APPENDIX **B**

用實例學Visual Basic 2012程式設計

Visual Basic 物件導向程式設計

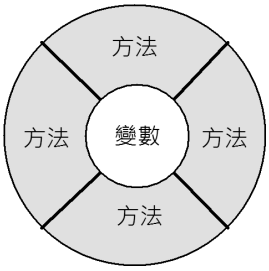
本章重點

B-1 物件導向的基礎

「物件導向程式設計」（Object-oriented Programming，OOP）是一種更符合人性化的程式設計方法，因為我們本來就生活在物件的世界，思考模式也遵循著物件導向模式。

B-1-1 物件的基本觀念

「物件」（Object）是物件導向技術的關鍵，以程式角度來說，物件是資料與相關程序與函數結合在一起的組合體，如下圖所示：



上述圖例的資料被使用介面的方法包裹成一個黑盒子，物件方法就是 Visual Basic 程序與函數。對於程式設計者來說，我們並不用考慮黑盒子內部儲存是哪一種資料；方法的程式碼是如何撰寫，只需知道物件提供什麼介面和如何使用它即可。

換到現實生活，物件範例隨處可見，例如：車子、電視、書桌和貓狗等都是物件。在物件導向技術的物件擁有三種特性，如下所示：

- **狀態（State）**：物件屬性目前的狀態值，屬性儲存的是物件狀態，可以簡單的是布林值變數，也可能是另一個物件，例如：車子的車型、排氣量、色彩和自排或手排等屬性。
- **行為（Behavior）**：行為是物件可見部分提供的服務，即可作什麼事，例如：車子可以發動、停車、加速和換擋等。
- **識別字（Identity）**：識別字是用來識別不同物件，每一個物件都擁有獨一無二的識別字。

因為，開車不需要了解車子是如何發動，變速箱擁有多少個齒輪才能正常的運作，車子對於我們而言是一個黑盒子，唯一要作的是學習如何開好車。同理，沒有幾個人了解電視如何能夠收到訊號，但是我們知道打開電源，更換頻道就可以看到影像。

B-1-2 物件導向程式分析

在 1970~80 年間主要的軟體工程分析方法是「由上而下分析法」（Top-down Design），不過這種分析方法的主要問題，如下所示：

- 由上而下分析法的整個處理過程，只是找出解決問題的程序或函數，也就是各別程序或函數的程式碼，並沒有真正考量到程式使用的資料本身。
- 由上而下分析法得到的程序或函數很難被重複使用，因為程序或函數都是針對特定問題所量身定製，需要大幅修改才能使用在其他問題上。

為了解決上述問題，由上而下分析法經常隨著「由下而上分析法」（Bottom-up Design），這種方法是由下而上，先尋找可重複使用的軟體元件，然後由下而上組合起來，以便解決整個問題。

這些可重複使用的軟體元件是一個個模組，如同電腦硬體的「隨插即用」（Plug and Play），將模組插入軟體系統就可以馬上運作，而不用考慮模組本身的詳細內容。

換句話說，只需符合系統需求，就可以把實際處理的資料隱藏起來，稱為「資訊隱藏」（Information Hiding）。物件是一種包含資料和處理這些資料的程式與函數的模組，以便達到資訊隱藏的目的。

物件導向程式分析是將原來專注於演算法的程序和函數分解，轉換成了解問題本質的物件，將整個軟體視為一個個定義完善的物件，整個程式是由物件組成，強調物件的重複使用，在建立下一層的每一個物件後，由下而上組合成整個應用程式的物件，以便解決整個問題。

B-1-3 物件導向程式語言

物件導向程式語言的精神是物件，程式語言之所以稱為「物件導向程式語言」（Object-oriented Language），主要是指程式語言支援封裝、繼承和多形。

封裝（Encapsulation）

封裝是將資料和處理資料的程序與函數組合成物件。在 Visual Basic 定義物件是使用「類別」（Class），屬於一種抽象資料型態，換句話說，就是替程式語言定義新的資料型態。

繼承（Inheritance）

繼承是物件的再利用，當定義類別後，其他類別就可以繼承此類別的資料和方法，新增或取代繼承物件的資料和方法。

多形（Polymorphism）

多形屬於物件導向最複雜的特性，類別如果需要處理各種不同的資料型態，我們並不需要針對不同資料型態來建立專屬類別，可以直接繼承基礎類別，也就是繼承此類別建立同名方法來處理不同的資料型態，因為方法的名稱相同，只是程式碼不同，也稱為「同名異式」。

B-2 類別與物件

Visual Basic 語言的類別是物件的原型，即物件藍圖。類別宣告可以分為兩個部分，如下所示：

- **成員資料（Data Member）**：物件的資料部分或稱為「成員變數」（Member Variables）。Visual Basic 語言可以使用宣告成 Private 變數配合宣告成 Public 的 Property 程序屬性來存取，或直接宣告成 Public 的變數，稱為欄位（Fields）。
- **成員方法（Method Member）**：物件處理資料的程式與函數，在物件導向語言稱為「方法」（Methods）。

簡單的說，Visual Basic 語言的類別就是由欄位（Fields）、屬性（Properties）、程序與函數組成，欄位與屬性是成員資料；程序與函數是成員方法。

B-2-1 宣告類別與建立物件

在 Visual Basic 程式碼宣告的類別是物件的藍圖，可以建立物件。事實上，類別的主要目的是建立完善的軟體元件，以便提供其他程式碼來使用或擴充其功能。

不過，類別有些方法是屬於類別本身，並不需讓其他程式碼來呼叫，有些則是類別的使用介面，此時可以使用存取修飾子控制類別的哪些成員可以讓其他程式碼存取；哪些不允許，以達成資訊隱藏的目的。

類別宣告

類別宣告是使用 Class/End Class 關鍵字，通常我們是將類別宣告獨立成程式檔案，即在專案加入類別檔案來宣告類別。例如：在 Class1.vb 類別檔案宣告時間資料的 MyTime 類別，如下所示：

```
Public Class MyTime
    ..........
End Class
```

上述程式碼宣告名為 MyTime 類別，內含存取修飾子宣告的成員資料（含屬性或欄位），或成員方法的程序或函數。成員主要是使用 Public 和 Private 兩種存取修飾子來定義成員，如下所示：

- **Private 修飾子**：類別成員只能在類別本身呼叫或存取。例如：MyTime 類別的 intHour、intMinute 和 intSecond 欄位是宣告成 Private，只能讓屬性的程序來存取。
- **Public 修飾子**：類別成員沒有存取限制，它是此類別建立物件的對外使用介面，可以讓其他程式碼呼叫其成員方法或存取成員資料。例如：MyTime 類別的 SetTime()和 GetTime()方法是宣告成 Public。

