

第 8 章 程式語言與邏輯

最

近在台灣巡迴的「機器人教育博覽會」中，展出了美女機器人、醫療機器人、寵物狗、鐵人 28 號……等，引發大批參觀人潮。這些有趣的機械人已進化到能與人類密切地互動。電影「機械公敵」、「人工智慧」中的情節，說不定很快就會在我們的日常生活中上演喔！

機器人之所以能夠模仿人類的各種動作，主要是因為科學家及工程師們在設計它們時，撰寫了許許多用來指揮機器人動作及接收外界訊號的程式，然後再將這些程式燒錄在記憶晶片上，才能讓機器人越來越像「人」，甚至還能變成「超人」！

我們平常所使用的電腦軟體，多是電腦科學家為解決人類各種不同的問題，而以程式語言撰寫出來的程式。如果你希望自己能指揮電腦解決問題，學好本章的內容，將能讓你從一個「使用」電腦的高手，變成更高竿會「指揮」電腦的專家！

| | | |
|-----|-----------------------|-----|
| 8-1 | 認識演算法 | 260 |
| 8-2 | 程式語言簡介 | 266 |
| 8-3 | Visual Basic 簡介 | 273 |
| 8-4 | 程式基本結構 | 311 |
| 8-5 | 結構化程式設計 | 329 |
| 8-6 | 程式的偵錯與除錯 | 339 |



8-1 認識演算法

自己動手製作蛋糕並不難，只要我們依循食譜記載的步驟來進行，即可烘製出美味可口的蛋糕（如圖 8-1）。

在日常生活中，當我們遇到較複雜的問題時，若也能依照事前規劃的特定步驟來處理，通常比較容易將問題順利解決。

- 1** 依特定的數量比率準備製作蛋糕所需的材料（例如麵粉、雞蛋、糖等）



- 2** 將材料放入容器中並攪拌均勻



- 4** 將模型放進烤箱，並以一定的時間及火候來烘焙



- 3** 將攪拌後的材料倒入蛋糕模型中



圖 8-1 生活中的演算法實例 蛋糕烘製的步驟也可算是一種演算法

8-1.1 演算法簡介

演算法（algorithm）是一組用來解決特定問題的有限指令或步驟。我們可以依循這些指令，在有限的步驟內逐步地解決問題或完成特定的工作。演算法中的每一個步驟都必須非常明確，不可模稜兩可，才不會出現執行結果與預期不一致的情形。

演算法不僅可用來解決數學或科學上的問題，也可應用在日常生活中，以解決特定的問題或完成特定的工作。圖 8-2 為一個計算段考成績總分的演算法實例。



圖 8-2 演算法實例 計算段考成績總分的演算過程

8-1.2 演算法的表示

在設計演算法時，我們常以流程圖表示法與敘述表示法兩種方式來表示處理的步驟，分別說明如下。

流程圖表示法

流程圖是透過簡明的圖示符號來表達解決問題步驟的示意圖。下表是常用的流程圖符號及其所代表的意義。

一點靈

我們通常會先使用流程圖來描繪解決問題的步驟，再參照流程圖的內容來撰寫程式碼。

表 8-1 常用的流程圖符號與說明

| 符號 | 代表意義 | 作用 |
|----|-------|-----------------------|
| | 開始或結束 | 表示流程圖的開始或結束 |
| | 螢幕 | 表示將資料輸出於螢幕上 |
| | 輸入或輸出 | 表示資料的輸入或輸出 |
| | 處理符號 | 表示執行某些工作 |
| | 決策或判斷 | 表示以符號內的條件式作判斷，決定執行的流向 |
| | 流向符號 | 表示程式的執行方向和順序 |
| | 連接符號 | 表示流程圖的出口或入口 |
| | 副程式符號 | 表示執行副程式 |
| | 列印符號 | 表示資料由印表機輸出 |
| | 磁碟符號 | 表示由磁碟輸入或輸出資料 |

圖 8-3 為使用流程圖表示法來描述日常生活中到商場購物及計算段考成績總分的流程。

! 一點靈

流程圖中統一以 T 表示 True (真)；F 為 False (假)。

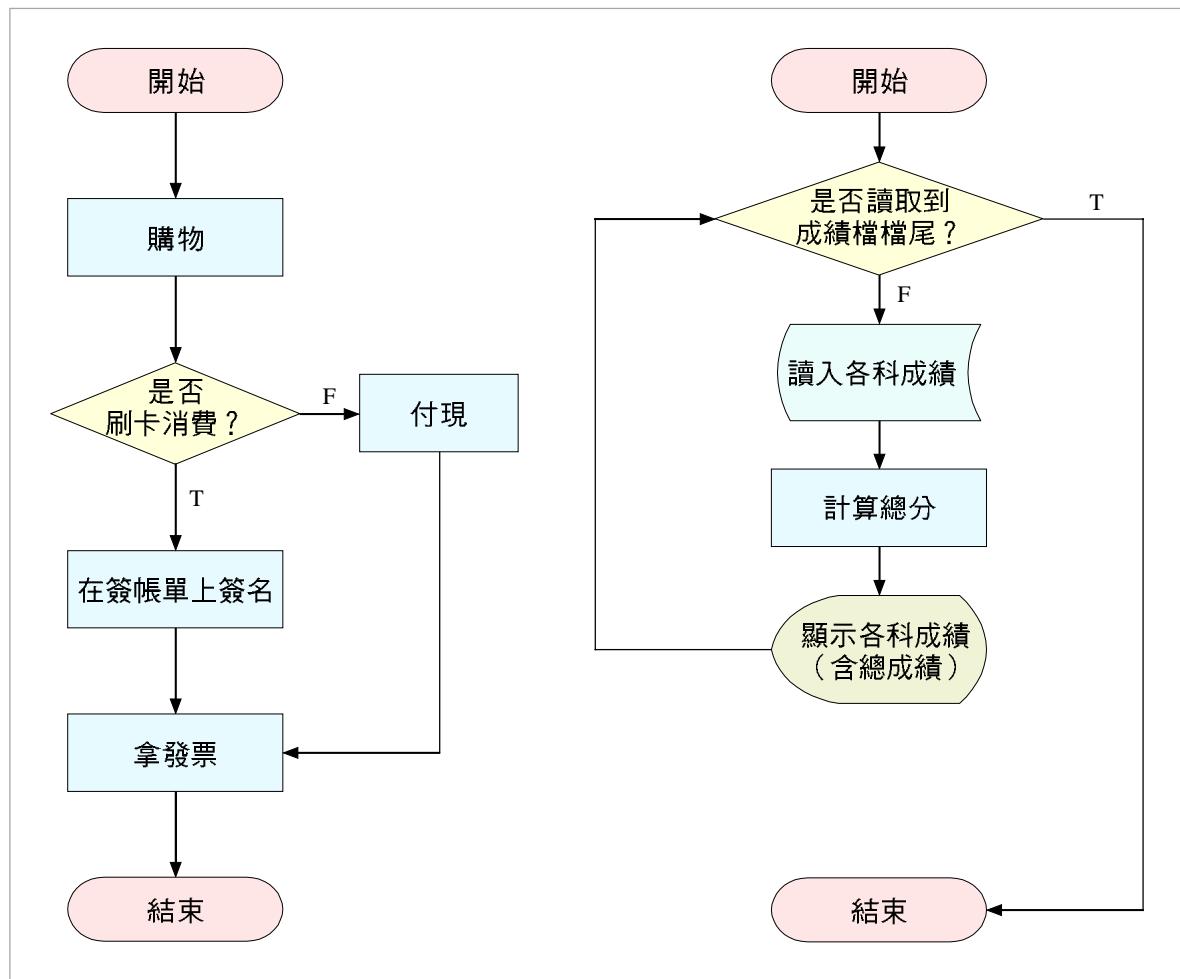


圖 8-3 流程圖表示 到商場購物（左）及計算段考成績總分（右）之流程

敘述表示法

敘述表示法是利用一般語言的敘述方式來表達演算法的處理步驟；我們常使用接近程式語言的格式及語法來描述演算法，這些用來描述處理步驟的敘述稱為**虛擬碼**（pseudo code）。下頁的圖 8-4 是將圖 8-3 中的演算法以敘述表示法來描述處理的步驟。

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. 到商場購物 2. 判斷是否使用信用卡付費？ 如果是，就在簽帳單上簽名；否則就付現金 3. 拿發票離開 | <ol style="list-style-type: none"> 1. 判斷是否讀取到成績資料檔的檔尾，如果是，就離開；否則就執行步驟 2 2. 從磁碟中讀入學生的各科成績 3. 加總各科成績 4. 顯示各科成績及總分，執行步驟 1 |
|---|--|

圖 8-4 敘述表示 到商場購物（左）及計算段考成績總分（右）之敘述表示

8-1.3 演算法的三種結構

演算法包含循序、條件及重複等 3 種基本結構。

- **循序結構** (sequence structure)：是一種由上而下依序執行解決問題步驟的控制結構。其執行流程如圖 8-5(a)所示。
- **條件結構** (selection structure)：是一種依照特定的條件或測試的結果，來決定不同執行路徑的控制結構。其執行流程如圖 8-5(b)所示。
- **重複結構** (repetition structure)：是一種反覆執行解決問題的步驟，直到特定條件出現才停止執行的控制結構。其執行流程如圖 8-5(c)所示。

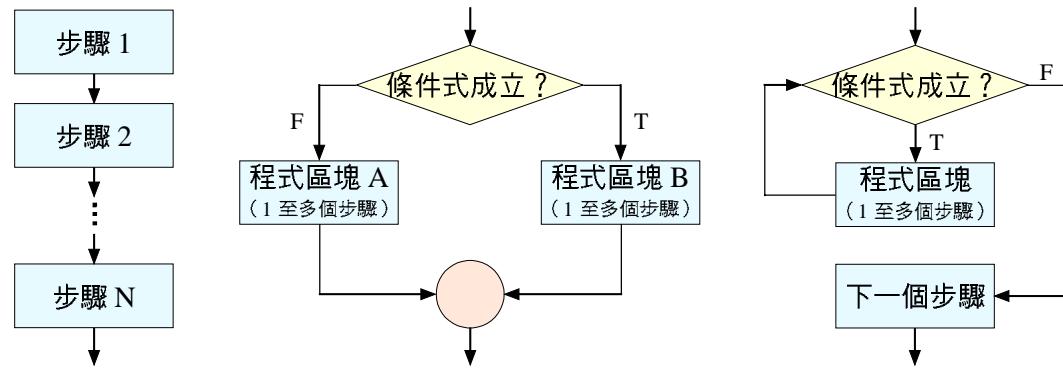


圖 8-5 演算法的控制結構 演算法是由循序、條件及重複等 3 種結構組成，這 3 種結構就是程式設計的基本結構

8-1.4 演算法與程式設計的關係

一個優良的演算法可幫助我們在有限的步驟內，以最有效率的方法來解決問題。因此，在設計程式來解決某項問題之前，我們通常會先使用流程圖或虛擬碼來設計解決問題的演算法，再參照演算法的內容來撰寫程式。

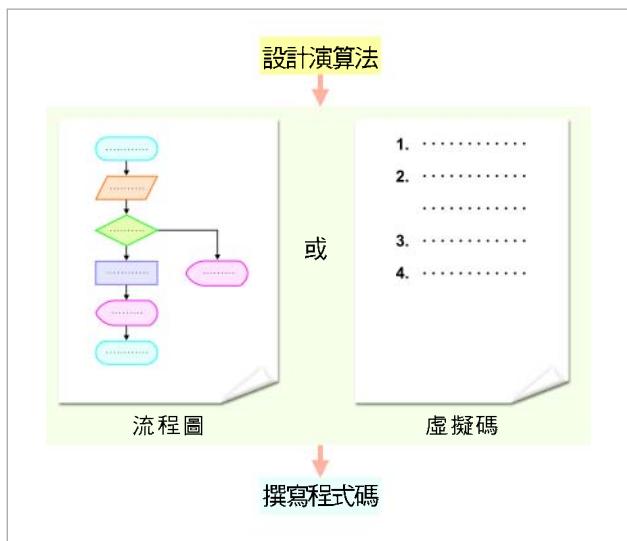
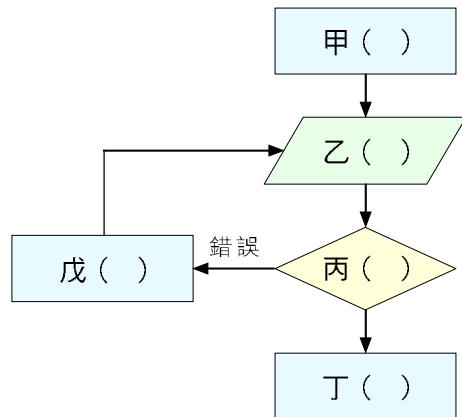


圖 8-6 演算法與程式設計的關係

隨堂練習

- () 1. 透過一些圖形符號來表示處理問題步驟的示意圖稱為 (A)虛擬碼 (B)演算法 (C)流程圖 (D)卡諾圖。
- () 2. 在流程圖的符號中，輸入或輸出的圖形為下列何者？ (A)圓形 (B)菱形 (C)橢圓形 (D)平行四邊形。
- () 3. 體育老師要求同學跑操場 5 圈，跑完後才可以休息。試問連續跑操場的過程，可用下列哪一種控制結構表示？ (A)循序 (B)條件 (C)索引 (D)重複。
4. 右圖是判斷輸入之密碼是否正確的演算法，請配合流程圖上的甲乙丙丁戊，填入正確的步驟代碼，步驟代碼依序為：
 ①判斷密碼是否正確
 ②輸入密碼
 ③提示輸入訊息
 ④提示錯誤訊息
 ⑤進入下一程序



8-2 程式語言簡介

有部感人的電影「再見了，可魯」，劇中的盲人渡邊先生必須學習及使用可魯聽得懂的語言（受訓時用的簡易英文單字），才能順利指揮「牠」。

在電腦世界中，我們要命令電腦從事各項資料處理的工作時，也必須使用電腦可以接受的特定語言來下達指令，才能指揮「它」運作。這種具有特定格式的語言稱為**程式語言**（programming language）。

8-2.1 程式語言的類別

程式語言若以其接近人類語言的程度來區分，可分為**低階語言**與**高階語言**兩大類（圖 8-7）。低階語言與人類所用的語言差異極大；而高階語言則與人類所使用的語言比較接近，如圖 8-8 所示。



圖 8-7 程式語言的分類 以接近人類語言的程度來區分，可分為低階語言與高階語言兩大類

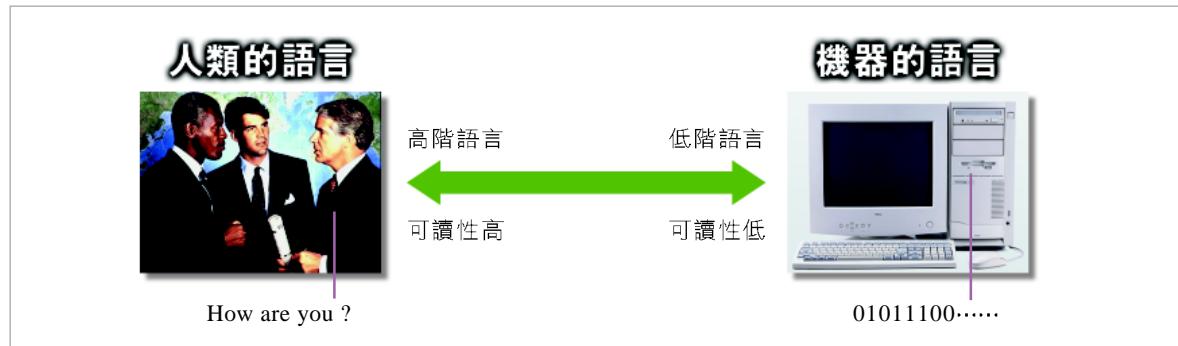


圖 8-8 低階語言與高階語言的區分 使用高階語言所撰寫的程式敘述較接近人類的語句，因此可讀性較高，低階語言則相反

註：高階語言除了圖 8-7 所列的程序性與物件導向語言之外，還有許多其它的語言，如專門用來開發人工智慧軟體的人工智慧語言（如：**Scheme**、**Prolog**）；專門用來查詢資料庫內容之查詢語言（如：**SQL**）……等。

低階語言

低階語言（low-level language）是在電腦發展初期就開發出來的程式語言，這種語言具有**機器依存**（machine-dependent）的限制，意即能在某一種廠牌的CPU上執行的程式，通常無法在另一種不同廠牌的CPU上執行。

軟體開發廠商已經很少使用低階語言來開發應用程式，但是由於低階語言與電腦硬體元件的運作關聯性較為密切，因此製造硬體設備的廠商，仍會使用這種語言來開發控制電腦硬體設備運作的驅動程式等軟體。低階語言可分為**機器語言**與**組合語言**兩種，分別說明如下。

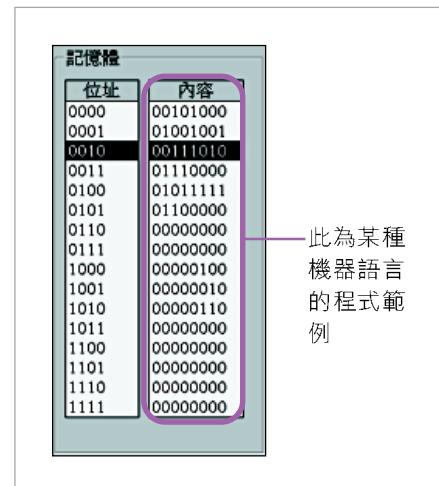
機器語言

機器語言（machine language）是唯一不需翻譯，電腦就能直接執行的語言。這種語言的指令是以0與1兩種符號組合而成（圖8-9），在學習上極為不容易。由於每種電腦的機器語言都是依據該電腦的特性所設計出來，不同CPU的電腦所使用的機器語言也不相同，因此即使學會了一種機器語言，也無法直接使用到其它廠牌的CPU上。

不過，由於機器語言的指令不需經過翻譯，電腦即能執行，可省去程式翻譯所需花用的時間及佔用的記憶體。因此利用機器語言所撰寫出來的程式，具有佔用記憶體較少、執行速度較快的特性。

組合語言

組合語言（assembly language）是一種使用接近人類語言的字組，來代替以0與1表示的機器語言符號，例如以"ADD"、"SUB"、"MUL"、"DIV"來代表算術運算中的加、減、乘、除。這種類似英文簡寫的字組，在電腦科學中稱為**助憶碼**（mnemonic code），可使程式的設計比機器語言來得容易，可讀性也較高。



| 位址 | 內容 |
|------|----------|
| 0000 | 00101000 |
| 0001 | 01001001 |
| 0010 | 00111010 |
| 0011 | 01110000 |
| 0100 | 01011111 |
| 0101 | 01100000 |
| 0110 | 00000000 |
| 0111 | 00000000 |
| 1000 | 00000100 |
| 1001 | 00000010 |
| 1010 | 00000110 |
| 1011 | 00000000 |
| 1100 | 00000000 |
| 1101 | 00000000 |
| 1110 | 00000000 |
| 1111 | 00000000 |

圖8-9 機器語言的程式範例 以0與1二進位方式編碼

但以組合語言撰寫的程式，必須經過**組譯器**（assembler）翻譯成機器語言之後，電腦才能執行。圖 8-10 為組合語言的範例。

| | | |
|-----------|------|------|
| 126E:0000 | LDA | 1000 |
| 126E:0001 | SUB | 1001 |
| 126E:0010 | ADD | 1010 |
| 126E:0011 | PSC | 0000 |
| 126E:0100 | STA | 1111 |
| 126E:0101 | HTL | 0000 |
| 126E:0110 | ES : | 0000 |

圖 8-10 組合語言範例 第 1、2、3 欄分別表示為記憶體位址、指令及運算的資料

高階語言

高階語言（high-level language）是一種比較接近人類語言的程式語言，這種程式語言會將許多常用的低階語言指令組合並隱含在一個程式敘述（statement）中，因此當我們使用一個高階語言的程式敘述來指揮電腦工作時，往往就相當於下達了許多個機器語言指令來要求電腦工作，如圖 8-11 所示。

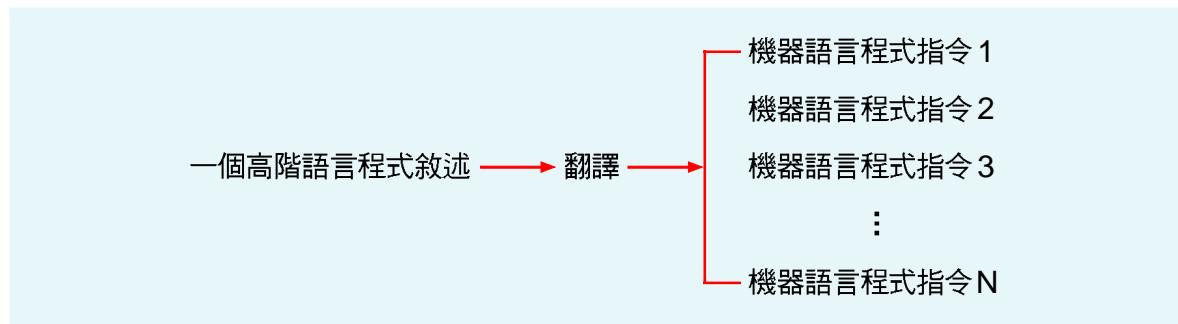


圖 8-11 高階語言程式敘述的示意圖 每一個高階語言的程式敘述是由多個機器語言指令組成

由於電腦只能執行機器語言指令，所以使用高階語言撰寫的程式，必須經過**編譯器**（compiler）或**直譯器**（interpreter）等軟體翻譯成機器語言指令後，電腦才能執行。下頁表 8-2 為高階語言與低階語言的比較。

表 8-2 高階語言與低階語言的比較

| 項目 | 高階語言 | 低階語言 |
|------------------|------|------|
| 程式的撰寫 | 較容易 | 較困難 |
| 佔用記憶體的空間 | 較大 | 較小 |
| 可讀性 | 較高 | 較低 |
| 可攜性 ^註 | 較高 | 較低 |
| 除錯 | 較容易 | 較困難 |
| 維護 | 較容易 | 較困難 |
| 學習難易度 | 較容易 | 較困難 |

程序性語言

程序性語言（procedural language）是一種依照程式敘述的先後順序及流程來執行的程式語言，程式設計者在使用這種語言設計程式時，必須利用安排好順序的程式敘述來告知電腦所要完成工作的流程。常見的程序性語言如下表所示：

表 8-3 常見的程序性語言

| 程式語言 | 適用的領域 | 簡介 |
|---------|---------------|--|
| FORTRAN | 科學及工程計算 | 1. 針對科學及工程計算用途所發展出來的語言 2. 具有運算速度快及準確度高的特色 |
| COBOL | 商業資料處理 | 1. 具有完善的檔案處理能力 2. 可處理大量的商業資料及製作各種商業報表 |
| BASIC | 教學使用 | 1. 具有易學易用的優點，適合初學者使用 2. 後來陸續發展出多種不同的版本，例如 GWBASIC 、 QBASIC 、 Visual Basic 等 3. Visual Basic 除適用於教學之外，也適用於一般視窗應用程式的開發 |
| Pascal | 教學使用 | 1. 為紀念法國數學家 Blaise Pascal 而命名 2. 具有結構化程式設計的概念 |
| C | 系統軟體及一般應用程式開發 | 1. 兼具組合語言與高階語言的特性 2. 可攜性佳，可在不同的電腦系統上執行 3. UNIX 作業系統核心部分大都是使用 C 語言所開發出來的 |

註：使用某種語言所撰寫的程式若無法在不同的電腦系統上執行，我們稱該種程式語言的可攜性低，反之則稱其可攜性高。

物件導向語言

物件導向語言（object-oriented language）是一種以設計個別物件功能的方式來開發程式的語言。使用這種語言來設計程式時，通常必須透過定義類別（class）及設計個別物件（object），來開發解決問題的程式。下面是有關類別及物件的說明。

- **類別**：從字面上來說，類別是指某些具有相同特性的物件集合，例如圖 8-12 中操場施工工人的卡車、訓導主任的跑車、校長的轎車……等不同的車輛物件，都具有可載人（物）、有輪子、可轉彎……等相同的特性，這些車輛物件都可歸屬為「車子」這個類別。在物件導向程式語言中，類別可視為某類物件的樣板，它定義了這類物件所具有的共同特性。

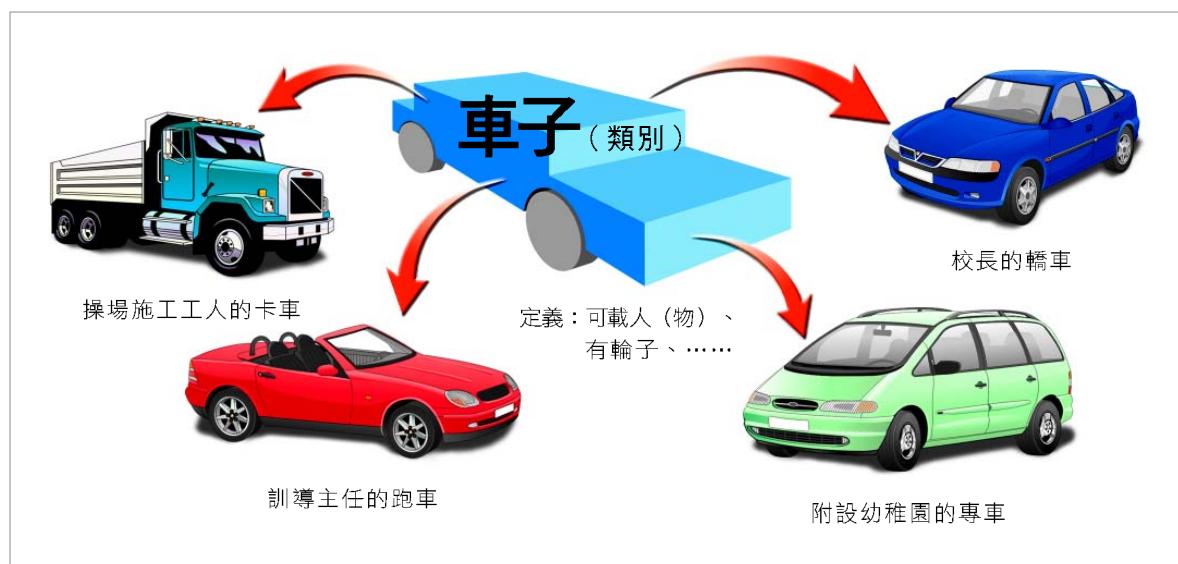


圖 8-12 類別的比喻說明 類別可視為某類物件的樣板，它定義了這類物件所具有的共同特性

- **物件**：是指具有特定屬性、方法及事件的個體，例如汽車具有長、寬、高、顏色等「屬性」；具有換檔變速、煞車、轉彎……等「方法」；還有爆胎、撞擊、下雨等可能發生的「事件」，如下頁圖 8-13。



圖 8-13 物件的比喻說明 物件是指具有特定屬性、方法及事件的個體。如圖中的跑車，它有車的「屬性」、開車的「方法」及可能發生的「事件」。

常見的物件導向語言有 C++、Java、Visual Basic .NET 等，這些語言皆適用於一般應用程式的開發、視窗應用程式的開發……等領域，有關這 3 種語言的簡介列表說明如下：

表 8-4 常見的物件導向語言

| 程式語言 | 簡介 |
|-------------------|---|
| C++ | 1. 以 C 語言為基礎，加入物件導向的特性 2. 提供視窗介面的程式設計環境 3. 具有高度的可攜性及重複使用性 |
| Java | 1. 以 C++ 為基礎簡化而成 2. 具有可攜性高及安全性佳的優點 |
| Visual Basic .NET | 1. 以 Visual Basic 為基礎，加強物件導向、網頁製作、多媒體等功能 2. 提供視覺化的程式設計環境 |

隨堂練習

- () 1. 若要在電腦中執行利用組合語言所撰寫的程式，必須利用下列何者，將組合語言轉換為機器語言？ (A)組譯器 (B)直譯器 (C)編譯器 (D)翻譯器。
- () 2. 下列哪一個程式語言具有「物件導向」的相關特性？ (A)COBOL (B) Visual Basic. NET (C)FORTRAN (D)BASIC。
- () 3. 在物件導向的觀念中，下列敘述何者可用來表示某類別之屬性？ (A)電視在播映 DVD 影片 (B)電腦在編譯 VB 程式 (C)電鍋使用 110V 伏特電壓 (D)電子雞在唱 KTV。

8-2.2 程式的翻譯

每一個國家都有自己的語言，當使用不同語言的兩國人民要交談時，就必須透過翻譯人員的翻譯，才能順利地進行溝通。

在電腦的世界中，電腦只能識別機器語言，當我們（人）使用高階語言來撰寫程式時，也要先透過翻譯程式將高階語言轉換成機器語言，才能讓電腦識別及執行。高階語言的翻譯方式有**直譯**與**編譯**兩種，分別說明如下：

編譯

編譯是將使用高階語言所撰寫的整個程式**原始碼**（source code）翻譯成**目的碼**（object code），但不會立即執行的一種程式翻譯方式。目的碼是一種機器語言格式的程式碼，在經過使用**連結程式**（linker）連結與該程式相關的檔案（例如：程式中所用到的函數、副程式等），即可產生**可執行的目的碼**（executable object code）。圖 8-14 是利用編譯器翻譯的流程。

