

單元 14

VB 程式進階

單元目標

- ☑ 瞭解 IF...Then...Else 的意義與用法。
- ☑ 瞭解 Select...Case 的意義與用法。
- ☑ 瞭解 For...Next 的意義與用法。
- ☑ 瞭解 Do...Loop 的意義與用法。

14-1 VB 中的選擇結構

如果明天不下雨，我就要出去玩；凡消費滿 1000 元就可享 9 折優惠，2000 元以上 8 折，3000 元以上 7 折。日常生活中充滿這類選擇性的問題，我們必須根據現實條件選擇一個最適合的途徑。

撰寫程式時也是一樣，如果程式只能由上而下循序執行而不具備「選擇」的能力，將會使程式的功用受到限制。因此 VB 提供了「If...Then...Else」以及「Select...Case」二種選擇結構，藉此可以執行各種判斷並執行相關處理，使程式撰寫更加靈活有彈性。

選擇結構

測試某一條件並按其結果來改變執行路徑的結構，舉例來說，當我們寫一個計算購書總價的程式時，可先判斷數量是否達到折扣標準，若達到某一標準，則對應某一折扣價而得總價為（定價 × 數量 × 折扣）。

14-1.1 具判斷力的 If...Then...Else

If...Then...Else 結構顧名思義就是「如果...就...否則...」的意思。例如「If 明天是個好天氣 Then 去陽明山賞楓 Else 在家看電視」或是「If 分數 \geq 60 Then Print "成績及格" Else Print "成績不及格"」等，都是 If...Then...Else 結構的例子。

指令語法與說明

**If...Then...Else**

格式 If 條件運算式 Then

敘述 1

敘述 2

...

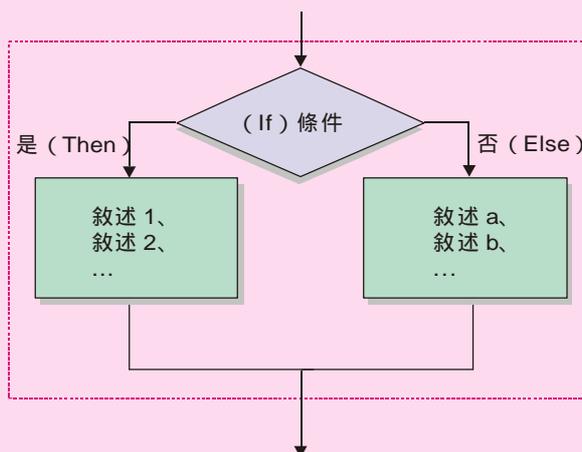
Else

敘述 a

敘述 b

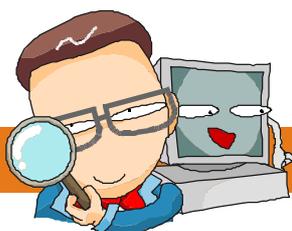
...

End If



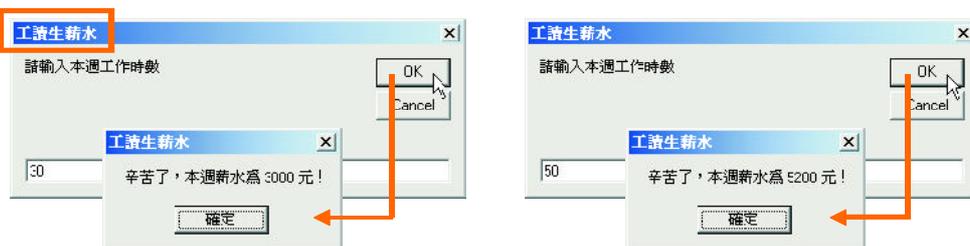
功能 利用關係或邏輯運算式進行判斷，有條件地指示程式去執行特定的敘述。

說明 當電腦執行此敘述時，會先測試 If 後面的運算式值，如果其值為真，便執行 Then 之後的敘述；如果其值為假，則執行 Else 之後的敘述，也就是「如果...就...否則...」的意思。例如「IF 分數 \geq 60 Then Print "成績及格" Else Print "成績不及格"」。



》》 計算工讀生薪水 (IF...Then...Else)

請撰寫一個「計算工讀生薪水」的程式：當工作時數小於等於40小時，每小時的薪水為100元；若超過40小時，則超過的部分每小時再加發20元加班費，如下圖。【範例檔案：工讀生薪水.frm】



```

Project1 - Form1 (程式碼)
Form Load
Private Sub Form_Load()
    Dim Hours, Money As Integer
    Hours = InputBox("請輸入本週工作時數")
    If Hours <= 40 Then
        Money = 100 * Hours
    Else
        Money = 100 * Hours + 20 * (Hours - 40)
    End If
    MsgBox "辛苦了，本週薪水為 " & Money & " 元！"
End Sub
    
```

為什麼我的標題列是「Project1」而不是「工讀生薪水」？

當我們使用Msgbox或InputBox指令時，若沒有特別以參數指定標題列文字，則系統預設會以專案檔名作為標題列文字，因此若同學是開啟範例檔案「工讀生薪水.vbp」進行練習，就會看到如課本上的畫面，而若是自行開新檔案並輸入程式碼，在未存檔前便可能看到類似「Project1」、「Project2」之類的標題列文字。

當然，我們也可以自行指定合適的標題列文字（相關語法請參閱13-3節），但為避免過多文字模糊程式碼畫面中的重點，因此本書在說明時皆統一省略，而標題列則會顯示該範例檔檔名。

14-1.2 巢狀 If...Then...Else

有時候我們的設定條件可能不只一個，例如「如果有很多人就叫 Pizza 來吃，如果 3 到 4 個就點炸雞全家餐，如果只有一個人就叫便當」，這時就必須要使用**巢狀 If...Then...Else** 敘述才能解決。

指令語法與說明



巢狀 If...Then...Else

格式

```

If 條件運算式 1 Then
    敘述 1
    敘述 2 ...
Else
    If 條件運算式 2 Then
        敘述 a
        敘述 b ...
    Else
        敘述 x
        敘述 y ...
    End If
End If
  
```

符合條件1時。

不符合條件1但符合條件2時。

不符合條件1也不符合條件2時。

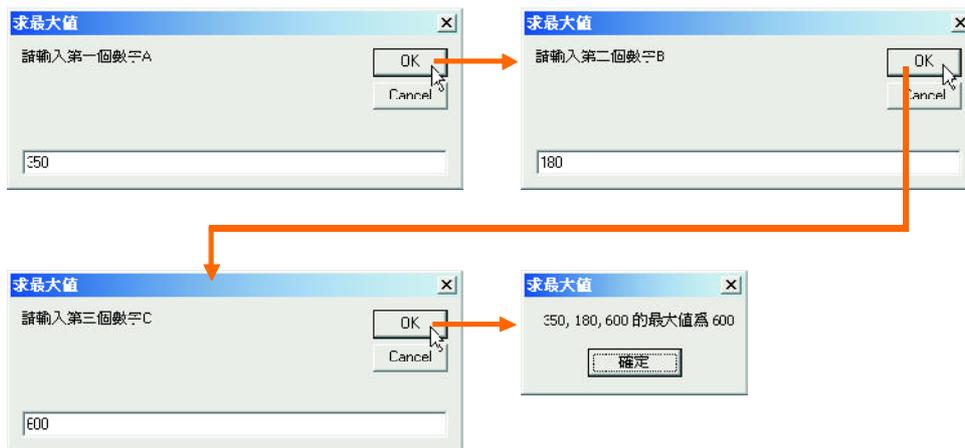
功能 利用巢狀的關係或邏輯運算式，有條件地指示程式依照不同的情況去執行特定的敘述。