物件屬性的宣告

Visual Basic 語言的屬性比直接宣告成 Public 變數的欄位擁有更多的控制能力，例如：額外加上程式碼檢查資料範圍。屬性可以存取宣告成 Private 的變數，用來隱藏類別的資訊，如下所示：

```
Public Class MyTime
    Private intHour As Integer
```

```
    Private intMinute As Integer
    Private intSecond As Integer
    Property Hour() As Integer
        Get
            Return intHour
        End Get
        Set(Value As Integer)
            intHour = Value
        End Set
    End Property
    ……
End Class
```

上述 MyTime 類別有 3 個宣告成 Private 變數：intHour、intMinute 和 intSecond 的欄位，每一個屬性是一個 Property/End Property 程序，以此例是 Hour 屬性。在 Property 程序擁有 Get 和 Set 程式區塊，其說明如下所示：

- **Get 程式區塊**：取得欄位值，以此例是傳回 intHour 變數值。
- **Set 程式區塊**：指定欄位值，程式碼使用指定敘述將欄位值指定成參數 Value 的值。

唯讀屬性需要使用 ReadOnly 修飾子，如下所示：

```
ReadOnly Property Hour() As Integer
    Get
       Return intHour
    End Get
End Property
```

上述程式碼宣告唯讀屬性 Hour，所以只有 Get 程式區塊，而沒有 Set 程式區塊。

物件方法的宣告

在 Visual Basic 類別宣告方法就是第 9 章的程序或函數，如下所示：

```
Public Class MyTime
    Private intHour As Integer
    Private intMinute As Integer
    Private intSecond As Integer
    ……
    Public Function GetTime() As String
        Dim str As String
```

```
        str = intHour & ":" & intMinute & ":" & intSecond
        Return str
    End Function
    Public Sub SetTime(h As Integer, _
      ByVal m As Integer, s As Integer)
        intHour = h
        intMinute = m
        intSecond = s
    End Sub
End Class
```

上述程序與函數是類別的成員方法，SetTime()程序可以指定時間資料；GetTime()函數取得時間字串。

建立物件

在 Visual Basic 程式碼建立物件是使用 New 關鍵字，可以依照類別藍圖建立物件，傳回指向此物件的參考（因為物件屬於參考資料型態），如下所示：

```
Dim now, open As MyTime
now = New MyTime()
open = New MyTime()
```

上述程式碼在宣告 MyTime 類別的物件變數 now 和 open 後，使用 New 關鍵字建立物件，因為同一個類別可以建立多個物件，此時每一個物件稱為類別的「實例」（Instances）。

請注意！物件變數 now 和 open 的值並不是物件本身，而是參考到此物件配置的記憶體位址。

存取物件屬性和方法

在 Visual Basic 程式碼建立物件後，我們可以存取物件屬性與方法，物件的屬性和方法都是使用「.」運算子。例如：存取 MyTime 物件 open 的屬性，如下所示：

```
open.Hour = 10
open.Minute = 30
open.Second = 30
```

上述程式碼使用指定敘述指定物件屬性 Hour、Minute 和 Second 的值。我們可以使用同樣方式來呼叫方法，如下所示：

```
txtOutput.Text = "開張時間: " & open.GetTime() &
                          vbNewLine
```

上述程式碼呼叫物件 open 的 GetTime()方法。因為同一個類別可以建立多個物件，所以每一個物件都可以呼叫自己的方法和存取屬性，如下所示：

```
open.GetTime()
close.GetTime()
```

範例專案 AppB-2-1\我的主控台程式

這個 Windows 應用程式是修改第 5-4 節的【我的主控台程式】，加入類別檔案的 MyTime 類別宣告後，在 Form1_Load()事件處理程序建立 MyTime 物件，以便使用物件屬性和方法顯示時間資料，其執行結果如下圖所示：

上述執行結果顯示 MyTime 物件的時間資料，因為 close 和 now 物件變數是指向同一個 MyTime 物件，可以看到時間資料是相同的。

表單設計工具：Form1.vb

請建立 Windows Form 專案【我的主控台程式】，這是直接複製和貼上第 5-4 節範例專案資料夾，然後刪除上下 2 個 Button 控制項，其建立步驟如下所示：

Step 1 請在「方案總管」視窗按二下 Form1.vb，可以開啟表單設計工具。

Step 2 選上下 2 個 Button 按鈕控制項，按 Delete 鍵刪除後，可以看到我們建立的使用介面，如右圖所示：

程式碼編輯器：Class1.vb

在「方案總管」視窗新增 Class1.vb 類別檔後，可以在程式碼編輯器輸入 MyTime 類別宣告的程式碼，請繼續上面步驟，如下所示：

Step 3 在「方案總管」視窗的【我的主控台程式】專案上，執行右鍵快顯功能表的「加入＞類別」指令，可以看到「加入新項目」對話方塊。

Step 4 【名稱】欄的預設檔名是【Class1.vb】，按【新增】鈕，可以看到開啟 Class1.vb 程式碼編輯器，輸入 MyTime 類別宣告。


```
01: Public Class MyTime
02:     Private intHour As Integer
03:     Private intMinute As Integer
04:     Private intSecond As Integer
05:     ' 物件屬性
06:     Property Hour() As Integer
07:         Get
08:             Return intHour
09:         End Get
10:         Set(Value As Integer)
11:             intHour = Value
12:         End Set
13:     End Property
14:     Property Minute() As Integer
15:         Get
16:             Return intMinute
```

```
17:         End Get
18:         Set(Value As Integer)
19:             intMinute = Value
20:         End Set
21:     End Property
22:     Property Second() As Integer
23:         Get
24:             Return intSecond
25:         End Get
26:         Set(Value As Integer)
27:             intSecond = Value
28:         End Set
29:     End Property
30:     ' 物件方法: 取得時間字串
31:     Public Function GetTime() As String
32:         Dim str As String
33:         str = intHour & ":" & intMinute & ":" & intSecond
34:         Return str
35:     End Function
36:     ' 物件方法: 設定時間
37:     Public Sub SetTime(h As Integer, m As _
                        Integer, s As Integer)
38:         intHour = h : intMinute = m : intSecond = s
39:     End Sub
40: End Class
```

程式碼解說

- 第 1~40 列：宣告 MyTime 類別，在第 2~4 列是宣告成 Private 變數的欄位，第 6~29 列宣告 Hour、Minute 和 Second 屬性，在第 31~39 列宣告 GetTime() 和 SetTime()方法。

