

以 C# 語言設計視覺化的氣泡排序程式

國立台灣師大附中 李啓龍老師

➤ 程式語言與 C#簡介：

程式語言就跟中文、英文這些語言一樣，用來當作與電腦溝通的媒介，例如當我們想與美國人對話時，可能會使用英文，那如果我們想和電腦溝通時，我們就得使用程式語言，讓電腦幫助我們完成想做的事情。

程式語言的種類非常多，例如被歸類為低階語言的機器語言、組合語言...等等，或是被歸類為高階語言的：C/C++、Java、Python、C#、Basic...等語言。其中低階語言在電腦中的執行效率比高階語言的效率，且對於電腦硬體的控制程度也較高，不過由於低階語言的語法結構與人類的語言使用習慣相去較遠，故低階語言在開發、閱讀與維護上，都比高階語言困難許多。而高階語言則是較偏敘述性的語言，其語法結構與人類的語法邏輯使用習慣較為接近，因此也比較易於開發、閱讀、除錯與維護，也是一般專案開發時較常使用的程式語言種類。然而高階語言的種類繁多，C#語言又有什麼特色值得我們選擇使用呢？

C#語言起源於微軟的.NET 架構計畫，當時微軟內部開發.NET 架構使用的程式語言，就是 C#語言的前身，也可以說 C#語言是為了.NET 架構而生的產物。相對於一般校園或公司開發使用率較高的 C/C++或是 Java、Basic 語言來說，C#可以說是非常年輕的程式語言。

微軟在設計 C#時，廣納百川並擷取各既存語言的優點，努力推動 C#成為微軟新一代程式設計的標準語言。從 C#名稱上就可以看出來，C#語言建立在 C/C++語法基礎上，然而 C#也兼有 Java 的精簡、嚴謹以及 C++/Java 兩種語言都強調的物件導向概念，C#一次包含了幾乎所有 C++與 Java 當中你能想到的特色與功能。另外在.NET 架構的推動下，微軟也透過 Visual C#程式設計軟體讓 C#同時擁有和 Visual Basic 一樣簡單方便的微軟視窗程式設計能力。

我們將 C#語言的優點條列如下：

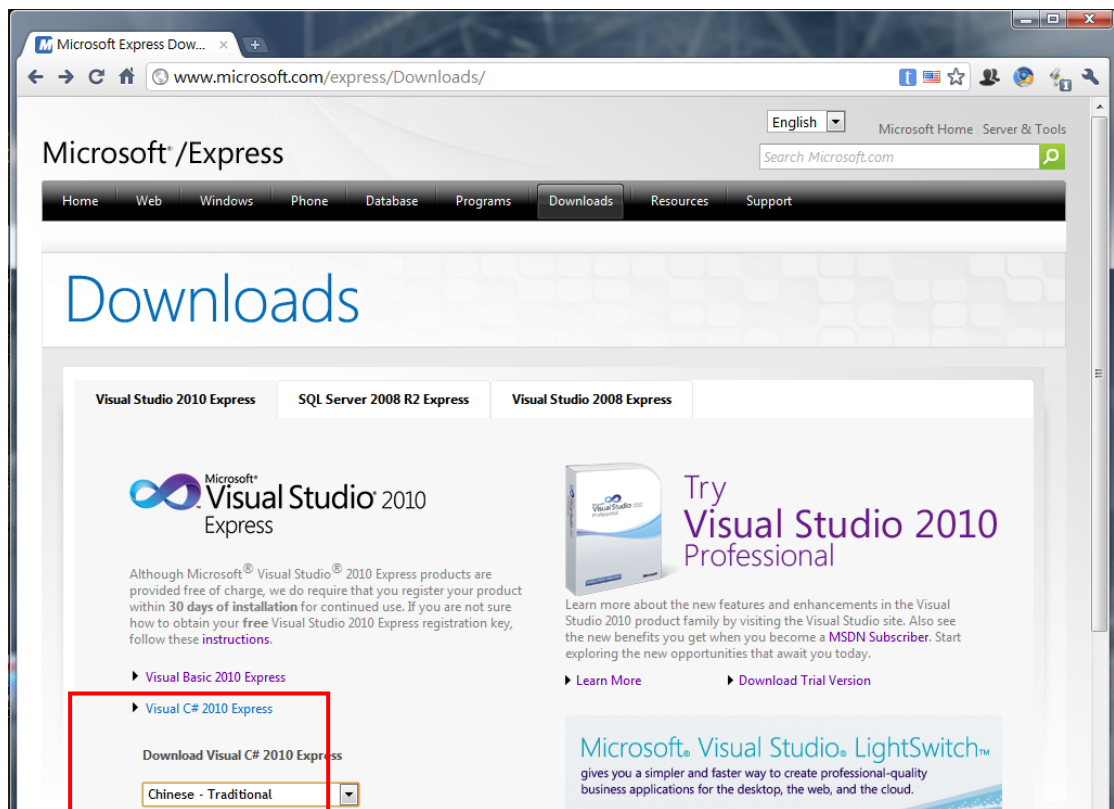
1. 程式初學者或指導者常常在 C/C++語言以及 Visual Basic 語言之中猶豫，C#語言正好兼有兩者的優點，學 C#可以讓我們透過 Visual C#得到與 Visual Basic 一樣強大的 Windows 視窗程式開發能力，由於語法與 C/C++、Java 幾乎相同，也不用擔心以後與主流 C/C++語言、Java 語言的銜接問題。
2. 對於已經有 C/C++或是 Java 基礎的程式設計師來說，轉換到 C#平台更是沒有障礙，程式語法與使用概念幾乎相同，使用 Visual C#更大幅簡化了在 Windows 底下開發視窗程式的難度，由於 Windows 與 C#都出自於微軟之手，使用 C#較其他語言，更能得到 Windows 作業系統的原生支援。
3. C#語言已經成為微軟的新一代開發標準，如微軟新一代網頁設計方案 Web