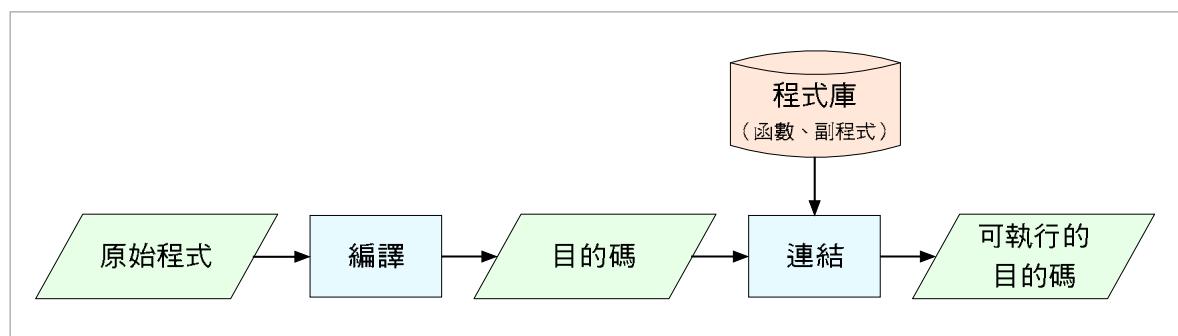


圖 8-14 利用編譯器翻譯的流程 將整個程式翻譯成機器語言程式，並連結與程式相關的檔案

當我們要執行程式時，尚需透過作業系統的**載入程式**（loader），將可執行的目的碼載入到主記憶體內，程式才能開始執行，如圖 8-15 所示。

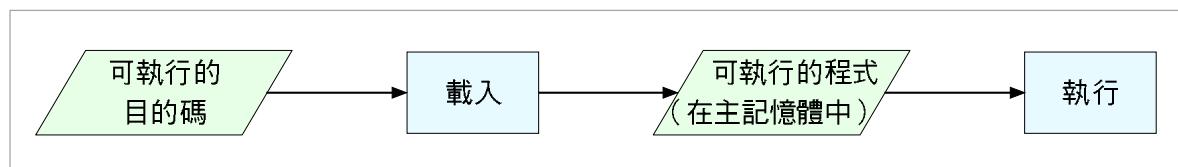


圖 8-15 使用編譯器翻譯高階語言程式的執行流程 需透過載入程序才能執行

直譯

直譯是將使用高階語言所撰寫的程式原始碼（source code），逐行翻譯成機器語言指令，並立即執行的一種程式翻譯方式，這種程式翻譯方式須使用**直譯器**（又稱**直譯程式**）來進行程式翻譯的工作^註。直譯的方式可讓程式設計者逐行檢查程式執行的結果，適合在程式開發階段的偵錯與測試使用，由於此種翻譯方式每次要執行時都必須重新翻譯，因此程式執行的速度相對地較慢。圖 8-16 是利用直譯器翻譯的流程。

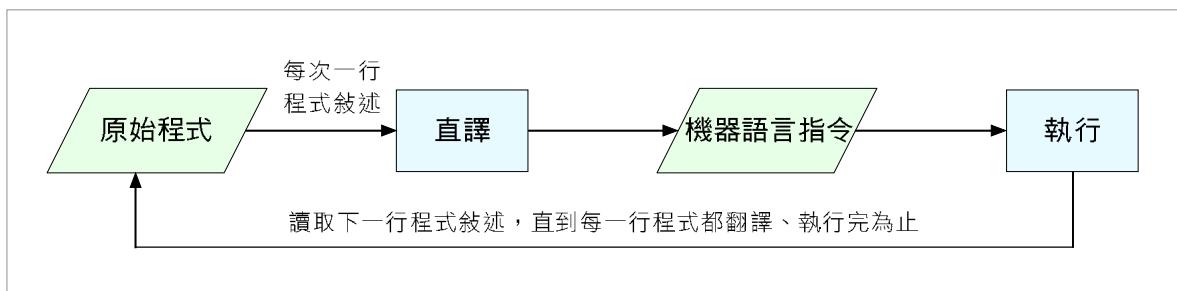


圖 8-16 利用直譯器翻譯的流程 逐行翻譯成機器語言指令並立即執行

8-3 Visual Basic 簡介

Visual Basic（簡稱 VB）是美國微軟公司以 BASIC 語言為基礎，再加入許多視窗程式所須使用的圖形介面工具及程式開發工具，於 1991 年所開發出來的程式語言。Visual Basic 提供圖形化的操作介面，讓使用者可以在這些介面上，以點選按鈕或圖示（icon）的方式來進行程式設計的工作。本章將以 Visual Basic 6.0 來介紹程式設計的應用與概念。

註：為了改善每次執行程式時都需重新翻譯程式的缺點，有些直譯器採先將程式原始碼翻譯成中間碼（intermediate code），待要執行程式時，再進行解譯的工作。

8-3.1 工作環境

Visual Basic 的工作環境包含了主視窗、工具箱、表單視窗、專案總管視窗、屬性視窗、表單配置視窗等主要區域，如圖 8-17 所示。

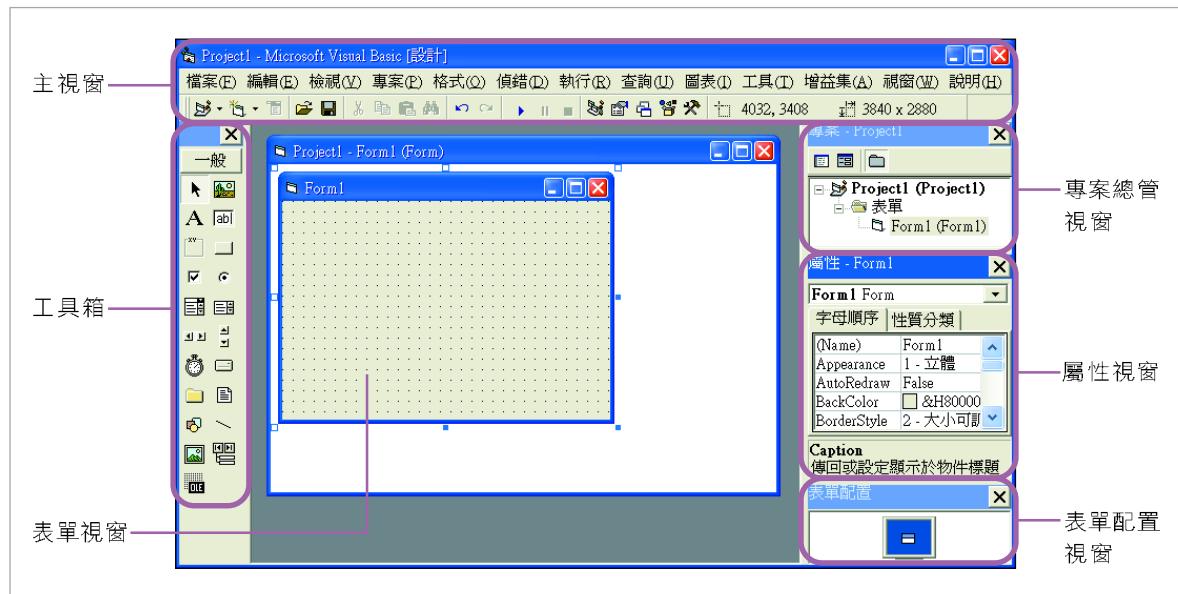


圖 8-17 Visual Basic 的工作環境 選按『開始 / 所有程式 / Microsoft Visual Basic 6.0/Microsoft Visual Basic 6.0』選項，即可啟動 Visual Basic 程式

- **主視窗**：是由**標題列**、**功能表列**及**工具列**所組成，其中**標題列**會顯示程式專案的名稱與工作模式（如圖 8-18 所示）。

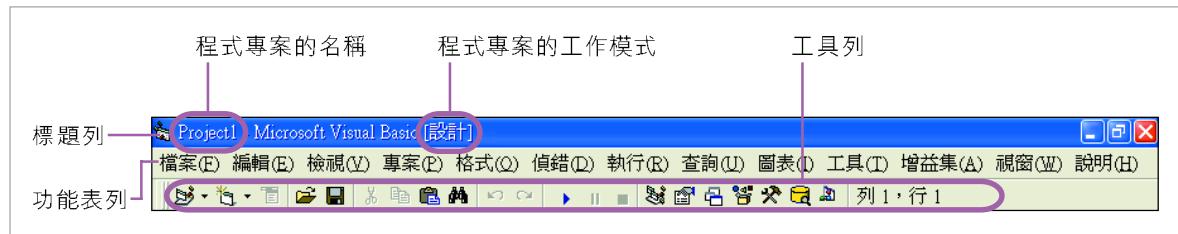


圖 8-18 Visual Basic 的主視窗 Visual Basic 主視窗和微軟Office軟體的操作介面很相似，都具有標題列、功能表列及工具列等視窗元件

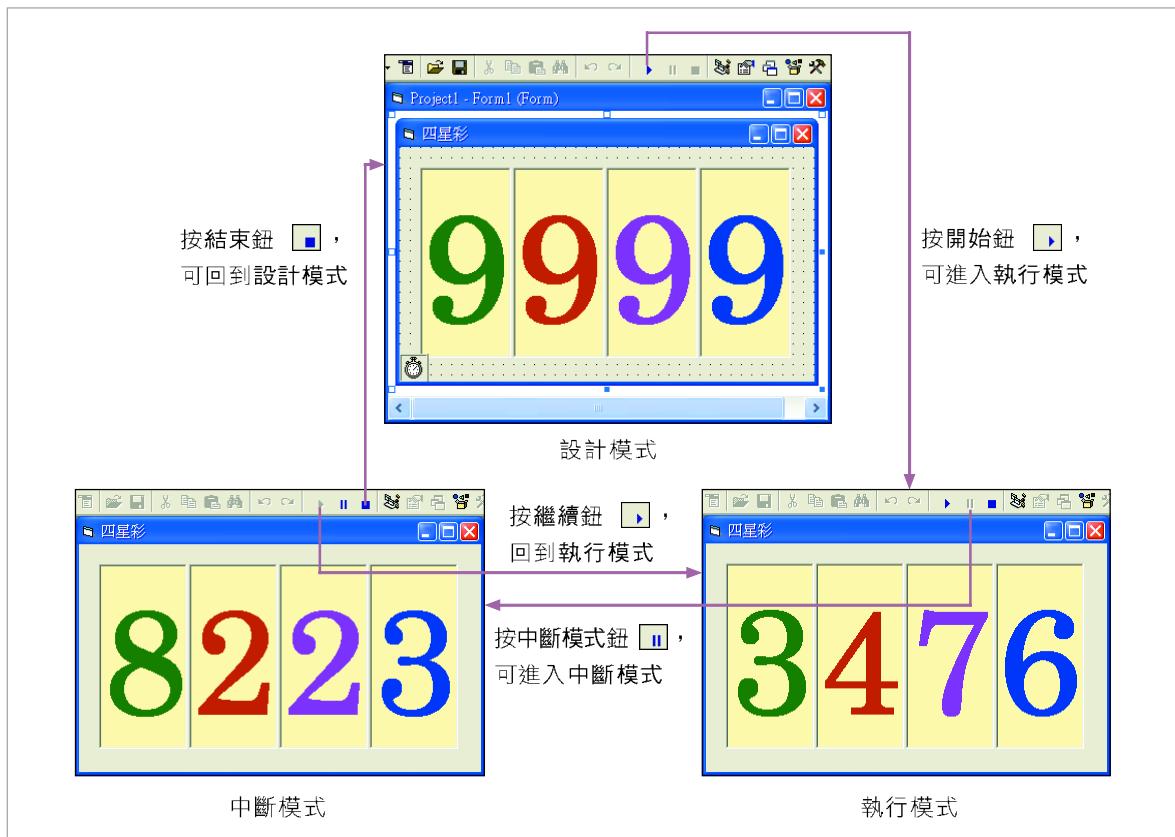
- **工具箱**：預設有 20 種用來開發視窗應用程式的控制物件按鈕。
- **表單視窗**：用來佈建控制物件的平台，常被設計用來作為視窗程式的主畫面或交談窗。

- 專案總管視窗：用來管理程式專案下的所有表單及程式。
- 屬性視窗：用來瀏覽及設定控制物件的屬性（如顏色、大小、字型等）。
- 表單配置視窗：用來顯示及設定程式執行時，表單相對於桌面的位置。

程式專案的工作模式

在 Visual Basic 中，程式專案的工作模式可分為設計、中斷及執行等 3 種模式。圖 8-19 為這 3 種工作模式切換的關係圖。

- 設計模式：可進行程式的設計與編修。
- 執行模式：可執行程式。
- 中斷模式：可暫時中斷程式的執行，以檢視程式的執行情形。



(可開啟範例檔案 "Ch08-3a.vbp")

圖 8-19 工作模式切換的關係圖 在設計模式下，表單上會顯示許多黑色控點，這些控點可用來幫助程式設計者對齊所佈建之控制物件的位置

隨堂練習

- () 1. 在 Visual Basic 的工作環境中，下列哪一個區域會顯示程式專案的工作模式？ (A)工具箱 (B)工具列 (C)狀態列 (D)標題列。
- () 2. 當撰寫好 Visual Basic 程式碼後，可按下列哪一個按鈕來執行程式？ (A)開始鈕 (B)中斷模式鈕 (C)結束鈕 (D)檢視程式碼鈕。
- () 3. 在 Visual Basic 的哪一種工作模式下，可佈建控制物件及撰寫程式碼？ (A)中斷模式 (B)設計模式 (C)執行模式 (D)標準模式。

8-3.2 程式的組成元素

Visual Basic 的程式敘述是由常數 (constant)、變數 (variable)、保留字 (reserved word)、運算子 (operator)、註解 (comment)……等元素所組成（如圖 8-20 計算圓面積的程式敘述所示），分別說明如下：

! 一點靈

Visual Basic 語言的不同組成元素，在程式敘述中會呈現不同的顏色，例如：
 藍色→保留字（如 Dim）
 黑色→一般文字（如 Command1）
 綠色→註解（如 '圓面積的計算）

```

Private Sub Command1_Click()
    ' 圆面積的計算
    Const pi = 3.14159
    Dim r As Single
    Dim area As Single
    r = Val(Text1.Text)
    area = pi * r ^ 2
    Text2.Text = area
End Sub

```

圖 8-20 計算圓面積的程式碼 每一行程式敘述都必須符合程式語言的語法。常數、變數、保留字、運算子、註解……等是組成每行程式敘述的基本要素

- **常數**：是一種在程式執行過程中，其值不會改變的資料項目。例如上頁圖 8-20 中的圓周率 pi。
- **變數**：是一種在程式執行過程中，資料內容可變動的資料項目。例如上頁圖 8-20 中的半徑 r。
- **保留字**：是程式中具有特別意義的字元組合，例如「Dim」便是一個 Visual Basic 的保留字，用來宣告變數的資料型別。
- **運算子**：是用來處理資料的運算符號，例如「*」運算子表示乘法運算。
- **註解**：是用來作為程式的說明（須以單引號或 "Rem" 為開頭），不會影響程式的執行（如上頁圖 8-20 中的第 2 行敘述）。適當地在程式中加入註解，可使程式較易於閱讀與維護。

8-3.3 資料型別與運算子

不同類型的資料項目其運算方式、儲存方式、佔用記憶體空間的大小等有所不同，因此每一種高階程式語言皆提供有許多不同的資料型別，讓程式設計者可針對程式所需使用的不同類型資料項目，賦予適當的資料型別。本小節將介紹 Visual Basic 的資料型別、常數與變數的宣告及用來進行資料運算的各種不同運算子。

資料型別

你看過生活智慧王的電視節目嗎？在「改造收納到你家」的單元中，收納專家總是會利用不同大小的置物箱，來分類收納屋內各種不同的物品。

好的程式設計者必須扮演類似前述收納專家的角色，依據程式所要處理的資料特性，使用適當資料型別的變數或常數來存放各種不同的資料。

Visual Basic 將資料分成許多種類型，例如：數值（value）、字串（string）、布林（boolean）、日期（date）等。以下將利用提供以直譯方式來翻譯並立即執行單行程式敘述的即時運算視窗^註，來演練這些資料型別的使用。

數值資料型別

在 Visual Basic 中，常用的數值資料型別有整數、長整數、單精度與倍精度等 4 種。不同的數值資料型別，所佔用的記憶體大小可能不同，表 8-5 為這 4 種數值資料型別的說明。

表 8-5 常用的數值資料型別

| 資料型別 | 型別的英文名稱 | 佔用空間 | 可表示的數值範圍 |
|------|---------|-------|--|
| 整數 | Integer | 2 位元組 | -32768 ~ 32767 |
| 長整數 | Long | 4 位元組 | -2147483648 ~ 2147483647 |
| 單精度 | Single | 4 位元組 | 正數：1.4E - 45 ~ 3.4E + 38 負數：-3.4E + 38 ~ -1.4E - 45 |
| 倍精度 | Double | 8 位元組 | 正數：4.94E - 324 ~ 1.79E + 308 負數：-1.79E + 308 ~ -4.94E - 324 |

註：“E”表示科學記號中之 10 的指數；例如 3.4E+38 即 = 3.4×10^{38} 。

字串資料型別

字串資料是由若干個中文字、英文字母、數字或特殊符號所組成。在撰寫程式時，字串前後須使用雙引號（"），來表示字串的範圍，如圖 8-21 所示。

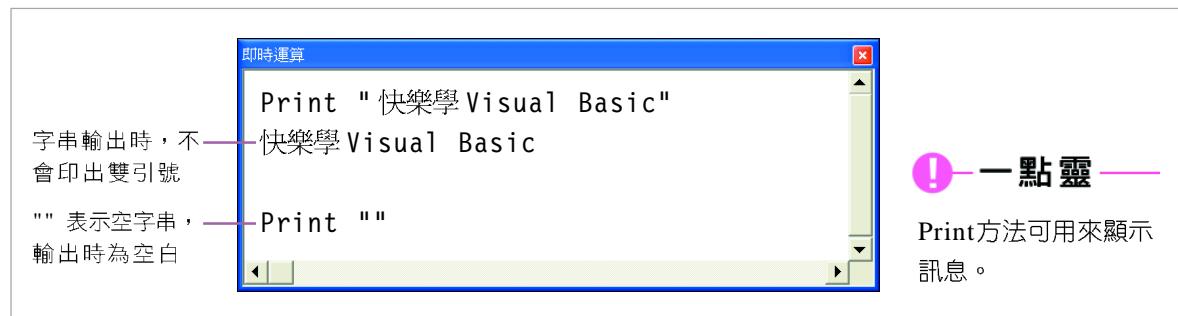


圖 8-21 字串資料表示法 字串前後須使用雙引號來標示出字串資料的範圍

註：選按『檢視 / 即時運算視窗』選項，即可開啟即時運算視窗。

布林資料型別

布林資料通常用來表示條件式的成立與否；True（對，或稱真）表示條件式成立；False（錯，或稱假）表示條件式不成立，如圖 8-22 所示。

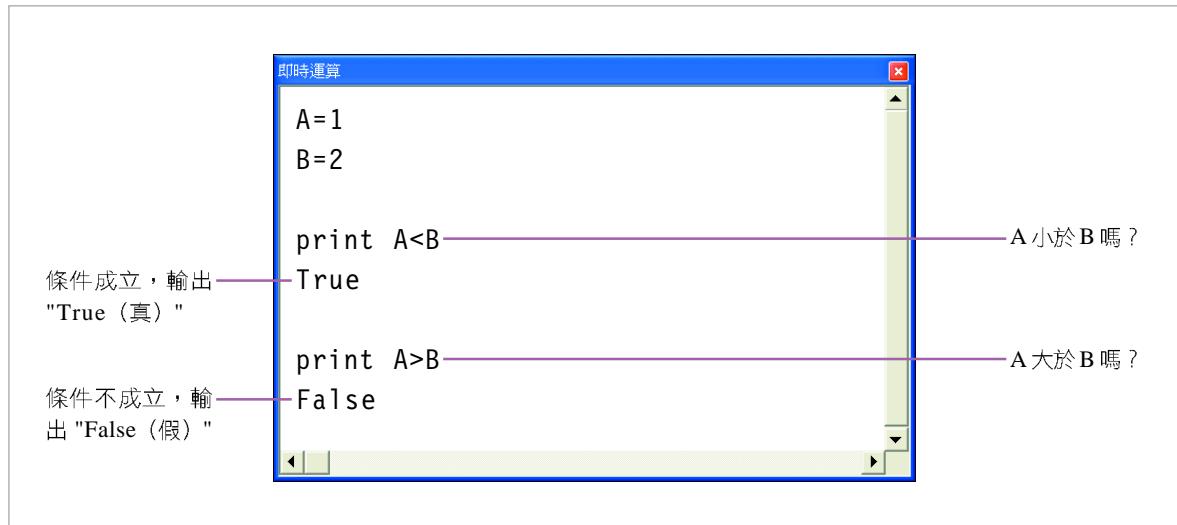


圖 8-22 布林資料的運算 常用來表示條件式的成立與否

日期資料型別

日期資料是用來表示日期與時間的資料型別，在撰寫程式時，日期前後必須以井字符號 (#) 包圍，如圖 8-23 所示。

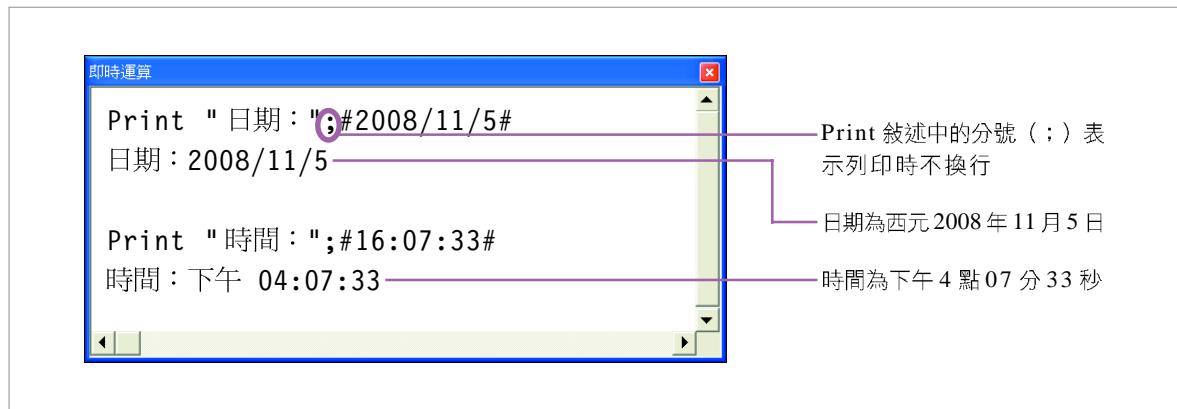


圖 8-23 日期資料的表示法 日期前後須使用井字符號來標示出日期資料的範圍

常數與變數的宣告

常數是一種程式中其值固定的資料；而**變數**則是用來存放資料內容可能會隨著程式執行而改變的資料。

命名規則

在 Visual Basic 中，當我們要宣告某一個常數或變數時，必須遵照以下的命名規則來命名，否則將出現如圖 8-24 所示的警告訊息。

- 長度必須在 255 個字元以內。
- 不可與 Visual Basic 的保留字相同（如：Dim、Print、Exit、For、If 等）。
- 必須以英文字母（大小寫皆可）或中文字開頭，後面則可以是英文字母、中文字、數字或底線。
- 名稱中不能有句點（.）或者是型別宣告字元（!、@、#、\$、%、& 等）。

一點靈

在中文版的VB中，常數或變數名稱除了可以使用英文字母來命名之外，也可以使用中文字來命名。

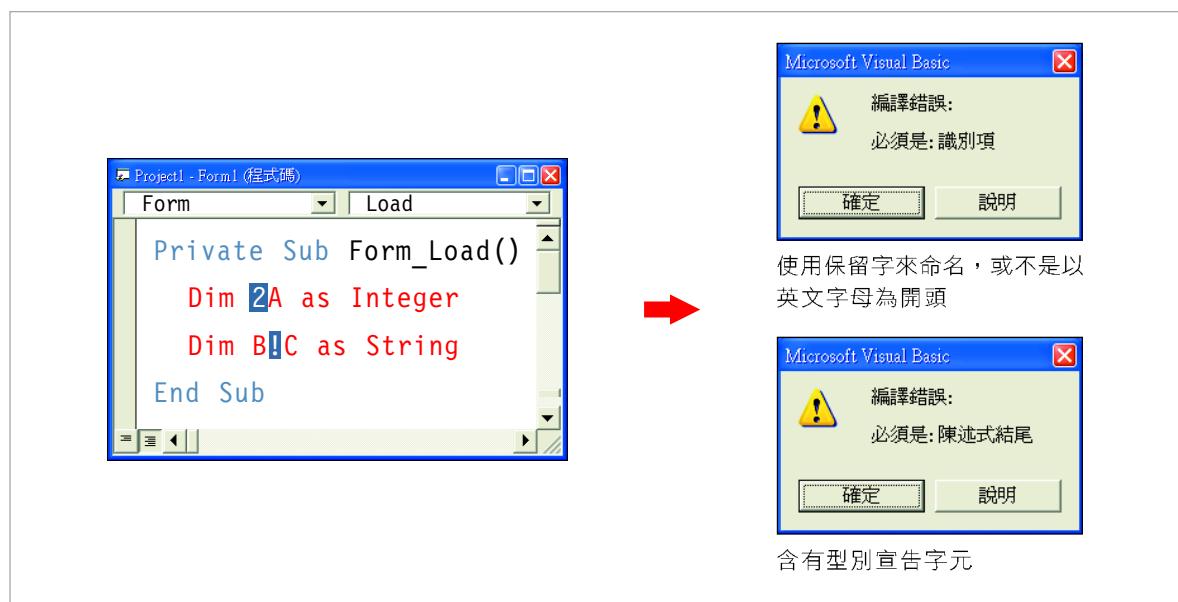


圖 8-24 警告訊息 變數或常數名稱不符合命名規則時，即會出現如上圖所示的警告訊息

表 8-6 列出了幾個 Visual Basic 正確與錯誤變數名稱的例子。

表 8-6 Visual Basic 正確與錯誤的變數名稱例子

| 變數名稱 | 正確 / 錯誤 | 說明 |
|--------|---------|--------------------------|
| HI_BOY | 正確 | |
| year96 | 正確 | |
| 2A | 錯誤 | 不可使用數字作為開頭 |
| B!C | 錯誤 | 不可使用含有資料型別的宣告字元 |
| Print | 錯誤 | Print 為 Visual Basic 保留字 |

宣告語法

程式中所使用到的常數與變數，都應該事先宣告。如果變數沒有宣告資料型別，VB 會將此變數視為可變資料型別（Variant）註。

關於常數與變數的宣告語法說明如下：

- 常數的宣告：

Const 常數名稱 As 資料型別 = 設定值

例 ◀ **Const pi As Single = 3.14** ' 將 pi 宣告為代表圓周率的常數

- 變數的宣告：

Dim 變數名稱 As 資料型別

或

Dim 變數名稱 + 型別宣告符號

例 1 **Dim r As Single** ' 將 r 宣告為單精度型別變數

例 2 **Dim r!** ' 將 r 宣告為單精度型別變數

註：可變資料型別的變數可儲存各種類型的資料，但在使用這類變數運算時，卻容易因變數間的資料型別不同，而產生運算錯誤。

表 8-7 為常用的資料型別說明：

表 8-7 常用的資料型別說明

| 資料型別 | 型別的英文名稱 | 型別宣告符號 | 記憶體佔用空間 |
|------|---------|--------|---------|
| 字串 | String | \$ | 可由使用者指定 |
| 整數 | Integer | % | 2 位元組 |
| 長整數 | Long | & | 4 位元組 |
| 單精度 | Single | ! | 4 位元組 |
| 倍精度 | Double | # | 8 位元組 |
| 日期時間 | Date | 無 | 8 位元組 |
| 布林 | Boolean | 無 | 2 位元組 |
| 可變 | Variant | 無 | 視變數內容而定 |

由上面的說明可得知，變數、常數的名稱及資料型別是由程式設計者透過宣告敘述來指定；而其所佔用的儲存空間，則視宣告的資料型別而定。例如宣告一個整數變數，Visual Basic 會分配 2 位元組的記憶體空間來儲存該變數資料。

! 一點靈

變數經過宣告為數值資料型別後，其初始值（即變數內容）為 0；若宣告為字串型別，則其初始值為空字串。

隨堂練習

- 在 Visual Basic 中，下列哪些是屬於合法的變數名稱？（請以勾選作答）

| | | |
|--------------------------------|--------------------------------|--------------------------------|
| <input type="checkbox"/> Hello | <input type="checkbox"/> SUM_1 | <input type="checkbox"/> 1A |
| <input type="checkbox"/> S% | <input type="checkbox"/> Print | <input type="checkbox"/> Dim |
| <input type="checkbox"/> b-2-c | <input type="checkbox"/> girl | <input type="checkbox"/> g2000 |
- () 在 Visual Basic 中宣告變數 a%，則此變數屬於下列哪一種變數？(A) 整數變數 (B) 字串變數 (C) 單精度變數 (D) 倍精度變數。
- () Visual Basic 敘述 "Dim a As String"，代表要將變數 a 宣告為下列哪一種資料型別？(A) 整數 (B) 字串 (C) 單精度 (D) 倍精度。