說明 當符合條件1時便執行敘述1、2...，若不符則進行第二個IF判斷，若符合條件2則執行敘述a、b...，而若不符合條件1也不符合條件2則執行敘述x、y...。理論上巢狀If...Then...Else敘述的的層次數目是沒有限制的，也就是在敘述1、2...，敘述a、b...，或敘述x、y...中還可以再加入巢狀If...Then...Else的敘述，但太多層次會造成閱讀及維護上的困難。



》 求最大值 (巢狀 IF...Then...Else)

請撰寫一個「求最大值」的程式：使用者透過輸入方塊輸入三個數字A、B、C，程式處理後再以訊息方塊顯示三者的最大值，如下圖。【範例檔案：求最大值.frm】



```

Project1 - Form1 (程式碼)
Form Load
Private Sub Form_Load()
    Dim A, B, C, Max As Integer '宣告整數變數
    A = InputBox("請輸入第一個數字A") '輸入 A
    B = InputBox("請輸入第二個數字B") '輸入 B
    C = InputBox("請輸入第三個數字C") '輸入 C

    If A > B Then '若 A > B
        If A > C Then '且若 A > C
            Max = A 'A 為最大值
        Else '否則
            Max = C 'C 為最大值
        End If
    Else '否則 (即 A < B)
        If B > C Then '且若 B > C
            Max = B 'B 為最大值
        Else '否則
            Max = C 'C 為最大值
        End If
    End If

    MsgBox A & ", " & B & ", " & C & " 的最大值為 " & Max '顯示最大值
End Sub
    
```

14-1.3 多重選擇 Select...Case

假設某一運動比賽需以年齡區分為 10 級，雖然我們可以使用巢狀的 If... Then...Else 敘述來完成，但這會使程式變得十分複雜混亂，而且執行上也比較沒有效率。為此 Visual Basic 提供了 **Select...Case** 敘述，可作為這類多重選擇的問題的解決方案。

指令語法與說明

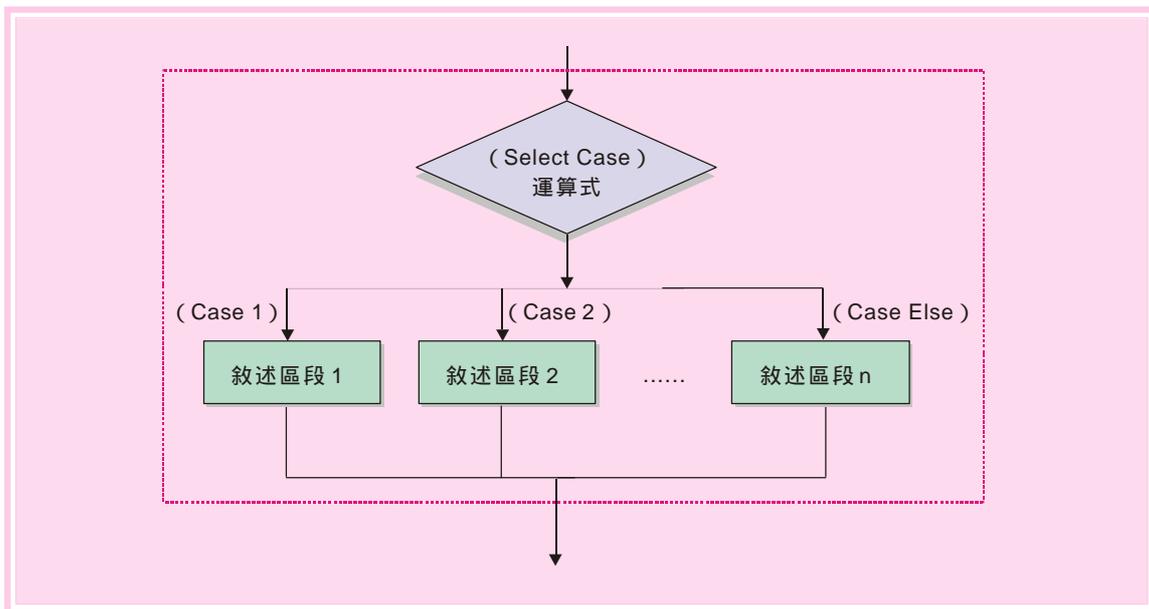
**Select...Case**

格式 Select Case 運算式
 Case 運算式值串列 1
 敘述區段 1
 Case 運算式值串列 2
 敘述區段 2
 :
 Case Else
 敘述區段 n
End Select

功能 利用關係或邏輯運算式進行判斷，有條件地指示程式去執行特定的敘述。

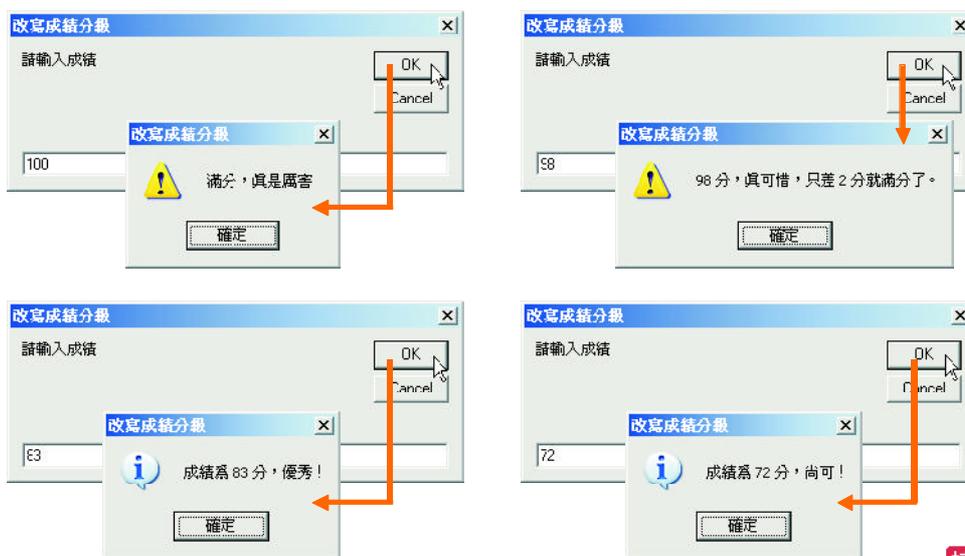
- 說明**
1. 求出 Select Case 運算式之值，並逐一與 Case 運算式值串列比對，若符合則執行該 Case 之後的敘述區段。
 2. 如果所有的 Case 運算式值串列都不符合，則執行 Case Else 的敘述區段。
 3. 如無符合條件的 Case 且無 Case Else 的敘述，則直接執行 End Select 的下一列敘述。
 4. Select Case 後的運算式可為數值變數、字串變數或是算術運算式。例如：Case 1 或 Case "a"、Case 1 to 5 或 Case "a" to "k"、Case is > 5 或 Case is > "a"...等。

