第 5 章　我的小算盤

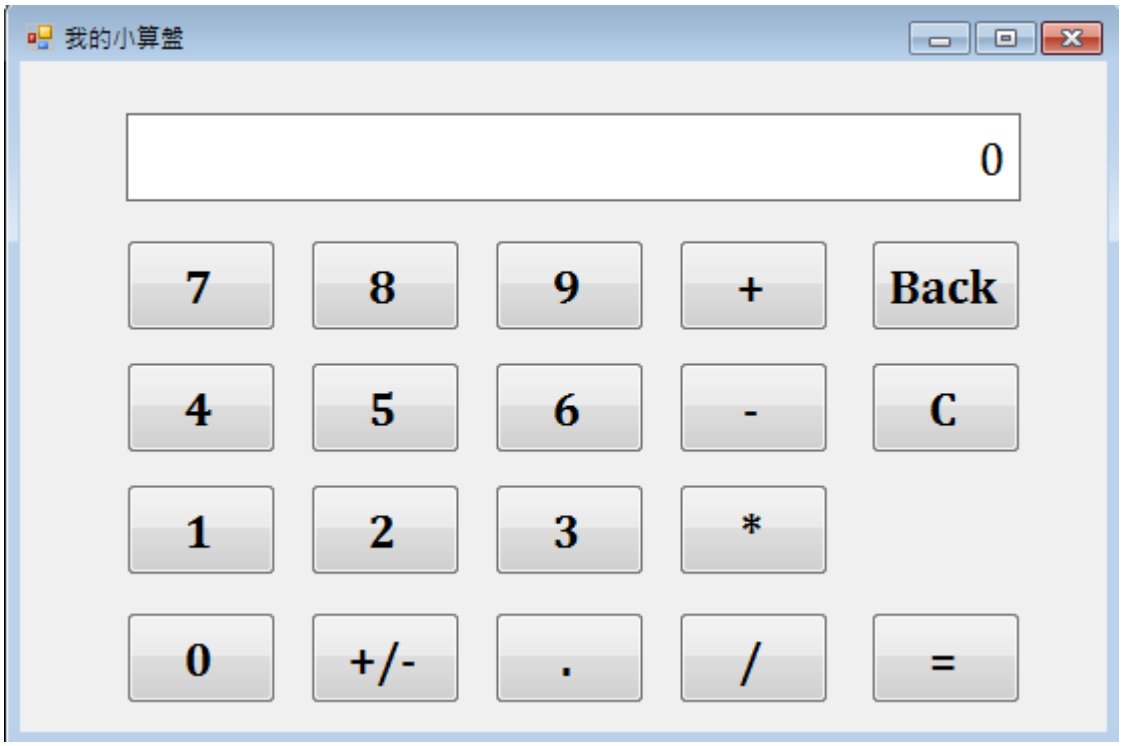
簡介：

小算盤也就是掌上型計算機的電腦版，本單元內容要完成的只是四則運算的數字輸入與計算流程，但以此為基礎，其他各式數學運算功能應該都不難逐步加入。此範例中許多按鍵功能有重複的部分，因此會講到事件副程式的共用；多數按鍵也必須可以使用對應的電腦鍵盤輸入，所以鍵盤事件也是重點。此外，還有一些文數字處理函數的運用，以及許多解決複雜操作流程的巧思，都值得仔細的研讀。

5-1 建立程式介面

[建立小算盤表單介面]

請開啓新專案，建立如下圖的程式操作介面：



請注意：上圖的數字面版應該使用 Label 物件製作，方法是將 Label 的 AutoSize 屬性設為 False，背景設為白色，TextAlign 設為靠右置中，邊框(BorderStyle)設為單線固定(FixedSingle)。雖然它長得很像 TextBox 但是實際上它應該不能直接用鍵盤鍵入數字，且 TextBox 物件的內部文字對齊功能不如 Label 完整，用 TextBox 其實無法呈現文字上下置中的效果。當然如果你堅持使用 TextBox 也可以，只是應該將 ReadOnly(唯讀)屬性設為 True，讓它不能直接打字，否則使用者在裡面打"ABC.XYZ"時要程式不當掉都難！