運算子

運算子 (operator) 是用來表示資料間運算方式的符號；**運算元** (operand) 是運算子所要運算的資料項目，包含常數、變數等；**運算式** (expression) 則是運算子與運算元的組合，如圖 8-25 所示。

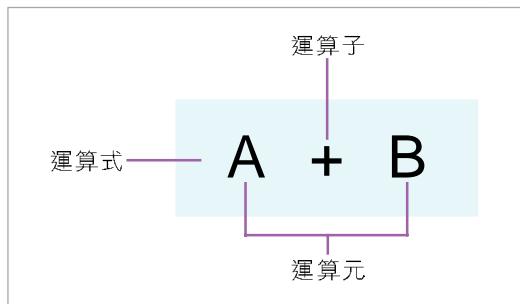


圖 8-25 運算式的組成 由運算元和運算子組成

在 Visual Basic 中，運算子可區分為算術、串接、比較、邏輯等 4 類，其中**比較運算子**與**邏輯運算子**由於多半應用於條件結構的程式敘述中，因此我們將在本章第 4 節介紹條件結構時再作介紹，以下先介紹**算術運算子**與**串接運算子**的用法。

算術運算子

算術運算子是用來執行數值間的運算，運算規則大致與數學上的規定相同，例如：由左至右進行運算的規則、先乘除後加減的運算優先順序等。表 8-8 為 Visual Basic 的 8 種算術運算子說明與運算實例，其中運算優先順序欄中的數字，數字越小表示運算優先順序越高。

表 8-8 各種算術運算子說明與運算實例

| 算術運算子 | 功能說明 | 運算實例 | 運算結果 | 運算優先順序 |
|-------|-----------------------|--------------------|----------|--------|
| $^$ | 計算某數的次方值 | 2^3 | 8 | 1 |
| - | 表示負數 | -6 | -6 | 2 |
| * | 兩數相乘 | $2 * 3$ | 6 | 3 |
| / | 兩數相除，取商數 | $8 / 6$ | 1.333... | 3 |
| \ | 兩數四捨六入成整數後相除，取商數的整數部份 | $8 \backslash 6$ | 1 | 4 |
| Mod | 兩數四捨六入成整數後相除，取餘數 | $8 \text{ Mod } 6$ | 2 | 5 |
| + | 兩數相加 | $1 + 2$ | 3 | 6 |
| - | 兩數相減 | $2 - 1$ | 1 | 6 |

請注意！使用 "\\" 運算子進行整數除法運算時，當除數與被除數是帶有小數的數值，VB 會先將兩數四捨六入後再相除。若除數或被除數小數點後僅有一個數值，且其數值為 5，則會以整數部份是奇數或偶數來作為進位的依據（奇數進位；偶數捨棄）；若小數點後有一個以上的數值，則無條件進位至整數部份。圖 8-26 為利用即時運算視窗顯示各種算術運算子的運算結果。

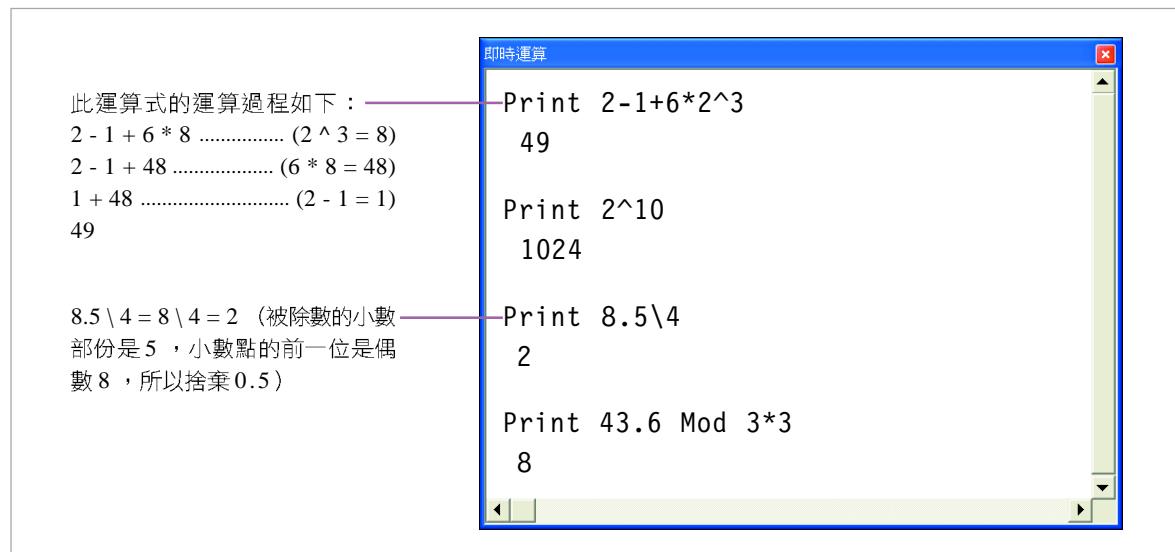


圖 8-26 各種算術運算子的運算 運算規則大致和數學上的規定相同

串接運算子

串接運算子是用來連接多個資料。若要連接同屬字串型別的多個字串資料，可使用串接運算子 "+"；若要連接不同型別的資料，則須使用串接運算子 "&"。圖 8-27 為利用即時運算視窗顯示串接運算子的運算範例。



圖 8-27 串接運算子的運算 "+" 用於字串間的串接，"&" 用於不同資料型別間的資料串接

若使用錯誤的串接運算子來連接資料，將會出現如下圖所示的錯誤訊息。

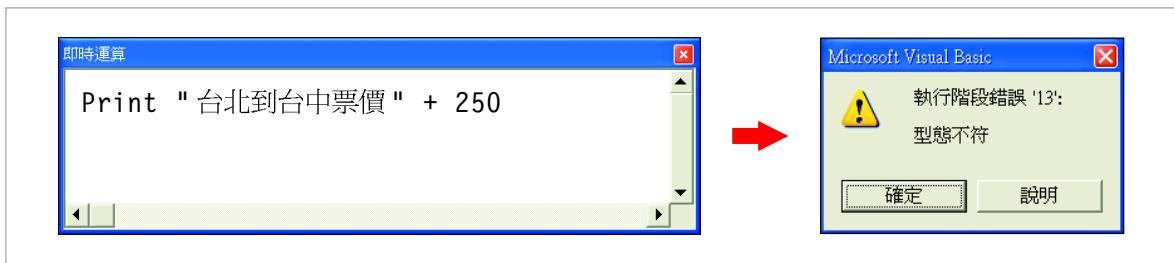


圖 8-28 串接運算子的錯誤用法 "+" 只能用在字串間的串接

"+" 運算子運用在字串與字串間為串接運算，若運用在數值與數值間則為算術運算，兩種運算所使用的符號雖相同，但運算功能不同，必須區分清楚。

隨堂練習

1. 請利用即時運算視窗輸入以下程式，並檢視其執行的結果。

- | | | | |
|--|---|---------------------------------|---|
| (1) Print $3 + 9 ^ 0.5$: _____ | 。 | (4) Print "幸運數字：" & 7 : _____ | 。 |
| (2) Print $64 \backslash 8 - 10$: _____ | 。 | (5) Print "123" + "456" : _____ | 。 |
| (3) Print $7 * 4 / 2 ^ 2$: _____ | 。 | (6) Print $123 + 456$: _____ | 。 |

操作實例 1 計算自己的平均成績

計算自己的平均成績，並判斷是否高於全班的平均成績。

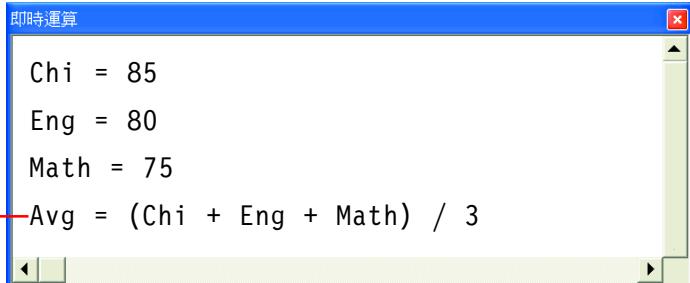
- 1** 在即時運算視窗中輸入：
Chi = 85，Chi 為國文成績，並按 **Enter** 鍵

```
即時運算
Chi = 85
Eng = 80
Math = 75
```

- 2** 輸入：Eng = 80，Eng 為英文成績，並按 **Enter** 鍵

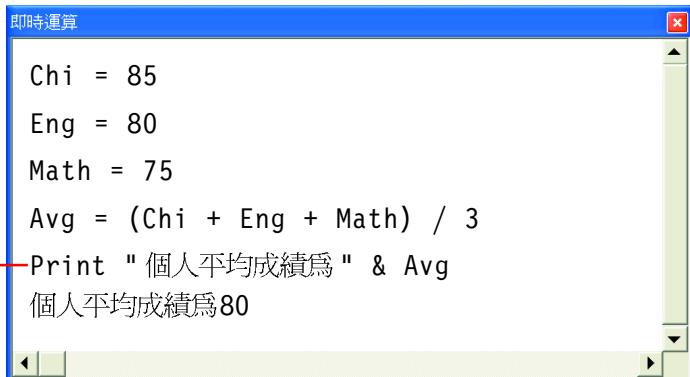
- 3** 輸入：Math = 75，Math 為數學成績，並按 **Enter** 鍵

- 4** 輸入平均成績的計算公式，並按 **Enter** 鍵



```
即时运算
Chi = 85
Eng = 80
Math = 75
Avg = (Chi + Eng + Math) / 3
```

- 5** 輸入：Print "個人平均成績為" & Avg，並按 **Enter** 鍵
，輸出自己的平均成績



```
即时运算
Chi = 85
Eng = 80
Math = 75
Avg = (Chi + Eng + Math) / 3
Print "個人平均成績為" & Avg
個人平均成績為80
```

8-3.4 內建函數

在 Visual Basic 中，函數分為內建函數（built-in function）及自定函數（user-defined function）兩種。其中**內建函數**是 Visual Basic 預先設計好的程式，使用者只要直接輸入函數名稱及參數，即可使用這些函數的功能，而不須自行撰寫程式；**自定函數**則是使用者依照自己的需求，所自行撰寫的函數。

內建函數依其處理的資料類型，可分為**字串函數**、**數值函數**及**時間日期函數**等 3 類，下面將介紹較常使用到的字串函數與數值函數之常用函數，並利用**即時運算視窗**來實際演練。

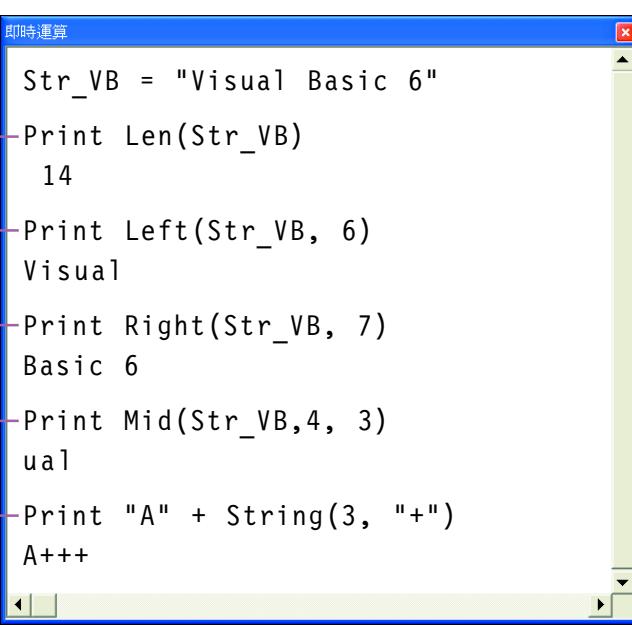
字串函數

字串函數是專門用來處理字串資料的函數，表 8-9 是幾種常用字串函數的說明。

表 8-9 常用的字串函數及語法說明

| 字串函數 | 傳回值說明 | 範例 | 傳回值 |
|---------------|---------------------------------------|--------------------|-------|
| Len(字串) | 字串的字元長度 | Len("Hello") | 5 |
| Left(字串, N) | 由左算起的 N 個字元 | Left("Hello", 2) | He |
| Right(字串, N) | 由右算起的 N 個字元 | Right("Hello", 2) | lo |
| Mid(字串, M, N) | 由左起第 M 個字元算起的 N 個字元 | Mid("Hello", 3, 2) | ll |
| Ltrim(字串) | 去除開頭（左邊）的所有空白字元 | Ltrim(" Hello") | Hello |
| Rtrim(字串) | 去除結尾（右邊）的所有空白字元 | Rtrim("Hello ") | Hello |
| Trim(字串) | 去除開頭及結尾的所有空白字元 | Trim(" Hello ") | Hello |
| String(N, 字元) | N 個字元；若第 2 個參數為字串，則將重複 N 個字串中的第 1 個字元 | String(3, "A") | AAA |
| Space(N) | N 個空白字元 | Space(2) + "Hi" | Hi |
| Str(數值) | 將數值轉換成字串型別的資料 | Str(10) | 10 |

例 使用字串函數計算字串長度、產生子字串及輸出特定字串（圖 8-29）。



The screenshot shows the Immediate window with the following code and output:

```

即時運算
Str_VB = "Visual Basic 6"
Print Len(Str_VB)
14
Print Left(Str_VB, 6)
Visual
Print Right(Str_VB, 7)
Basic 6
Print Mid(Str_VB,4, 3)
ual
Print "A" + String(3, "+")
A+++
```

Annotations from left to right:

- 計算字串變數 Str_VB 的長度
- Print Len(Str_VB)
- 14
- 輸出字串變數 Str_VB 由左算起的 6 個字元
- Print Left(Str_VB, 6)
- Visual
- 輸出字串變數 Str_VB 由右算起的 7 個字元
- Print Right(Str_VB, 7)
- Basic 6
- 輸出字串變數 Str_VB 由第 4 個字元算起的 3 個字元
- Print Mid(Str_VB,4, 3)
- ual
- 輸出字母 "A" 及 3 個加號 (+)
- Print "A" + String(3, "+")
- A+++

圖 8-29 字串函數的使用 計算字串長度、產生子字串及輸出特定字串的運算

數值函數

數值函數是專門用來處理數值資料的函數，下表是幾種常用數值函數的說明。

表 8-10 常用的數值函數及語法說明

| 數值函數 | 傳回值說明 | 範例 | 傳回值 |
|--------------|-----------------------------|----------------|----------------|
| Val(字串) | 將字串轉換成數值型別的資料 | Val("10") | 10 |
| Int(數值) | 小於等於數值的最大整數值 | Int(-3.2) | -4 |
| Fix(數值) | 整數部份 | Fix(4.8) | 4 |
| Sqr(數值) | 平方根 | Sqr(4) | 2 |
| Rnd() | 產生 1 個 ≥ 0 且 < 1 的亂數 | Rnd() | 介於 0 與 1 之間的數值 |
| Round(數值, N) | 將數值四捨五入，並取至小數位數第 N 位 | Round(3.15, 1) | 3.2 |

例 使用數值函數轉換數值型別及計算簡易的數學式（圖 8-30）。

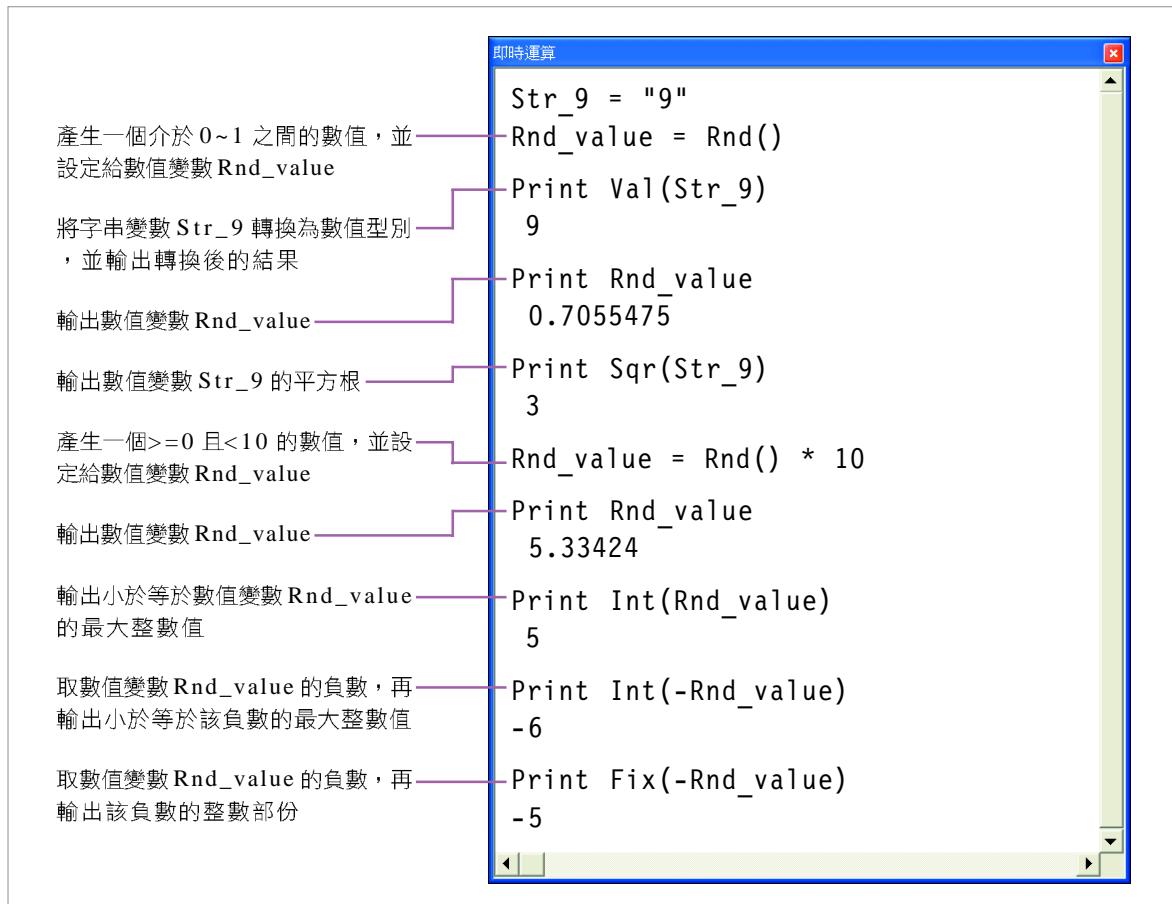


圖 8-30 數值函數的使用 轉換數值型別、產生介於不同範圍的亂數及計算簡易數學式的運算

8-3.5 輸入與輸出

在學習每一種程式語言時，大多需要使用輸入與輸出敘述來輸入資料與輸出程式運算的結果。以下分別介紹 Visual Basic 的 InputBox 函數、Print 方法及 MsgBox 敘述等基本輸出入敘述。

InputBox 函數

InputBox 函數是用來顯示一個供使用者輸入資料的輸入交談窗，其語法如下：

變數 = InputBox (訊息文字[, 標題文字][, 輸入欄位預設值])

使用 InputBox 函數時，會傳回使用者在輸入交談窗中所輸入的資料。標題文字及輸入欄位預設值可省略不寫。

圖 8-31 是利用 InputBox 函數所產生的輸入交談窗，我們可以將 InputBox 函數所傳回的資料在程式中使用。



圖 8-31 InputBox 函數的使用 產生一個讓使用者輸入圖書館帳號的交談窗，按確定鈕後，會把圖中所輸入的資料 "s8806024" 傳回給變數 UserID

Print 方法

Print (輸出) 可用來將訊息顯示在表單上，其語法有 3 種，如下頁表 8-11 所示。

表 8-11 Print 方法的 3 種格式

| 語法 | 例子 | 結果 |
|------------|--------------------------|------------------|
| Print "字串" | Print "快樂學 Visual Basic" | 快樂學 Visual Basic |
| Print 運算式 | Print 1+2 | 3 |
| Print 變數 | Pi = 3.14 Print Pi | 3.14 |

使用 Print 方法時，若以逗號 (,) 來連接多個要輸出的資料，則輸出的資料間會有多個空白字元出現；若以分號 (;) 來連接，則輸出的資料間不會有空白字元。圖 8-32 為在即時運算視窗中，使用 Print 方法搭配逗號及分號的範例。

一點靈

若使用 "?" 來替代 Print 也可以達到相同的效果。

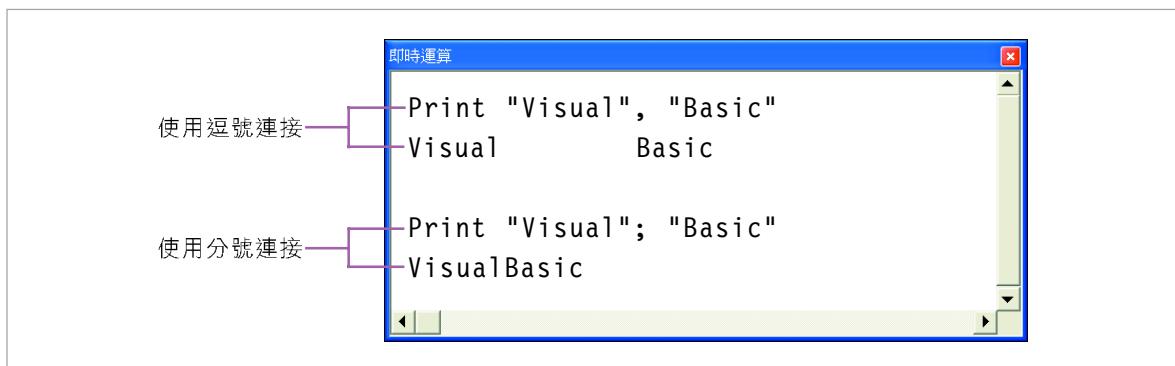


圖 8-32 Print 方法的使用 搭配逗號及分號來連接所要輸出字串的範例

但若分號後的資料為數值型別，則在輸出該資料時，會先列印該數值的正負符號（正數：列印一個空白字元；負數：列印一個 "-" 號），再列印該數值，並在該數值末端多列印一個空白字元（如圖 8-33 所示）。



圖 8-33 Print 方法搭配分號的使用 連接字串及數值資料的差異

MsgBox 敘述

MsgBox 敘述可用來產生一個訊息交談窗，以告知或提醒使用者程式執行的狀態，其語法如下：

MsgBox 訊息文字 [, 訊息圖示設定值][, 回覆鈕設定值][, 標題文字]

訊息圖示及回覆鈕的設定方法有兩種，一種是輸入 VB 內建常數，另一種是輸入代表數值。若要同時設定訊息圖示及回覆鈕等設定值，須以加號（+）連接。表 8-12 及表 8-13 分別是訊息圖示及回覆鈕的常用設定值。

表 8-12 訊息圖示常用設定值

| VB 內建常數 | 代表數值 | 訊息圖示 |
|---------------|------|------|
| vbCritical | 16 | × |
| vbQuestion | 32 | ? |
| vbExclamation | 48 | ! |
| vbInformation | 64 | i |

表 8-13 回覆鈕常用設定值

| VB 內建常數 | 代表數值 | 回覆鈕 |
|---------------|------|--------------|
| vbOKOnly | 0 | 確定 |
| vbOKCancel | 1 | 確定、取消 |
| vbYesNoCancel | 3 | 是(Y)、否(N)、取消 |
| vbYesNo | 4 | 是(Y)、否(N) |

圖 8-34 是利用 MsgBox 敘述所產生的訊息交談窗。

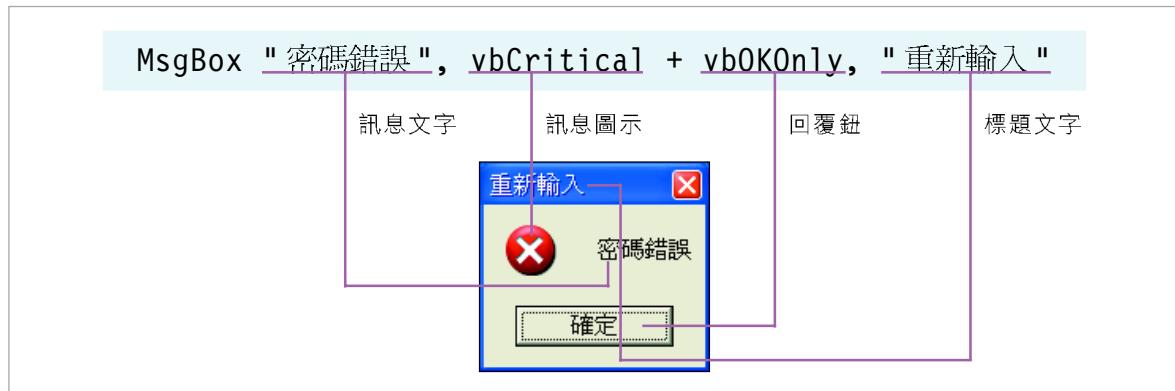


圖 8-34 MsgBox 敘述的使用 產生一個密碼輸入錯誤的訊息交談窗

操作實例 2

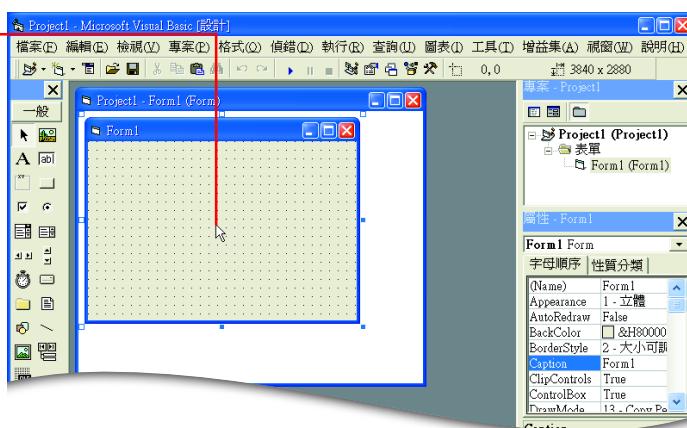
歡迎使用者登入的程式

建立一個新專案，撰寫程式碼，顯示一個供使用者輸入姓名的輸入交談窗，再顯示一個歡迎使用者登入的訊息交談窗。

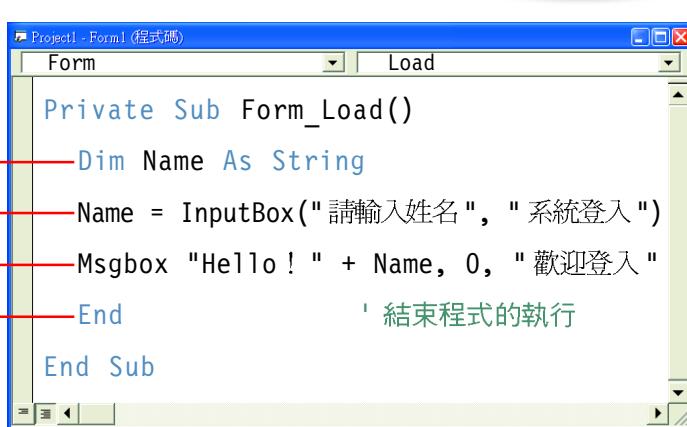
- 選按『開始 / 所有程式 / Microsoft Visual Basic 6.0/Microsoft Visual Basic 6.0』選項，以啟動 Visual Basic 程式