接下頁

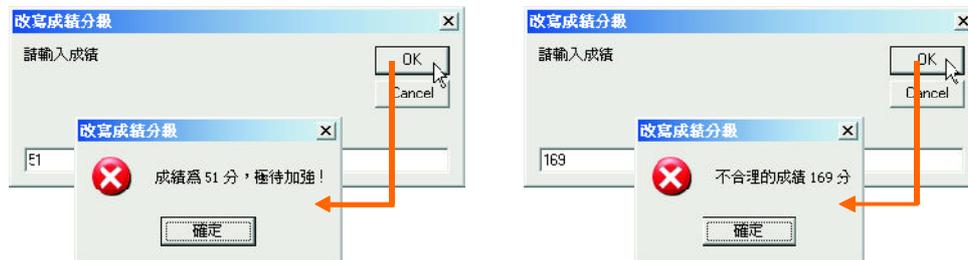


》改寫成績分級 (Select...Case)

請利用Select...Case改寫「成績分級」的程式，使成績區分為更多等級(100、99、80 96、60 79、0 59、其他)，並分別顯示不同的回應訊息如下圖。【範例檔案：改寫成績分級.frm】



接下頁



```

Project1 - Form1 (程式碼)
Form Load
Private Sub Form_Load()
    Dim Grade As Integer
    Grade = InputBox("請輸入成績")
    '宣告整數變數 Grade
    '輸入成績並存入變數 Grade

    Select Case Grade
    '根據 Grade 的值
    Case 100
    '若是 100
    MsgBox "滿分，真是厲害！", vbExclamation
    '顯示訊息
    Case 97, 98, 99
    '若介於 97~99
    MsgBox Grade & " 分，真可惜，只差 " & 100 - Grade & "
    " 分就滿分了。", vbExclamation
    '顯示訊息 (上一行的 _ 代表延續下一行)
    Case 80 To 96
    '若介於 80~96
    MsgBox "成績為 " & Grade & " 分，優秀!", vbInformation
    '顯示訊息
    Case 60 To 79
    '若介於 60~79
    MsgBox "成績為 " & Grade & " 分，尚可!", vbInformation
    '顯示訊息
    Case 0 To 59
    '若介於 0~59
    MsgBox "成績為 " & Grade & " 分，極待加強!", vbCritical
    '顯示訊息
    Case Else
    '除了以上選項
    MsgBox "不合理的成績 " & Grade & " 分", vbCritical
    '顯示訊息
    End Select

    End
    '結束程式
End Sub

```

14-1.4 選擇結構的應用範例

利用上面介紹的 If... Then...Else 以及 Select... Case 的觀念，我們要來撰寫一個「登入畫面」的程式，請參考以下範例。



》登入畫面(選擇結構的應用)

請練習撰寫一個「登入畫面」的程式：當使用者輸入正確的ID與密碼時便顯示問候語，如果錯誤時會要詢問使用者是否重新輸入，答「是」則可重新輸入，答「否」則結束程式。

【範例檔案：登入畫面進階.frm】

輸入的密碼會顯示為 * 號。

登入

登入成功

May, 愈來愈漂亮了!

確定

錯誤

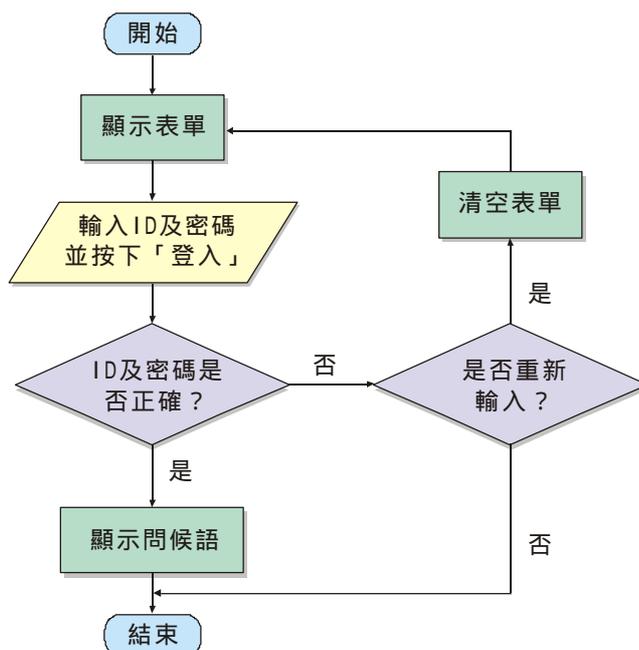
不合法的身份或密碼錯誤，是否要重新輸入?

是(Y) 否(N)

輸入錯誤的使用者ID或密碼時。

結束程式。

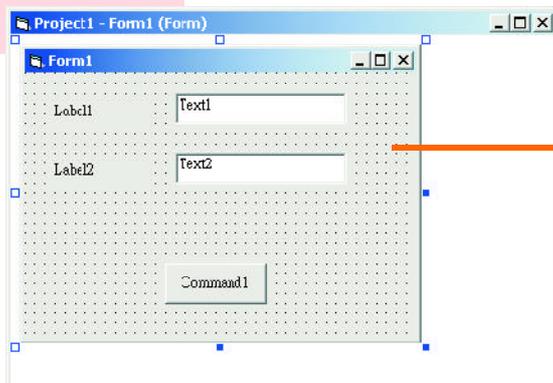
【流程圖】



接下頁

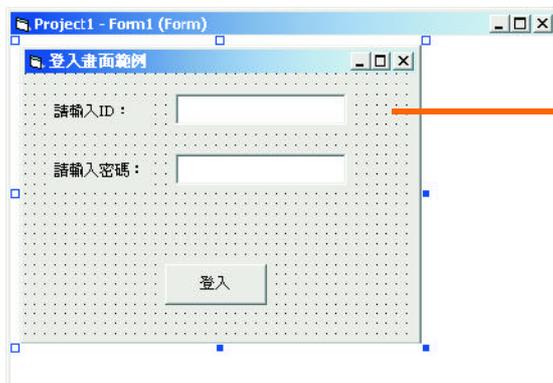
216 第五篇

基本視窗程式應用



1 建立一個新專案，並利用工具箱上的各種控制項建立如圖的表單，然後分別設定各控制項屬性如下表。

物件名稱	屬性	設定值	備註
Form1	Caption	登入畫面範例	表單標題列
Label1	Caption	請輸入ID：	標籤文字
Label2	Caption	請輸入密碼：	標籤文字
Text1	MaxLength	10	限制最大輸入長度
	Text	(空字串)	清空文字盒
Text2	MaxLength	10	限制最大輸入長度
	PasswordChar	*	以 * 取代輸入的密碼
	Text	(空字串)	清空文字盒
Command1	Caption	登入	按鈕文字