程式碼編輯器：Form1.vb

Step 5 切換至程式碼編輯器，可以建立 Form1_Load()事件處理程序，和刪除 Button1_Click()和 Button2_Click()事件處理程序。

```
01: Private Sub Form1_Load(sender As Object, e _
                    As EventArgs) Handles Me.Load
02:     ' 宣告 myTime 類別的物件變數
03:     Dim now, open, close As MyTime
04:     ' 建立物件實例
```

```
05:     now = New MyTime()
06:     open = New MyTime()
07:     close = now
08:     ' 設定 open 物件的成員變數
09:     open.Hour = 10
10:     open.Minute = 30
11:     open.Second = 30
12:     txtOutput.Text="開張時間: " & open.GetTime() & vbNewLine
13:     ' 設定 close 物件的成員變數
14:     close.SetTime(22, 30, 0)
15:     txtOutput.Text &= "結束時間: " & close.GetTime() &
                        vbNewLine
16:     txtOutput.Text &= "現在時間: " & now.GetTime()
17:     txtOutput.SelectionStart = 0
18: End Sub
```

程式碼解說

- 第 3~7 列：宣告 MyTime 物件變數和建立 MyTime 物件，在第 7 列將 close 指定成 now，換句話說，兩個物件變數是指向同一個 MyTime 物件。
- 第 9~16 列：使用屬性和 SetTime()方法指定物件的時間資料後，分別使用 GetTime()方法取出時間字串，然後顯示在 TextBox 控制項。
- 第 17 列：指定 TextBox 控制項的 SelectionStart 屬性來取消顯示反白的文字內容。

B-2-2 方法的過載

Visual Basic 語言的程序與函數允許擁有兩個以上的同名方法，只需傳遞的參數個數或資料型態不同即可，稱為「過載」（Overloads）或重載。在類別建立過載方法需在方法名稱前使用 Overloads 關鍵字（模組不需要），如下所示：

```
Public Overloads Sub SetTime(h As Integer,
                 m As Integer, s As Integer)
Public Overloads Sub SetTime(h As Integer,
                 m As Integer
```

上述同名方法的參數個數分別是 3 個和 2 個。

範例專案 AppB-2-2\我的主控台程式

這個 Windows 應用程式是修改第 B-2-1 節的【我的主控台程式】，新增 MyTime 類別的過載方法，可以使用 MyTime 物件方法指定時間資料，其執行結果如下圖所示：

表單設計工具：Form1.vb

請建立 Windows Form 專案【我的主控台程式】，這是直接複製和貼上第 B-2-1 節範例專案資料夾，其步驟如下所示：

Step 1 請在「方案總管」視窗按二下 Form1.vb 開啟表單設計工具，可以看到反白顯示唯讀的 TextBox 多行文字方塊控制項。

程式碼編輯器：Class1.vb

Step 2 在「方案總管」視窗按二下開啟 Class1.vb 類別檔後，可以在程式碼編輯器修改 MyTime 類別宣告的程式碼。

```
01: Public Class MyTime
02:     Public Hour As Integer
03:     Public Minute As Integer
04:     Public Second As Integer
05:     ' 物件方法: 取得時間字串
06:     Public Function GetTime() As String
07:         Dim str As String
08:         str = Hour & ":" & Minute & ":" & Second
09:         Return str
10:     End Function
11:     ' 物件方法: 設定時間(1)
12:     Public Overloads Sub SetTime(h As Integer,
                m As Integer, s As Integer)
13:         Hour = h : Minute = m : Second = s
14:     End Sub
15:     ' 物件方法: 設定時間(2)
```

```
16:     Public Overloads Sub SetTime(h As Integer,
                m As Integer)
17:         Hour = h : Minute = m : Second = 0
18:     End Sub
19: End Class
```

程式碼解說

- 第 1~19 列：宣告 MyTime 類別，在第 2~4 列改為 Public 欄位，第 12~18 列宣告 2 個過載方法 SetTime()，只是參數個數不同。

程式碼編輯器：Form1.vb

Step 3 切換至程式碼編輯器，可以修改 Form1_Load()事件處理程序。

```
01: Private Sub Form1_Load(sender As Object, e _
                    As EventArgs) Handles Me.Load
02:     ' 宣告 myTime 類別的物件變數
03:     Dim open, close As MyTime
04:     ' 建立物件實例
05:     close = New MyTime()
06:     open = New MyTime()
07:     ' 設定 open 物件的成員變數
08:     open.SetTime(10, 30)
09:     ' 設定 close 物件的成員變數
10:     close.SetTime(22, 30, 0)
11:     txtOutput.Text="開張時間: " & open.GetTime() & vbNewLine
12:     txtOutput.Text &= "結束時間: " & close.GetTime() &
                                    vbNewLine
13:     txtOutput.SelectionStart = 0
14: End Sub
```

程式碼解說

- 第 8 列和第 10 列：使用過載 SetTime()方法指定物件的時間資料，可以看到參數的個數不同。

B-2-3 類別的建構子

當 Visual Basic 程式碼使用基本資料型態宣告變數時，編譯程式會配置所需記憶體空間和指定預設的初值。如果類別沒有「建構子」（Constructors），或

稱建構函數，在使用 New 關鍵字建立物件時，物件預設建構子只會配置記憶體空間，並不會指定成員資料的初值，即初始物件。

如果希望類別在建立物件時能夠指定初值，類別需要宣告建構子。建構子是物件的初始方法，在建立物件時會自動呼叫此方法（如果沒有，就使用預設建構子）。建構子的特點如下所示：

- 建構子沒有傳回值，換句話說，它一定是 Sub 程序。
- 不論類別名稱，建構子一定是名為 New 的程序，例如：類別 MyTime 的建構子是 New()。
- 建構子通常是使用 Public 修飾子進行宣告，事實上，建構子可以使用任何存取修飾子。
- 建構子支援過載，可以擁有多個同名建構子，只是擁有不同參數型態和個數，如下所示：

```
Public Sub New()
Public Sub New(h As Integer, m As Integer)
Public Sub New(h As Integer, m As Integer,
               s As Integer)
```

上述第 1 個建構子沒有參數；第 2 個擁有 2 個參數；第 3 個有 3 個參數。建構子程式碼的撰寫方式和其他成員方法相同，不過，建構子過載，並不需要使用 Overloads 關鍵字。

範例專案　AppB-2-3\我的主控台程式

這個 Windows 應用程式是修改第 B-2-2 節的【我的主控台程式】，在 MyTime 類別將 SetTime()程序改為類別過載的建構子後，使用不同建構子來初始物件，其執行結果如下圖所示：