- 點選標準執行檔圖示，按開啟鈕，以建立標準執行檔類型的專案



- 雙按表單，開啟程式碼視窗
- 輸入一行宣告敘述，宣告變數 Name 為字串型別

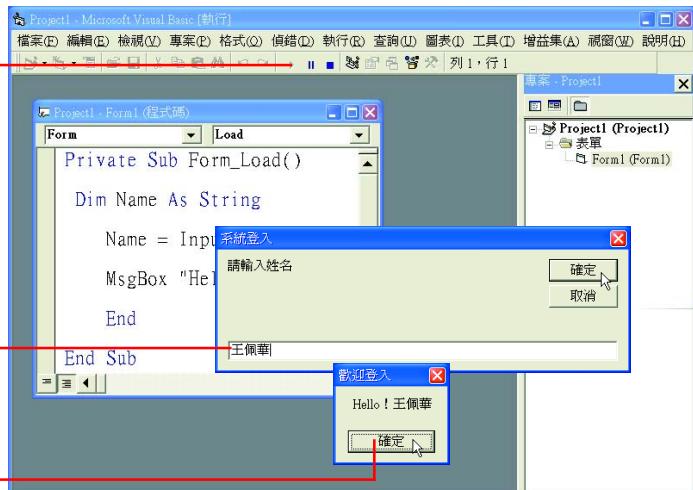


- 輸入一行敘述將 InputBox 函數所傳回的姓名資料，設定給字串變數 Name

- 輸入一行敘述，顯示一個歡迎使用者登入的訊息交談窗

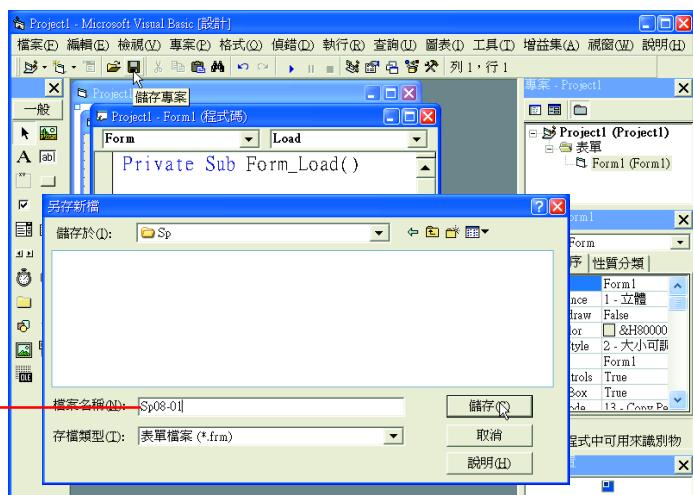
- 輸入End敘述，設定結束程式的執行

8 按開始鈕，執行程式



9 輸入姓名 "王佩華" 並按確定鈕
10 在顯示含有姓名的訊息交談
窗，按確定鈕

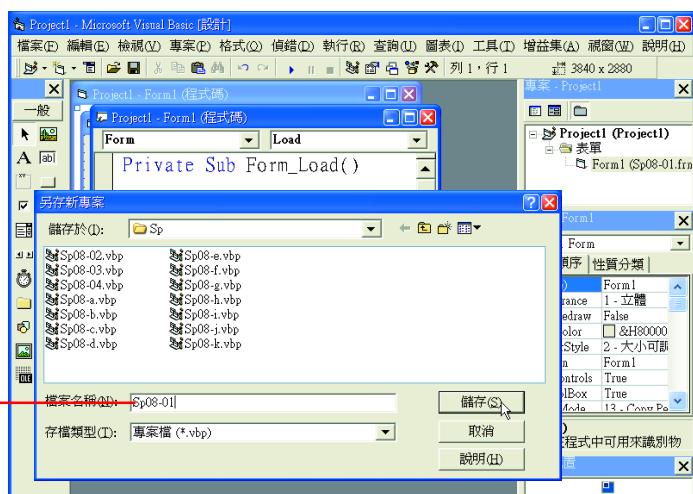
11 按儲存專案鈕 ，在另存
新檔交談窗中輸入表單名稱
"Sp08-01" 並按儲存鈕，儲
存表單



12 接著再輸入專案名稱 "Sp08-
01" 並按儲存鈕，儲存整個
專案

! 一點靈

在Visual Basic中，每一個程式都是以專案的型態來儲存，由於1個專案可包含1個以上的表單，因此在儲存Visual Basic程式時，須分別儲存專案與表單內容。



隨堂練習

- () 1. 執行 Visual Basic 敘述 "Print Minute(Time())"，不可能出現下列哪一個輸出結果？(A)12 (B)28 (C)56 (D)90。
- () 2. 在 Visual Basic 中，使用 Print 來顯示資料時，若不想讓輸出的資料之間列印距離相隔較遠，可使用下列哪一個符號來達成？(A), (B). (C); (D)@。
- () 3. 在 Visual Basic 程式中撰寫：MsgBox "記得來投票 ", 48, " 模特兒選拔 " 敘述，當執行程式時，" 記得來投票 " 將出現在訊息交談窗中的哪一個位置？(A)訊息文字 (B)標題文字 (C)輸入欄的預設值 (D)說明檔代碼。

8-3.6 控制物件

日常生活中的有形物品，例如手機、隨身碟……等都可以稱為物件。在 Visual Basic 中，物件則是指設計程式時所需使用的視窗元件，這些元件一般通稱為**控制物件**（control object，或稱**控制項**），例如 Form（表單）及工具箱中的 CommandButton（按鈕）、Label（標籤）等。我們可以使用這些控制物件來快速完成程式視窗畫面的外觀設計。

控制物件的特徵與功能

Visual Basic 控制物件的特徵與功能是由**屬性**（property）、**方法**（method）及**事件**（event）等 3 種要素所決定，分別說明如下。

屬性

就日常生活中的有形物品而言，屬性是指物品本身所具有的各種性質，例如電視機有不同的廠牌、型號、顏色、大小等屬性。在 Visual Basic 中，屬性是指控制物件所具有的性質或特性。程式設計者可視需要更改控制物件的屬性設定值，以改變控制物件的外觀、大小或樣式。

設定或改變控制物件屬性的方法有 2 種：第 1 種方法是選定控制物件後，在屬性視窗中更改屬性的設定值（如圖 8-35 所示）。

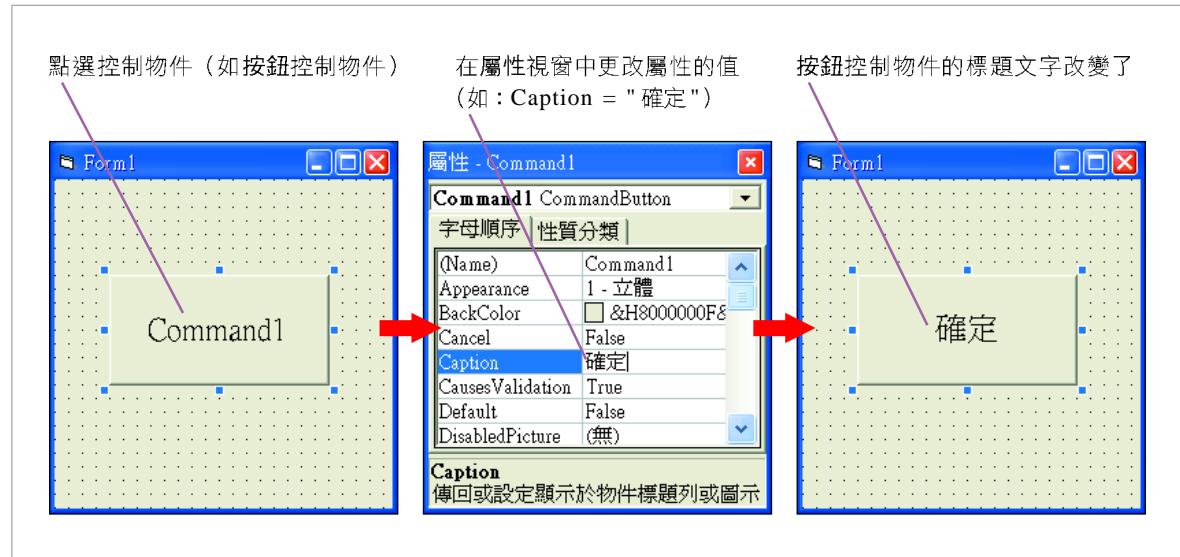


圖 8-35 屬性的設定 利用屬性視窗改變按鈕控制物件的標題屬性

第二種方法則是利用程式敘述來設定控制物件的屬性，其語法如下：

控制物件名稱.控制物件的屬性 = 設定值

例 `Command1.Caption = "確定"`

' 將 Command 控制物件的標題文字設定為 "確定" '

隨堂練習

- 練習在表單上佈建 1 個按鈕控制物件，然後利用屬性視窗將其 Caption 屬性設為 "確定"。
- 延續上題，若要改以程式敘述來設定 Command1 的 Caption 屬性，則程式敘述應如何撰寫？

方法

在日常生活中，方法是指要達到某種目的所必須採取的手段，例如為了將聯絡人的資料儲存到手機中，可以利用輸入的方法，將通訊資料記錄到手機電話簿中。

在 Visual Basic 中，方法是指內建在控制物件中的程序，當程式設計者在程式中使用某一種控制物件的方法時，內建在該控制物件的程序就會被啓動起來執行，以完成某項特定的工作。

使用控制物件之方法的語法如下：

控制物件名稱.方法 參數資料

例 ◆ Form1.Cls

‘ 使用表單控制物件之 Cls 方法，清除表單上所顯示的文字

事件

對日常生活中的人或物而言，事件是指發生在人或物上的事項，例如「背上」書包、「按下」電腦的電源鈕、「旋轉」喇叭的音量鈕等。在 Visual Basic 中，事件是指可以被控制物件偵測到的動作，例如在表單控制物件上按滑鼠左鍵一下，表單即會偵測到這個動作，產生一個 Click 事件。

● **事件程序**：在 Visual Basic 中，要使程式對某個事件有所反應，意即啓動一段程式碼執行特定的工作，必須在程式碼視窗中撰寫與事件相對應的程式碼。這些當事件發生時所要執行的程式碼稱為**事件的程序**，或稱為**事件程序**（event procedure）。

當我們要撰寫事件程序（例如 Form_Activate 事件程序）時，必須先開啓程式碼視窗，並切換到對應的控制物件選項及事件選項後，才能開始撰寫該事件的程式碼（如圖 8-36 所示）。

! 一點靈

雙按控制物件或按專案視窗中的檢視程式碼鈕 ，都可開啟程式碼視窗。

網路資源

http://yes.nctu.edu.tw/vb/0_Basis/Sect/Sect4.htm 瀏覽編寫 Visual Basic 程式的基本技巧

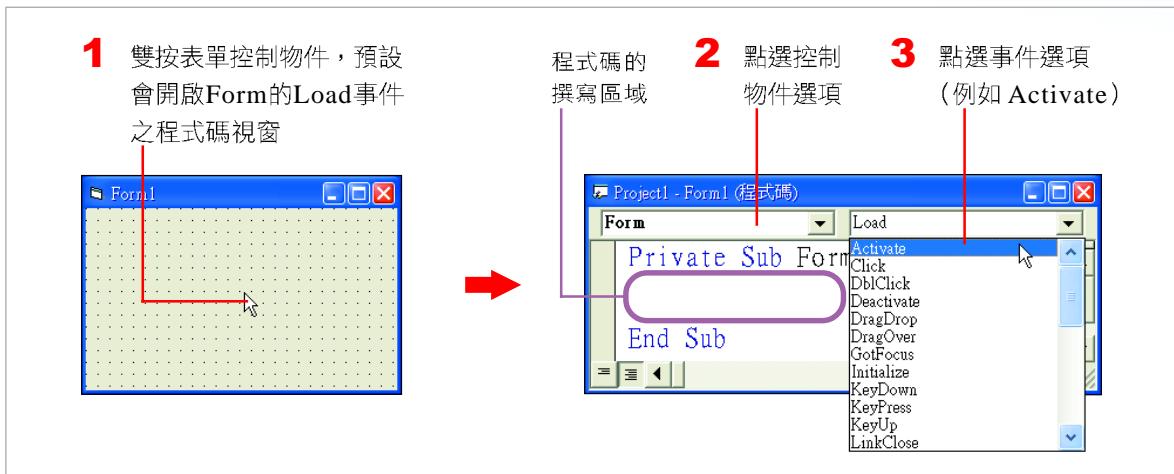


圖 8-36 撰寫事件程序的步驟 以設計 Form_Activate 事件程序為例

● **一般程序**：是指撰寫在程式碼視窗之**一般**選項內的程式碼，如圖 8-37 所示。這些撰寫在一般程序內的程式碼，通常是副程式、自定函數及全域變數的宣告敘述等。

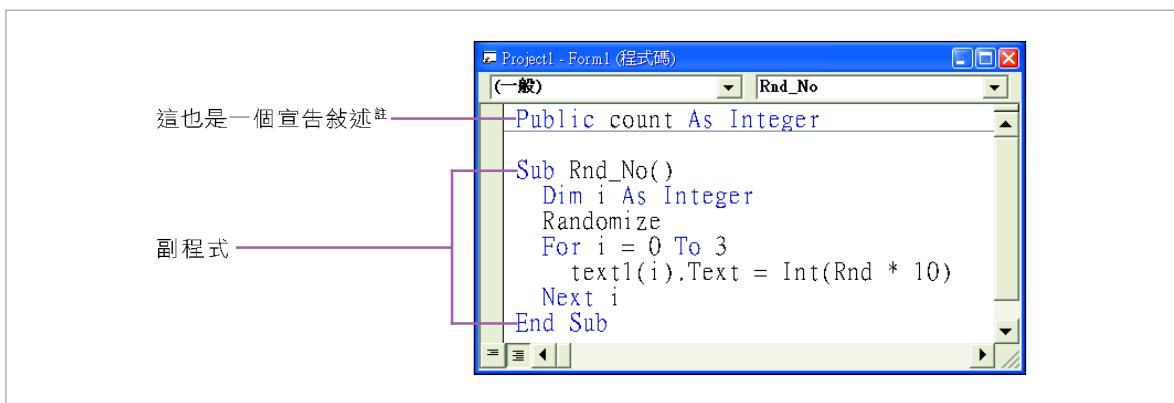


圖 8-37 一般程序 通常用來撰寫程式所需使用的副程式、自定函數及全域變數的宣告敘述等

隨堂練習

- () 1. 在 Visual Basic 中，下列何者是指內建在控制物件中的程序？ (A)事件 (B)方法 (C)屬性 (D)事件程序。
- () 2. 「在表單上雙按滑鼠左鍵，可顯示特定訊息文字」，在此敘述中何者為表單的事件？ (A)表單 (B)雙按滑鼠左鍵 (C)顯示特定訊息文字 (D)滑鼠。

註：全域變數（global variable）可被專案中的任何程式敘述使用。宣告全域變數時，必須使用保留字 "Public" 來宣告。

基本控制物件

在 Visual Basic 中，**Form**（表單）及工具箱的各個工具鈕皆屬於控制物件。

圖 8-38 為 **Form** 控制物件及工具箱中常用控制物件的說明。

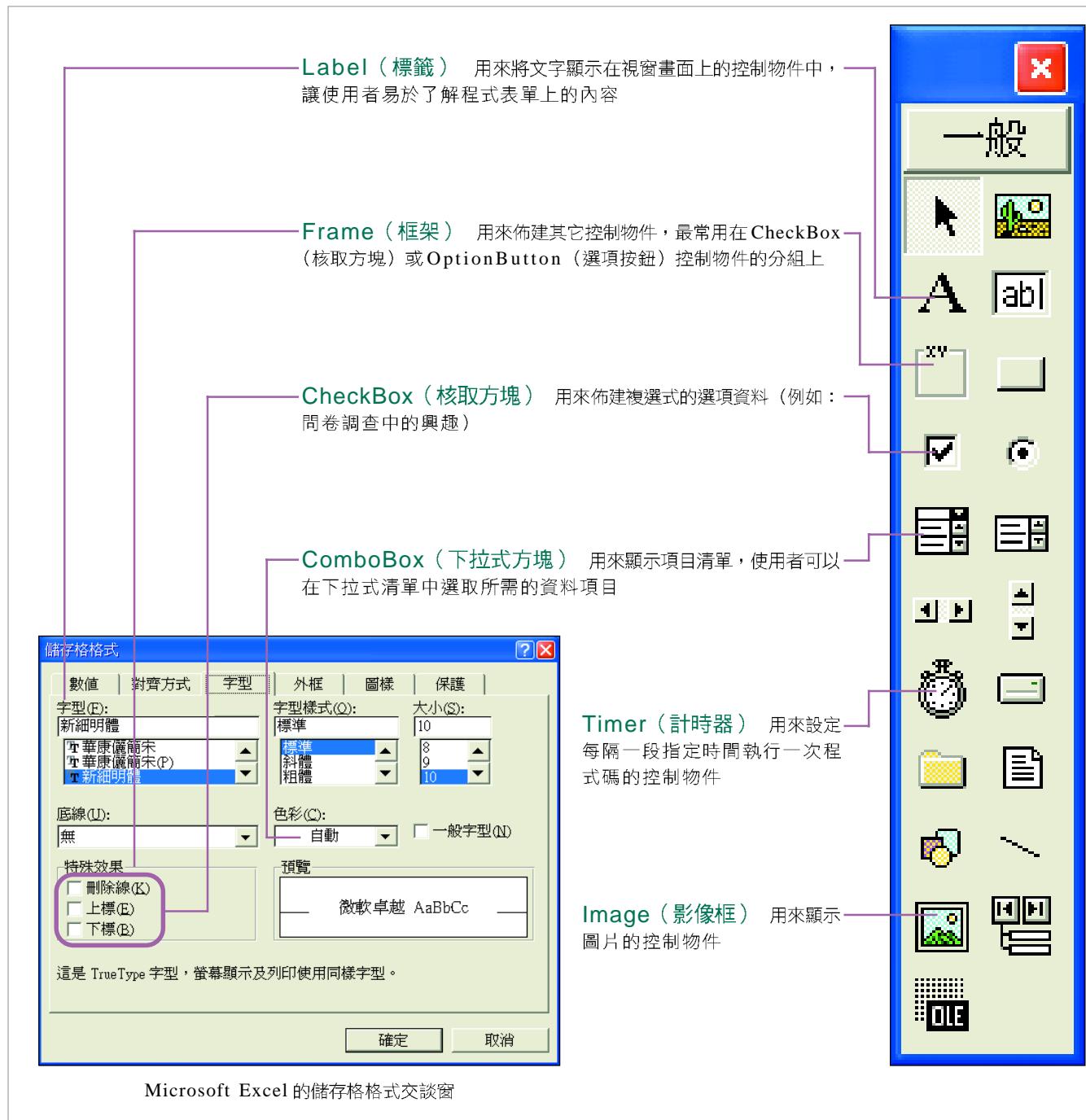
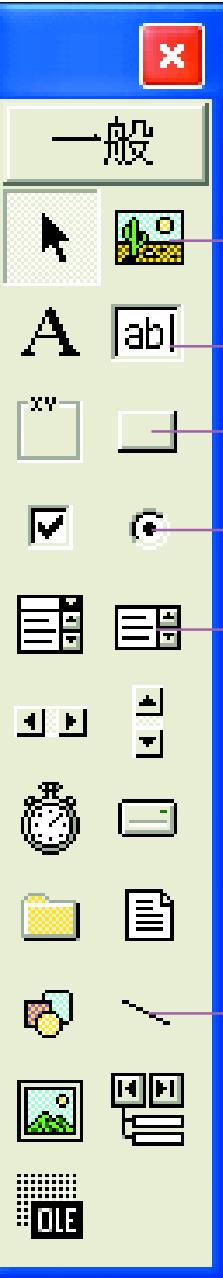


圖 8-38 Visual Basic 常用控制物件 表單及工具箱中常用控制物件在應用程式中的使用範例

Form (表單) 用來佈建控制物件的平台，常被用來做為視窗程式的主畫面或交談窗



PictureBox^{註1} (圖片框) 用來顯示圖片或繪製圖案的控制物件，常被用在需要圖片說明的程式視窗中

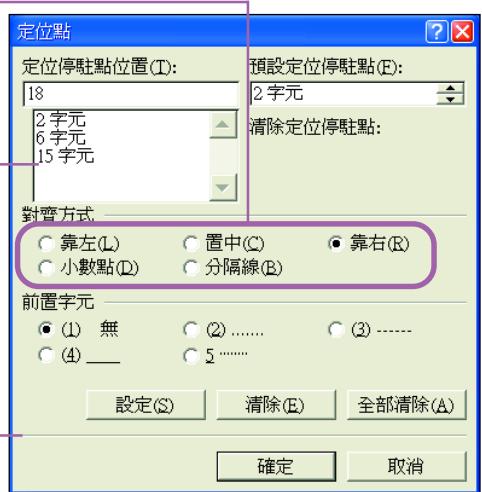
TextBox (文字方塊) 用來提供使用者輸入資料的控制物件，在視窗程式中，我們經常使用它來作為使用者輸入資料的區域，以便將相關資料傳送到程式中儲存或處理

CommandButton (按鈕) 用來操控程式運作的控制物件，例如在 Yahoo! 奇摩即時通的交談視窗中按傳送鈕，可將我們輸入的訊息傳送出去

OptionButton (選項按鈕) 用來佈建單選式的選項資料（例如：問卷調查中的性別）

ListBox^{註2} (清單) 用來顯示項目清單，使用者可以在清單中選取所需的資料項目

Line (線) 用來繪製直線，使用者只要按此鈕，即可以在表單上，拉曳出一條直線



Microsoft Word 的定位點交談窗

Visual Basic 工具箱

註1 : Image 與 PictureBox 的最大差異在於 Image 提供 Stretch 屬性，可讓圖片隨著 Image 控制物件的大小自動縮放。

註2 : ComboBox 與 ListBox 最大差異在於 ComboBox 所佔用的表單空間較小，適合使用在資料項目很多的清單中。

每一個控制物件有其特定的功能與適用的場合，下面將介紹程式視窗設計時，經常須使用的 **Form**（表單）、**CommandButton**（按鈕）、**Label**（標籤）及 **TextBox**（文字方塊）等 4 個控制物件。

Form（表單）

Form 控制物件是用來佈建控制物件的平台，常被用來作為視窗程式的主畫面或交談窗。

● 常用屬性：圖 8-39 是 Form 控制物件的常用屬性說明。

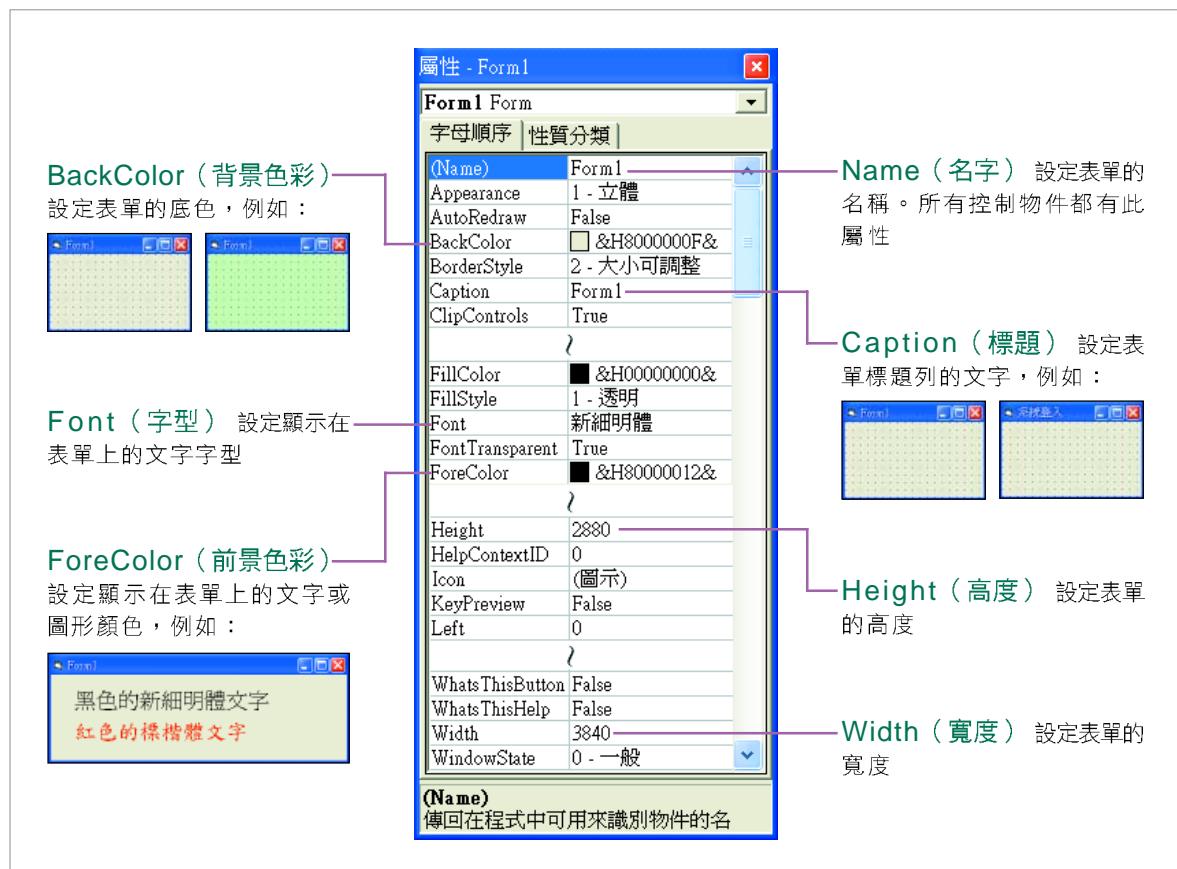


圖 8-39 Form 常用的屬性 BackColor、Caption、Font、ForeColor、Height、Name、Width 等屬性說明

- 常用方法：表 8-14 是 Form 控制物件的常用方法說明。

表 8-14 Form 控制物件常用的方法說明

| 方法 | 說明 |
|-------|----------------|
| Cls | 清除表單上所顯示的文字及圖形 |
| Hide | 隱藏表單 |
| Show | 顯示表單 |
| Print | 在表單上輸出資料 |

- 常用事件：表 8-15 是 Form 控制物件的常用事件說明。

表 8-15 Form 控制物件的常用事件說明

| 事件 | 說明 |
|----------|---|
| Activate | 當表單成為作用中的視窗時發生；一般常在此事件中撰寫設定表單初始化畫面的程式敘述 |
| Click | 當使用者在表單上按滑鼠左鍵一下時發生 |
| DblClick | 當使用者在表單上連按滑鼠左鍵二下時發生 |
| Load | 當表單載入時發生；一般常在此事件中撰寫設定控制物件屬性的程式敘述 |

當程式執行時，首次使用到表單，便會引發表單的 Load 事件；另外，當表單成為作用視窗時，則會引發表單的 Activate 事件，例如當我們由 Form1 切換到 Form2 時，Form2 便成為作用視窗，此時即會引發 Form2 的 Activate 事件。

若 VB 程式專案只包含 1 個表單，且表單的 Load 及 Activate 事件中，皆有撰寫對應的程式碼，則當程式執行時，將會先執行 Load 事件程序，再執行 Activate 事件程序。

一點靈

作用視窗指的是顯示在桌面最上層的視窗。

- 表單的範例：撰寫簡易程式，讓使用者在表單上單按滑鼠左鍵（Click）時，可輸出一行空自行及一串文字，如下頁圖 8-40 所示。

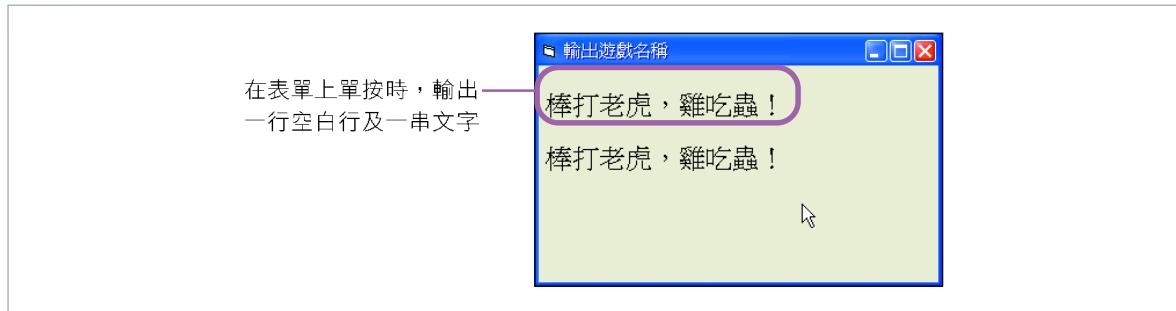


圖 8-40 表單範例的執行結果 單按滑鼠左鍵時，會輸出遊戲名稱

► 設定控制物件的屬性：

1. 設定 **Form1** 的 **Caption**（標題）屬性為輸出遊戲名稱。
2. 在 **Form1** 的 **Font**（字型）屬性中，將大小設定為 **14**。

► 撰寫程式碼：（可開啓範例檔案 "Ch08-3b.vbp"）

在表單的 **Click** 事件中，利用 **Print** 方法輸出一行空白行及一串文字。

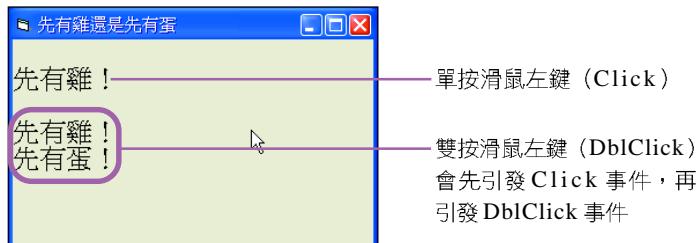
《程式碼》

```
Private Sub Form_Click()
    Print                               ' 輸出一行空白行
    Print "棒打老虎，雞吃蟲！"           ' 輸出一串文字
End Sub
```

隨堂練習

1. 若希望程式執行時，在表單上單按滑鼠左鍵（Click）時，可以輸出字串 "先有雞！"；在表單上雙按滑鼠左鍵（DblClick）時，可輸出字串 "先有蛋！"，則上述「輸出遊戲名稱」範例之程式碼該如何修改呢？

成果預覽：



註：請注意，在 Visual Basic 中，每個專案檔案可包含 1 個以上的表單，第 1 個表單預設的名稱為 **Form1**、第 2 個表單預設的名稱為 **Form2**……以此類推。但不論在哪一個表單中，撰寫 **Form** 控制物件的事件時，其事件程序的命名方式為：**Form_**事件名稱，因此，在本例中 **Form1** 的 **Click** 事件程序名稱為 **Form_Click()**。

CommandButton (按鈕)