2 全部完成後，表單應該會像左圖一樣。

3 雙按「登入」按鈕開啟【程式碼】視窗並輸入程式碼。



接下頁

```

Select Case ID                                     '根據輸入的 ID
Case "Kevin"                                       'ID 為「Kevin」時
    If Password = "1234" Then                       '若輸入密碼為「1234」
        MsgBox "Kevin, 又見到你真好!", vbInformation, "登入成功" '顯示訊息
    End If
Case "May"                                         'ID 為「May」時
    If Password = "13579" Then                     '若輸入密碼為「13579」
        MsgBox "May, 愈來愈漂亮了!", vbInformation, "登入成功" '顯示訊息
    End If
Case Else                                          '輸入其他 ID 時
    Click = MsgBox("不合法的身份或密碼錯誤，是否要重新輸入?", _ '顯示錯誤訊息並傳回回應值
        vbExclamation + vbYesNo, "錯誤")
    If Click = vbYes Then                           '若回應值為「是」
        Text1.Text = ""                             '將 Text1 清空
        Text2.Text = ""                             '將 Text2 清空
        Form1.Show                                  '重新呼叫並顯示表單
    Else                                           '否則 (即回應值為「否」)
        End                                         '直接結束程式
    End If
End Select
End
End Sub

```

在這個範例中，我們用了幾個控制項，包括用以顯示標籤文字的「Label」以及用來輸入文字的「TextBox」。另外，我們也針對對話方塊的問題按鈕做了不同回應，這些基本的範例類型都可以作為往後撰寫程式的參考。在這個範例中幾個比較需要說明的地方是：

1. Label 的 **Caption** 屬性 (通常以 Label.Caption 代表) 表示要顯示在 Label 所在位置的文字。
2. **Text.MaxLength** 表示文字方塊中所能輸入字元的最大長度，在這個範例中設定的最大長度是 10。
3. **Text.PasswordChar** 表示輸入文字方塊中的字元將以指定的密碼字元代替，在這個範例中是符號「*」。
4. 程式碼

```
ID = Text1.Text           ' 將 Text1 輸入值存入變數 ID
Password = Text2.Text     ' 將 Text1 輸入值存入變數 Password
```

的目的是以較有意義的變數名稱來存放輸入值，避免在撰寫程式時產生混淆，此外也使程式較易維護，因為一旦要更改 ID 或密碼的輸入來源時，只要更動此二行程式碼即可。

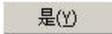
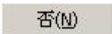
5. 程式碼

```
Case "Kevin"              ' ID 為「Kevin」時
  If Password = "1234" Then ' 若輸入密碼為「1234」
    MsgBox "Kevin, 又見到你真好！", vbInformation, "登入成功"
  End If
```

先以 Select...Case 結構判斷 ID 輸入值是否屬於內建的 ID (Kevin 或 May)，如果是的話，再以 If...Then 敘述判斷密碼是否正確，正確的話就以對話方塊顯示問候語並跳出 Select...Case 結構。

6. 如果在 Select...Case 中找不到符合的 ID 時會執行下列敘述

```
Click = MsgBox(" 不合法的身份或密碼錯誤，是否要重新輸入？", _
vbExclamation + vbYesNo, " 錯誤 ")           ' 顯示錯誤訊息並傳回回應值
```

這會顯示一個對話方塊。當使用者按下按鈕  是(Y) 時會傳回值 vbYes 至變數 Click 中，而按下按鈕  否(N) 則會傳回值 vbNo (請參閱 13-3.1 節)，藉此我們可以進行下列處理

```

If Click = vbYes Then           ' 若回應值為「是」
    Text1.Text = ""            ' 將 Text1 清空
    Text2.Text = ""            ' 將 Text2 清空
    Form1.Show                  ' 重新呼叫並顯示表單
Else                             ' 否則（即回應值為「否」）
    End                          ' 直接結束程式
End If

```

其中 `Text1.Text = ""` 以及 `Text2.Text = ""` 的目的在於清除前一次輸入的資料，方便使用者直接輸入新資料。而 `Form1.Show` 的功用則是再次顯示表單 `Form1` 以便使用者能夠再次進行輸入 ID 與密碼的動作，其中 `Show` 是 `Form` 的一種「方法」，功能是顯示表單。

14-2 VB 中的重複結構

我們在撰寫程式時，經常需要一再使用程式中的某些片段，而**重複結構**便是因應此種需求而產生的結構。重複結構有時又稱**迴圈**，藉此程式設計師不需要一直撰寫重複的程式碼，就可以重複執行某段程式，而且可以很方便地設定起始或停止的條件，這使得程式更加簡潔有條理，並有效地控制程式執行的流程。

VB 中主要提供 `For...Next` 以及 `Do Loop` 二種重複結構，以下亦將簡單介紹其觀念與應用方式。

重複結構

測試某一條件是否成立以決定是否重複執行某一段程式，以「計算購書總價」的例子來說，由於書店顧客不只一位，因此程式將會不斷重複執行直到某一個特定的條件（如打烊）成立為止。

14-2.1 計數式迴圈 For...Next

當程式中某個敘述區段要重覆執行的次數是固定且可計數時（例如重複執行某程式片段 100 次），這類已知起始值，每執行一次增減多少，一直到超過或等於終止值才停止的問題，都可以使用 **For...Next** 敘述。

指令語法與說明

**For...Next**

格式 For 控制變數 = 初始值 To 終止值 [Step 增加值]

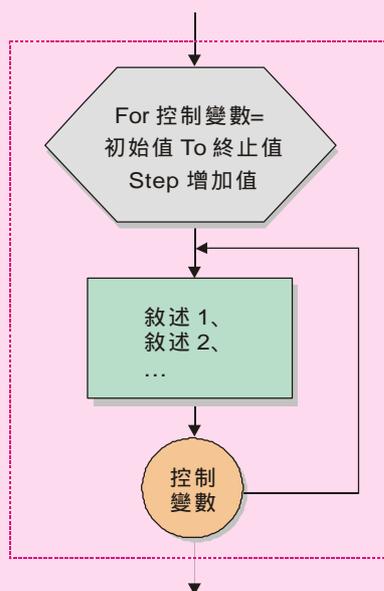
敘述 1

敘述 2

Next [控制變數]