Developer 或是與 Adobe Flash 對抗的 Silverlight，還有能夠設計 PC、Xbox360 遊戲的 Game Studio 軟體套件，C# 語言都對這些微軟的新產品，有著全面或甚至獨佔性的支援。

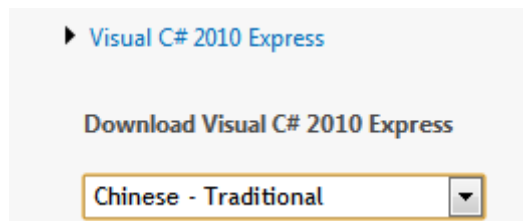
➤ 下載與安裝 Visual C# 2010 Express 版軟體：

Visual C# 2010 Express 版軟體是一套免費的中文化軟體，下載 Visual C# 2010 Express 安裝檔與安裝的程序說明如下：

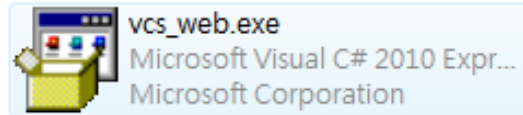
Step1: 請直接在瀏覽器輸入網址 <http://www.microsoft.com/express/Downloads/>，即可到達下載頁面。



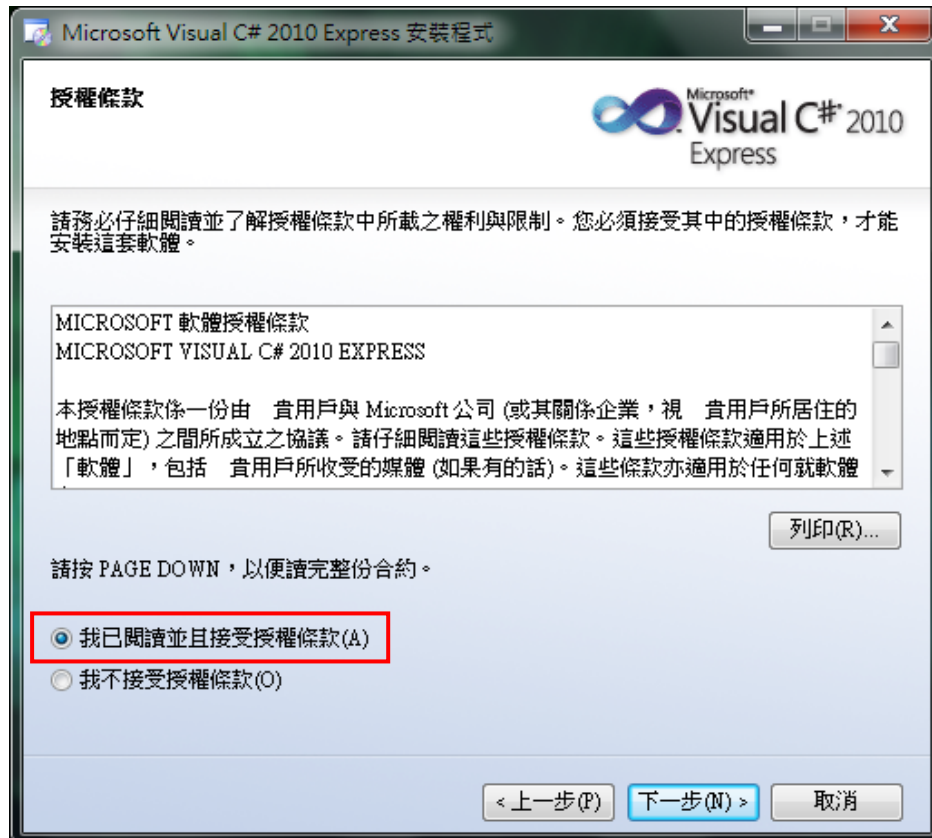
Step2: Visual C# 2010 Express 提供多國語系版本。若要下載繁體中文版，請在下載的頁面上，選擇「Chinese - Traditional」選項，就可以下載繁體中文版。選取好「Chinese - Traditional」選項後，瀏覽器就會自動開始下載安裝檔案。



Step3: 下載完成後會得到一個安裝檔「vcs_web.exe」，用滑鼠左鍵雙擊即可開始安裝。



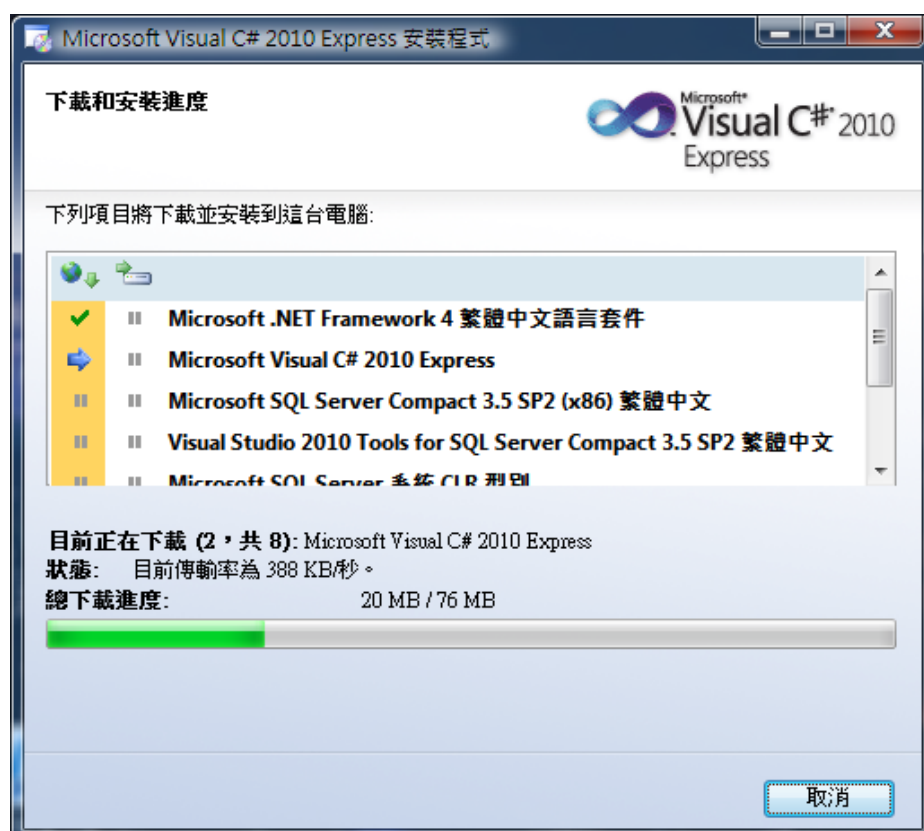
Step4: 安裝的過程大致都只需按「下一步」按鈕即可進行，但其中幾個步驟可選擇的安裝細節，必須稍微注意一下。首先，在「使用者授權合約」部分，我們要記得勾選「我已閱讀並且接受授權合約中的條款」選項，才能繼續安裝。