CommandButton 控制物件是用來操控程式運作的控制物件，例如在一般交談窗中常見的確定鈕、取消鈕等。

- 常用屬性：圖 8-41 是 CommandButton 控制物件的常用屬性說明。

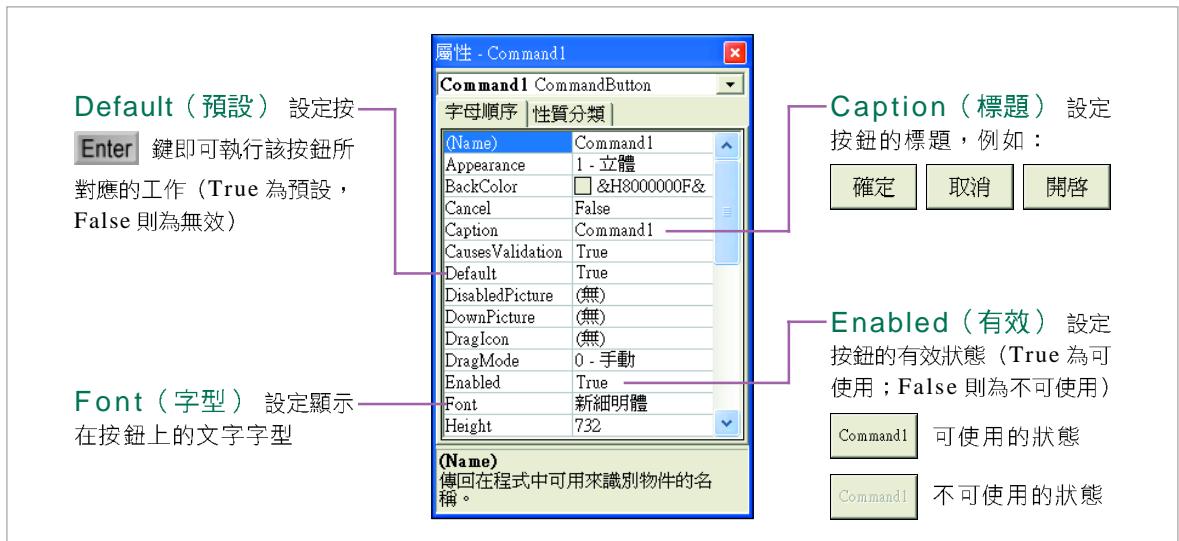


圖 8-41 CommandButton 常用的屬性 Caption 、Default 、Enabled 、Font 等屬性說明

- 常用事件：Click 是 CommandButton 控制物件最常使用的事件。當程式設計者希望使用者在按下按鈕時，可以執行特定的工作，可將對應執行該工作的程式碼撰寫在按鈕的 Click 事件中（如圖 8-42 中的儲存鈕）。

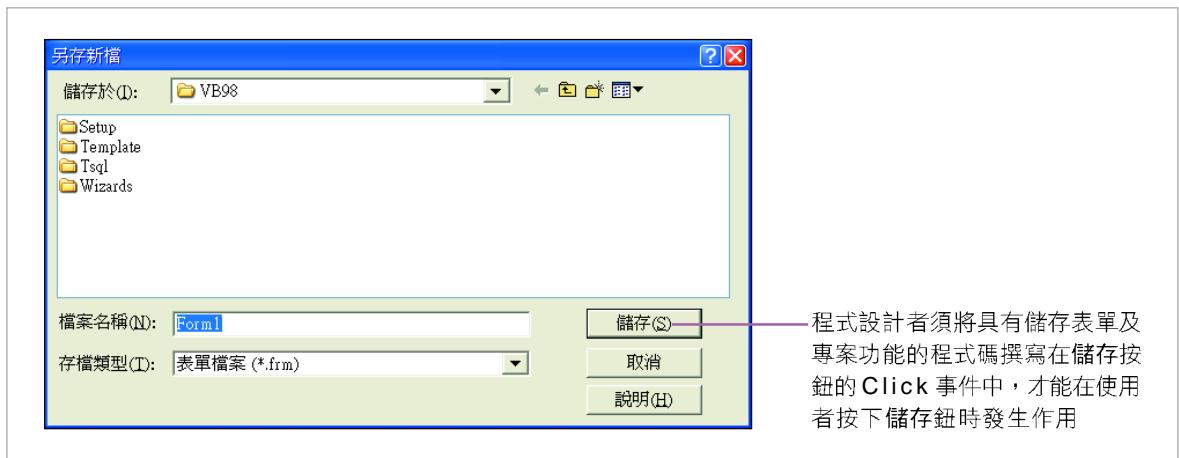


圖 8-42 CommandButton 的 Click 事件實例 為了讓使用者可透過不同的按鈕來操控程式的運作，程式設計者需在每一個按鈕（事件）中，設計對應的程式碼

- 按鈕的範例：修改上一個範例的程式碼使該程式具有清除表單畫面的功能（如圖 8-43 所示）。

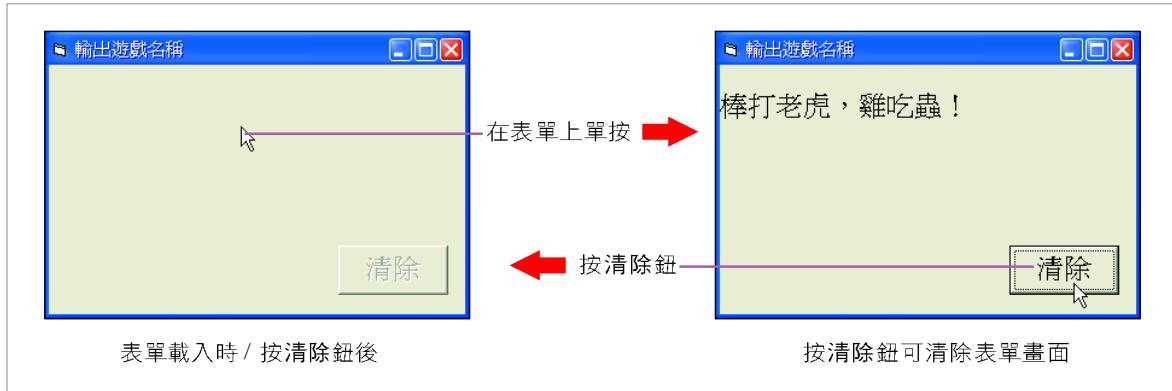


圖 8-43 按鈕範例的執行結果 佈建按鈕控制物件，並增加清除輸出在表單上的文字功能

► 佈建控制物件及設定屬性：

1. 在表單上佈建 1 個按鈕（Command1）控制物件。
2. 設定 **Command1** 的 **Caption** 屬性為 "清除"，並設定其 **Enabled**（有效）屬性為 **False**。

► 撰寫程式碼：（可開啓範例檔案 "Ch08-3c.vbp"）

1. 在表單的 **Click** 事件中，設定 **Command1** 的 **Enabled** 屬性為 **True**。
2. 在清除鈕（Command1）的 **Click** 事件中，使用 **Cls** 方法清除表單畫面，並設定 **Command1** 的 **Enabled** 屬性 **False**。

《程式碼》

```

Private Sub Command1_Click()
    Cls
    Command1.Enabled = False
End Sub

Private Sub Form_Click()
    :
    Command1.Enabled = True
End Sub

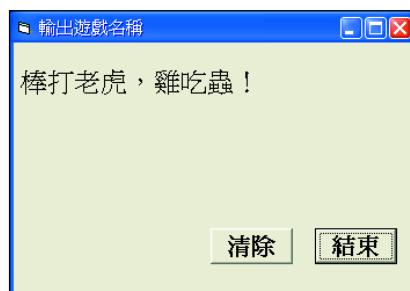
```

' 清除表單畫面
' 設定不可重複按清除鈕
' 設定可按清除鈕

隨堂練習

1. 延續前面的範例，若要在表單上再佈建 1 個按鈕控制物件，使程式具有結束執行的功能，則程式碼該怎麼修改？

成果預覽：



Label (標籤)

Label 控制物件可用來將文字顯示在視窗畫面上，其用途是讓使用者瞭解表單上的內容或操作項目。

- 常用屬性：圖 8-44 是 Label 控制物件的常用屬性說明。

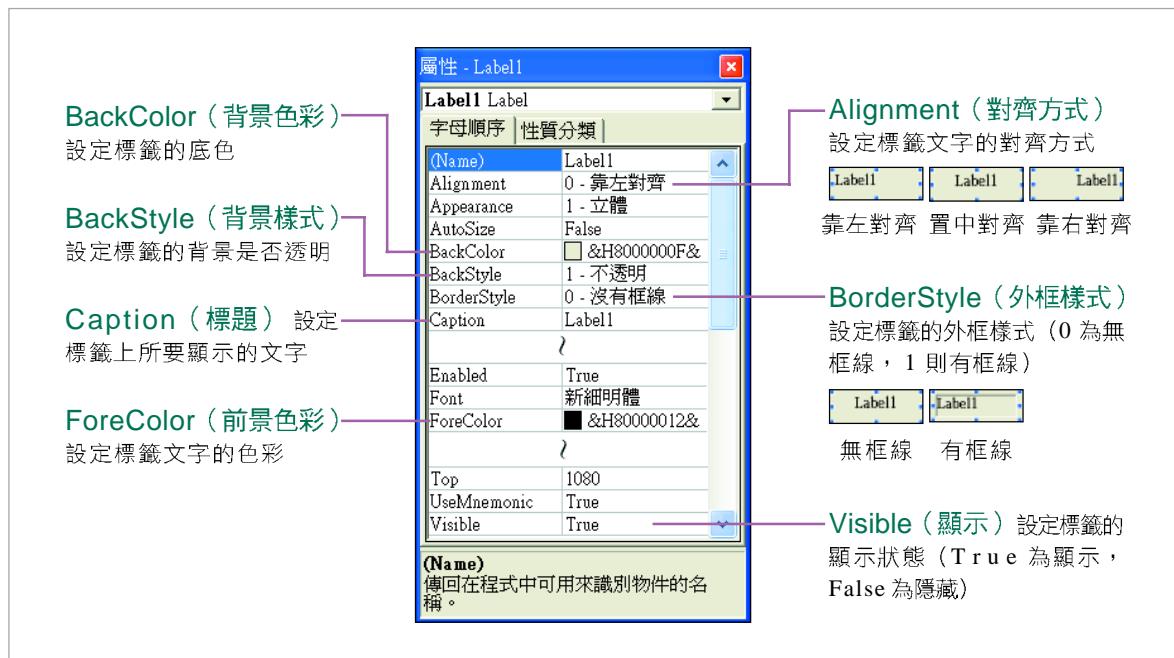


圖 8-44 Label 常用的屬性 Alignment、BackColor、BackStyle、BorderStyle、Caption、ForeColor、Visible 等屬性說明

- 標籤的範例：使用一個標籤來顯示腦筋急轉彎的題目，並在使用者按下謎底揭曉鈕後，在另一個標籤中顯示解答（如圖 8-45 所示）。

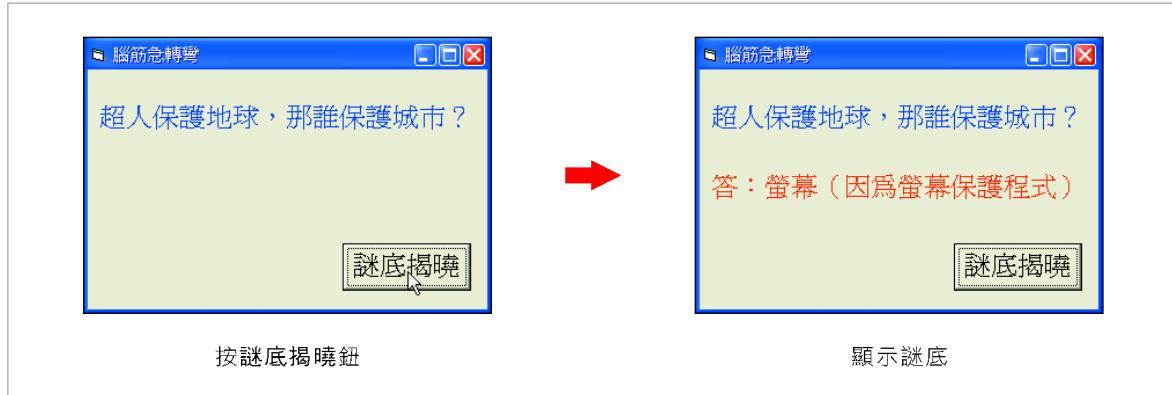


圖 8-45 標籤範例的執行結果 顯示腦筋急轉彎的題目及解答

► 佈建控制物件及設定屬性：

1. 在表單上佈建 2 個標籤（Label1 及 Label2）及 1 個按鈕（Command1）控制物件。
2. 依表 8-16 設定控制物件的相關屬性。

表 8-16 標籤範例的屬性設定

| 控制物件名稱 | 屬性 | 設定值 |
|----------|-----------|------------------|
| Form1 | Caption | 腦筋急轉彎 |
| Label1 | Caption | 超人保護地球，那誰保護城市？ |
| Label1 | ForeColor | 藍色 (&H00FF0000&) |
| Label2 | Caption | 答：螢幕（因為螢幕保護程式） |
| Label2 | ForeColor | 紅色 (&H000000FF&) |
| Command1 | Caption | 謎底揭曉 |

► 撰寫程式碼：（可開啓範例檔案 "Ch08-3d.vbp"）

1. 在 **Form1** 的 **Load** 事件中，設定 **Label2** 的 **Visible** 屬性為 **False**，使解答文字在表單載入時隱藏。
2. 在謎底揭曉鈕（Command1）的 **Click** 事件中，設定 **Label2** 的 **Visible** 屬性為 **True**，顯示解答文字。

《程式碼》

```

Private Sub Form_Load()
    Label2.Visible = False      ' 隱藏 Label2
End Sub

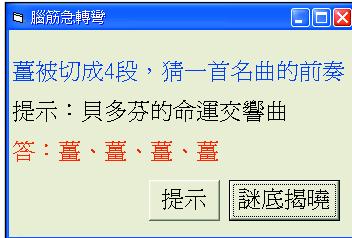
Private Sub Command1_Click()
    Label2.Visible = True       ' 顯示 Label2
End Sub

```

隨堂練習

1. 某一腦筋急轉彎的題目、提示及解答如下圖。請仿照上述腦筋急轉彎的範例，在表單上再佈建 1 個標籤及按鈕控制物件，讓使用者在按下提示鈕後，可顯示相關的提示文字。

成果預覽：



2. 開啟 "Ex08-d.vbp" 專案檔，撰寫一個可讓使用者在影像的相異部份，以單按滑鼠左鍵方式找出相異的部份。

成果預覽：



- 提示：(1) 參考上圖紅框部份，在影像相異部份佈建 4 個 Label（標籤）控制物件（Label2~Label5），並設定控制物件的 Caption 屬性為 ""（空字串）、BackStyle 屬性為 0- 透明
 (2) 在 4 個 Label 控制物件中撰寫程式敘述：
`MsgBox "好棒喔！", , "找到了"`

TextBox（文字方塊）

TextBox 是專門用來供使用者輸入資料的控制物件。在視窗程式中，我們經常使用它來作為使用者輸入資料的區域，以便將相關資料傳送到程式中處理或儲存。

- 常用屬性：圖 8-46 是 TextBox 控制物件的常用屬性說明。

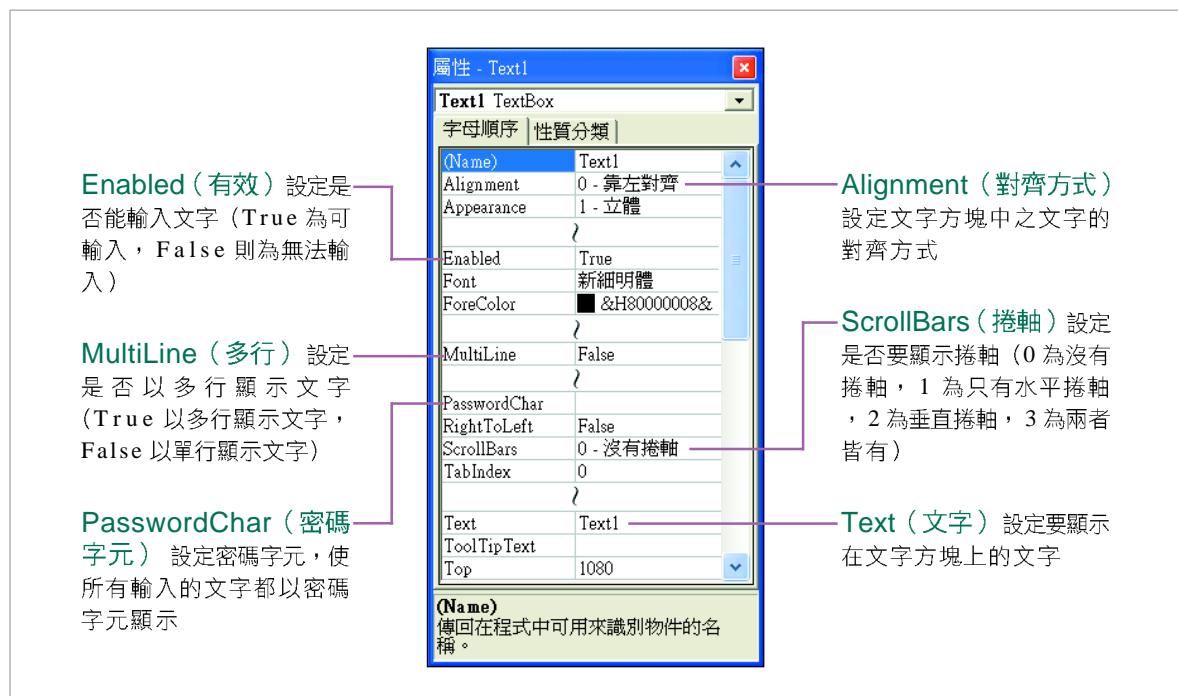
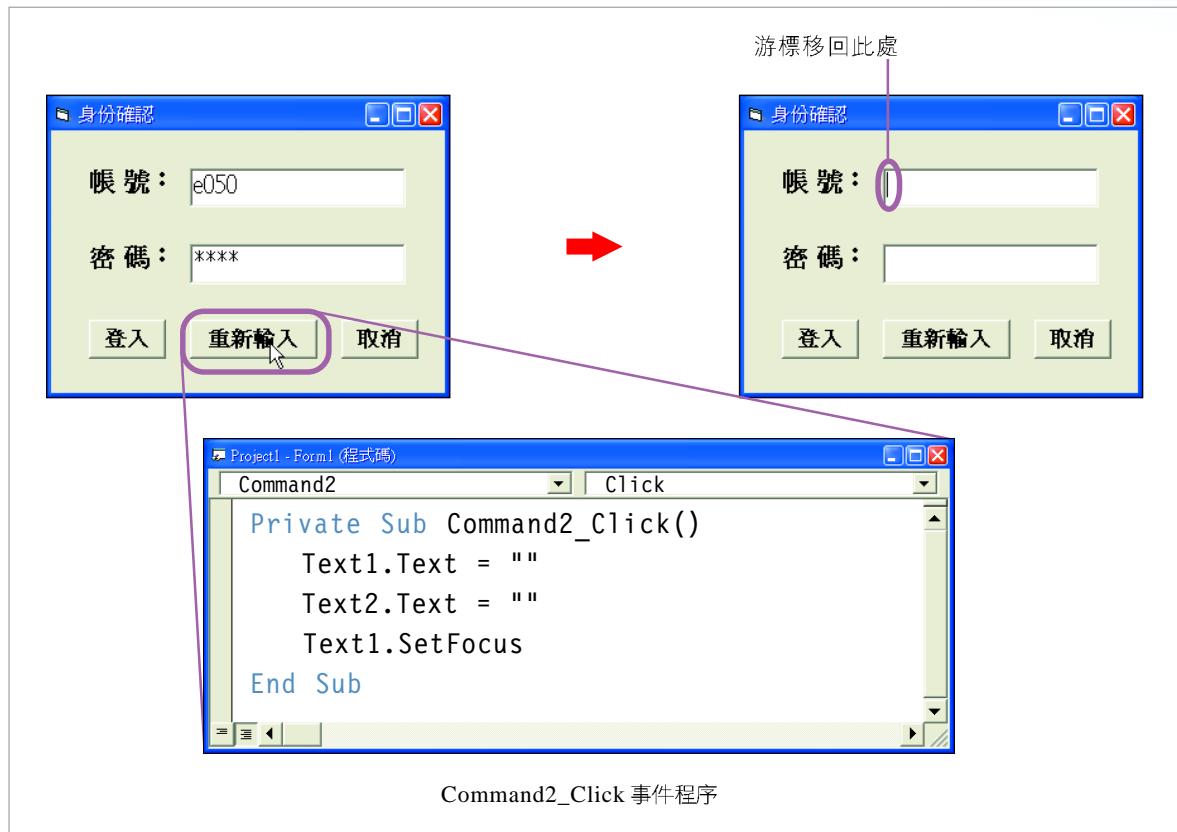


圖 8-46 TextBox 常用的屬性 Alignment、Enabled、MultiLine、PasswordChar、ScrollBars、Text 等屬性說明

- 常用方法：**SetFocus** 是 TextBox 控制物件常使用的方法，它可將游標移至文字方塊中，以省去使用者必須利用滑鼠（或按 Tab 鍵）來移動游標位置的操作。下頁圖 8-47 是一個使用 SetFocus 方法的範例，當使用者按下重新輸入鈕時，程式除了會清除文字方塊中的資料外，還會將游標移回第 1 個文字方塊中。



(範例檔案 "Ch08-3e.vbp")

圖 8-47 TextBox 之 SetFocus 方法的使用 身份確認的範例；當按下重新輸入時，會清除 2 個文字方塊中的文字，並將游標移至第 1 個文字方塊內

- **文字方塊的範例**：使用 2 個文字方塊控制物件，讓使用者輸入兩個整數值 a、b，並在按下計算鈕後，顯示 a 的 b 次方值（如圖 8-48 所示）。

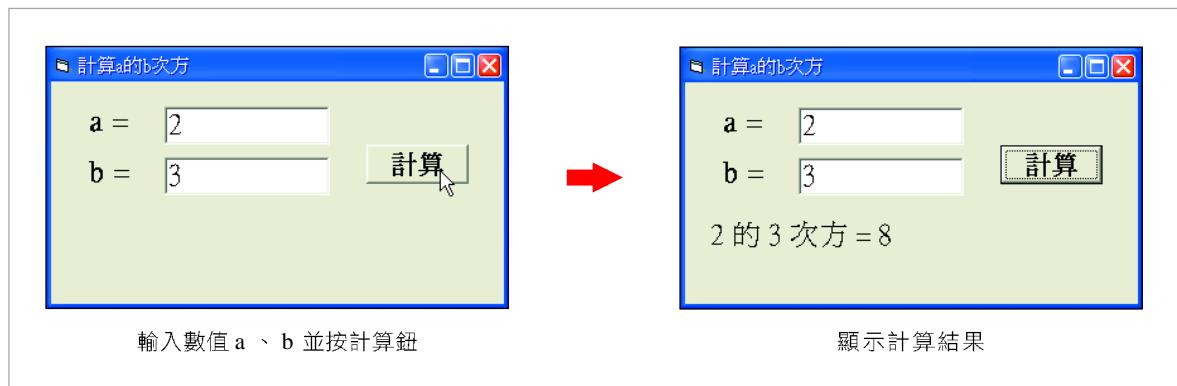


圖 8-48 文字方塊範例的執行結果 計算 a 的 b 次方

► 佈建控制物件及設定屬性：

1. 在表單上佈建 3 個標籤（Label1、Label2、Label3）、2 個文字方塊（Text1、Text2）及 1 個按鈕（Command1）控制物件。
2. 依表 8-17 設定各控制物件的相關屬性。

表 8-17 文字方塊範例的屬性設定

| 控制物件名稱 | 屬性 | 設定值 |
|----------|---------|--------------|
| Form1 | Caption | 計算 a^b 的次方 |
| Label1 | Caption | $a =$ |
| Label2 | Caption | $b =$ |
| Label3 | Caption | (空白) |
| Text1 | Text | (空白) |
| Text2 | Text | (空白) |
| Command1 | Caption | 計算 |

► 撰寫程式碼：（可開啓範例檔案 "Ch08-3f.vbp"）

1. 在計算鈕（Command1）的 Click 事件中，宣告 a、b 兩個整數變數及 Ans 一個長整數變數。
2. 將使用者在 2 個文字方塊（Text1、Text2）中所輸入的資料轉換成數值型別，並分別設定給變數 a、b。
3. 設定變數 Ans 為計算 a^b 的值。
4. 設定 Label3 的標題（Caption）屬性為計算 a^b 的結果。

《程式碼》

```

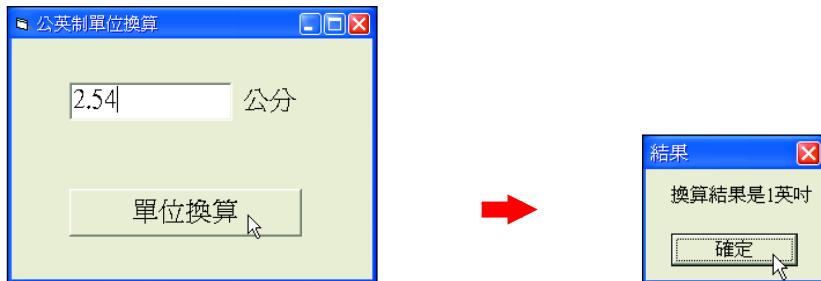
Private Sub Command1_Click()
    Dim a As Integer, b As Integer, Ans As Long
    a = Val(Text1.Text)                      ' 將輸入的值轉換成數值型別
    b = Val(Text2.Text)                      ' 將輸入的值轉換成數值型別
    Ans = a ^ b
    Label3.Caption = a & " 的 " & b & " 次方 = " & Ans
End Sub

```

隨堂練習

1. 撰寫一個公英制單位換算的程式，將使用者輸入的公分數值，利用 MsgBox 訊息交談窗顯示對應的英吋數值（需四捨五入至小數點第 2 位）。

成果預覽：



- 提示：
- (1) 佈建 Text、Label 與 CommandButton 控制物件並設定其相關屬性
 - (2) 利用敘述：Val(Text1.Text)，將使用者在 Text 控制物件輸入的值，轉換成數值型別的資料
 - (3) 1 英吋 = 2.54 公分
 - (4) 利用 Round() 內建函數將計算後的英吋數值，四捨五入至小數點第 2 位
 - (5) 利用 MsgBox 敘述，顯示計算後的英吋數值

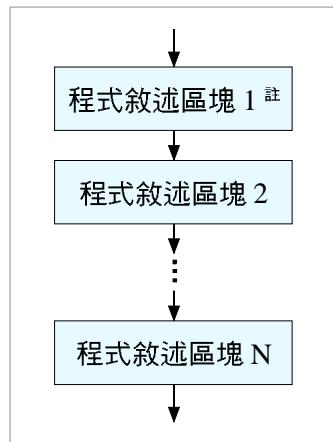
8-4 程式基本結構

一個程式大多是由**循序**、**條件**及**重複**等 3 種基本結構所組合，這 3 種結構也就是本章第 1 節所介紹之演算法的 3 種結構。

8-4-1 循序結構

在日常生活中，我們經常會遇到需要依序進行的問題，例如：要搭乘火車時，必須先購買火車票，然後才能進到月台搭車；要觀賞電影時，必須先挑選想看的電影，然後到上映該電影的電影院購票，最後才能進場觀看電影。

在程式語言中，遇到需要依序執行的問題，可使用循序結構來處理。**循序結構**是程式中最基本的結構，其執行順序是由程式的第一行敘述開始，然後逐行執行到最後一行。圖 8-49 為循序結構的執行流程；圖 8-50 為循序結構的範例。



```

Private Sub Form_Load()
    Dim born_year As Integer
    Dim age As Integer
    born_year = InputBox("請輸入出生之西元年份")
    age = Year(Date) - born_year
    MsgBox "你今年" & age & "歲"
End
End Sub

```

(範例檔案 "Ch08-4a.vbp")

圖 8-49 循序結構的執行流程
循序結構是一種由上而下依序執行程式敘述的程式結構

圖 8-50 循序結構的範例 使用 InputBox 函數讓同學輸入出生之西元年份，並以 MsgBox 來顯示年齡的範例

8-4.2 條件結構

在日常生活中，我們經常會遇到需要抉擇的問題，例如逛書店時，如果身上有足夠的錢，才可以選購想要閱讀的書籍；買火車票時，如果尚有座位，就會買座票，否則就須買站票；週末夜晚我們可以選擇從事各種不同的休閒活動，例如到 KTV 唱歌、去看電影或是和家人相聚。

在程式設計時，經常會遇到有關程式執行路徑需要抉擇的問題，此時可使用條件結構來處理。**條件結構**是一種利用條件判斷敘述來選擇程式執行路徑的程式結構。

在條件判斷敘述中，經常需使用**比較運算子**與**邏輯運算子**兩種運算子，以下先介紹這 2 種運算子的用法，再介紹 If-Then-Else 與 Select Case 判斷敘述。

註：本章以下將以「**程式敘述區塊**」來代表一行以上（含）的程式敘述。

比較運算子

比較運算子是用來比較兩個相同資料型別之運算元間的大小關係，若比較後的結果成立（為真），則傳回 True（對，或稱真，一般常以 T 表示）；若不成立則傳回 False（錯，或稱假，一般常以 F 表示）。表 8-18 為 Visual Basic 的 6 種比較運算子的說明與運算實例。