功能 依據使用者所設定的次數，重複執行迴圈內的敘述區段。

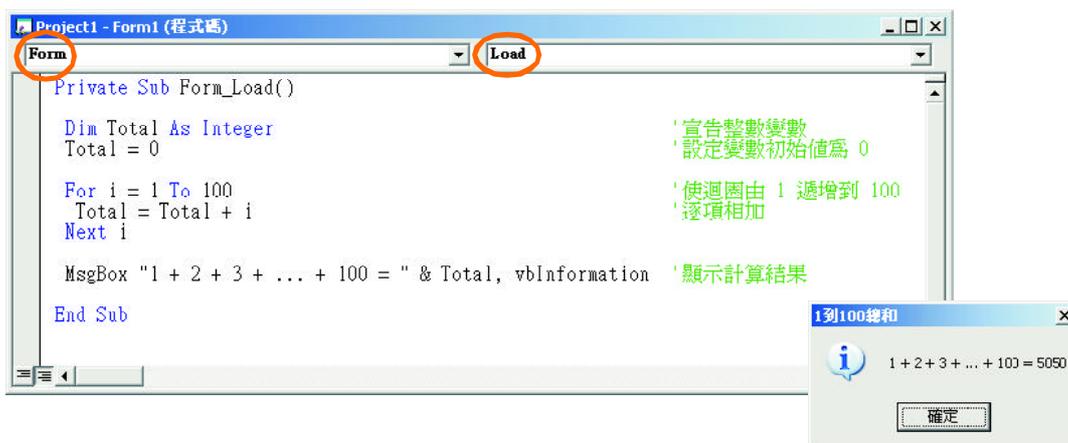
- 說明**
1. 控制變數必須是數值變數。
 2. 初始值、終止值與增加值可以為數值常數、數值變數或數值運算式。若增加值為 1，則 Step 可以省略。
 3. 如果初始值小於終止值，則增加值應為正數，反之，若初始值大於終止值，則增加值應為負數，如此才會執行迴圈內的敘述。
 4. 迴圈的執行次數 = (終止值 - 初始值) \ 增加值 + 1，如果執行次數小於 0，則迴圈內的敘述不會被執行。





》》 計算 1 到 100 的總和 (For...Next)

請利用 For...Next 敘述撰寫一個「計算 1 到 100 總和」的程式，如下圖。【範例檔案：1 到 100 總和.frm】



在這個程式中，我們首先宣告一個整數變數 total 用以存放加總後的結果，並藉由 total = 0 設定其初始值為 0，然後

1. 程式碼

```
For i = 1 To 100
```

省略 STEP 敘述表示增加值為 1，即控制變數 i 每次加 1 直到 i=100，所以這個迴圈會執行 100 次。

2. 由於變數 i 每次加 1，所以 Total=Total+i 相當於

```
Total = Total + 1
Total = Total + 2 .....
Total = Total + 100
```

222 第五篇

基本視窗程式應用

所以便能達到我們想要計算 $1 + 2 + 3 + \dots + 100$ 之目的。



》》 計算 1 到 100 的奇數和 (For...Next 增加值為 2)

請利用 For...Next 敘述撰寫一個「計算 1 到 100 奇數和」的程式，如下圖。【範例檔案：1 到 100 奇數和.frm】

```
Project1 - Form1 (程式碼)
Form
Load
Private Sub Form_Load()
    Dim Total As Integer           '宣告整數變數
    Total = 0                     '設定變數初始值為 0
    For i = 1 To 100 Step 2       '使迴圈由 1 到 100 每次遞增 2
        Total = Total + i        '逐項相加
    Next i
    MsgBox "1 + 3 + 5 + ... + 99 = " & Total, vbInformation '顯示計算結果
End Sub
```

1到100奇數和

1 + 3 + 5 + ... + 99 = 2500

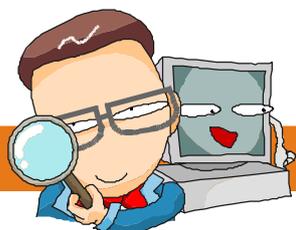
確定

這個程式唯一的差別在於，為了只計算奇數之和，變數 i 必須每次加 2，這可以利用程式碼 For $i=1$ To 100 Step 2 來達成，如此便能達到我們想要計算 1 到 100 之奇數和的目的。

14-2.2 巢狀 For...Next

一個 For...Next 結構之中也可以再包含一個或數個 For...Next 結構，我們稱之為巢狀 For...Next。下面的例子，便是利用二層迴圈建立的「九九乘法表」。由程式中可以看出，當 $i=1$ 時， j 依序由 1 至 9，當 $i=2$ 時， j 一樣依序由 1 至 9，如此重複下去直到 $i=9$ 。另外程式中

二個 Print 指令的目的是，當 j 跑完一個輪迴時（剛好是一行），讓游標換行並空一行。



》》九九乘法表 (巢狀 For...Next)

請利用使用巢狀 For...Next 結構在表單上建立「九九乘法表」，如下圖。【範例檔案：九九乘法表.frm】

```

Private Sub Form_Click()
    For i = 1 To 9           '第一層迴圈
        For j = 1 To 9     '第二層迴圈
            Print j; "x"; i; "="; j * i, '印出一橫列 (如 1x1=1、2 x1=2、3x1=3...)
        Next j
        Print: Print       '換列並空一列
    Next i
End Sub
    
```

1×1 = 1	2×1 = 2	3×1 = 3	4×1 = 4	5×1 = 5	6×1 = 6	7×1 = 7	8×1 = 8	9×1 = 9
1×2 = 2	2×2 = 4	3×2 = 6	4×2 = 8	5×2 = 10	6×2 = 12	7×2 = 14	8×2 = 16	9×2 = 18
1×3 = 3	2×3 = 6	3×3 = 9	4×3 = 12	5×3 = 15	6×3 = 18	7×3 = 21	8×3 = 24	9×3 = 27
1×4 = 4	2×4 = 8	3×4 = 12	4×4 = 16	5×4 = 20	6×4 = 24	7×4 = 28	8×4 = 32	9×4 = 36
1×5 = 5	2×5 = 10	3×5 = 15	4×5 = 20	5×5 = 25	6×5 = 30	7×5 = 35	8×5 = 40	9×5 = 45
1×6 = 6	2×6 = 12	3×6 = 18	4×6 = 24	5×6 = 30	6×6 = 36	7×6 = 42	8×5 = 48	9×6 = 54
1×7 = 7	2×7 = 14	3×7 = 21	4×7 = 28	5×7 = 35	6×7 = 42	7×7 = 49	8×7 = 56	9×7 = 63
1×8 = 8	2×8 = 16	3×8 = 24	4×8 = 32	5×8 = 40	6×8 = 48	7×6 = 56	8×3 = 64	9×8 = 72
1×9 = 9	2×9 = 18	3×9 = 27	4×9 = 36	5×9 = 45	6×9 = 54	7×6 = 63	9×2 = 72	9×9 = 81

14-2.3 條件式迴圈 Do...Loop

條件式迴圈可重覆執行某一程式區段，至於重覆的次數則視設定條件是否符合而定。其中先作測試再決定是否執行迴圈者稱為**前測式迴圈**。而**後測式迴圈**則是先執行一次迴圈敘述後再行判斷是否重覆執行。



Do...Loop(前測式)

格式 Do While 條件式

敘述 1

敘述 2 ...