表單設計工具：Form1.vb

請建立 Windows Form 專案【我的主控台程式】，這是直接複製和貼上第 B-2-2 節範例專案資料夾，其步驟如下所示：

Step 1 請在「方案總管」視窗按二下 Form1.vb 開啟表單設計工具，可以看到反白顯示唯讀的 TextBox 多行文字方塊控制項。

程式碼編輯器：Class1.vb

Step 2 在「方案總管」視窗按二下開啟 Class1.vb 類別檔後，可以在程式碼編輯器修改 MyTime 類別宣告的程式碼。

```
01: Public Class MyTime
02:     Public Hour As Integer
03:     Public Minute As Integer
04:     Public Second As Integer
05:     ' 建構子(1)
06:     Public Sub New()
07:         Hour = 0 : Minute = 0 : Second = 0
08:     End Sub
09:     ' 建構子(2)
10:     Public Sub New(h As Integer,
11:                    m As Integer)
12:         Hour = h : Minute = m : Second = 0
13:     End Sub
14:     ' 建構子(3)
15:     Public Sub New(h As Integer,
16:                m As Integer, s As Integer)
17:         Hour = h : Minute = m : Second = s
18:     End Sub
19:     ' 物件方法: 取得時間字串
20:     Public Function GetTime() As String
21:         Dim str As String
22:         str = Hour & ":" & Minute & ":" & Second
23:         Return str
24:     End Function
25: End Class
```

程式碼解說

- 第 1~25 列：宣告 MyTime 類別，在第 6~18 列宣告 3 個過載建構子，只是參數的個數不同。

程式碼編輯器：Form1.vb

Step 3 切換至程式碼編輯器，可以修改 Form1_Load()事件處理程序。

```
01: Private Sub Form1_Load(sender As Object, e _
                    As EventArgs) Handles Me.Load
02:     ' 宣告 myTime 類別的物件變數
03:     Dim now, open, close As MyTime
04:     ' 建立物件實例
05:     now = New MyTime()
06:     close = New MyTime(10, 30)
07:     open = New MyTime(22, 30, 0)
08:     txtOutput.Text="現在時間: " & now.GetTime() & vbNewLine
09:     txtOutput.Text &= "開張時間: " & open.GetTime() &
                                    vbNewLine
10:     txtOutput.Text &= "結束時間: " & close.GetTime() &
                                    vbNewLine
11:     txtOutput.SelectionStart = 0
12: End Sub
```

程式碼解說

- 第 5~7 列：分別使用 3 個過載建構子建立 3 個物件。

B-2-4 類別的靜態成員

類別的靜態成員（Static Member）屬於一種特殊成員，這些成員不需要建立物件，就可以使用類別名稱來存取和呼叫，例如：在第 9 章說明的 Math 數學類別和 Console 類別的 WriteLine()方法。

Visual Basic 語言是使用 Shared 關鍵字宣告類別的靜態成員，或稱為類別成員。例如：在 Student 類別宣告靜態成員 Count 和 NumOfStudents()方法，如下所示：

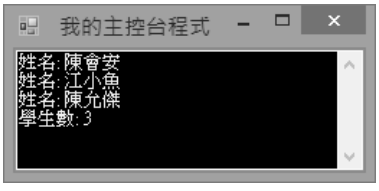
```
Class Student
    Private Name As String
    Private Shared Count As Integer
    ……
    Public Shared Function NumOfStudents() As Integer
        Return Count
    End Function
End Class
```

上述程式碼使用 Shared 關鍵字宣告 Count 變數和 NumOfStudents()方法。因為是靜態成員，所以程式碼可以直接使用類別名稱 Student 來呼叫，如下所示：

```
txtOutput.Text &= "學生數: " & Student.NumOfStudents() &
                  vbNewLine
```

範例專案 AppB-2-4\我的主控台程式

這個 Windows 應用程式是修改第 B-2-3 節的【我的主控台程式】，在類別檔宣告 Student 類別，內含靜態成員 Count 和 NumOfStudents()方法來計算學生數，其執行結果如下圖所示：



表單設計工具：Form1.vb

請建立 Windows Form 專案【我的主控台程式】，這是直接複製和貼上第 B-2-3 節範例專案資料夾，其步驟如下所示：

Step 1 請在「方案總管」視窗按二下 Form1.vb 開啟表單設計工具，可以看到反白顯示唯讀的 TextBox 多行文字方塊控制項。

程式碼編輯器：Class1.vb

Step 2 在「方案總管」視窗按二下開啟 Class1.vb 類別檔後，可以在程式碼編輯器刪除 MyTime 類別後，改為 Student 類別宣告的程式碼。

```
01: Public Class Student
02:     Private Name As String
03:     Private Shared Count As Integer
04:     ' 建構子
05:     Public Sub New(Name As String)
06:         Me.Name = Name
07:         Count += 1
08:     End Sub
09:     ' 成員方法
10:     Public Function GetStudentName() As String
11:         Return Name
12:     End Function
13:     ' 靜態成員方法，也稱為類別方法
14:     Public Shared Function NumOfStudents() As Integer
15:         Return Count
16:     End Function
17: End Class
```

程式碼解說

- 第 1~17 列：Student 類別宣告的第 3 列是靜態成員 Count，在第 14~16 列是靜態成員方法 NumOfStudents()，在第 5~8 列是建構子，因為參數與成員變數同名，所以第 6 列使用 Me 關鍵字指明是成員變數 Name；而不是參數 Name。

程式碼編輯器：Form1.vb

Step 3 切換至程式碼編輯器，可以修改 Form1_Load()事件處理程序。

```
01: Private Sub Form1_Load(sender As Object, e _
                    As EventArgs) Handles Me.Load
02:     ' 宣告 Student 類別的物件變數
03:     Dim joe, jane, jason As Student
04:     ' 建立物件實例
05:     joe = New Student("陳會安")
06:     jane = New Student("江小魚")
07:     jason = New Student("陳允傑")
08:     ' 呼叫物件方法
09:     txtOutput.Text = "姓名: " & joe.GetStudentName() &
                         vbNewLine
10:     txtOutput.Text &= "姓名: " & jane.GetStudentName() &
                         vbNewLine
```

```
11:     txtOutput.Text &= "姓名: " & jason.GetStudentName() &
                          vbNewLine
12:     txtOutput.Text &= "學生數: " & Student.NumOfStudents()
13:     txtOutput.SelectionStart = 0
14: End Sub
```

程式碼解說

- 第 12 列：呼叫靜態成員方法 NumOfStudents()顯示學生數。

B-2-5 部分類別

部分類別（Partial Class）允許同一個類別分割成多個檔案。主要類別的宣告如同一般類別，例如：MyName 類別宣告，如下所示：

```
Public Class MyName
    Private FirstName, LastName As String
    Public Sub New(f As String, l As String)
        FirstName = f : LastName = l
    End Sub
    Public Function GetName() As String
        Return "姓名: " & LastName & FirstName
    End Function
End Class
```