表 8-18 各種比較運算子說明與運算實例

| 比較運算子 | 功能說明 | 運算實例 | 結果 |
|-------|-----------|--------|----|
| = | 等於 | 1 = 2 | F |
| > | 大於 | 1 > 2 | F |
| < | 小於 | 1 < 2 | T |
| >= | 大於等於（不小於） | 1 >= 2 | F |
| <= | 小於等於（不大於） | 1 <= 2 | T |
| <>或>< | 不等於 | 1 <> 2 | T |

比較運算子在比較字串資料的大小時，會先以兩字串的第 1 個字元之 ASCII 碼比較，若相等，則再依序以字串中其它相同位置的 ASCII 碼來做比較（如圖 8-51 所示）。

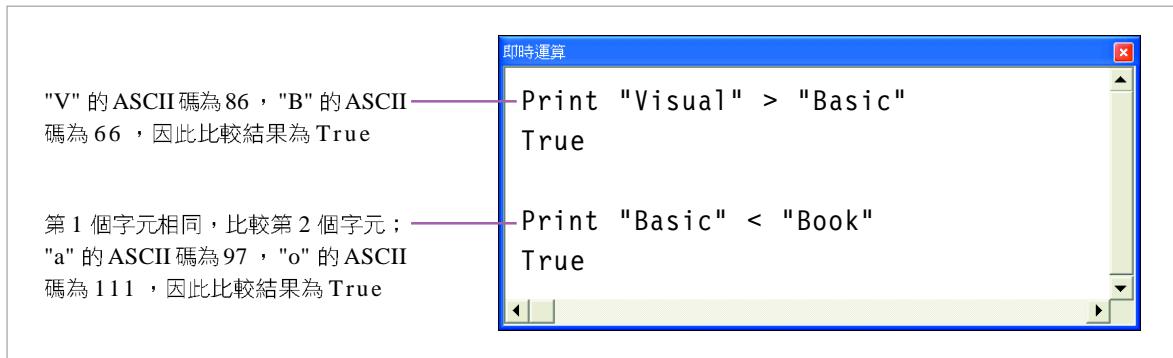


圖 8-51 比較運算子的運算 比較字首字母的 ASCII 碼大小，若相同再逐一往下比較

邏輯運算子

邏輯運算子是用來執行運算式間的邏輯運算，Visual Basic 的邏輯運算子中較常見有 Not、And、Or 等 3 種，下頁表 8-19 為這 3 種邏輯運算子的說明與運算實例。

表 8-19 各種邏輯運算子說明與運算實例（假設 A、B 運算式的運算結果皆為 T）

| 邏輯運算子 | 功能說明 | 運算實例 | 結果 | 運算優先順序 |
|---------|----------------------|---------|----|--------|
| Not (非) | 傳回與運算式相反的結果 | Not A | F | 1 |
| And (且) | 當兩邊的運算式皆為真時，傳回 True | A And B | T | 2 |
| Or (或) | 只要有一邊的運算式為真，即傳回 True | A Or B | T | 3 |

邏輯運算子經常和比較運算子結合使用。當運算式中同時含有算術運算子、比較運算子和邏輯運算子時，其運算的優先順序依序為：算術運算子、比較運算子、邏輯運算子。

If-Then-Else 判斷敘述

If-Then-Else 敘述是用來判斷條件式成立時，所要執行的程式敘述區塊，以及條件式不成立時，須執行的程式敘述區塊，其執行流程若以流程圖表示，可繪製如圖 8-52 所示。

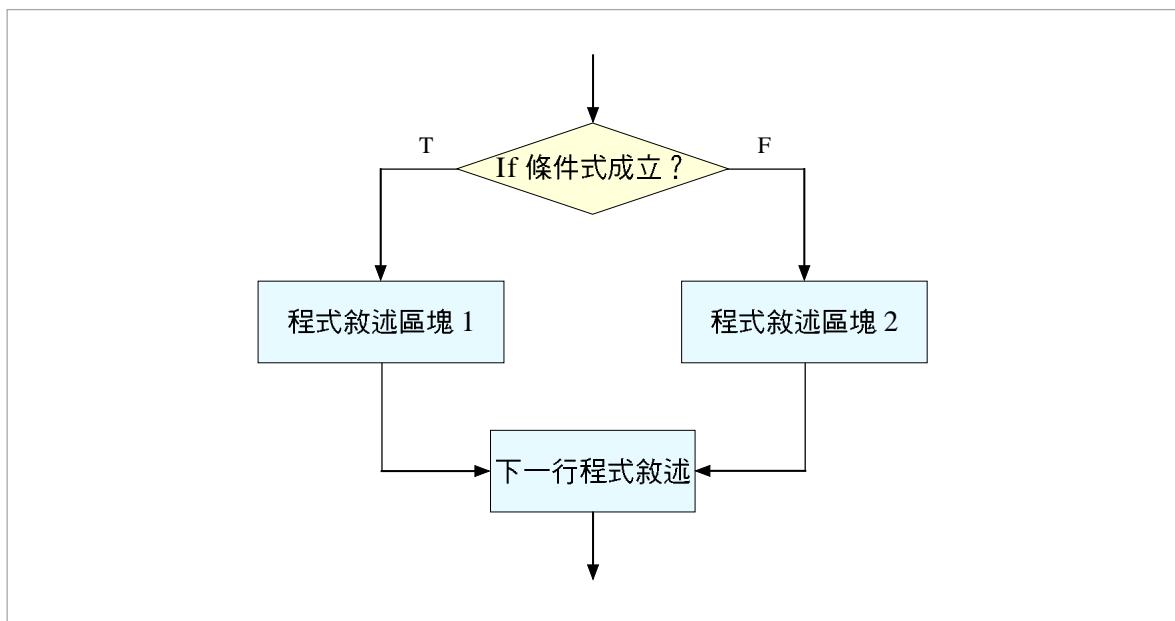


圖 8-52 If-Then-Else 判斷敘述的流程圖 條件式成立時，執行左側的程式敘述區塊；條件不成立時，執行右側的程式敘述區塊

● 語法說明：

```

語法：If 條件式 Then
      程式敘述區塊 1
      Else
      程式敘述區塊 2
End If

```

《說明》

- 當「條件式」成立時，會執行 **Then** 之後的程式敘述區塊；當「條件式」不成立時，會執行 **Else** 之後的程式敘述區塊。
- If-Then-Else 敘述必須與 End If 成對使用。

● 範例：

圖 8-53 為使用 If-Then-Else 敘述來判斷視力是否正常的範例；當輸入的值介於 0.8~1.2 之間時，會輸出 " 視力正常 " 的訊息，否則會輸出 " 視力可能需矯正 " 的訊息。

```

Dim eye_sgt As Single
eye_sgt = InputBox("請輸入視力檢測值：")
If eye_sgt > 0.8 And eye_sgt <= 1.2 Then
    Print " 視力正常 "
Else
    Print " 視力可能需矯正 "
End If

```

(範例檔案 "Ch08-4b.vbp")

圖 8-53 If-Then-Else 敘述的範例（I） 判斷視力是否要矯正

圖 8-54 為使用 If-Then-Else 敘述來判斷是否要吃素的範例；當輸入的農曆日期為 1 或 15，會輸出 " 要吃素喔 " 的訊息，否則會輸出 " 董素不拘 " 的訊息。

```
Dim today As Integer
today = InputBox("請輸入今天的農曆日期：")
If today = 1 Or today = 15 Then
    Print "要吃素喔"
Else
    Print "董素不拘"
End If
```

(範例檔案 "Ch08-4c.vbp")

圖 8-54 If-Then-Else 敘述的範例（II） 判斷是否要吃素

隨堂練習

- 下面是使用 If-Then-Else 敘述，來判斷足月、早產兒（懷孕至分娩的週數未滿 37 週，稱為早產兒）的程式片段，請在空格處填入正確的條件式。

```
week = InputBox("請輸入懷孕至分娩的週數：")
If _____ Then
    Print "足月兒"
Else
    Print "早產兒"
End If
```

- 假設班上的計概平均分數為 75 分，請撰寫一個程式，判斷使用者所輸入的計概成績是否高於全班的平均分數，最後將專案、表單儲存為 "Sp08-f.vbp" 、 "Sp08-f.frm" 。



巢狀 If 判斷敘述

在程式設計實務中，當抉擇不只兩種時，我們可以使用巢狀 If 判斷敘述來處理。所謂巢狀 If 判斷敘述，是指 If 判斷敘述中還有 If 判斷敘述。圖 8-55 為使用巢狀 If 判斷敘述的範例。

```

year = InputBox("請輸入出生年次：")
If year >= 60 Then
    If year <= 69 Then          '60~69 年間出生者
        Print "草莓族"
    Else
        Print "水蜜桃族"
    End If
Else
    Print "芭樂族"             '59 年（含）前出生者
End If

```

(範例檔案 "Ch08-4d.vbp")

圖 8-55 巢狀 If 判斷敘述的範例 刪斷出生年次所屬世代代稱

Select Case 判斷敘述

Select Case 判斷敘述是專門用來處理多重選擇的條件判斷敘述，利用這種敘述可讓我們一次設定多個不同的條件，使程式敘述的層次變得簡潔而易於閱讀。

```

If income >= 100 Then
    Print "可申請白金卡"
Else
    If income >= 40 Then
        Print "可申請金卡"
    Else
        Print "可申請普卡"
    End If
End If

```

使用巢狀 If 敘述

```

Select Case income
Case Is >= 100:Print "可申請白金卡"
Case Is >= 40:Print "可申請金卡"
Case Else:Print "可申請普卡"
End Select

```

使用 Select Case 敘述

圖 8-56 使用巢狀 If 敘述及 Select Case 敘述的比較 判斷可申請之信用卡卡別；income 代表年收入（單位為萬元）， ≥ 100 可申請白金卡，40 ~ 99 可申請金卡， < 40 可申請普卡

Select Case 判斷敘述的執行流程，若以流程圖表示，可繪製如圖 8-57 所示。

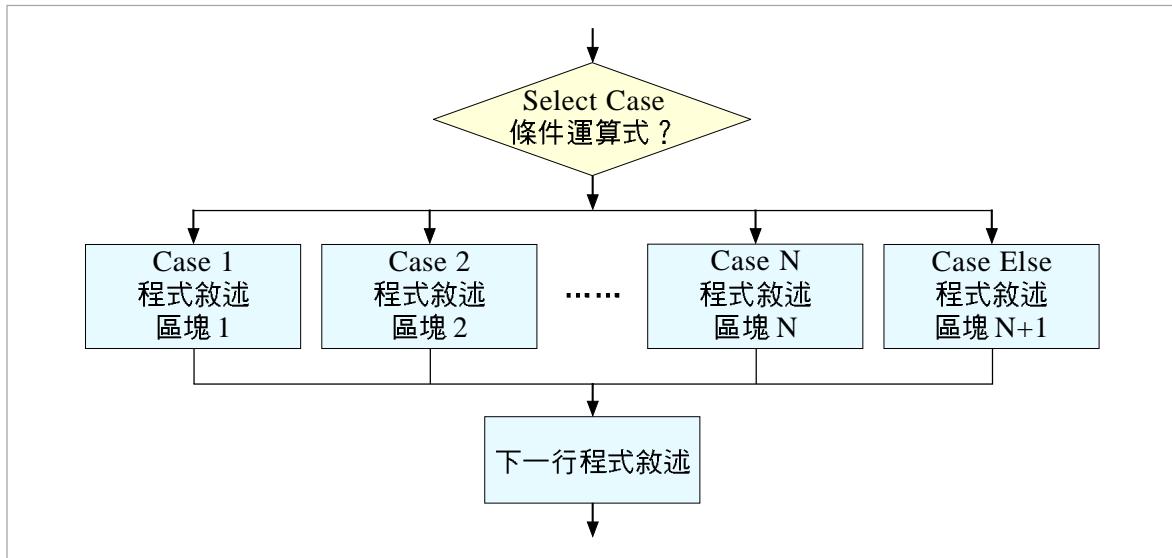


圖 8-57 Select Case 判斷敘述的流程圖 依據條件運算式的值，執行符合條件的 Case 程式敘述區塊

● 語法說明：

語法：Select Case 資料或運算式

Case 條件式 1

 程式敘述區塊 1

Case 條件式 2

 程式敘述區塊 2

Case 條件式 3

 程式敘述區塊 3

⋮

Case 條件式 N

 程式敘述區塊 N

Case Else

 程式敘述區塊 N+1

End Select

《說明》

1. 當資料或運算式符合「條件式 1」時，執行「程式敘述區塊 1」；符合「條件式 2」，則執行「程式敘述區塊 2」；……，依此類推。
2. 若不符合任何條件式，則執行「程式敘述區塊 N+1」。
3. Select Case 判斷敘述中的 Case 敘述可有下列幾種語法表示：
 - (1) 數值或字串：例如 Case 1、Case 2, 3, 4、或 Case "A", "B"。
 - (2) 指定範圍：例如 Case 1 To 10、Case 15 To 25。
 - (3) 關係運算式：例如 Case Is > 2、Case Is = 1。
4. 若所有條件式皆不符合時不需執行任何程式敘述，則 Case Else 可省略不寫。

● 範例：

圖 8-58 為使用 Select Case 判斷敘述來判斷輸入之年齡可觀賞影片等級的範例。

```
Dim age As Integer
age = InputBox("請輸入年齡：")
Select Case age
    Case Is < 6                                ' 未滿 6 歲者
        Print "可看普遍級"
    Case Is < 12                               ' 6~11 歲者
        Print "可看普遍級及保護級"
    Case Is < 18                               ' 12~17 歲者
        Print "可看非限制級的影片"
    Case Else                                 ' 18 歲（含）以上者
        Print "可看各級影片"
End Select
```

(範例檔案 "Ch08-4e.vbp")

圖 8-58 Select Case 判斷敘述的範例 (I) 依年齡判斷可觀賞的影片等級

圖 8-59 為改用 Select Case 判斷敘述來判斷輸入之年次所屬世代代稱的範例（改寫後的範例，較 P.317 使用巢狀 If 判斷敘述的範例簡潔許多）。

```
Dim year As Integer
year = InputBox("請輸入出生年次：")
Select Case year
    Case Is <= 59 : Print "芭樂族"      '59 年（含）前出生者
    Case 60 To 69 : Print "草莓族"      '60~69 年間出生者
    Case Is >= 70 : Print "水蜜桃族"    '70 年（含）後出生者
End Select
```

（範例檔案 "Ch08-4f.vbp"）

圖 8-59 Select Case 判斷敘述的範例（II）判斷出生年次所屬的世代代稱

隨堂練習

1. 判斷颱風強度（根據氣象局的規定，風速每小時在 62~117 公里為輕度颱風；118~183 公里為中度颱風；184~220 公里為強烈颱風）。

```
speed = InputBox ("請輸入風速，單位：公里 / 時 ")
Select Case speed
    Case 62 To 117:Print "輕度颱風"
    Case _____:Print "中度颱風"
    Case _____:Print "強烈颱風"
End Select
```

2. 請撰寫一個依據降雨量多寡來判斷雨量定義的程式，並將專案、表單儲存為 "Sp08-g.vbp"、"Sp08-g.frm"。

提示：(1) 大雨：五十毫米以上
 (2) 豪雨：一百三十毫米以上
 (3) 大豪雨：二百毫米以上
 (4) 超大豪雨：三百五十毫米以上

8-4-3 重複結構

在日常生活中，我們經常會遇到需要反覆執行的問題，例如學校在舉行 4000 公尺賽跑時，只要請選手們反覆繞著 400 公尺長的跑道跑 10 圈，即可跑完全程，而不須大費周章地舖設一條長為 4000 公尺的跑道。

重複結構是一種利用迴圈敘述來重複執行一段程式敘述的結構。在程式設計時，我們也經常會遇到類似前述需要反覆執行的問題，此時便可使用重複結構來處理。以下將介紹 For-Next 迴圈敘述。

For-Next 迴圈敘述

For-Next 迴圈敘述就是一種用來處理必須重複執行，且已確定須執行次數的程式敘述，其執行流程若以流程圖表示，可繪製如圖 8-60 所示。

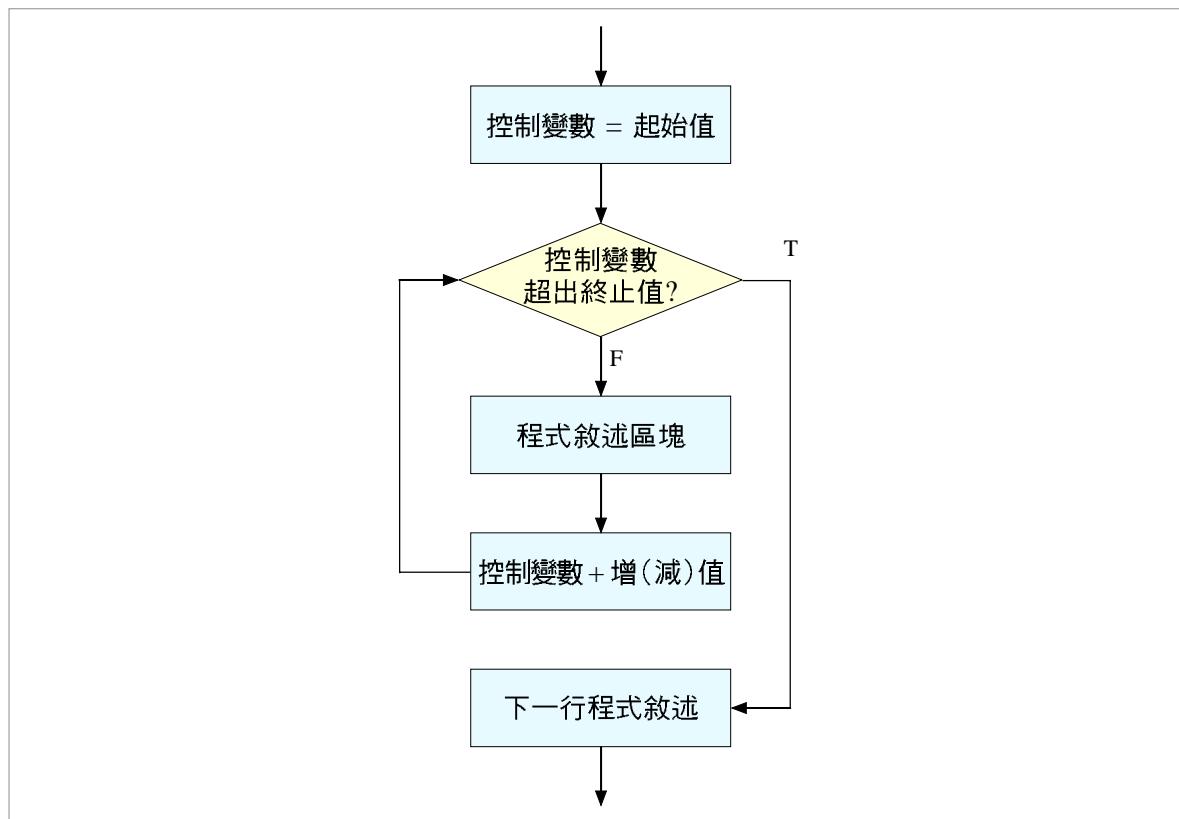


圖 8-60 For-Next 迴圈敘述的流程圖 當控制變數的值未超過終止值時，反覆執行中間的程式敘述區塊；超過時，則執行下一行程式敘述

● 語法說明：

語法：**For** 控制變數 = 起始值 **To** 終止值 **Step** 增值（減值）
程式敘述區塊
Next 控制變數

《說明》

1. 第 1 次進入 For-Next 迴圈時，控制變數會被設為起始值。
2. 每次執行 For-Next 迴圈時，程式都會判斷控制變數是否超過終止值，若未超過，則繼續執行迴圈內的敘述，若超過，則跳出迴圈。
3. 當執行到 Next 敘述時，程式會將目前控制變數的值加上增值（或減值）。
4. 增值為 1 時，Step 可省略不寫。

● 範例：

圖 8-61 為使用 For-Next 迴圈敘述來計算定期存款之本利和的範例。

```
N = InputBox("請輸入定存的年數：")
Sum = InputBox("請輸入定存的金額：")
rate = 0.02           ' 年利率為 2%
For i = 1 To N Step 1
    Sum = Sum * (1 + rate)
Next i
Print "本利和 = " & Sum
```

! 一點靈

左圖程式敘述 `Sum = Sum * (1 + rate)` 中的「=」符號代表指定運算，表示要將符號右側的運算結果指定存入到左側的變數 Sum 中，與數學的等號 (=) 意義完全不同。

(範例檔案 "Ch08-4g.vbp")

圖 8-61 For-Next 迴圈敘述的範例 (I) 計算定期存款的本利和

圖 8-62 為使用 For-Next 迴圈敘述將使用者輸入的字串反向輸出的範例。

```
s = InputBox("請輸入任意字串")
For i = Len(s) to 1 Step -1
    Print Mid(s, i, 1);
Next i
```

(範例檔案 "Ch08-4h.vbp")

圖 8-62 For-Next 迴圈敘述的範例 (II) 將使用者輸入的字串反向輸出



Do-Loop 迴圈敘述

當程式中的某段程式敘述必須重複執行，且無法事先確定執行次數時，我們可以使用 Do-Loop 迴圈敘述來處理。Do-Loop 迴圈敘述分為 Do-Loop 前測式迴圈敘述及 Do-Loop 後測式迴圈敘述兩類，表 8-20 是這 2 種敘述的比較說明。

表 8-20 前測式迴圈敘述與後測式迴圈敘述的比較說明

| 種類說明 | 前測式迴圈敘述 | 後測式迴圈敘述 |
|--------------------|---|---|
| 語法說明 | 語法： Do While 條件式 程式敘述區塊 Loop 說明：條件式成立時，執行迴圈內的程式敘述區塊 | 語法： Do 條件式 程式敘述區塊 Loop While 說明：先執行迴圈內的程式敘述區塊，再判斷條件式；條件式成立時，繼續執行迴圈內的程式敘述區塊 |
| 流程圖 | <pre> graph TD A{條件式？} -- T --> B[程式敘述區塊] B --> C(()) C -- F --> D{條件式？} </pre> | <pre> graph TD A[程式敘述區塊] -- T --> B{條件式？} B -- F --> A </pre> |
| 範例 (計算1加到10的總和) | <pre> sum = 0 i = 1 Do While i<=10 sum = sum + i i = i + 1 Loop Print "1+2+3+ +10=" & sum </pre> | <pre> sum = 0 i = 1 Do sum = sum + i i = i + 1 Loop While i<=10 Print "1+2+3+ +10=" & sum </pre> |

(範例檔案 "Ch08-4i.vbp"、"Ch08-4j.vbp")

註：若迴圈之條件式設計不當，可能會使程式產生無法跳離迴圈的情形，而形成無窮迴圈。

隨堂練習

1. 下面是一個使用 For-Next 敘述，來計算全班 40 位同學的國文總平均的程式片段，請在空格處填入正確的起始值、終止值、及增（減）值。

```
sum = 0 : chi = 0 : avg = 0
For i = _____ To _____
    chi = InputBox("請輸入第 " & i & " 位同學的國文成績：")
    sum = sum + chi
Next i
avg = sum / 40
Print "全班國文總平均為 " & avg
```

2. 請使用 For-Next 敘述，撰寫一個模擬樂透彩開獎的程式。

提示：隨機產生 6 個介於 1~42 的整數

8-4-4 程式設計實例

本實例將利用 If-Then-Else 判斷敘述、Select Case 判斷敘述及 For-Next 迴圈敘述來隨機輸出 3 個介於使用者指定範圍的整數（如圖 8-63）。下頁圖 8-64 是本實例的流程圖。

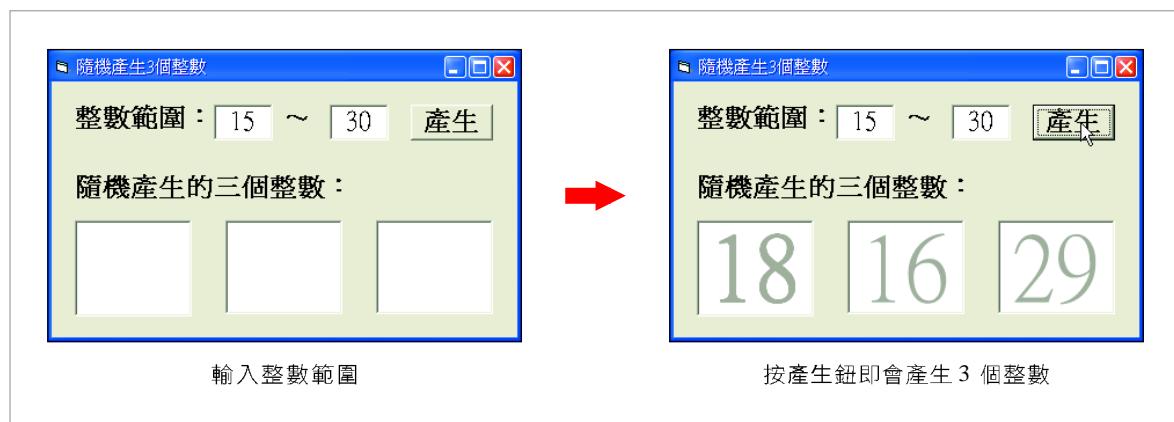


圖 8-63 隨機產生 3 個整數的執行結果 整數範圍由使用者輸入決定

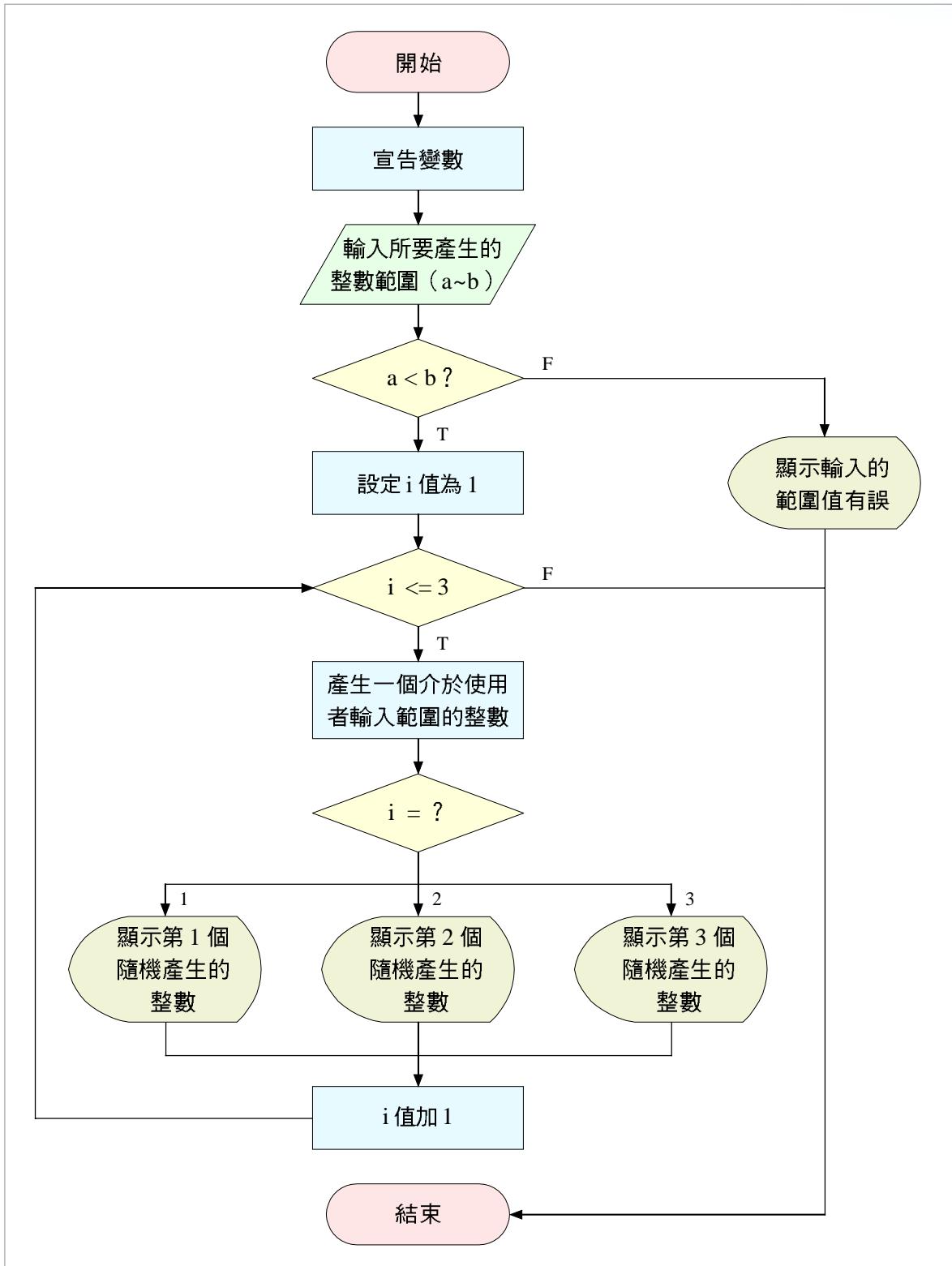


圖 8-64 隨機輸出 3 個整數的流程圖 利用 If-Then-Else 判斷敘述、Select Case 判斷敘述、及 For-Next 迴圈敘述來隨機產生 3 個介於使用者指定範圍的整數