Loop

或

Do Until 條件式

敘述 1

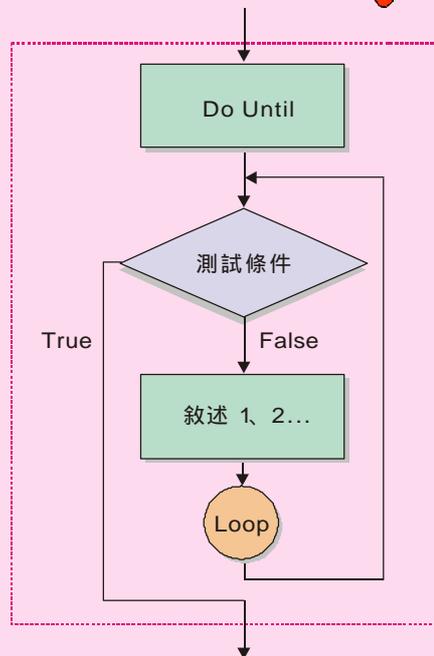
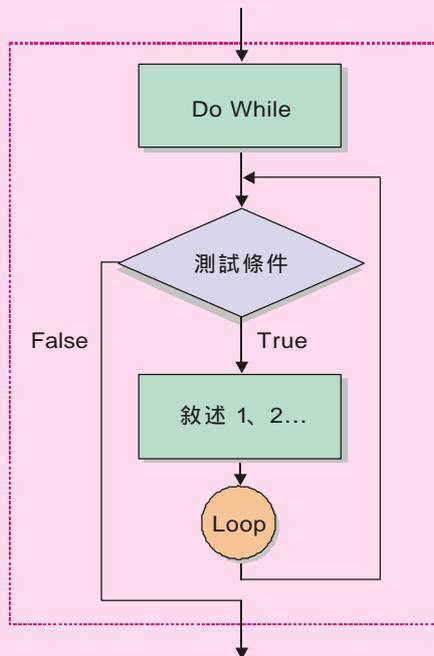
敘述 2 ...

Loop

功能 當條件為真時(While)，或是條件為真以前(Until)，重複執行迴圈內的敘述區段，所以依條件限制有可能一次也沒有執行迴圈內的敘述。

說明 Do While...Loop 或 Do Until...Loop 亦可構成巢式迴圈。

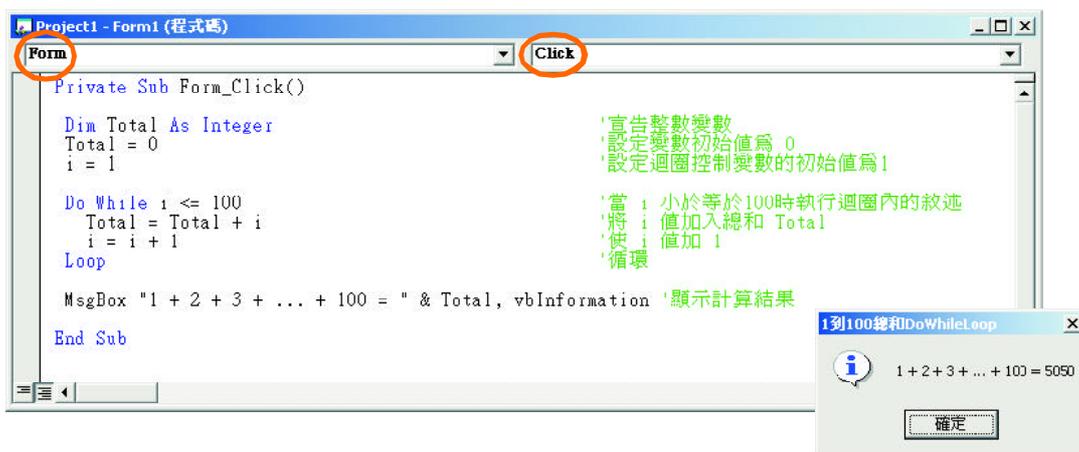
早期的 QBASIC 中有一種 While...Wend 結構，其作用相當於 VB 中的 Do While...Loop 結構。





》計算 1 到 100 的總和 (Do While...Loop)

請利用 Do While...Loop 敘述撰寫一個「計算 1 到 100 總和」的程式，如下圖。【範例檔案：1 到 100 總和 DoWhileLoop.frm】

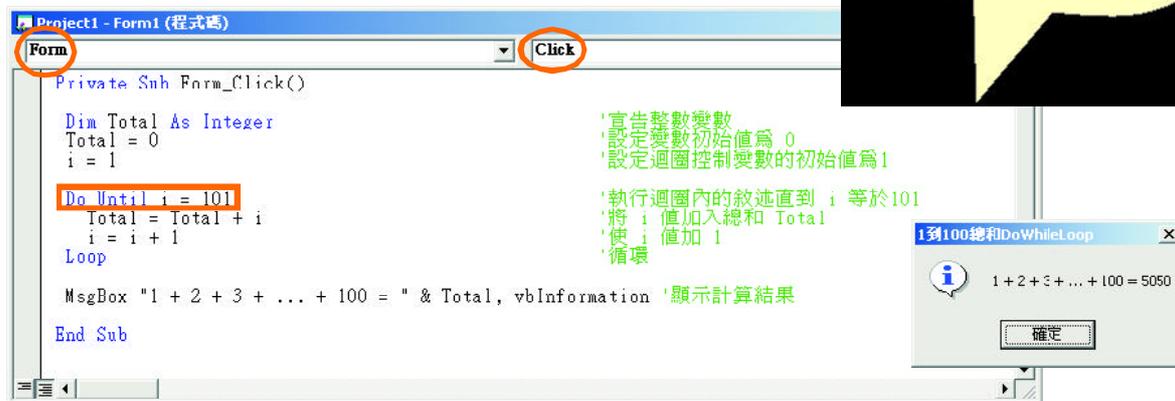


由於前測式迴圈的條件式為 Do While $i \leq 100$ ，所以迴圈內的敘述會一直執行直到 $i=101$ 為止。當 $i=100$ 時，會執行迴圈內的敘述，並逐一將 i 的值加到變數 Total 中，也就是計算 1 到 100 的總和。圖 14-2.1 中的範例也是同樣的原理，只是我們將條件式改為 Do Until $i = 101$ 。（範例檔案：1 到 100 總和 DoWhileLoop.frm）