上述程式碼的類別宣告和一般類別宣告並沒有什麼不同。不過，部分類別需要使用 Partial 關鍵字進行類別宣告，如下所示：

```
Partial Class MyName
    Public Function GetFirstName() As String
        Return "名: " & FirstName
    End Function
End Class
```

上述程式碼是部分類別宣告，類別名稱與主要類別相同，只是新增 GetFirstName()成員方法。請注意！雖然原始程式碼分割成多個檔案，但在編譯時，仍然是組合成完整類別後才進行編譯。

部分類別在解決程式問題上並沒有任何幫助，其主要目的是方便 Visual Studio 整合開發工具的程式碼管理。例如：VSE 的 Windows Form 應用程式專案，在表單設計視窗自動建立的控制項程式碼，就是位在 Form1.Designer.vb 檔案的部分類別。

範例專案 AppB-2-5\我的主控台程式

這個 Windows 應用程式是修改第 B-2-4 節的【我的主控台程式】，在類別檔宣告 MyName 部分類別來儲存姓名資料，其執行結果如下圖所示：

表單設計工具：Form1.vb

請建立 Windows Form 專案【我的主控台程式】，這是直接複製和貼上第 B-2-4 節範例專案資料夾，其步驟如下所示：

Step 1 請在「方案總管」視窗按二下 Form1.vb 開啟表單設計工具，可以看到反白顯示唯讀的 TextBox 多行文字方塊控制項。

程式碼編輯器：Class1.vb

Step 2 在「方案總管」視窗按二下開啟 Class1.vb 類別檔後，可以在程式碼編輯器刪除 Student 類別後，改為 MyName 類別宣告的程式碼。

```
01: Public Class MyName
02:     Private FirstName, LastName As String
03:         Public Sub New(f As String, _
                        l As String)
04:         FirstName = f : LastName = l
05:     End Sub
06:     Public Function GetName() As String
07:         Return "姓名： " & LastName & FirstName
08:     End Function
09: End Class
10:
11: Partial Class MyName
12:     Public Function GetFirstName() As String
13:         Return "名： " & FirstName
14:     End Function
15: End Class
```

程式碼解說

- 第 1~15 列：MyName 類別宣告，分成 2 個部分類別，各擁有一個成員方法，在第 1~9 列是主要類別，第 11~15 列是部分類別的宣告。

程式碼編輯器：Form1.vb

Step 3 切換至程式碼編輯器，可以修改 Form1_Load()事件處理程序。

```
01: Private Sub Form1_Load(sender As Object, e _
                    As EventArgs) Handles Me.Load
02:     ' 宣告 MyName 類別的物件
03:     Dim name As New MyName("會安", "陳")
04:     ' 呼叫物件方法
05:     txtOutput.Text = name.GetName() & vbNewLine
06:     txtOutput.Text &= name.GetFirstName() & vbNewLine
07:     txtOutput.SelectionStart = 0
08: End Sub
```

程式碼解說

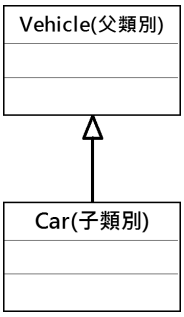
- 第 3~6 列：在建立 MyName 物件後，呼叫成員方法顯示姓名，2 個方法是位在不同部分類別的宣告。

B-3 類別的繼承

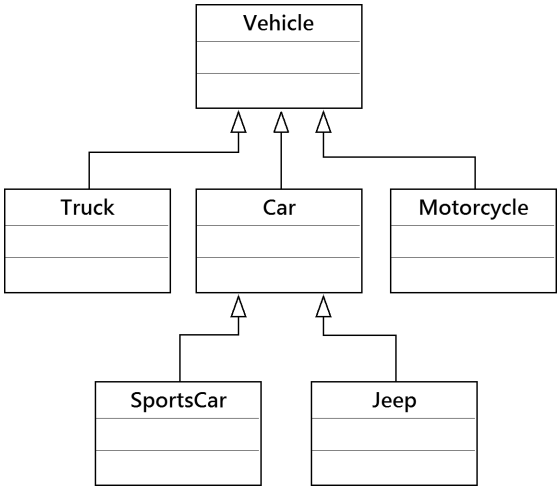
「繼承」（Inheritance）是物件導向程式設計的重要觀念，一個類別可以直接繼承現存類別的部分或全部成員資料和方法，擴充功能來新增額外的成員資料或方法，或覆寫繼承類別的方法來修改其功能。

B-3-1 類別繼承的基礎

類別如果是繼承自其他類別，我們稱此類別為繼承類別的「子類別」（Subclass）或「延伸類別」（Derived Class），繼承的類別稱為「父類別」（Superclass）或「基礎類別」（Base Class）。例如：類別 Car 是繼承自類別 Vehicle，其繼承關係如下圖所示：



上述圖例的 Vehicle 類別是 Car 類別的父類別，反之，類別 Car 是類別 Vehicle 的子類別。如果多個子類別繼承同一個父類別，則每一個子類別稱為「兄弟類別」（Sibling Classes），如下圖所示：



上述圖例的類別 Truck、Car 和 Motocycle 是兄弟類別。當然我們可以繼續繼承類別 Car，此時的類別 Jeep 也是類別 Vehicle 的子類別，不過，並不屬於直接繼承的子類別。

B-3-2 實作類別的繼承

筆者準備直接使用範例來說明類別繼承。在父類別 TextLine 定義一列文字字串，如下所示：

```
Public Class TextLine
    Public Line As String
```

```
    Public Sub New()
        Line = ""
    End Sub
    Public Sub New(text As String)
        Line = text
    End Sub
    ……
End Class
```

上述 TextLine 類別擁有 Line 的成員資料、建構子和成員方法。

類別的繼承

在專案可以建立新類別檔（也可以在同一個類別檔）繼承已經存在的類別，繼承是使用 Inherits 關鍵子擴充父類別的宣告。例如：MyLine 子類別是繼承自 TextLine 類別，其宣告如下所示：

```
Public Class MyLine
    Inherits TextLine
    Public Sub New()
        MyBase.New()
    End Sub
    ……
End Class
```

上述程式碼的 MyLine 子類別繼承自 TextLine 父類別，子類別可以使用 MyBase 呼叫父類別的建構子、方法和屬性，以此例是建構子。然後在子類別新增所需的屬性、方法或覆寫父類別的方法。

雖然子類別可以繼承父類別所有成員資料和方法，但是在存取時仍然有一些限制，如下所示：

- 子類別不能存取父類別宣告成 Private 的成員資料和方法。
- 父類別的建構子並不屬於類別成員，所以子類別並不能繼承父類別的建構子，只能使用 MyBase 呼叫父類別的建構子。