其他物件當然都是 Button 物件，請先依畫面所示寫上各物件應顯示的文字，再用格式對齊功能將按鍵們排列整齊，最後一一更改其名稱以方便後續程式寫作時辨認其角色：數

字面板取名為 T；0~9 按鍵依序取名為 b0~b9；正負號取名為 bSign；小數點鍵取名 bDot；加號為 bAdd；減號為 bSub；乘號為 bMul；除號為 bDiv；倒退鍵(Back)為 bBack；清除鍵(C)為 bClear；等號為 bEQ。

[變數名稱可以用中文嗎？]

當然程式內的變數或物件名稱完全可以自定，上面的取名只是簡化名稱，讓它們易於辨識，程式碼也可以簡短一些而已。有趣的是：其實如果全部都用中文取名也是可以的！因為目前的.NET 程式內建是使用雙位元組的 Unicode 編碼，變數名稱可以使用多國文字。以筆者的經驗建議是：雖然內部變數或物件使用非英文字很安全，但是專案、表單、類別或網頁等等較大的程式單元名稱最好避免用非英文字，原因是很多作業系統或舊版軟體對於非英文的名稱還是會不相容，不必因為堅持用中文而自找麻煩，讓程式安裝或執行時出現異常。

5-2 建立 0~9 數字的按鍵輸入

[數字按鍵的程式]

首先我們要製作可以輸入數字到顯示面板的程式，請先雙擊 b0 按鍵寫程式如下：

```
private void b0_Click(object sender, EventArgs e)
{
    if (T.Text == "0") { T.Text = ""; }
    T.Text += ((Button)sender).Text;
}
```

當我們輸入 0~9 的數字時，如果目前是預設值的數字"0"，那麼應該視為沒有內容(空字串"")，所以鍵入 1 就是 1 了，不應該是"01"。接著就是將 1 字碼加到目前內容(T.Text)的最後。程式中的 sender 就是事件觸發者，預設狀況下其實這段程式未必只能給 b0 專用，也可以讓其他按鍵的事件共用，所以事件觸發者不一定是 b0。我們稍後要讓其他數字鍵也共用此副程式，所以就必須用 sender 這個代名詞寫這段程式，如果你寫成 T.Text += b0.Text 就真的只會顯示數字"0"了！上面的寫法是先明確告訴電腦 sender 是一個 Button 物件，且將該按鍵上面的數字加入 T.Text(看板)。

[共用事件副程式]

因為其他 1~9 數字鍵的功能除了輸入的數字之外都一模一樣，可以共用前面的 b0 副程式，不必重複寫九個一樣的副程式。怎麼做呢？請先到方案總管，點選 Form1 項目下的 Form1.Designer.cs 檔案，打開檔案後展開『設計工具產生的程式碼』(按程式碼前面的加號)，如下圖：

接著在有關 b0 物鍵的程式碼中找到黑體字那一行，加以複製：

```
//
// b0
//
…
this.b0.TabIndex = 2;
this.b0.Text = "0";
this.b0.UseVisualStyleBackColor = true;
this.b0.Click += new System.EventHandler(this.b0_Click);
```

這就是讓我們按 b0 按鍵(觸發 b0.Click 事件)時會去找 b0_Click 副程式執行的關鍵程式碼。它的語法是 b0.Click 事件要加入一個新的事件副程式(EventHandler)，名稱為 this.b0_Click。請回到表單設計頁面，雙擊表單空白處進入 Form_Load 事件副程式，貼上這行程式，多次複製並修改程式變成這樣：

```
private void Form1_Load(object sender, EventArgs e)
{
    this.b1.Click += new System.EventHandler(this.b0_Click);
    this.b2.Click += new System.EventHandler(this.b0_Click);
    this.b3.Click += new System.EventHandler(this.b0_Click);
    this.b4.Click += new System.EventHandler(this.b0_Click);
    this.b5.Click += new System.EventHandler(this.b0_Click);
    this.b6.Click += new System.EventHandler(this.b0_Click);
    this.b7.Click += new System.EventHandler(this.b0_Click);
    this.b8.Click += new System.EventHandler(this.b0_Click);
    this.b9.Click += new System.EventHandler(this.b0_Click);
}
```

