do/Loop迴圈至少會執行一次，如下所示：

```
intI = 1
intTotal = 0
Do
    intTotal=intTotal + intI
    intI = intI + 1
Loop While intI <=10
```

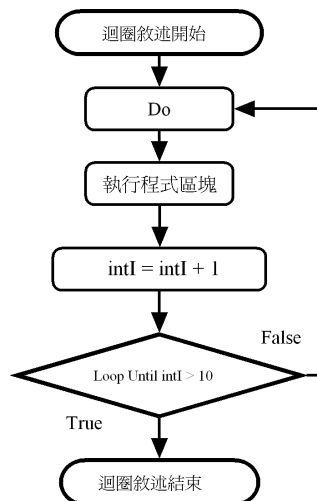


9

條件迴圈-Do/Loop Until

- Do/Loop迴圈如果使用 Until條件在迴圈結尾檢查，這個迴圈也一樣至少執行一次，如下所示：

```
intI = 1
intTotal = 0
Do
    intTotal = intTotal + intI
    intI = intI + 1
Loop Until intI > 10
```



10

巢狀迴圈

- 巢狀迴圈是在迴圈內擁有其它的迴圈結構。
- 例如：在For/Next迴圈擁有For/Next和Do/Loop迴圈，同樣的，Do/Loop迴圈內也可以擁有For/Next和Do/Loop迴圈。

11

For/Next巢狀迴圈

- 最簡單的巢狀迴路是在For/Next迴圈中擁有其它的For/Next迴圈（因為不用自行處理計數器變數），一個二層的巢狀迴圈結構的範例，如下所示：

```
' 第一層
For intI = 1 To 10
  ' 第二層
  For intJ = 1 To intI
    output &= "*"
  Next intJ
  output &= vbNewLine
Next intI
```

12

混合的巢狀迴圈

- VB.NET的巢狀迴圈可以混合For/Next和Do/Loop Until迴圈結構，例如：在For/Next迴圈內擁有Do/Loop Until迴圈，如下所示：

```
' 第一層
For intI = 1 To 9
  ' 第二層
  intJ = 1
  Do
    .....
    intJ += 1
  Loop Until intJ > 9
Next intI
```

13

While/End While迴圈

- While/End While就是舊版VB 6的While/Wend迴圈，這是一種在迴圈開頭檢查的迴圈，例如：從1加到10的程式碼，如下所示：

```
Dim total As Integer = 0
Dim counter As Integer = 1
While counter <= 10
  total += counter
  counter += 1
End While
```

14

跳出迴圈

- 不論使用For/Next或Do/Loop迴圈，程式都會依條件**重複執行**一定的次數，如果因程式所需要**中斷迴圈**的執行，即跳出迴圈。
- For/Next使用**Exit For**指令
- Do/Loop使用**Exit Do**指令

15

Exit For指令

- 如果在For/Next迴圈尚來到達結束條件時，可以使用Exit For指令強迫跳出For/Next迴圈，如此馬上就會結束迴圈的執行，如下所示：

```
For intI = 1 To 100 Step 1
    ...
    If amount <= 0 Then
        Exit For ' 離開For迴圈
    End If
    ...
Next
```

16

Exit Do指令

- Do/Loop迴圈需要自己處理迴圈的結束條件且確認迴圈能夠結束，如果沒有使用While或Until指令在迴圈頭尾測試條件，單純Do/Loop迴圈是一個無窮迴圈，如果不指定終止條件，可以使用Exit Do指令結束迴圈，如下所示：

```
Do
    ....
    If str = "" Then
        Exit Do
    End If
    ....
Loop
```

17

VB.NET的錯誤處理

- 錯誤處理是指在程式執行時，如果發生不正常的執行狀態時，可以在程式碼中處理這些錯誤，錯誤處理的目的是為了讓程式能夠更加的強壯(Robust)，如此這算程式遇到不尋常的情況，也不會造成程式崩潰(Crashing)，進而導致整個系統當機的情況。

18

VB.NET的錯誤處理

- VB.NET提供更結構化的錯誤處理敘述，即 Try/End Try，其語法如下所示：

```
Try
    ' 測試的錯誤程式碼
    程式區塊
Catch e As Exception
    ' 錯誤處理的程式碼
    程式區塊
Finally
    程式區塊
End Try
```

19

執行結果

- For i=1 to 10 step 2 : total=total+I : next , total=?
- Dim n as integer=1
do while n<=64
 n=2*n : txt.text &= n &vbnewline
loop , txt.text=?
- Dim total as integer=0, I as integer
for i=1 to 10
if (I mod 2) <> 0 then
 total+=I : txt.text&=I & vbnewline
else
 total -= 1
end if
txt.text &= total & vbnewline , txt.text=?

20

作業

- 以兩個文字方塊，結合for/next來顯示4的倍數，且有多行及捲軸
- 信用卡利率18%，請以文字方塊、標籤及按鈕，計算借1萬元，從第2個月開始以每月固定金額方式還款，顯示共需幾個月才能還完
- 以一個文字方塊結合for/next來顯示從1到該數間的質數