圖 14-2.1

計算 1 到 100 的總和 (Do Until...Loop)。



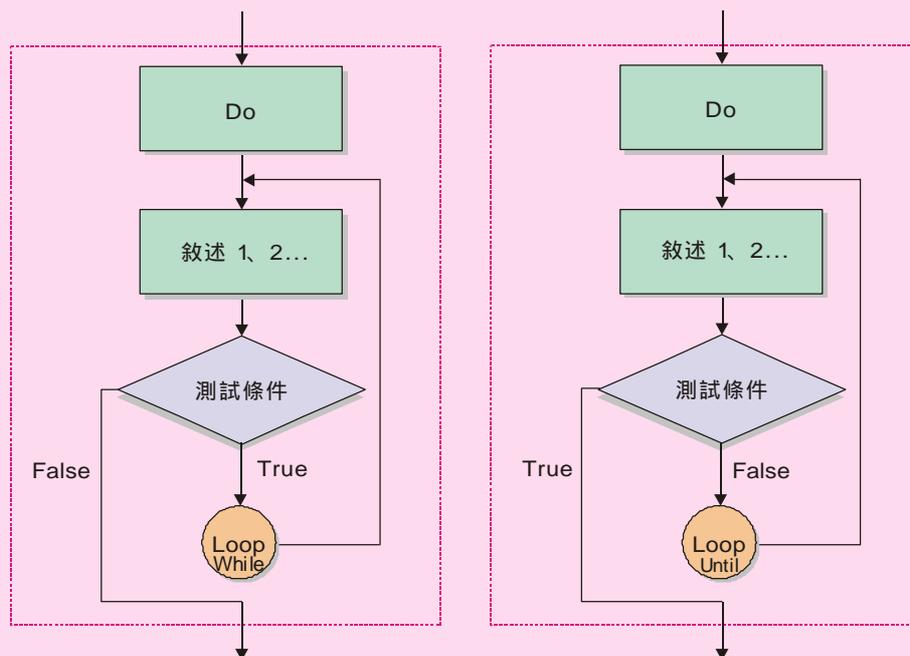
**Do...Loop(後測式)**

格式 Do
敘述 1
敘述 2 ...
Loop While 條件式

或

Do
敘述 1
敘述 2 ...
Loop Until 條件式

功能 至少先執行一次迴圈內的敘述再根據條件式測試是否重複迴圈內的敘述。



下面是二個後測式迴圈的簡單範例。此外，仔細觀察這四個例子會發現，前測試迴圈與後測試迴圈最大的差異處在於：前測試迴圈有可能連一次也不執行迴圈內的敘述，而後測試迴圈則最少會執行迴圈內的敘述一次。



》》 計算 1 到 100 的總和 (Do...Loop While)

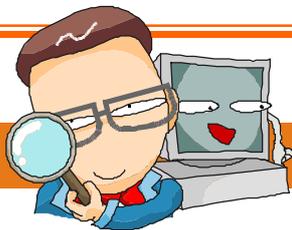
請利用 Do...Loop While 敘述撰寫一個「計算 1 到 100 總和」的程式，如下圖。【範例檔案：1 到 100 總和 DoLoopWhile.frm】

```

Project1 - Form1 (程式碼)
Form Click
Private Sub Form_Click()
    Dim Total As Integer
    Total = 0
    i = 1
    Do
        Total = Total + i
        i = i + 1
    Loop While i <= 100
    MsgBox "1 + 2 + 3 + ... + 100 = " & Total, vbInformation
End Sub
    
```

'宣告整數變數
 '設定變數初始值為 0
 '設定迴圈控制變數的初始值為 1
 '執行迴圈內的敘述
 '將 i 值加入總和 Total
 '使 i 值加 1
 '當 i 小於等於 100 時循環
 '顯示計算結果

1 到 100 總和 DoLoopWhile
 1 + 2 + 3 + ... + 100 = 5050
 確定



》》 計算 1 到 100 的總和 (Do...Loop Until)

請利用 Do...Loop Until 敘述撰寫一個「計算 1 到 100 總和」的程式，如下圖。【範例檔案：1 到 100 總和 DoLoopUntil.frm】

```

Project1 - Form1 (程式碼)
Form Click
Private Sub Form_Click()
    Dim Total As Integer
    Total = 0
    i = 1
    Do
        Total = Total + i
        i = i + 1
    Loop Until i = 101
    MsgBox "1 + 2 + 3 + ... + 100 = " & Total, vbInformation
End Sub
    
```

'宣告整數變數
 '設定變數初始值為 0
 '設定迴圈控制變數的初始值為 1
 '執行迴圈內的敘述
 '將 i 值加入總和 Total
 '使 i 值加 1
 '當 i 等於 101 時不再循環
 '顯示計算結果

1 到 100 總和 DoLoopUntil
 1 + 2 + 3 + ... + 100 = 5050
 確定

14-2.4 重複結構的應用範例

下面是利用 Do...Loop 結構設計的「公雞賽跑」遊戲，為了說明方便，範例中只使用了 Do Until...Loop 的方式，同學可自行嘗試使用其他不同的 Do...Loop 結構改寫。



》》公雞賽跑(重複結構的應用)

請參閱次頁的流程圖，練習撰寫一個「公雞賽跑」的程式，執行畫面如下圖，當使用者按下  鈕後，電腦會以亂數決定要使那隻公雞向前跑，而當任一隻公雞跑超過終點線時便結束比賽且顯示比賽結果。【範例檔案：公雞賽跑.frm】