就是讓 b1.Click~b9.Click 等事件也通通都去執行 b0_Click 這個副程式。試試看，你現在可以自由輸入 0~9 的數字了，程式也會避開產生顯示"05"這類的不合理情況。

[寫在 Form1.Designer 也 OK]

讀者或許有注意到上述作法我們是到 Designer 檔案取得複製樣版，那麼將此程式碼直接寫在 Designer 檔案的其他物件建構程式碼裡面可以嗎？當然可以！事實上 Form1.cs 與 Form1.Designer.cs 是同一個表單的程式碼，只是為了閱讀方便才加以分檔，即使你將 Form1.cs 裡面的事件副程式搬到 Designer 裡面，程式依然可以順利執行。在此將共用事件的程式碼集中在 Form1.cs 的 Form_Load 目的是讓編輯與事後閱讀方便，大家很容易就可以知道這些按鍵共用一個副程式，不必去 Designer 裡面非常多的程碼中去找到它們。

5-3 建立 0~9 數字的鍵盤輸入

[建立鍵盤輸入程式]

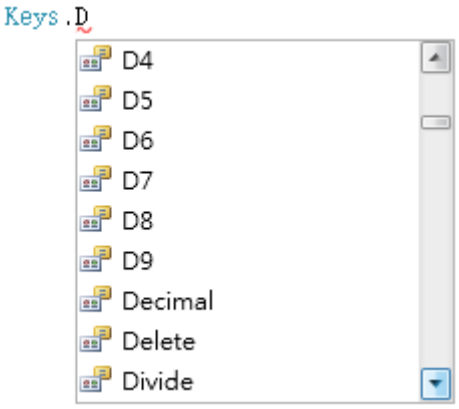
要建立鍵盤輸入機制，如前面的小小電子琴單元所示，應該先開啟表單的 KeyPreview 屬性，然後建立 KeyDown 事件程式框架。接下來你可能認為要像小小電字琴單元一樣，寫 10 個狀況(switch…case)來處理使用者分別輸入 0~9 數字的情形，甚至加上標準鍵盤的右方數字輸入區也有數字輸入鍵，那就應該有 20 個 case 了！其實用一點巧思創意，程式也可以簡短很多，請先填入以下程式碼：

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    String S = e.KeyData.ToString();//鍵盤名稱
    String D = S.Substring(S.Length - 1);//最後一個字元
    String Q = S.Substring(0, S.Length - 1);//前面的其他字元
    if ((Q == "D" && S.Length == 2) || Q == "NumPad"){
        if (T.Text == "0") { T.Text = ""; }
        T.Text += D;
    }
}
```

[用鍵盤名稱辨識數字鍵]

先測試一下，是不是鍵盤上所有的數字鍵(共 20 個)都可以用於數字輸入了？怎麼做到的呢？首先說明 e.KeyData 是鍵盤的『名稱』，如小小電子琴單元中介紹的 D1，D2…等就是鍵盤上方數字鍵的名稱(KeyData)，NumPad0~NumPad9 則是右側數字盤的鍵盤名稱。所以我們需要回應的是名稱以"D"或"NumPad"開頭的鍵盤，將它們末尾的數字(0~9)加入數字面板即可，但是以 D 或 NumPad 開頭的鍵盤就一定是數字嗎？如果不是的話，我的數字看板輸入一個非數字的東西不就當掉了？

確實如此！你可以去查資料看看一百多個鍵盤的 KeyData，也可以簡單的請問一下智慧感知(IntelliSense)系統，如下在程式碼合法區域內打 Keys.就會出現以下反應，你可以據此知道是否可能有其他名稱會混進來，再據以增加篩選條件。



結果是發現 D 重複的可能極高，但是限制名稱只有兩字元的鍵盤時，就確定只有 D0~D9 了！NumPad 部分則沒有重複的可能，所以篩選條件式就變成：D 開頭只有兩字元名稱的鍵盤與 NumPad+任何字元為名的鍵盤。

程式首先用 S 字串記下 e.KeyData，就是使用者按下的鍵盤名稱，接著取得 S 的最後一字元 D，以及此字元之前的其他字元組成的字串 Q。然後檢查 Q 是不是"D"而且 S 為二字元，或者 Q 是"NumPad"，兩種狀況之一成立時就是輸入數字了！可以將 D 變數視為輸入的 0~9 數字碼加入數字看板。當然碰到預設值 0 的時候也應忽略，與前面使用 Button 物件輸入時一樣。

於是乎，好像不必用到龐大的 switch…case 語法就將 20 個鍵盤的輸入程式寫好了！

當然接下來的加減乘除、小數點或倒退 Buton 鍵是不是也可以加入鍵盤的相對鍵呢？當然都可以！你應該視情況選擇程式解決方案，有時用 switch 也許比前面這個方法更方便。

5-4 加入小數點與改變正負鍵

[加入小數點的程式]

加小數點時只有一條規則，就是不能重複！雙擊小數點按鍵寫程式如下