操作實例 3 輸出 3 個介於使用者指定範圍的整數

請開啟已佈建好控制物件的範例檔案 "Ex08-02.vbp"，並依照下面的說明，撰寫「輸出 3 個介於使用者指定範圍的整數」的程式，然後測試程式執行的結果。

1. 編寫程式碼

- 1.1 雙按表單上的產生鈕，開啟Command1的Click事件程序之程式碼視窗
- 1.2 宣告變數lower、upper、i、Rnd_value為整數資料型別
- 1.3 將使用者在Text1及Text2中輸入的整數範圍，存入變數lower、upper中
- 1.4 判斷lower值是否小於upper值，若是，便設定執行For-Next迴圈3次，每次皆產生1個介於lower~upper之間的整數
- 1.5 設定依據變數i值介於不同範圍時，所要在文字方塊中顯示的整數；結束For-Next迴圈敘述
- 1.6 若lower值不小於upper值時，顯示錯誤訊息；設定結束If-Then-Else判斷敘述

```

Project1 - Form1 (程式碼)
Command1 Click

Private Sub Command1_Click()
    Dim lower As Integer, upper As Integer
    Dim i As Integer, Rnd_value As Integer
    lower = Val(Text1.Text)
    upper = Val(Text2.Text)
    If lower < upper Then
        For i = 1 To 3
            Randomize '用來設定亂數的序列，'以產生不同一組的亂數
            Rnd_Value = lower + Int(Rnd() * _
                (upper - lower + 1))
            Select Case i
                Case 1: Text3.Text = Rnd_value
                Case 2: Text4.Text = Rnd_value
                Case 3: Text5.Text = Rnd_value
            End Select
        Next i
    Else
        MsgBox "整數範圍輸入錯誤，重新輸入", _ 
            16, "輸入錯誤"
    End If
End Sub

```

續下頁 ➤

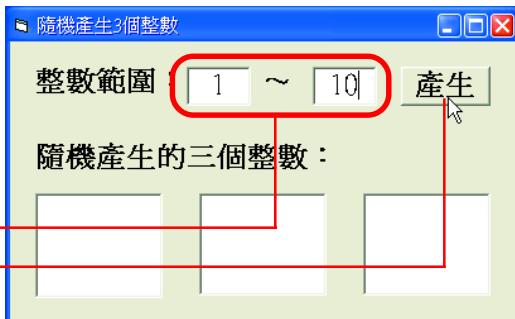
註：當程式敘述太長而無法撰寫於同一行時，可使用底線（_）來接續程式。

2. 測試程式

2.1 按開始鈕 ，執行程式

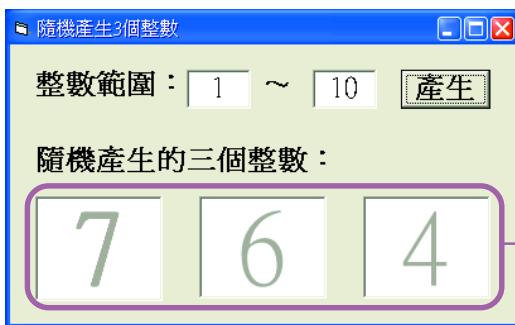
2.2 輸入所要產生的整數範圍，例如1~10

2.3 按產生鈕



3. 儲存專案

3.1 將表單儲存為 "Sp08-02.frm"，專案儲存為 "Sp08-02.vbp"



隨堂練習

1. 操作實例3的程式可能會產生重複的數值，你知道怎麼修改程式碼，來避免這種情形嗎？

提示：在行號90~100的Case2與Case3中，加入Do-Loop迴圈敘述來控制數值不重複。

2. 改寫操作實例3的程式碼，使程式可以產生6個介於1~49且不重複的整數數值，以模擬大樂透的開獎數字，最後將表單儲存為 "Sp08-j.frm"、專案儲存為 "Sp08-j.vbp"。

成果預覽：





程式碼撰寫的良好習慣

為了設計出正確且易於維護的程式，在程式設計時應儘量遵守下列幾點原則：

- 養成宣告變數及常數的習慣：**程式中所使用到的變數及常數若未事先宣告，將會因其資料型別不定，而增加程式偵錯的複雜度與困難度。
- 遵照一定的命名規則：**宣告變數及常數時，應該使用有意義的字元來命名，以利程式的閱讀及維護。例如：宣告一個存放國文成績的變數時，可將變數命名為 "chinese_grade"。
- 善用內縮格式：**程式語言中的某些程式敘述（例如 If 判斷敘述），須在開頭及結尾處使用特定的關鍵字將其中的敘述包圍起來；適當地將敘述內縮，可提高程式的可讀性（圖 8-65）。

```

    If a > b Then
        Print "a = ", a;
        Print "b = ", b
        Print "a 大於 b"
    End If
  
```

圖 8-65 善用內縮格式 適當地將程式敘述內縮，可提高程式的可讀性

- 善用副程式或自定函數：**對於需重複使用的程式敘述，應儘量寫成副程式或自定函數（將在下一節中介紹），以避免重複撰寫，而使得程式過於冗長。
- 避免使用 GoTo 敘述：**在程式設計時，使用 GoTo 敘述將會強制改變程式執行的流程，而使程式執行順序的追蹤及維護都較為困難；因此在撰寫程式時，應儘量避免使用此種敘述。
- 適當使用註解：**在宣告變數及撰寫特定功能的程式敘述區塊旁，加上適當的註解說明，有助於程式的閱讀與維護。

8-5 結構化程式設計

要建造一棟大樓時，通常必須由建築師進行整體的規劃設計，並繪製建築藍圖，再由建設公司依據藍圖，將土木、水電、裝潢等工程發包給專業人員來施工。若未能「整體規劃、按圖施工」，而僅憑一時之興，想到那兒，蓋到那兒，這種建築物有誰敢住呢？

結構化程式設計 (structured programming) 的技術與前述建築大樓必須先規劃再施工的概念很相近。這種技術可用來將整體規劃後的程式功能加以細分，並將細分後可共用的功能，撰寫成副程式；再將這些副程式組成完整的程式，以改善重複撰寫相同程式敘述，所造成程式不易維護的問題。圖 8-66 是使用結構化程式設計及未使用結構化程式設計之程式的比較。

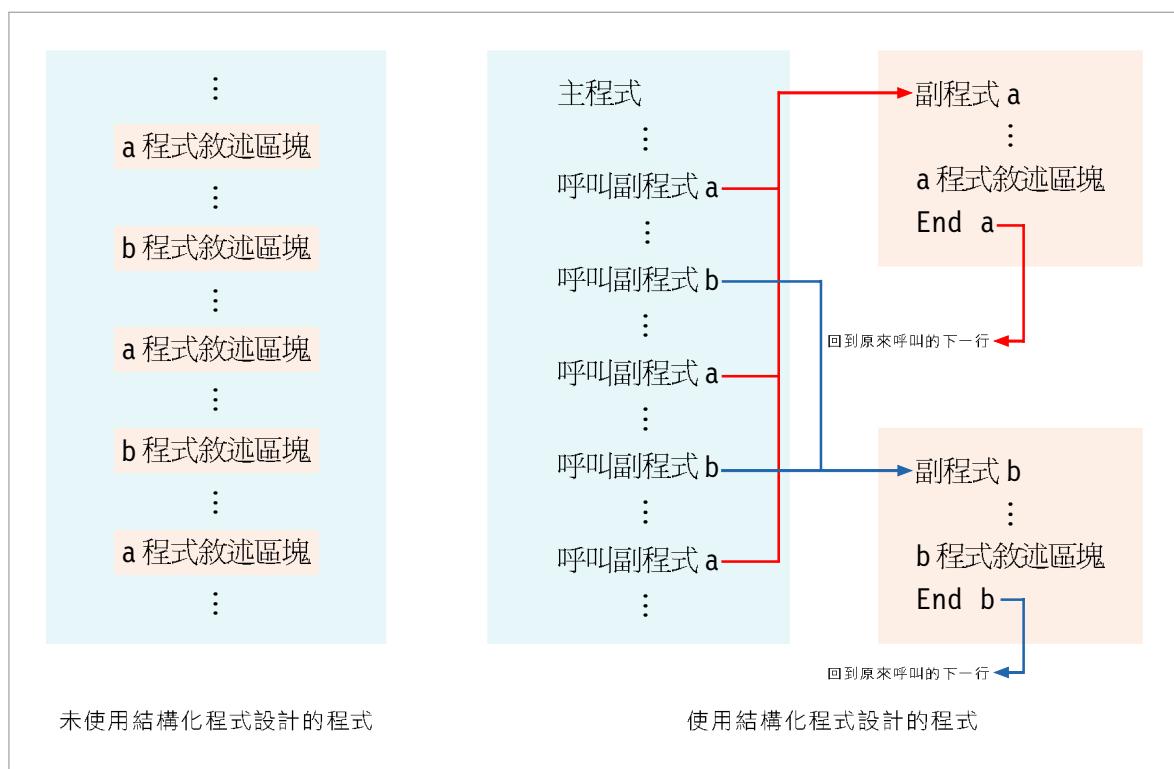


圖 8-66 使用及不使用結構化程式設計之比較 未使用結構化設計的程式，重複撰寫相同的程式敘述，造成程式不易維護；使用結構化設計的程式，可避免重複撰寫相同程式敘述，執行流程較簡潔，程式易於維護

8-5.1 結構化程式設計的概念

結構化程式設計技術通常是遵循由上到下的程式設計（top-down programming）、模組化的程式設計（modular programming）及使用程式基本結構等3個原則來進行程式設計的工作。下面將介紹這3個結構化程式設計的原則。

由上到下的程式設計

學校舉辦園遊會時，老師會先將班上同學分成佈置攤位、採購、販售……等組別，每組再由組長分配組員所需負責的工作。這種由上到下分派工作的方法，就類似由上到下的程式設計概念。

由上到下的程式設計是指在進行程式設計時，先由整個程式的主要功能開始設計，然後再依序往下設計各個子功能，直到最底層的功能設計完成為止。圖8-67即為一個由上到下的「統一入學測驗成績處理系統」之程式設計實例。

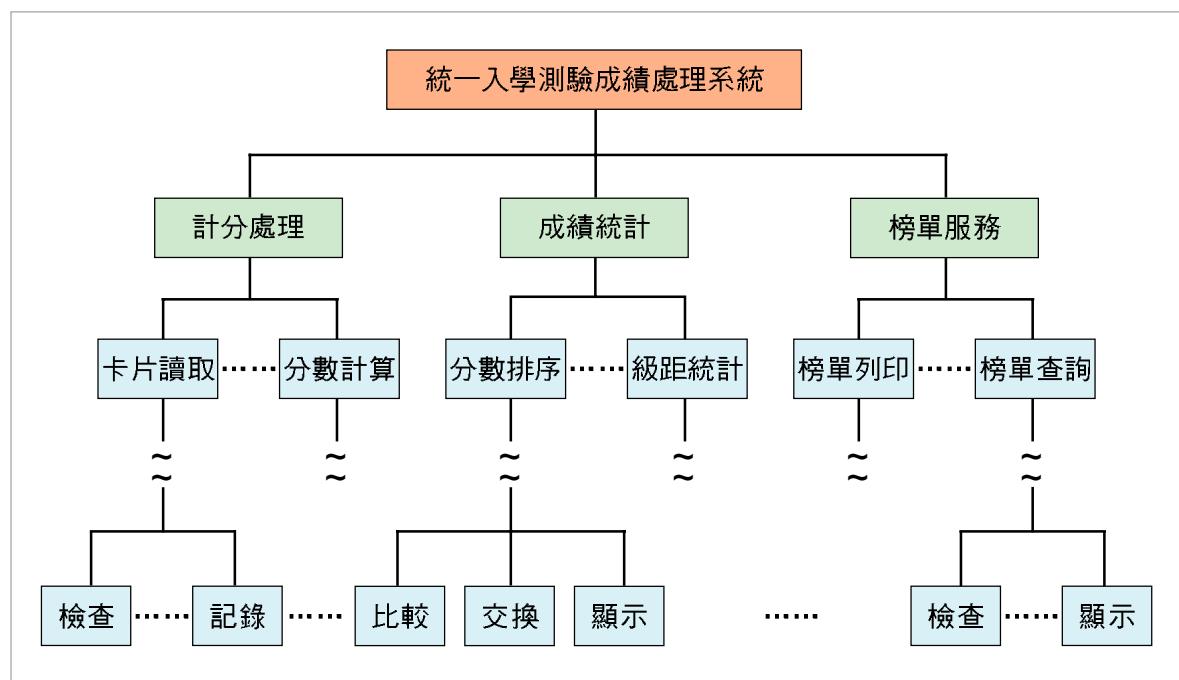


圖8-67 由上到下的設計實例 此系統分割為計分處理、成績統計、榜單服務等3個主要功能，再往下層層分割直到最底層的子功能為止，例如檢查、比較、交換、顯示等

模組化的程式設計

模組（module）通常是指一段具有某種特定功能、大小適中、容易閱讀及維護的程式。當我們遵循「由上到下的程式設計」原則，將程式切割成許多個子功能後，便可將這些子功能分別撰寫成模組程式（圖 8-68）。



圖 8-68 模組程式 模組就像積木，
由多個模組可組合成一個大型的程式

為了讓模組可供許多程式使用，並且易於測試及維護，在設計模組時應儘量依循以下 3 項原則：

- 設計功能單純的模組程式，以便提供給更多的程式使用。
- 儘量減少模組之間相互影響的程度，例如應避免模組在執行過程中，跳至另一模組執行，或模組間相互更改彼此的資料設定等，以使模組成為獨立的程式基本單元。
- 將模組控制在適當的大小（例如不超過 50 行敘述），以方便閱讀與維護。

使用程式基本結構

循序、條件及重複是程式的 3 種基本結構，在撰寫程式時，應儘量使用這 3 種程式的基本結構，並避免使用 GoTo 敘述來改變程式的執行流程，讓每一個程式敘述區段都只有一個入口與出口，如下頁圖 8-69 所示。

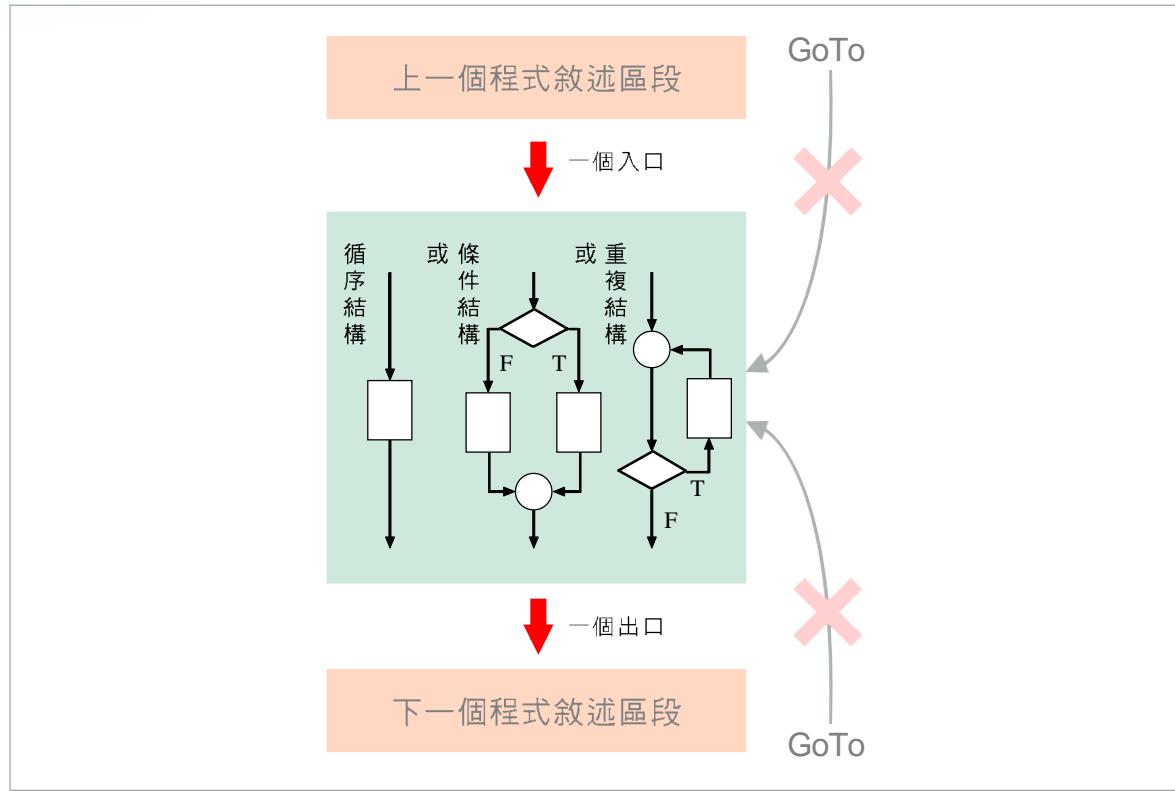


圖 8-69 結構化程式設計 使用程式基本結構，避免使用 GoTo 敘述，並遵守一個入口與出口的程式設計原則

隨堂練習

- () 1. 下列哪三個程式基本結構是結構化程式設計常用的控制結構 ?

 (A)循序、索引、選擇 (B)循序、條件、重複

 (C)索引、選擇、重複 (D)樹狀、循序、重複。
- () 2. 下列何者不是結構化程式設計的原則（方法）？

 (A)由上到下的設計方式

 (B)在程式中配合問題解決的需求，重複撰寫功能相同的程式敘述

 (C)避免使用 GoTo 敘述

 (D)採用模組化的設計技術。
- () 3. 下列何者不是模組化程式的優點？

 (A)可重複使用 (B)程式開發易於分工

 (C)程式不須經過翻譯即可執行 (D)程式易於維護與除錯。

*8-5.2 副程式與自定函數

模組化的程式設計是結構化程式設計的重要方法；在 Visual Basic 中，我們可以使用副程式（subprogram）及自定函數（user-defined function）將大型的程式分割成許多個功能單純的小型模組，以利程式撰寫、測試及維護。下面我們將分別說明副程式及自定函數的使用方法。

副程式

副程式是一段獨立的程式區塊，由 Sub 與 End Sub 之間的程式敘述所組成。每個副程式都有一個專屬的名稱；當程式中的其它區塊需要使用副程式的功能時，可以使用**呼叫**（call）的方式來執行這一段副程式（如圖 8-70 所示）。

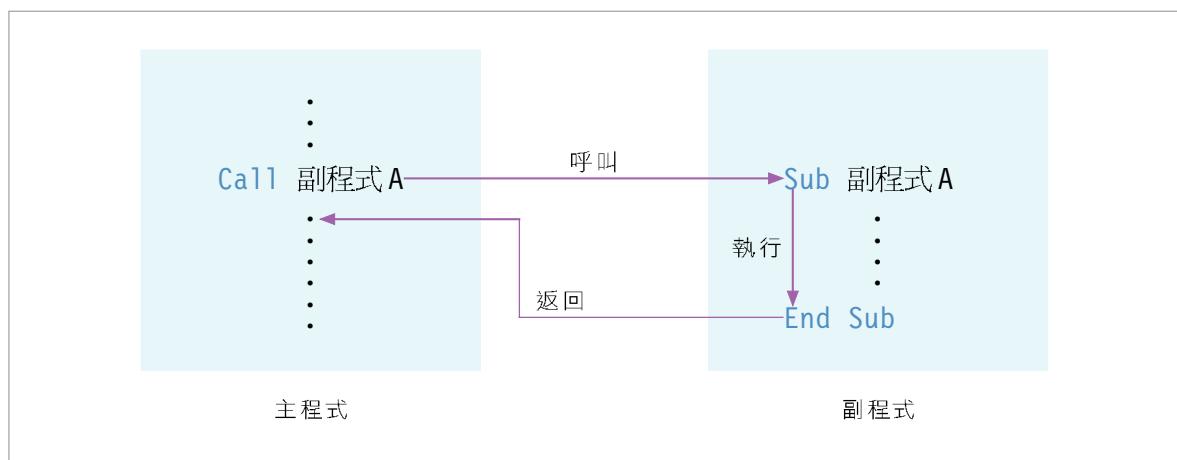


圖 8-70 主程式呼叫副程式的示意圖 主程式呼叫副程式時，會執行副程式中的程式碼；執行結束後，便會返回主程式，繼續執行下一行程式敘述

● 語法說明：

《定義副程式的語法》

```
Sub 副程式名稱[(參數 1 [As 資料型別], 參數 2 [As 資料型別], ……)]
```

 程式敘述區塊

```
End Sub
```

《呼叫副程式的語法》

Call 副程式名稱[(參數 1, 參數 2, ……)]

《說明》

- 定義副程式的敘述，必須撰寫在一般程序中。
- 每一個副程式都有一個專屬名稱，其命名規則與變數的命名規則相同，但須注意在同一個 Visual Basic 模組（如表單）中，副程式名稱不可重複。
- 撰寫副程式時可視程式設計的需要，決定是否使用參數。在 Sub 敘述中加入的參數，稱為**形式參數**（formal parameters）；在 Call 敘述中代入的參數，稱為**實際參數**（actual parameters）。
- 在 Call 敘述中的「實際參數」，會在執行 Call 敘述時傳入副程式中，並由「形式參數」承接。

範例：

圖 8-71 為具有「華氏溫度換算為攝氏溫度」功能的副程式。

```
Sub Change_temp(F As Integer)           'F 為形式參數
    Dim C As Integer
    C = (F - 32) * 5 / 9
    MsgBox "華氏 " & F & " 度 = 攝氏 " & C & " 度", , "溫度換算"
End Sub

Private Sub Form_Activate( )
    Dim F_temp As Integer
    F_temp = InputBox("請輸入華氏溫度：", "溫度換算")
    Call Change_temp(F_temp)             'F_temp 為實際參數
End Sub
```

(範例檔案 "Ch08-5a.vbp")

圖 8-71 副程式的範例 溫度換算

隨堂練習

1. 下面是兩個使用副程式的程式片段，請在空格處填上正確的敘述。

(1) 將攝氏溫度換算為華氏溫度（華氏溫度 = 攝氏溫度 * 9 / 5 + 32）。

```
Sub Change_temp(C As Integer)
    Dim F As Integer
    F = _____
    Print "攝氏 " & C & " 度 = 華氏 " & F & " 度"
End Sub

Private Sub Form_Activate( )
    Dim C_temp As Integer
    C_temp = InputBox("請輸入攝氏溫度：", "溫度換算")
    Call _____
End Sub
```

(2) 依據輸入的身份證字號來判斷性別（身份證字號中的第 2 個字元為 1，代表男性；為 2，代表女性）。

```
Sub Check_Id(S_id As String)
    Dim gender As Integer
    gender = Int(Mid(S_id, 2, 1))
    If _____ Then
        Print "男性"
    Else
        If _____ Then Print "女性"
    End If
End Sub

Private Sub Form_Activate( )
    Dim id As String
    id = InputBox("請輸入身份證字號：", "判斷性別")
    Call _____
End Sub
```

自定函數

在 Visual Basic 中，**自定函數**是使用者依照自己的需求，所自行撰寫的函數。其語法與副程式非常類似，差別在於自定函數被呼叫執行後，會傳回一個值給呼叫它的主程式（如圖 8-72 所示），而副程式不會。自定函數是由 Function 與 End Function 之間的程式敘述所組成。當程式中的其它區塊需要使用自定函數的功能時，便可以呼叫的方式來執行函數。

!—一點靈—

VB 的函數分為內建函數及自定函數兩種，其中內建函數已於 8-3.4 節中介紹。

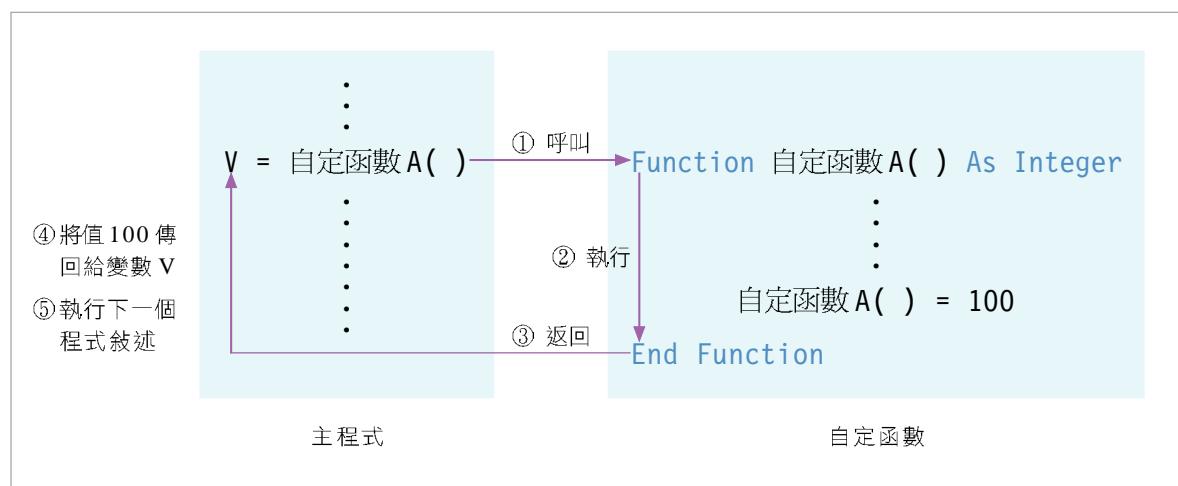


圖 8-72 主程式呼叫自定函數的示意圖 主程式呼叫自定函數時，會執行自定函數中的程式碼；執行結束後，會傳回一個值給呼叫它的程式敘述，再繼續往下執行

● 語法說明：

《定義自定函數的語法》

```

Function 自定函數名稱[(參數 1 [As 資料型別], 參數 2_
[As 資料型別], ……)] As 資料型別
    程式敘述區塊
    自定函數名稱 = 傳回值
End Function

```

《呼叫自定函數的語法》

變數名稱 = 自定函數名稱[(參數 1, 參數 2, ……)]

《說明》

1. 定義自定函數的敘述，必須撰寫在一般程序中。
2. 每一個自定函數都有一個專屬名稱，其命名規則與變數的命名規則相同，但須注意在同一個 Visual Basic 模組（如表單）中，自定函數名稱不可重複。
3. 在 Function 敘述中加入的參數，稱為**形式參數**；在呼叫自定函數敘述中代入的參數，稱為**實際參數**。
4. 在 Function 敘述的最後，可宣告自定函數的資料型別，亦即傳回值的資料型別。

● 範例：

圖 8-73 為具有「計算圓面積」功能的自定函數。

```
Function Cal_Circle(r As Single) As Single      'r 為形式參數
    Const pi = 3.14159
    Cal_Circle = r ^ 2 * pi
End Function

Private Sub Form_Activate()
    Dim rad As Single
    Dim area As Single
    rad = InputBox("請輸入圓的半徑：", "計算圓面積")
    area = Cal_Circle(rad)                      'rad 為實際參數
    MsgBox "半徑為 " & rad & " 的圓面積 = " & area, , "計算圓面積"
End Sub
```

(範例檔案 "Ch08-5b.vbp")

圖 8-73 自定函數的範例 依據使用者輸入之半徑，呼叫自定函數來計算圓面積

隨堂練習

1. 下面是 2 個使用自定函數的程式片段，請在空格處填入正確的敘述。

(1) 計算圓周長。

```
Function Cal_Cycle(r As Single) As _____
    Const pi = 3.14159
    Cal_Cycle = 2 * pi * r
End Function
```

```
Private Sub Form_Activate( )
    Dim rad As Single, p As Single
    rad = InputBox("請輸入圓半徑：", "計算圓周長")

    _____
    MsgBox "半徑為 " & rad & " 的圓周長 = " & p
End Sub
```

(2) 產生一個介於特定範圍的亂數。

```
Function Rnd_value(lower As Integer, upper As Integer) As _____
    Rnd_value = Int(Rnd * (upper - lower + 1)) + lower
End Function
```

```
Private Sub Form_Activate( )
    Dim Rnd_L As Integer, Rnd_U As Integer
    Dim Rnd_M As Integer
    Rnd_L = InputBox("亂數值的下限：", "設定亂數的範圍")
    Rnd_U = InputBox("亂數值的上限：", "設定亂數的範圍")

    _____
    Print Rnd_M
End Sub
```

8-6 程式的偵錯與除錯

當我們在撰寫程式時，可能會因為輸入的指令錯誤，或擬定解題時的邏輯思考錯誤，而造成程式無法順利執行，或產生錯誤的結果。因此，除了要學會如何寫程式之外，學會程式的偵錯與除錯也是很重要的。