Step5: 選擇安裝的目的資料夾，預設的資料夾為「C:\Program Files\Microsoft Visual Studio 10.0/」，使用者也可以依據使用需求，另外設定檔案安裝的資料夾位置。



以下就是下載和安裝的進度畫面。



Step6: 安裝程式完成後按下「結束」鍵，即可順利完成「Microsoft Visual C# 2010

Express」的安裝。



➤ 氣泡排序法簡介：

氣泡排序法（Bubble Sort）是一個相當有名的排序方式，其實際執行過程如下：假設今天有 N 個元素要排序，我們要將最大的元素放置在最後 1 位，實作的步驟為：

Step 1：比較陣列中第 1 與第 2 個元素，若第 1 個比第 2 個大，則兩者交換。

Step 2：比較陣列中第 2 與第 3 個元素，若第 2 個比第 3 個大，則兩者交換。

...

Step N ：比較陣列中第 $N-1$ 與第 N 個元素，若第 $N-1$ 個比第 N 個大，則兩者交換。

氣泡排序法的作法就類似氣泡往上浮一般，將大的數字一路換到陣列最後方，故做完此步驟後，陣列中最大的元素就放置在第 N 個元素了，然後重複這個步驟，讓次大的元素「浮到」 $N-1$ 的位置，總共做 $N-1$ 次類似的動作後，就可將整個陣列由小至大排序了。

➤ 氣泡排序法演示視窗程式：

程式範例：氣泡排序法演示視窗程式

【學習重點】：控制項元件的陣列宣告

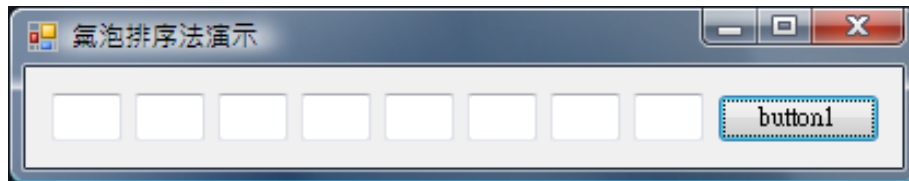
一、程式設計目標

在視窗程式中設定 8 個文字方塊讓使用者自行填入數字，代表將被排序的數

列，按下「開始排序」的按鈕後，配合計時器元件，將氣泡排序法的過程以每 1 秒一步的方式，直接對程式畫面上文字方塊的數列進行改變，程式會利用改變文字方塊底色的方式，以及將按鈕文字顯示為目前步驟的作法，讓使用者可以清楚地看到整個氣泡排序的變化過程。

二、表單配置

新增 8 個 TextBox 與 1 個 Button，大約配置如下，最後記得也要新增 1 個 Timer 元件來計時。

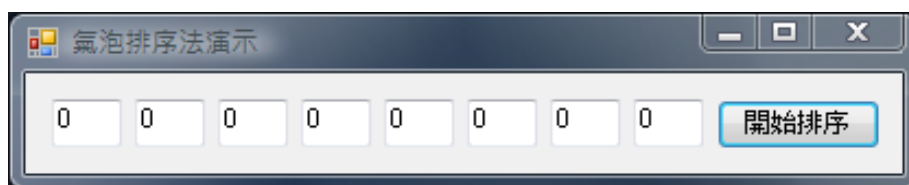


然後依照以下表格更改各物件屬性值：

物件	Name 屬性	Text 屬性
button1	startButton	開始排序
textBox1~8	numText1~8	0
timer1	*可以不用變更	*無此項屬性

計時器元件 timer1 的屬性 Interval 預設值為 100，只有 0.1 秒，在這麼快的速度下不利於使用者觀察排序過程，所以我們在此根據設計目標修改 timer1 的 Interval 屬性值為 1000。