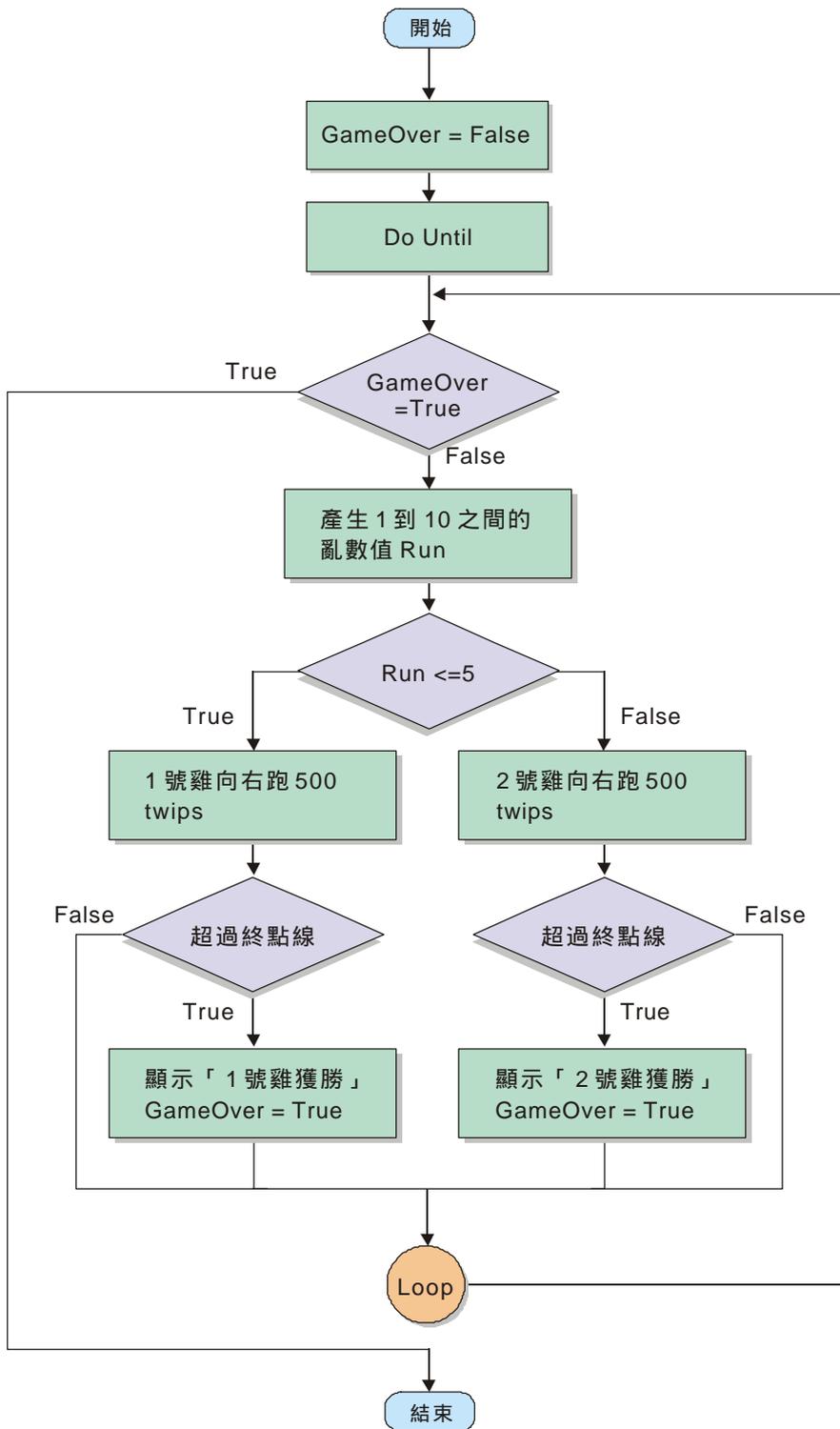


【提示】

1. 利用VB敘述Randomize與Int(N * Rnd)可以產生 0 到 N-1 之間的整數亂數。
2. 以上述方式產生1到10之間的亂數，且當亂數值小於等於5時，1號雞便向前跑一步，反之則2號雞向前跑一步。
3. 使用Do Until...Loop結構使公雞不斷向前跑。
4. 當(公雞圖片的X座標值 Image.Left + 圖片寬度) >= 終點線 X 座標值時，便代表公雞已跑過終點線，比賽結束。

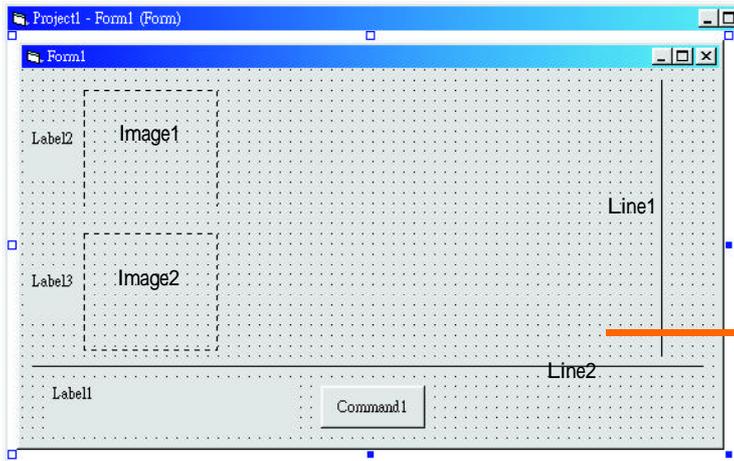
接下頁

【流程圖】



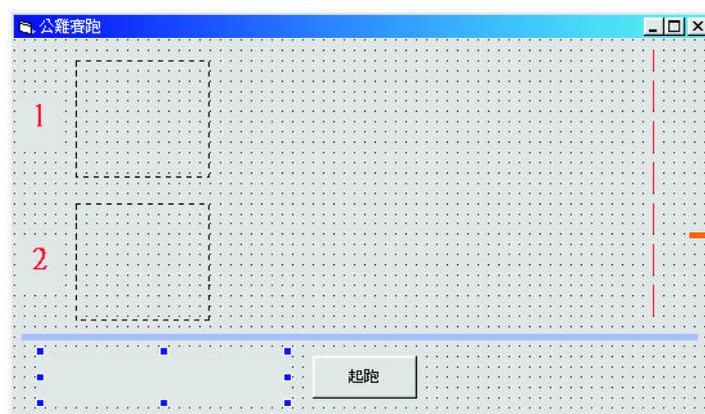
230 第五篇

基本視窗程式應用



1 建立一個新專案，並利用工具箱上的各種控制項建立如圖的表單，然後分別設定各控制項屬性如下表。

物件名稱	屬性	設定值	備註
Form1	Caption	公雞賽跑	表單標題列
Label1	Caption	(空字串)	標籤文字
	Font	14點	字體大小
Label2	Caption	1	標示1號公雞
	Font	粗體, 18點	字體樣式與大小
	ForeColor	&H000000FF&	將文字設為紅色
Label3	Caption	2	標示2號公雞
	Font	粗體, 18點	字體樣式與大小
	ForeColor	&H000000FF&	將文字設為紅色
Command1	Caption	起跑	按鈕文字
Line1	BorderColor	&H000000FF&	設定線條顏色為紅色
	BorderStyle	2-破折線	線條樣式
Line2	BorderColor	&H00008080&	設定線條顏色為紫色
	BorderWidth	5	線條粗細



設定完成後的表單外觀如左圖。

接下頁

```

Project1 - Form1 (程式碼)
Form Load
Private Sub Form_Load()
    Image1 = LoadPicture(App.Path + "\公雞.gif") ' 載入公雞圖片
    Image2 = LoadPicture(App.Path + "\公雞.gif") ' 載入公雞圖片
End Sub
    
```

2 雙按表單開啟【程式碼】視窗，並輸入表單 Form1 的 Load 事件程序如左。

```

Project1 - Form1 (程式碼)
Command1 Click
Private Sub Command1_Click()
    GameOver = False ' 設定控制變數的初始值
    Do Until GameOver = True ' 使用 Do...Loop 迴圈讓公雞不斷向前跑
        Randomize
        Run = Int(10 * Rnd) + 1 ' 產生 1 到 10 之間的亂數
        If Run <= 5 Then ' 當亂數小於等於5時
            Image1.Left = Image1.Left + 500 ' 使 1 號雞向右移動 500 twips
            If Image1.Left + 1560 >= 7440 Then ' 若 1 號雞跑過終點線則獲勝
                Labell.Caption = "1 號雞獲勝!" ' 顯示訊息
                GameOver = True ' 設定控制變數以結束遊戲
            End If
        Else
            Image2.Left = Image2.Left + 500 ' 使 2 號雞向右移動 500 twips
            If Image2.Left + 1560 >= 7440 Then ' 若 2 號雞跑過終點線則獲勝
                Labell.Caption = "2 號雞獲勝!" ' 顯示訊息
                GameOver = True ' 設定控制變數以結束遊戲
            End If
        End If
    Loop
End Sub
    
```

3 雙按  鈕開啟【程式碼】視窗，並輸入 Click 事件程序。