覆寫父類別的方法

當子類別繼承父類別擴充其功能時，對於父類別的方法，如果希望保留方法名稱，只更改其功能，在父類別的方法可以使用 Overridable 關鍵字來宣告，表示方法可以覆寫，如下所示：

```
Public Class TextLine
    Public Line As String
    ……
    Public Overridable Function GetText() As String
        Return LCase(Line)
    End Function
End Class
```

上述父類別的 GetText()方法是可覆寫的方法。在繼承子類別可以使用 Overrides 關鍵字來覆寫此方法，如下所示：

```
Public Class MyLine
    Inherits TextLine
    ……
    Public NotOverridable Overrides _
              Function GetText() As String
        Return MyBase.Line
    End Function
End Class
```

上述子類別擁有同名 GetText()方法覆寫父類別的方法，NotOverridable 關鍵字表示此方法不能再被子類別覆寫。當執行 MyLine 類別的物件方法 GetText()，就是執行覆寫方法，而不再是父類別的 GetText()方法。

事實上，在 VSE 建立的 Windows Form 應用程式專案，其自動建立的 Form1.Designer.vb 檔案是表單使用介面的程式碼，此部分類別是繼承自 Windows.Forms 的 Form 類別，如下所示：

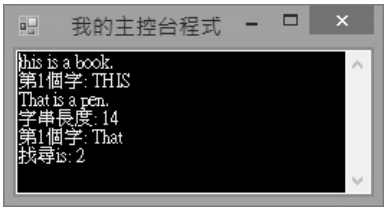
```
Partial Class Form1
    Inherits System.Windows.Forms.Form
    ……
End Class
```

上述程式碼的 Form1 類別是繼承 Windows.Forms.Form 的子類別，簡單的說，Windows.Forms.Form 類別是一個空視窗物件，所謂視窗應用程式就是繼承此類別，然後新增控制項，也就是在表單設計工具新增的控制項。

範例專案 AppB-3-2\我的主控台程式

這個 Windows 應用程式是修改第 B-2-5 節的【我的主控台程式】，擁有 Class1.vb 父類別檔和 Class2.vb 子類別檔，MyLine 類別是繼承自 TextLine 類別，

新增屬性 Length、方法 InStr()和覆寫 GetText()方法後，測試這 2 個類別建立的物件，其執行結果如下圖所示：



上述執行結果的第 1 個字串都是小寫，它是執行 TextLine 類別的 GetText()方法，第 2 個字串有大小寫之分，因為是執行 MyLine 子類別覆寫的方法。

表單設計工具：Form1.vb

請建立 Windows Form 專案【我的主控台程式】，這是直接複製和貼上第 B-2-5 節範例專案資料夾，其步驟如下所示：

Step 1 請在「方案總管」視窗按二下 Form1.vb 開啟表單設計工具，可以看到反白顯示唯讀的 TextBox 多行文字方塊控制項。

程式碼編輯器：Class1.vb

Step 2 在「方案總管」視窗按二下開啟 Class1.vb 類別檔後，可以在程式碼編輯器刪除 MyName 類別後，改為 TextLine 類別宣告的程式碼。

```
01: Public Class TextLine
02:     Public Line As String
03:     ' 建構子(1)
04:     Public Sub New()
05:         Line = ""
06:     End Sub
07:     ' 建構子(2)
08:     Public Sub New(text As String)
09:         Line = text
10:     End Sub
11:     ' 物件方法
12:     Public Function GetWord() As String
13:         Dim arrWords() As String = Split(Line, " ")
14:         Return (arrWords(0))
15:     End Function
```

```
    ' 可覆寫的方法
    Public Overridable Function GetText() As String
        Return LCase(Line)
    End Function
End Class
```

程式碼解說

- 第 1~20 列：TextLine 類別宣告，內含 2 個過載建構子、Line 欄位、GetWord() 方法和可覆寫的 GetText()方法。

程式碼編輯器：Class2.vb

Step 3 在「方案總管」視窗新增 Class2.vb 類別檔後，可以在程式碼編輯器輸入 MyLine 類別宣告的程式碼。

```
Public Class MyLine
    Inherits TextLine
    ' 建構子
    Public Sub New()
        MyBase.New()
    End Sub
    ' 新增屬性
    ReadOnly Property Length() As Long
        Get
            Return Len(MyBase.Line)
        End Get
    End Property
    ' 新增方法
    Public Function InStr(str As String) As Integer
        Dim arrWords() As String =
                    Split(MyBase.Line, " ")
        Dim i As Integer
        For i = 0 To UBound(arrWords)
            If arrWords(i) = str Then
                Exit For
            End If
        Next i
        If i < UBound(arrWords) Then
            Return i + 1
        End If
        Return -1
```

```
26:     End Function
27:     ' 覆寫方法
28:     Public NotOverridable Overrides _
              Function GetText() As String
29:         Return MyBase.Line
30:     End Function
31: End Class
```

程式碼解說

- 第 1~31 列：繼承自 TextLine 類別的 MyLine 子類別宣告，在第 5 列以 MyBase 呼叫父類別的建構子，第 8~26 列新增唯讀的 Length 屬性和 InStr()方法，屬性和方法都是使用 MyBase.Line 取得父類別的字串內容，在第 15 列使用 Split()函數以空白字元作為分隔符號，將字串內容分割轉換成字串陣列。
- 第 28~30 列：覆寫父類別的 GetText()方法，因為使用 NotOverridable 關鍵字，表示此方法不能再被子類別覆寫。

程式碼編輯器：Form1.vb

Step 4 切換至程式碼編輯器，可以修改 Form1_Load()事件處理程序。

```
01: Private Sub Form1_Load(sender As Object, e _
                    As EventArgs) Handles Me.Load
02:     Dim line As MyLine = New MyLine()
03:     Dim txtline As TextLine =
                New TextLine("THIS IS A BOOK.")
04:     txtOutput.Text = txtline.GetText() & vbNewLine
05:     txtOutput.Text &= "第 1 個字: " & txtline.GetWord() &
                    vbNewLine
06:     line.Line = "That is a pen."
07:     txtOutput.Text &= line.GetText() & vbNewLine
08:     txtOutput.Text &= "字串長度: " & line.Length &
                    vbNewLine
09:     txtOutput.Text &= "第 1 個字: " & line.GetWord() &
                    vbNewLine
10:     txtOutput.Text &= "找尋 is: " & line.InStr("is")
11:     line = Nothing : txtline = Nothing
12:     txtOutput.SelectionStart = 0
13: End Sub
```