更改後的表單配置大約如下：



三、程式碼撰寫

```

01 private void startButton_Click(object sender, EventArgs e)
02 {
03     timer1.Enabled = true;
04     startButton.Text = "1";
05 }
06
07 private void timer1_Tick(object sender, EventArgs e)
08 {
09     TextBox[] textArray = new TextBox[] { numText1, numText2, numText3,

```

```

numText4, numText5, numText6, numText7, numText8 };
10     for (int i = 0; i < 8; i++)
11         textArray[i].BackColor = SystemColors.Window;
12
13     int check = int.Parse(startButton.Text);
14     for (int j = 0; check > (7 - j); j++)
15         check -= (7 - j);
16
17     check--;
18     textArray[check].BackColor = Color.LightBlue;
19     textArray[check+1].BackColor = Color.LightBlue;
20     string tmp;
21     if (int.Parse(textArray[check].Text) > int.Parse(textArray[check+1].Text)) {
22         textArray[check].BackColor = Color.Pink;
23         textArray[check+1].BackColor = Color.Pink;
24         tmp = textArray[check+1].Text;
25         textArray[check+1].Text = textArray[check].Text;
26         textArray[check].Text = tmp;
27     }
28
29     if (startButton.Text == "28") {
30         timer1.Enabled = false;
31         textArray[check].BackColor = SystemColors.Window;
32         textArray[check + 1].BackColor = SystemColors.Window;
33         startButton.Text = "開始排序";
34     }
35     else
36         startButton.Text = (int.Parse(startButton.Text) + 1).ToString();
37 }

```

四、程式碼解說

先從功能比較單純的「開始排序」按鈕開始吧！程式碼內容只要做兩件事，第一件事是開始啟動計時器，也就是將計時器元件 timer1 的 Enabled 屬性設為 true，另一件事就是將按鈕的顯示文字從「開始排序」變成代表步驟的數值「1」，我們等於是利用按鈕的 Text 屬性來幫我們記住進行了多少步驟。

```

private void startButton_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;

```



```
startButton.Text = "1";  
}
```

控制氣泡排序過程的關鍵是計時器元件 timer1 的 Tick 事件，Tick 事件每一次被觸發，就代表排序的過程向前一步，由於文字方塊 numText1~8 並非陣列，會導致我們在之後的存取上會有許多不便，故 Tick 事件的一開始，我們就要另外宣告一個 TextBox 的陣列 textArray，並且初始設定文字方塊 numText1~8 為陣列成員，控制項元件陣列的宣告語法與一般一維陣列相同，這個運用可是本範例的一大重點！請同學務必了解這個用法的存在與目的。

前面提到我們之後會用改變文字方塊底色的方式來強化氣泡排序的過程演示，故我們立即可以運用剛剛宣告的文字方塊陣列 textArray 來將其內的所有文字方塊，也就是 numText1~8，全部的底色都先設定成預設值「SystemColors.Window」，以避免前一步驟的 Tick 事件之殘色留在顯示畫面上。

```
private void timer1_Tick(object sender, EventArgs e)  
{  
    TextBox[] textArray = new TextBox[] { numText1, numText2, numText3,  
    numText4, numText5, numText6, numText7, numText8 };  
    for (int i = 0; i < 8; i++)  
        textArray[i].BackColor = SystemColors.Window;
```

接下來我們要透過另一種形式來進行氣泡排序，氣泡排序法的過程實際上是數次的相鄰元素比較、交換，我們要做的就是藉由記錄在按鈕上的「步驟」數值，推斷現在應該要對哪一組相鄰元素進行檢查。我們先將「開始排序」按鈕上的文字讀進變數 check 當中，然後藉由一個 for 迴圈對 check 值做是否大於 7,6,5...1 的遞減式檢查及扣值，同學可以回顧前面的氣泡排序法，一旦步驟值大於 7，就代表將最大的氣泡(值)浮到第 8 個位置的工作已經完成，所以要減 7，如果 check 值還是大於 6，就代表將次大的氣泡(值)浮到第 7 個位置的工作已經完成，所以要再減 6... 依此原則持續往下做，最後剩下的 check 值，就意味著目前這個步驟該檢查的位置。

```
int check = int.Parse(startButton.Text);  
for (int j = 0; check > (7 - j); j++)  
    check -= (7 - j);
```