```
private void bDot_Click(object sender, EventArgs e)
{
    if (T.Text.IndexOf(".") < 0)
    {
        T.Text = T.Text + ".";
    }
}
```

其中的 IndexOf 就是搜尋函數，找到目標(".")時傳回該字元的索引位置，找不到就傳回-1！所以如果答案小於 0 就是沒找到小數點應該加上小數點。否則就是已經有小數點不准再加，也就是按下去沒反應了！

[加入正副號的程式]

(+/-)按鍵的功能就是改變數值的正負號，因為一般習慣下+號是不顯示的，所以程式寫法不是直接換正負號；且和數字不同，正負號應該出現在數值字串之前，不是之後。程式碼如下：

```
private void bSign_Click(object sender, EventArgs e)
{
    if (double.Parse(T.Text) != 0)
    {
        if (T.Text.IndexOf("-") >= 0)
        {
            T.Text = T.Text.Replace("-", "");
        }
        else
        {
            T.Text = "-" + T.Text;
        }
    }
}
```

程式首先檢查目前數值是不是 0？因為 0 是無關正負的，毋須處理。如果不是 0 時，再檢視數值字串中目前有無"-"號，有的話就刪除它！Replace("-","")的意思是將字串 T.Text 中的"-"變成""(空字串)，也就是刪除負號變成正數了！相反的，原字串中如果沒有負號("-")，表示是正數，就應該給它加上負號變成負數了！

5-5 輸入運算元

[記住＋－×÷與第一個數字]

前面程式已經可以完整輸入任何數字，接著就要輸入加減乘除四個運算元之一了！先想像一下，按下加號時一般的計算機會如何反應？是不是一定要記住現有的數字，並清除數字面板，等待輸入下一個數字？而且我們希望加減乘除都用同一個副程式，所以也要記得你按的是加減乘除哪個按鍵？等到按下等於("="鍵)時才知道兩數應該做哪種運算？請先在加減乘除按鍵內分別將其 Tag 屬性設為："+"、 "-"、 "*" 與 "/" 四個字元。雙擊加號鍵寫程式如下：

```
double A;
String op;
private void bAdd_Click(object sender, EventArgs e)
{
    A = double.Parse(T.Text);
    T.Text = "0";
    op = ((Button)sender).Tag.ToString();
}
```

程式先宣告倍準數 A 用來記住第一個要運算的數字，再宣告一個 op 字串用來記住運算元是加減乘除的哪一個？副程式內就是用 A 讀入目前看板的數值，然後清除看板為 0，接著讓 op 紀錄目前點選按鍵(sender)的 Tag 值，在此就是"+"了！

[＋－×÷共用副程式]

接下來與數字鍵一樣，我們要讓減(-)、乘(*)與除(/)三個鍵共用這個副程式，所以再模仿 Designer.cs 裡面 bAdd.Click 事件的程式碼，在 Form_Load 事件副程式裡面加入粗體字的三行程式碼如下：