程式碼解說

- 第 2~3 列：建立 MyLine 和 TextLine 類別的物件 line 和 txtline，TextLine 是使用建構子指定字串內容。
- 第 4~5 列：呼叫 GetText()方法顯示 txtline 的字串內容和執行 GetWord()方法。
- 第 7~10 列：顯示 line 的字串內容、取得 Length 屬性值、執行 GetWord()和 InStr()方法，其中 Length 屬性和 InStr()是新增的屬性和方法，GetWord()是繼承的方法。
- 第 11 列：將物件設為 Nothing，表示該物件變數並沒有值，即尚未建立物件。

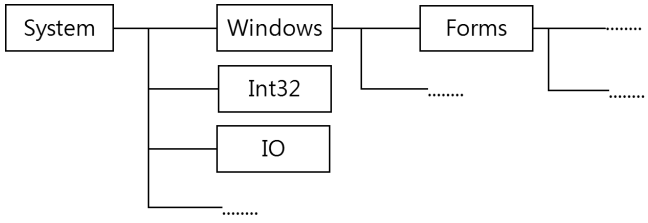
B-4 .NET Framework 類別函數庫

Visual Basic 語言是一種.NET Framework 程式語言，它是微軟新世代的程式開發平台，在本書使用的版本是 4.5 版。

B-4-1 .NET Framework 的基礎

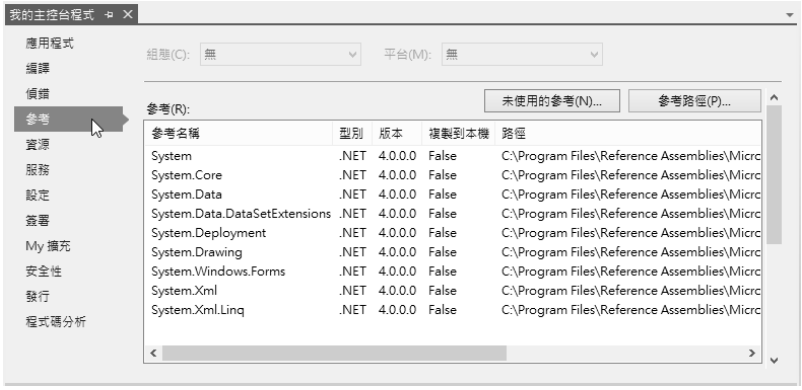
.NET Framework 是由 CLR(Common Language Rumtime)和.NET Framework 類別函數庫（.NET Framework Class Library，簡稱.NET FCL）所組成。

.NET FCL 是一個龐大的類別函數庫，只需支援.NET Framework 的程式語言都可以使用此類別函數庫的類別和方法。它是使用稱為命名空間（Namespace）的階層類別架構所組織成的函數庫，在每一個命名空間下擁有多個類別，如下圖所示：



上述圖例的 System 是最基礎的命名空間，其下擁有基本資料型態的 System.Int32（每一階層的命名空間是使用「.」運算子連接）、System.Byte、System.Single 和 System.String 等命名空間。

例如：Visual Basic 建立的 Windows 應用程式和主控台應用程式專案預設匯入相關的命名空間，請開啟專案的屬性頁，在左邊選【參考】標籤，如下圖所示：



上述圖例的上方是參考項目，即匯入的.NET DLL 檔案。請捲動視窗至最後，可以看到匯入的命名空間，如下表所示：

專案範本	匯入的命名空間	.NET DLL 檔案
主控台應用程式	Microsoft.VisualBasic System System.Collections System.Collections.Generic System.Data System.Diagnostics System.Linq System.Xml.Linq System.Threading.Tasks	 System.dll System.Core.dll System.Data.dll System.DataSetExtensions.dll System.Deployment.dll System.Xml.dll System.Xml.Linq.dll
Windows Form 應用程式	Microsoft.VisualBasic System System.Collections System.Collections.Generic System.Data System.Diagnostics System.Drawing System.Windows.Forms System.Linq System.Xml.Linq System.Threading.Tasks	 System.dll System.Core.dll System.Data.dll System.DataSetExtensions.dll System.Drawing.dll System.Deployment.dll System.Windows.Forms.dll System.Xml.dll System.Xml.Linq.dll

換句話說，為什麼我們可以在專案使用 Visual Basic 函數，就是因為匯入 Microsoft.VisualBasic 命名空間。Windows Form 應用程式可以繪圖和建立表單是因為匯入 System.Drawing 和 System.Windows.Forms 命名空間。

B-4-2 在 Visual Basic 程式匯入的命名空間

對於專案預設匯入的命名空間，我們可以直接在 Visual Basic 程式碼使用此命名空間下的類別，如果不屬於預設匯入的命名空間，就需要自行使用程式碼匯入所需的命名空間，其語法如下所示：

```
Imports 命名空間
```

上述程式碼使用 Imports 關鍵字匯入指定的命名空間，其功能如同在參考標籤勾選指定的命名空間，此後即可在程式碼使用此命名空間的類別。例如：在 Visual Basic 應用程式使用 StringBuilder 物件需要匯入 System.Text 命名空間，如下所示：

```
Imports System.Text
```

上述程式碼是位在類別宣告之外，例如：Form1 類別宣告，如下所示：

```
Imports System.Text

Public Class Form1
    ……
End Class
```

除了使用 Imports 關鍵字外，我們也可以在專案的屬性頁直接勾選所需匯入的命名空間。

B-5 My 命名空間的類別

對於.NET FCL 類別函數庫的龐大類別來說，程式設計師常常需要花費相當時間才能找到真正需要的類別，在 Visual Basic 語言提供名為 My 的命名空間，可以讓我們快速找到所需的類別。

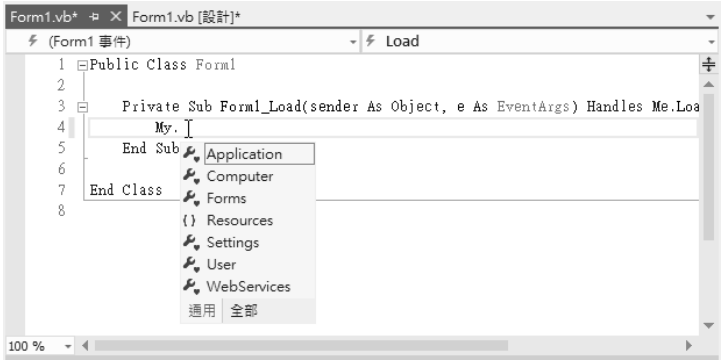
B-5-1 My 命名空間的基礎

如同 Windows 作業系統桌面上的捷徑，其主要目的是幫助使用者快速找到常用的應用程式和檔案。Visual Basic 語言的 My 命名空間，就是存取.NET Framework 類別函數庫的捷徑，可以讓我們快速取得常用類別來建立 Visual Basic 應用程式。