得到了我們該進行檢查的位置之後，在檢查之前有幾件事要做，首先是 check 值要減 1，因為我們是透過 textArray 陣列操作這些文字方塊，所以索引值相較位置要減 1，接著我們就把要被檢查的兩個索引位置對應的文字方塊背景顏色改成淺藍色，顯示目前比較的是這兩個元素，最後是宣告暫存用的字串 tmp。

用簡單的 if 判斷式就可以進行比較了，將要比較大小的兩個文字方塊內容轉換成整數型態後就可以比較了，如果前者比後者大，就要將兩者交換，交換時先

將文字方塊背景顏色改成粉紅色，讓使用者可以清楚的看到這兩個元素被交換位置了，最後再把兩個文字方塊的內容交換即可。

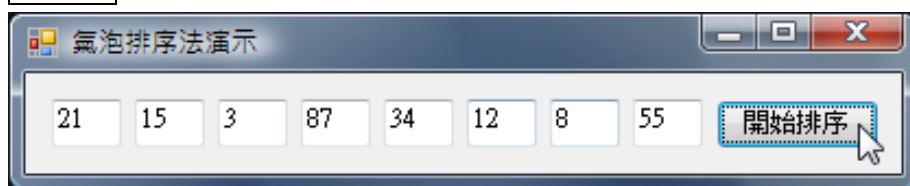
```
check--;
textArray[check].BackColor = Color.LightBlue;
textArray[check+1].BackColor = Color.LightBlue;
string tmp;
if (int.Parse(textArray[check].Text) > int.Parse(textArray[check+1].Text)) {
    textArray[check].BackColor = Color.Pink;
    textArray[check+1].BackColor = Color.Pink;
    tmp = textArray[check+1].Text;
    textArray[check+1].Text = textArray[check].Text;
    textArray[check].Text = tmp;
}
```

由於氣泡排序法的步驟是固定的，所以在計時器每一次 Tick 的最後，我們可以藉由目前的步驟數來判斷，這是否已是「最後一步」了，以 8 個元素來說，氣泡排序總共是 28 個步驟，所以如果按鈕上的文字已經是 28 的話，就代表已經排序結束了，我們要將計時器 timer1 給停止，並且將最後一次更改的文字方塊背景顏色和按鈕的文字給復原；若是其他情況，按鈕上文字還沒到 28，就要把按鈕上的文字轉換為數字後加 1 再轉回文字，完成步驟數增加的更新。

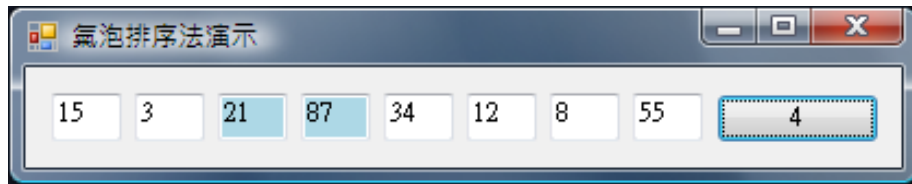
```
if (startButton.Text == "28") {
    timer1.Enabled = false;
    textArray[check].BackColor = SystemColors.Window;
    textArray[check + 1].BackColor = SystemColors.Window;
    startButton.Text = "開始排序";
}
else
    startButton.Text = (int.Parse(startButton.Text) + 1).ToString();
}
```

五、執行結果

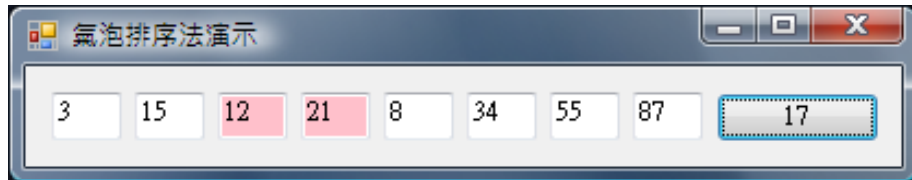
Step1：輸入要氣泡排序的 8 個元素資料，然後按下『開始排序』按鈕。



Step2：目前進行比較的兩組元素會用淺藍色標出。



Step3 : 有發生交換的兩組元素會用粉紅色標出。



Step4 : 完成排序後的樣子。