```
private void Form1_Load(object sender, EventArgs e)
{
    .........
    this.b8.Click += new System.EventHandler(this.bO_Click);
    this.b9.Click += new System.EventHandler(this.bO_Click);
    this.bSub.Click += new System.EventHandler(this.bAdd_Click);
    this.bMul.Click += new System.EventHandler(this.bAdd_Click);
    this.bDiv.Click += new System.EventHandler(this.bAdd_Click);
}
```

5-6 執行計算

[按下等號時的程式]

輸入運算元之後接著輸入第二個運算數字，然後按等號就應該把答案算出來了！雙擊等號按鍵寫程式如下：

```
private void bEQ_Click(object sender, EventArgs e)
{
    double B = double.Parse(T.Text);
    double C = 0;
    switch (op)
    {
        case "+": C = A + B; break;
        case "-": C = A - B; break;
        case "*": C = A * B; break;
        case "/": C = A / B; break;
    }
    T.Text = C.ToString();
    A = C;
}
```

程式首先用倍準數 B 取得輸入的第二個數字，也宣告變數 C 準備承接計算的答案，接著依據 op(之前輸入的加減乘除號內部的 Tag 值)決定做哪種運算？最後答案 C 會顯示於數值看板。同時也將原來儲存第一個計算數值的 A 變成 C，這是為了可以繼續做連續運算之用。譬如 2 加 2 之後變成 4，再按"+"與"2"鍵應該會算出 6 來，原因就是此時的第一個運算數值 A 已經變成 4 了！否則你會一直都算出一樣的答案 4！

5-7 清除與倒退鍵

[清除看板與內部記憶]

清除鍵的意義應該是重設所有的計算狀況為原始狀態，除了看板變回 0，內部與計算相關的參數也都應歸 0。程式碼如下：

```
private void bClear_Click(object sender, EventArgs e)
{
    T.Text = "0";
    A = 0;
    op = "";
}
```

[倒退鍵的程式]

倒退鍵(Back)就是將上一個輸入的數字字元刪除，但是如果已經退無可退時應該就是停在一個數字 0！而且如果退到將小數部分剛好刪完時，數字可能會呈現不符合慣例的"15."這樣的情況，就是有小數點但是後面沒有小數！此時應該自動也將小數點刪除。所以程式稍微囉嗦一點如下：

```
private void bBack_Click(object sender, EventArgs e)
{
    String S = T.Text;
    if (S.Length > 1)
    { T.Text = (double.Parse(S.Substring(0, S.Length - 1))).ToString(); }
    else
    { T.Text = "0"; }
}
```

程式首先用字串變數取得數值看板的內容，接著判斷其長度，如果大於 1 時做正常的刪除末尾字元的動作，Subsrting 是部分字串的意思，第一個參數是起始字元的索引，第二個參數是取出的字元數目，S.Length 是整個 S 字串的長度，所以(S.Substring(0, S.Length - 1)就是只排除末尾一個字元的字串了！相對的，如果字串長度只剩下一個字元時，不論內容為何都應該歸為 0 值！因此如果你在看板值已經為 0 時繼續按倒退鍵，程式就不會當掉，而是一直保持為 0！

大功告成，你的小算盤應該可以順利執行加減乘除四則運算了！但是不要期望太高，如果你輸入的數值有效位數大於 15 為，超過 double 資料型態的上限，程式會自動減少有效位數的。

5-8 進階挑戰

一、如何讓鍵盤的+、-、*、/、小數點與 Backspace 或 Esc 鍵也能加入程式反應？
提示：加入 KeyDown 程式裡各個鍵盤的回應程式碼。

二、如何加入倒數、開根號、次方與三角函數等功能？
提示：將目前數值(X)直接做相關計算(如 1/X)，複雜函數可使用 Math 數學函數物件。

三、如何模仿小算盤程式將目前顯示數值拷貝至其他程式？
提示：使用剪貼簿 Clipboard 物件，試試程式碼 Clipboard.SetDataObject(T.Text);。

課後閱讀

事件副程式的運作機制

[物件與事件副程式的連結]

幾個單元下來我們好像已經對事件副程式相當熟悉，就是對著某個物件點兩下滑鼠，產生一個副程式框架，然後寫程式就對了！但是本單元應該有讓你見識到事件副程式不簡單的一面！當你在雙擊物件，或到屬性視窗的閃電標記下找到事件種類，點擊產生程式框架時，有幾件事會同時發生：一個是產生程式碼框架標題(如 b0_Click)，一個是產生副程式中適當的參數群(如 object sender, EventArgs e)，另一個是在 Designer.cs 檔案中將此副程式註冊給我們設計中的物件。，如：