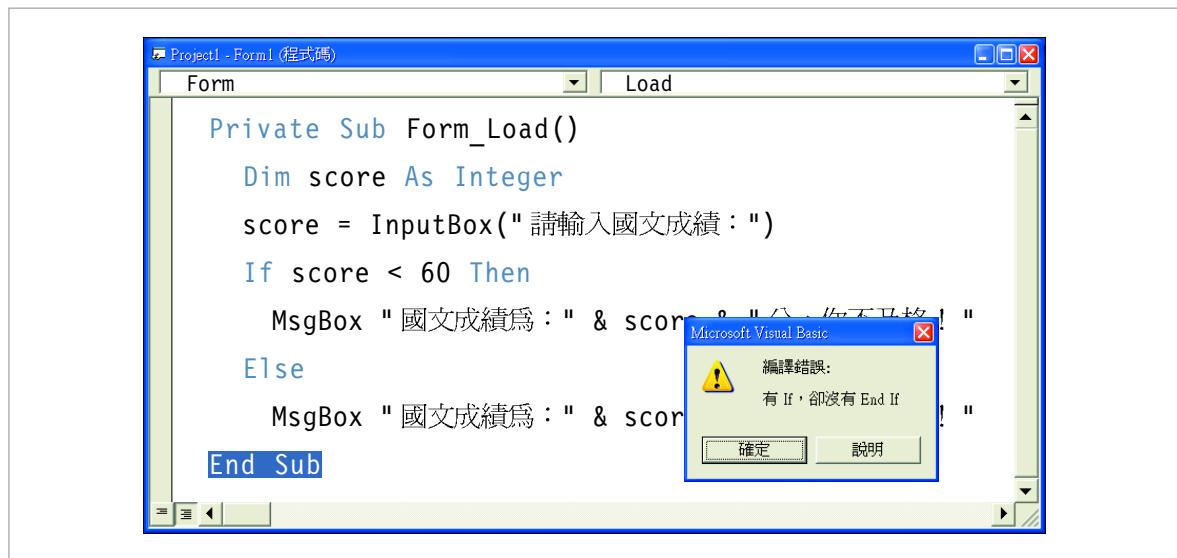
8-6.1 程式錯誤的種類

在 Visual Basic 中，主要的程式錯誤可分為編譯錯誤及執行期錯誤兩種，分別說明如下：

編譯錯誤

編譯錯誤是指在編譯程式時所發現的錯誤。這類型的錯誤通常是因為程式設計者未遵照程式語言的語法，輸入不正確的程式敘述所造成。

例如在 If 條件敘述之後，若未輸入 End If 的敘述時，就會產生編譯錯誤的訊息交談窗（如圖 8-74 所示）。



(範例檔案 "Ch08-6a.vbp")

圖 8-74 程式錯誤的範例 (I) 在 If 條件敘述之後需有 End If 的敘述

執行期錯誤

執行期錯誤則是指在執行程式時才出現的錯誤。這類型的錯誤可概分為兩種：第一種是系統錯誤，例如程式敘述中使用了不存在的硬體設備（如印表機）；第二種是邏輯錯誤，這種錯誤是因為程式設計者在構思程式執行流程時，邏輯思考出現錯誤所造成。例如圖 8-75 的程式中，由於 If 條件判斷敘述之 Then 及 Else 內的敘述錯置，因此產生了成績 85 分，卻顯示不及格的錯誤訊息。

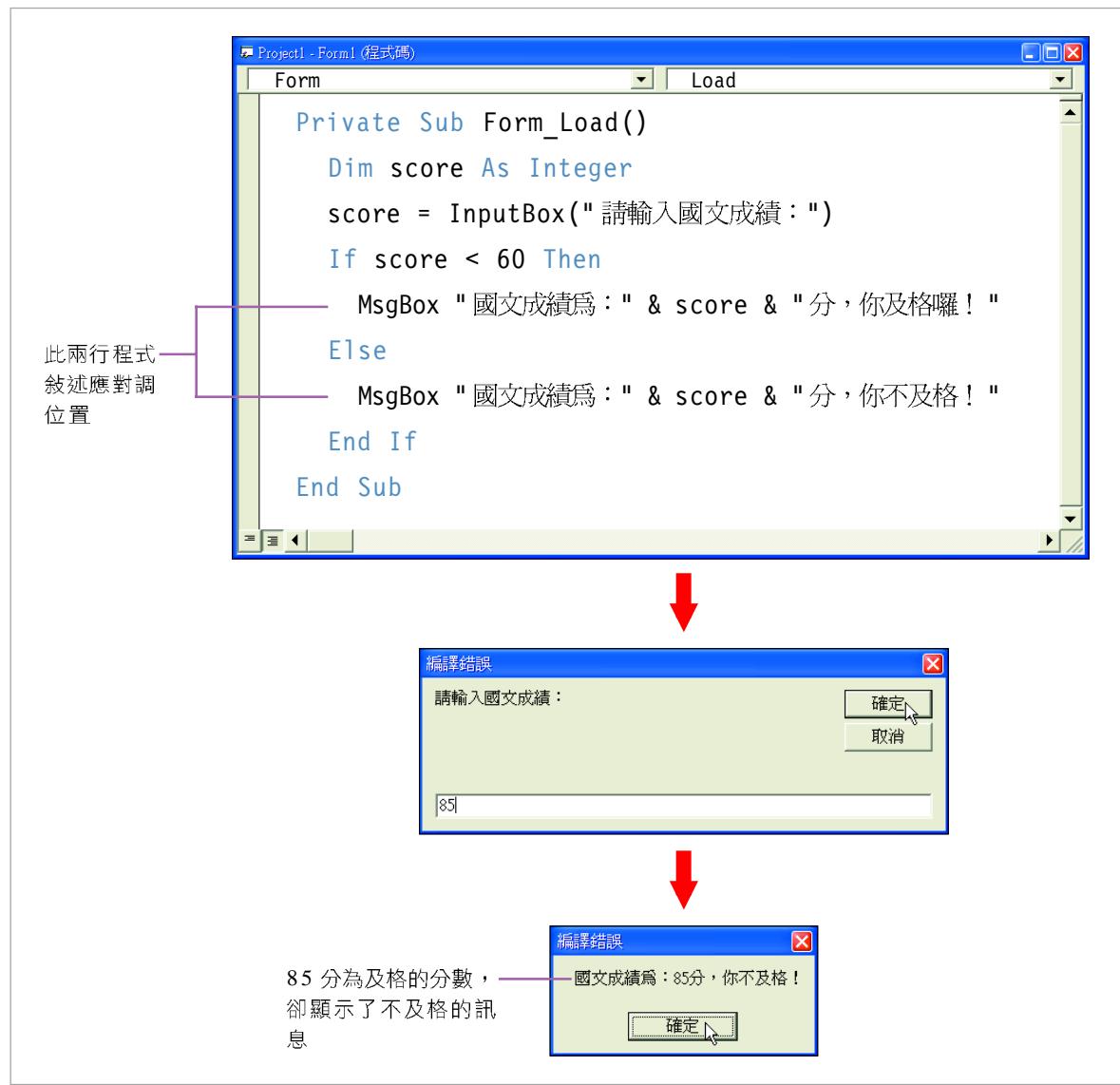


圖 8-75 程式錯誤的範例 (II) If 條件判斷敘述之 Then 及 Else 內的敘述錯置所產生的執行期錯誤

(範例檔案 "Ch08-6b.vbp")

8-6.2 程式除錯的工具

Visual Basic 提供的**偵錯**工具列（圖 8-76），可讓程式設計者逐步地對程式碼進行偵錯的工作，以便找出程式碼錯誤的地方，並掌握程式的執行結果。

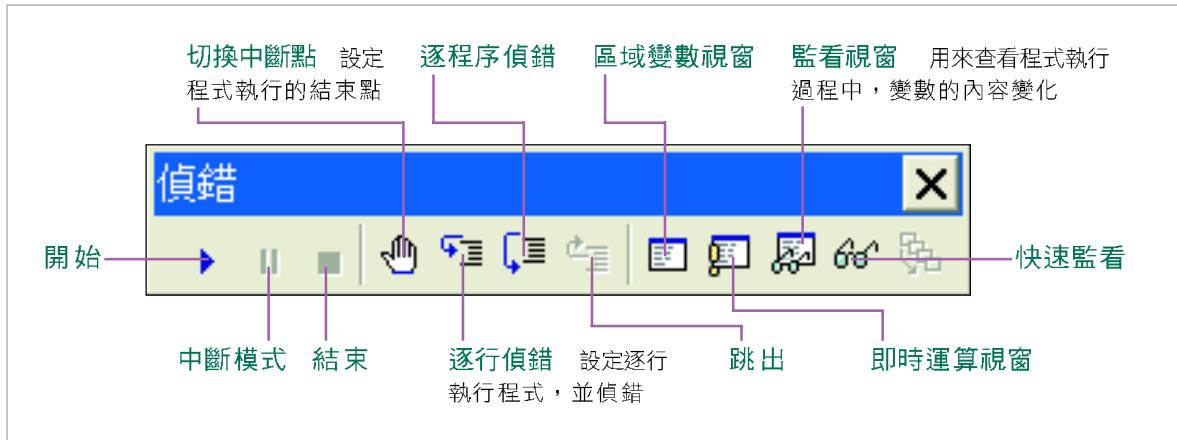


圖 8-76 偵錯工具列 Visual Basic 提供的程式除錯工具

操作實例 4 程式的偵錯與除錯

範例檔案 Ex08-03.vbp 具有判斷悠遊卡中的金額，並在金額不足時顯示一個交談窗讓使用者進行加值的功能。請開啟此範例檔案，並利用**偵錯**工具列逐行執行程式敘述，來查看程式執行過程中，變數 value_add 與變數 sum 的內容變化，並修改錯誤的程式碼。

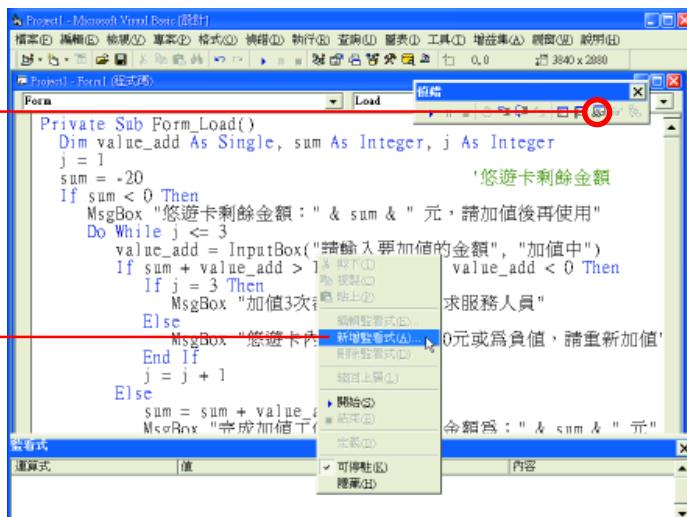
- 1 在工具列上按滑鼠右鍵，選按『**偵錯**』選項，以開啟**偵錯**工具列

```

Project - Microsoft Visual Basic [設計]
檔案(F) 檔案(I) 檔案(U) 車輛(D) 格式(O) 偵錯(D) 執行(E) 查詢(S) 單元(U) 工具(T) 增益集(G) 說明(M)
Project1 - Form1 (程式碼)
Form1 - Load
Private Sub Form_Load()
    Dim value_add As Single, sum As Integer, j As Integer
    j = 1
    sum = -20
    If sum < 0 Then
        MsgBox "悠遊卡剩餘金額：" & sum & " 元，請加值後再使用"
        Do While j <= 3
            value_add = InputBox("請輸入要加值的金額", "加值")
            If sum + value_add > 10000 Or sum + value_add < 0 Then
                If j = 3 Then
                    MsgBox "加值3次都未成功，請尋求服務人員"
                Else
                    MsgBox "悠遊卡內金額超過10,000元或為負值，請重新加值"
                End If
                j = j + 1
            Else
                sum = sum + value_add
                MsgBox "完成加值工作，悠遊卡目前金額為：" & sum & " 元"
            End If
        Loop
    Else
End Sub

```

2 按監看視窗鈕，開啟監看式視窗



3 在監看式視窗的任意處，按滑鼠右鍵，選按『新增監看式』選項，以開啟新增監看式交談窗，來新增要查看的變數名稱

4 在運算式欄輸入要查看的變數名稱 "sum"，並按確定鈕

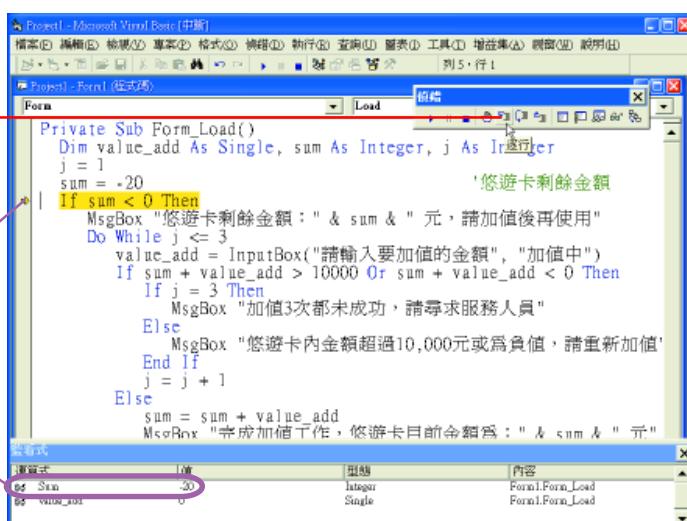


5 參照步驟3~4，在監看式視窗中，新增另一個要查看的變數名稱 "value_add"

6 連續按偵錯工具列中的逐行鈕（或 F8 鍵），逐行執行程式

此處會標示程式執行到的程式敘述

程式執行到第4行時，可發現在監看式視窗中的變數 "sum"，其值已變為 "-20"



- 7** 按偵錯工具列中的逐行鈕（或 F8 鍵），繼續執行程式；在出現提示使用者加值的交談窗按確定鈕

```

Private Sub Form_Load()
    Dim value_add As Single, sum As Integer, j As Integer
    j = 1
    sum = -20
    If sum < 0 Then
        MsgBox "悠遊卡剩餘金額：" & sum & " 元，請加值後再使用"
        Do While j <= 3
            value_add = InputBox("请输入要加值的金额", "加值中")
            If sum + value_add < 0 Then
                If j = 3 Then
                    MsgBox "悠遊卡剩餘金額：<0 元，請加值後再使用"
                    End If
                Else
                    MsgBox "悠遊卡內金額超過10,000元或為負值，請重新加值"
                End If
            j = j + 1
        Else
            sum = sum + value_add
            MsgBox "完成加值工作，悠遊卡目前金額為：" & sum & " 元"
        End If
    End Sub

```

- 8** 在 InputBox 交談窗中輸入要加值的金額（例如：1000），可發現程式會繼續執行迴圈內的敘述（又出現 InputBox 交談窗要使用者輸入金額）

```

Private Sub Form_Load()
    Dim value_add As Single, sum As Integer, j As Integer
    j = 1
    sum = -20
    If sum < 0 Then
        MsgBox "悠遊卡剩餘金額：" & sum & " 元，請加值後再使用"
        Do While j <= 3
            value_add = InputBox("请输入要加值的金额", "加值中")
            If sum + value_add < 0 Then
                If j = 3 Then
                    MsgBox "悠遊卡剩餘金額：<0 元，請加值後再使用"
                    End If
                Else
                    MsgBox "悠遊卡內金額超過10,000元或為負值，請重新加值"
                End If
            j = j + 1
        Else
            sum = sum + value_add
            MsgBox "完成加值工作，悠遊卡目前金額為：" & sum & " 元"
        End If
    End Sub

```

- 9** 在程式中，加入一行程式敘述 ($j = 4$)，其目的是要跳離迴圈

```

Private Sub Form_Load()
    Dim value_add As Single, sum As Integer, j As Integer
    j = 1
    sum = -20
    If sum < 0 Then
        MsgBox "悠遊卡剩餘金額：" & sum & " 元，請加值後再使用"
        Do While j <= 3
            value_add = InputBox("请输入要加值的金额", "加值中")
            If sum + value_add < 0 Then
                If j = 3 Then
                    MsgBox "悠遊卡剩餘金額：<0 元，請加值後再使用"
                    End If
                Else
                    MsgBox "悠遊卡內金額超過10,000元或為負值，請重新加值"
                End If
            j = j + 1
        Else
            sum = sum + value_add
            MsgBox "完成加值工作。悠遊卡目前金額為：" & sum & " 元"
        End If
    End Sub

```



在執行程式時，若只想針對某一段程式敘述進行偵錯，可以點選所要偵錯之程式敘述的最左端作為中斷點（如圖 8-77 所示），使程式執行至該行敘述時，可自動中斷，以供使用者監看程式的執行狀況。

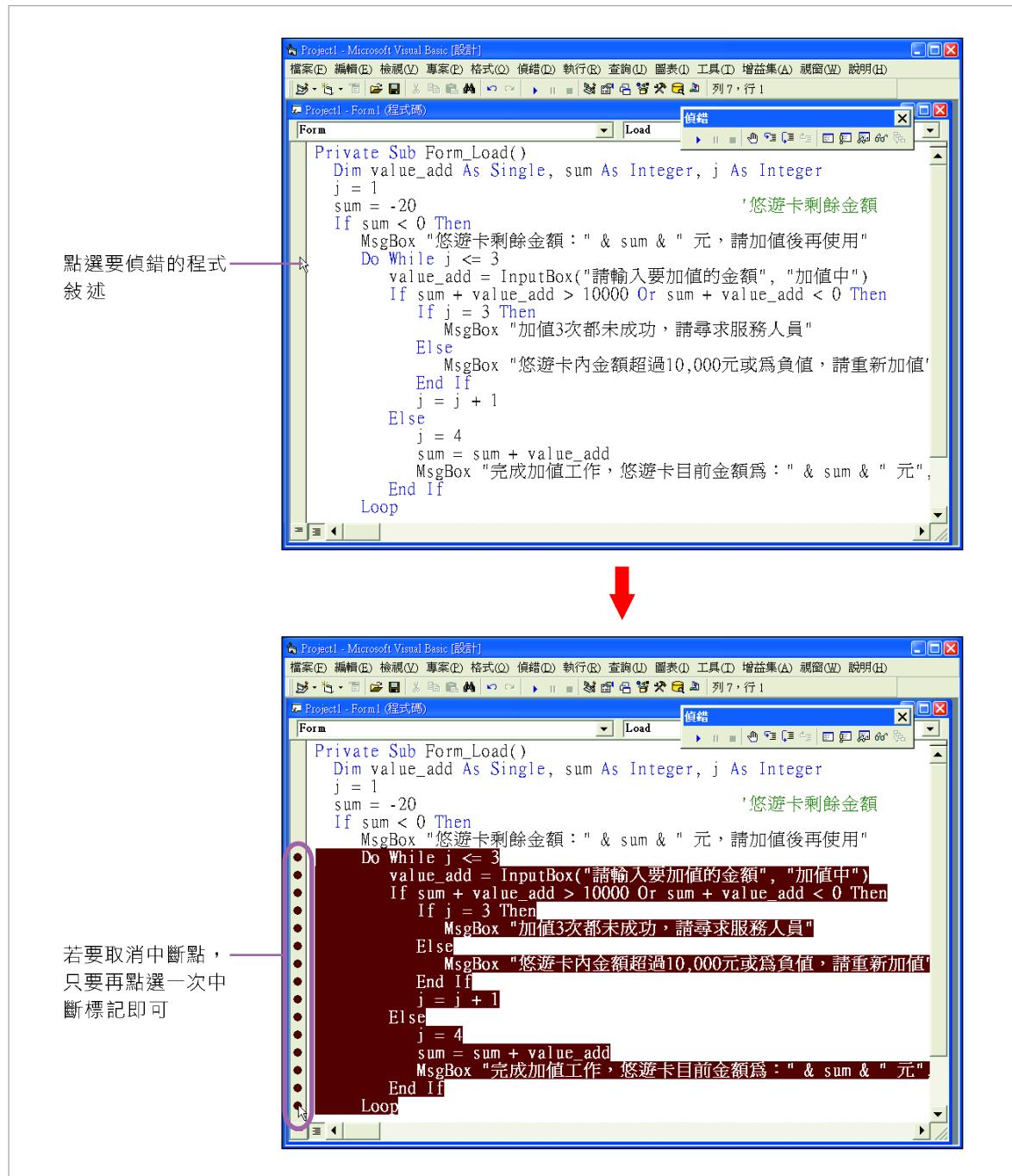


圖 8-77 使用中斷點進行程式的偵錯 點選程式敘述最左端可建立中斷點

課後評量

一、選擇題

- () 1. 有關高階語言與及低階語言的比較，下列哪一項正確？ (A)高階語言除錯性較困難 (B)低階語言學習較為容易 (C)高階語言可攜性較佳 (D)低階語言可讀性較高。
- () 2. 身份證統一編號適合使用下列哪一種類型的資料型別來表示？ (A)字串 (B) 數值 (C)布林 (D)日期。
- () 3. 偉聖和班上同學打算在園遊會中設置一個砸水球的攤位；為了增加砸水球的趣味性，偉聖想要利用 VB 撰寫一個隨機產生代表同學座號的程式，以指定同學來作為供人砸水球的活靶。請問他可以使用下列哪一個內建函數來隨機產生亂數值？ (A)Int() (B)Rnd() (C)Str() (D)Val()。
- () 4. 惠君正在考慮是否要參加夏令營；若好友 A 君和 B 君都要參加，他就要參加，但如果只有死黨 C 君要參加，他也會參加；請問下列哪一個 VB 運算式可用來表示此問題（假設 YN 表示惠君參加此項活動的意願，A、B、C 分別表示惠君 3 位好友的參加意願）？
 - (A)YN = (A And B) Or C (B)YN = A And (B Not C)
 - (C)YN = (A Or B) EQV C (D)YN = (A Xor B) and C。
- () 5. 儲存撰寫完成的 Visual Basic 程式，至少會產生哪兩種檔案？
 - (A).doc、.xls (B).dll、.sys (C).exe、.com (D).vbp、.frm。
- () 6. 每一個 Visual Basic 程式專案至少會包含幾個表單？ (A)1 個 (B)2 個 (C)3 個 (D)4 個。
- () 7. 執行下列 Visual Basic 程式片段的結果為何？ (A)12 (B)20 (C)24 (D)144。

```
Private Sub Form_Activate()
    Cls
    A = 4 : B = 2 : C = 3
    C = A + 2 * B ^ C
    Print C
End Sub
```

- () 8. 若使用 Visual Basic 來撰寫「最受歡迎的十大明星票選」程式，請問下列哪一個控制物件最有可能用來讓受訪者填寫自己欣賞明星的理由？ (A)Label (B) TextBox (C)CommandButton (D)Form。

()9. 執行下列 Visual Basic 程式片段的結果如何？ (A)5 (B)3 (C) $a > b$ (D) $a < b$ 。

```
a = 5 : b = 3
If a > b Then
    Print "a > b"
Else
    Print "a < b"
End If
```

()10. 執行下列 Visual Basic 程式片段的結果為何？ (A)0 (B)10 (C)-10 (D)120。

```
F = 1
For i = 5 To 1 Step -3
    F = F * i
Next i
Print F
```

二、多元練習

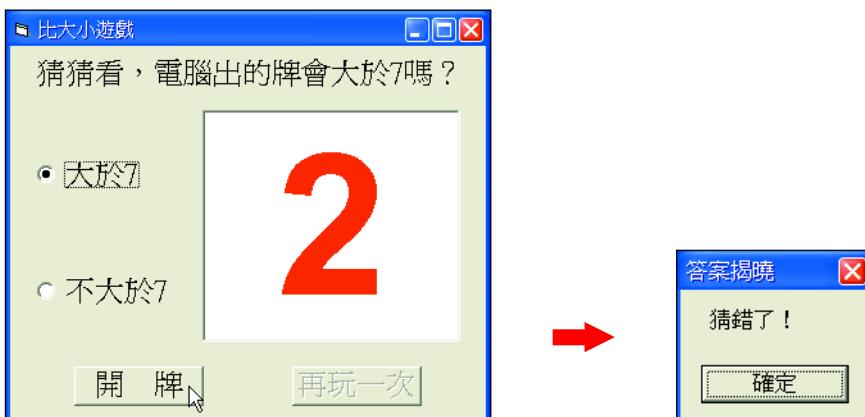
1. 想想看，下面 3 個日常生活問題的解決步驟，可使用哪一種控制結構來描述？

A.循序結構 B.條件結構 C.重複結構

- _____ (1) 外出帶傘：如果天氣預報的降雨指數超過 50%，就帶傘出門；否則就不帶傘出門。
- _____ (2) 隱形眼鏡配戴：首先要清潔雙手，再依序將隱形眼鏡配戴上左、右眼。
- _____ (3) 體能檢測：在計時 3 分鐘內，持續做仰臥起坐。

2. 開啟 "Ex08-k.vbp" 專案檔，撰寫一個「比大小遊戲」的程式，並利用 MsgBox 訊息交談窗顯示遊戲的勝負。

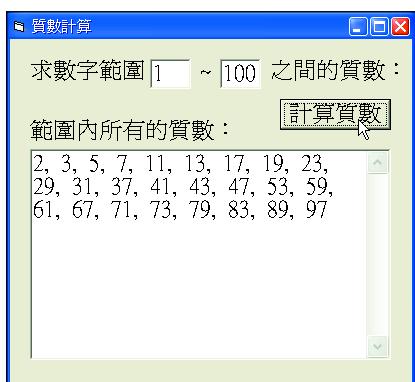
成果預覽：



- 提示：(1) 利用 Rnd()內建函數在 Label2 控制物件，隨機顯示 1~13 中的任一數字
- (2) 在開牌鈕控制物件中，利用 If 敘述判斷玩家是否猜對，並利用 MsgBox 敘述顯示遊戲的結果
- (3) 設定當開牌鈕被按下後，該按鈕即為不可使用的狀態，且再玩一次鈕更改為可使用的狀態
- (4) 設定當再玩一次鈕被按下後，該按鈕即為不可使用的狀態，且開牌鈕更改為可使用的狀態
3. 開啟 "Ex08-1.vbp" 專案檔，利用函數的功能，撰寫一個可用來解一元二次方程式 $ax^2 + bx + c = 0$ 的程式。
- 成果預覽：
-
- 提示：(1) 一元二次方程式的解為 $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ 與 $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$
- (2) 利用 If 敘述撰寫條件式：當 $b^2 - 4ac < 0$ 時，輸出無解；否則輸出 x 的值
- (3) 利用 Label 控制物件顯示解答
4. 開啟 "Ex08-m.vbp" 專案檔，撰寫一個可顯示使用者指定範圍內的所有質數之程式。

成果預覽：

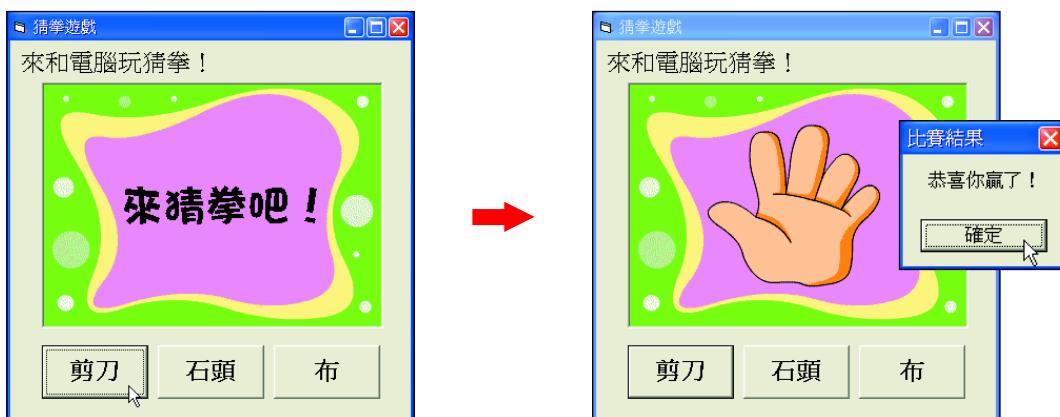
提示：



- (1) 質數是指只能被 1 與自己本身整除的數
(1 不是質數)
- (2) 利用 For-Next 敘述及 Mod 內建函數，判斷數字是否為質數
- (3) 利用 For-Next 及 If 敘述，計算數字範圍中有哪些質數
- (4) 將結果顯示在 TextBox 控制物件中

5. 開啟 "Ex08-n.vbp" 專案檔，撰寫一個「猜拳遊戲」的程式，最後利用 MsgBox 訊息交談窗顯示遊戲的勝負。

成果預覽：



提示：(1) 開啟之專案已佈建有下表所列之 4 個 PictureBox 控制物件（這些控制物件疊放在一起）

| 控制物件 | Picture1 | Picture2 | Picture3 | Picture4 |
|-------|----------|----------|----------|---------------|
| 顯示的圖片 | 剪刀 | 石頭 | 布 | 來猜拳吧！ (預設) |

- (2) 在副程式中依照猜拳的規則撰寫判斷遊戲輸贏的程式
 (3) 利用 Rnd() 內建函數，讓電腦隨機顯示剪刀、石頭或布的圖片
 (4) 利用 MsgBox 敘述顯示遊戲結果
6. 小傑利用 Select Case 敘述，撰寫一個可依據氣溫來提醒出門應穿著何種衣物的程式，但是程式無法正常執行，請幫他除錯，使程式可順利執行。

```
Private Sub Form_Activate()
    Dim temp As Integer
    temp = Int(InputBox("請輸入今天的氣溫 : "))
    Select temp
        Case Is < 15: Print "穿件厚外套"
        Case 15 ~ 20: Print "穿件薄外套"
        Case Is > 20: Print "穿件短袖 T 恤服裝"
    End Select
End Sub
```