在 My 命名空間的第一層物件，可以讓我們輕鬆存取網路設定、磁碟檔案、應用程式到 Windows 登錄資料等各種功能。其說明如下表所示：

物件	說明
My.Computer	提供目前電腦上的資訊，包含電腦本身、網路連線資訊、存取檔案系統、滑鼠和鍵盤狀態、播放音效、使用 Windows 剪貼簿和存取登入資料等
My.Application	取得目前執行應用程式或 DLL 相關聯的資料，包含版本、執行路徑、語系和命令列參數等
My.Forms	能夠存取目前專案所宣告的每一個 Windows Form 表單實例，換句話說，不用宣告表單物件變數，就可以存取指定表單的屬性
My.User	提供目前使用者相關資訊，可以檢查使用者帳號或所屬群組等
My.WebServices	直接存取目前專案所有 Web 參考的實例，只需透過 My.WebServices 物件的屬性，就可以存取這些 Web 服務
My.Settings	存取應用程式的相關設定，即存取 XML 格式的組態檔
My.Resources	提供應用程式資源的存取功能，通常是指本地化字串、圖形或音效檔案等

在 VSE 程式碼編輯器的 IntelliSense 智慧程式碼輸入功能，可以幫助我們瀏覽 My 命名空間的類別架構，快速建立所需的程式碼，如下圖所示：



在程式碼編輯視窗輸入 My 後，再輸入「.」運算子，就會顯示小視窗列出可用的第 1 層物件，選擇後，即可輸入第 2 層、3 層來存取 My 命名空間物件的屬性和方法。

B-5-2 使用 My 命名空間的物件

My 命名空間的物件主要可以分成兩大類；My.Computer、My.Application 和 My.User 是對應.NET Framework 的功能；My.Forms、My.WebServices、My.Settings 和 My.Resources 是針對目前的專案。

取得電腦和應用程式的相關資訊

在 Visual Basic 程式碼可以使用 My.Computer 和 My.Application 物件的屬性，來取得電腦和應用程式的相關資訊，如下所示：

```
txtOutput.Text = "電腦名稱: " & My.Computer.Name &
                 vbNewLine
txtOutput.Text &= "系統時間: " &
            My.Computer.Clock.LocalTime & vbNewLine
txtOutput.Text &= "程式名稱: " &
       My.Application.Info.AssemblyName & vbNewLine
txtOutput.Text &= "程式標題: " &
            My.Application.Info.Title & vbNewLine
txtOutput.Text &= "程式路徑: " &
      My.Application.Info.DirectoryPath & vbNewLine
```

上述程式碼使用 My.Computer.Name 屬性取得電腦名稱，Clock 物件的 LocalTime 屬性取得系統時間。My.Appication.Info 物件的屬性取得應用程式的相關資訊。

文字檔案處理

在 My.Computer.FileSystem 物件的屬性和方法，可以執行檔案與資料夾處理，例如：建立和寫入文字檔案內容，如下所示：

```
My.Computer.FileSystem.WriteAllText("Test.txt",
                "Visual Basic 程式設計", False)
```

上述程式碼的 WriteAllText()方法可以寫入第 2 個參數的字串至第 1 個參數的文字檔案，最後 1 個參數指定是否新增至檔尾，False 為不是。

ReadAllText()方法可以讀取整個文字檔案的內容，如下所示：

```
str = My.Computer.FileSystem.ReadAllText("Test.txt")
```

測試網路連線與播放音效

在 My.Computer.Network 物件的屬性與方法，可以測試連線、上傳和下載檔案。例如：測試網路連線，如下所示：

```
If My.Computer.Network.IsAvailable Then
    txtOutput.Text &= "電腦現在有網路連線..." &
                      vbNewLine
End If
```

上述 If 條件使用 IsAvailable 屬性測試網路連線，有連線傳回 True；否則為 False。至於播放音效，則是使用 My.Computer.Audio 物件的方法，如下所示：

```
My.Computer.Audio.PlaySystemSound(
                  Media.SystemSounds.Beep)
```

上述程式碼使用 PlaySystemSound()方法播放 Windows 的系統音效，參數是 SystemSounds 類別的成員。如果使用 Play()方法，可以播放參數指定的 WAV 音效檔案。

範例專案 AppB-5-2\我的主控台程式

這個 Windows 應用程式是修改第 B-3-2 節的【我的主控台程式】，使用 My 命名空間的物件顯示電腦和應用程式資訊、處理文字檔案、測試連線和播放音效，其執行結果如下圖所示：

表單設計工具

請建立 Windows Form 專案【我的主控台程式】，這是直接複製和貼上第 B-3-2 節範例專案資料夾，其步驟如下所示：

Step 1 請在「方案總管」視窗刪除 Class1.vb 和 Class2.vb 後，按二下 Form1.vb 開啟表單設計工具。

程式碼編輯器

Step 2 切換至程式碼編輯器，可以修改 Form1_Load()事件處理程序。

```
01: Private Sub Form1_Load(sender As Object, e _
                    As EventArgs) Handles Me.Load
02:     ' 電腦和應用程式資訊
03:     txtOutput.Text = "電腦名稱: " & My.Computer.Name &
                      vbNewLine
04:     txtOutput.Text &= "系統時間: " &
                   My.Computer.Clock.LocalTime & vbNewLine
05:     txtOutput.Text &= "程式名稱: " &
               My.Application.Info.AssemblyName & vbNewLine
06:     txtOutput.Text &= "程式標題: " &
                    My.Application.Info.Title & vbNewLine
07:     txtOutput.Text &= "程式路徑: " &
              My.Application.Info.DirectoryPath & vbNewLine
08:     ' 文字檔案處理
09:     Dim str As String
10:     My.Computer.FileSystem.WriteAllText(
                "Test.txt", "Visual Basic 程式設計", False)
11:     str = My.Computer.FileSystem.ReadAllText("Test.txt")
12:     txtOutput.Text &= "Test.txt 檔案內容: " & str &
                      vbNewLine
13:     ' 測試網路連線
14:     If My.Computer.Network.IsAvailable Then
15:         txtOutput.Text &= "電腦現在有網路連線..." &
                          vbNewLine
16:     End If
17:     ' 播放音效
18:     My.Computer.Audio.PlaySystemSound(
                   Media.SystemSounds.Beep)
19:     txtOutput.SelectionStart = 0
20: End Sub
```

程式碼解說

- 第 3~18 列：測試 My 命名空間的物件，主要是 My.Computer 和 My.Application 物件的屬性和方法。