```
private void b0_Click(object sender, EventArgs e)
```

副程式標題會顯示出物件名稱與事件的種類(b0_Click)，但是其實並不是因為這樣物件與事件就建立關係了！這個名字亂改都可以的，真正重要的是 Designer.cs 檔案中註冊副程式的那一行，如：

```
this.b0.Click += new System.EventHandler(this.b0_Click);
```

是這種程式碼讓某物件的事件與副程式相連結的，所以我們也可以刪除此行程式讓事件不再執行副程式，或者複製此行改寫讓其他物件使用同一個副程式。但是也必須注意副程式的參數，第一個永遠是 object sender，object 是指物件資料型別， sender 是觸發事件者，事實上所有事件副程式中此參數都一樣的，但是第二個參數就不同了！雖然都稱為 e，但是細看前面的資料型別很多樣的，鍵盤事件是 KeyEventArgs，Click 事件是 EventArgs，這些都代表不同的參數『集合』，也就是 e 裡面可以有很多次級參數，譬如鍵盤事件有 e.KeyCode 與 e.KeyData，也有 KeyValue。

所以其實完全手動建立事件副程式的框架與程式碼不是不可以，而是必須兼顧到寫對事件的 e 參數型別種類，還要自己到 Designer.cs 檔案中加入事件副程式的登記，將某物件的事件與它的事件副程式進行連結。

再看看連結副程式的程式碼 new System.EventHandler，我們也可以發現在此副程式也被當作一個物件，使用 new 關鍵字來建置，也就等於每個共用此副程式的物件，其實都是拿走一個實體拷貝，然後各用各的。『共用』這個名詞或許不是絕對精確，但是以使用者觀點執行結果是一樣的。對事件副程式有了如此的認識，以後處理它們的程式碼時，應該就不會再有害怕弄錯的不安全感了。

較複雜的邏輯判斷式

[多個條件時的”而且”與”或者”]

程式要變聰明就是要寫程式的人先夠聰明，想清楚處理事情的邏輯規則，之後還要將

想好的邏輯判斷用程式的 if…else 或 switch…case 等等語法表達出來。如果是簡單一點的狀況，學會邏輯比較的等於(==)、大於(>)、小於(<)和不等於(!=)等基本符號就夠了！但是邏輯複雜，條件狀況不只一種，或者有條件有優先次序時就有點累了！我們至少必須學會 AND(&&)與 OR(||)兩種記號，如果兩條件都必須符合才執行程式時就要用：if(條件一 && 條件二)。如果只需一種狀況符合就可執行時就是：if(條件一 || 條件二) 了！

在本範例中我們也有用到兩層的 if 如下：

```
if (T.Text != "0")
{
    if (T.Text.IndexOf("-") >= 0)
    {
        T.Text = T.Text.Replace("-", "");
    }
    else
    {
        T.Text = "-" + T.Text;
    }
}
```

這就是有條件有優先次序時的狀況，有些條件需要先符合之後再考慮其他條件，這種情況常常也可以使用 AND 的方式寫，譬如上面程式也可以寫成：

```
if (T.Text != "0" && T.Text.IndexOf("-") >= 0)
{
    T.Text = T.Text.Replace("-", "");
}
else
{
    if (T.Text != "0")
    {
        T.Text = "-" + T.Text;
    }
}
```

就像數學題，解法常常不只一種，就看你的腦袋思路如何進行了！重點是邏輯要正確，而且沒有漏洞！每一種可能狀況都應該想到，最忌諱的事情是程式在執行時遇到你沒想到的一種情形，因而當掉或反應與預期不同！所以竹科那些高薪工程師們容易過勞死，就是在加班努力測試出電子產品的所有可能運作狀況，永遠不會卡住當機，而且操作結果絕無異常的機器(程式)就是優良產品了！