

# .NET 程式設計入門（使用 C#）

# .NET 專案範本








# 檔案介紹

檔案	說明
cs_form.sln	儲存方案中使用到那些檔案資料等資訊
cs_form.csproj	儲存專案中使用到那些檔案資料等資訊
cs_form.exe	利用專案所編譯出來的執行檔
Form1.cs	儲存表單及程式碼相關資訊
Form1.resx	儲存表單中使用的相關資料
assemblyInfo.cs	描述組件的資訊
cs_form.csproj.user	專案目前編輯狀態
cs_form.suo	記錄方案的編輯狀態
App.ico	應用程式圖示

# 方案總管

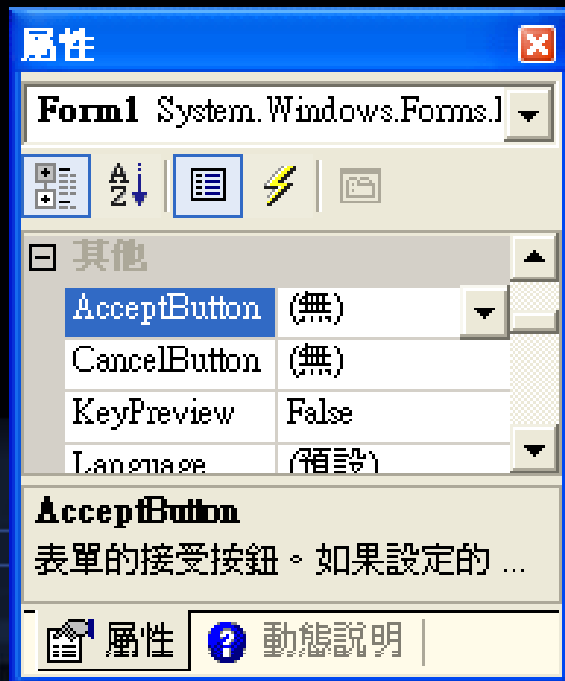
- 我們可以利用 [檢視] 中的 [方案總管] 來開啓方案總管視窗



圖示	功能說明
	切換到程式編輯視窗
	切換到表單編輯視窗
	重新整理
	顯示所有檔案
	打開屬性視窗

# 屬性視窗

- 在程式設計階段，可以透過 [檢視] 中的 [屬性視窗] 來設定修改表單上各控制項的屬性值



圖示	功能說明
	物件下拉選單
	依屬性類型分類顯示
	依屬性名稱排序顯示
	顯示屬性視窗
	顯示事件視窗

# 控制項

- 開啟 [檢視] 中的 [工具箱] 會列出常用控制項
- 要在表單中加入控制項的方式有二種
  - 點選控制項後拖曳至表單上
  - 連續點選控制項二下
- 重疊顯示二個控制項
  - 可利用右鍵中的提到最上層或移到最下層來調整控制項顯示順序



# 表單及控制項常用屬性

- 外觀
  - Text – 控制項的標題文字
  - BackColor – 背景色
  - BackgroundImage – 背景圖
  - Font – 字型設定
  - ForeColor – 前景色
- 行爲
  - Enable – 設定該控制項是否有作用
  - Visible – 是否顯示該控制項
- 配置
  - Location – 位置 (程式碼中可用 Left 及 Top 屬性設定)
  - Size – 大小

# 事件

- 在視窗程式設計中，程式的執行流程是以使用者觸發的事件順序來決定
- 表單及控制項個別提供了許多不同的事件，我們可以在不同的事件中撰寫不同的程式
- 當使用者觸發事件後，程式的流程會立即跳至該事件內的程式區塊中執行
- 在事件視窗中，點選事件名稱二下，即可跳至程式碼編輯視窗該事件內的程式區塊



# 表單常用事件

- **Load**
  - 當表單被載入開啓時，會執行此事件內的程式區段一次
- **Click**
  - 在表單上按下滑鼠左鍵，會執行此事件內的程式區段一次
- **Resize**
  - 調整表單大小時，會執行此事件內的程式區段一次

# Label 控制項

- Label 控制項主要是用來顯示文字
- 常用屬性
  - Text – 設定要顯示的文字
  - TextAlign – 設定文字顯示位置
  - Visible – 是否顯示控制項
- 常用事件
  - Click – 按一下時發生
  - MouseEnter – 滑鼠經過
  - MouseLeave – 滑鼠離開

# 實例探討 sample6-a1

- 程式功能
  - 猜謎語
- 程式內容

```
private void answerLB_Click(object sender,  
    System.EventArgs e)  
{  
    answerLB.Text = "灰熊厲害";  
}
```

# 課堂練習 sample6-b1

- 程式功能
  - 猜謎語
  - 當滑鼠移至題目上時，題目的字改成 "看解答"，當滑鼠移開後恢復成原本題目的文字
  - 當使用者點選 "看解答" 時，原本題目的文字變成解答
- 基本概念
  - 利用 Click、MouseEnter 及 MouseLeave 事件來完成

# Button 控制項

- Button 控制項提供按鈕功能
- 常用屬性
  - Text – 設定要顯示的文字
  - TextAlign – 設定文字顯示位置
  - Visible – 是否顯示控制項
- 常用事件
  - Click – 按一下時發生
  - MouseEnter – 滑鼠經過
  - MouseLeave – 滑鼠離開

# 實例探討 sample6-a2

- 程式功能
  - 設計結束視窗按鈕
  - 設定表單上的 CancelButton

- 程式內容

```
private void closeBTN_Click(object sender,  
    System.EventArgs e)  
{  
    Application.Exit();  
}
```

# TextBox 控制項

- TextBox 控制項用以讓使用者輸入資料
- 常用屬性
  - Text – 設定取得控制項中的文字
  - TextAlign – 設定文字顯示位置
  - PasswordChar – 輸入的資料以別的字元顯示
  - ReadOnly – 文字是否允許修改
  - MaxLength – 輸入文字最大寬度
  - MultiLine – 允許多行輸入
  - ScrollBar – 選擇要顯示的捲軸

# 課堂練習 sample6-b2

- 程式功能
  - 登入系統
  - 輸入帳號為 **alice** ，密碼為 **bob** 時，顯示登入成功，否則顯示資料錯誤
  - 另設清除按鍵，將輸入方塊的資料清除
- 基本概念
  - 在 **Click** 事件中判斷使用者輸入的資料是否正確



# Timer 控制項

- Timer 控制項供計時及週期事件使用
- 常用屬性
  - Enable – 是否啟動 Timer
  - Interval – Timer 事件發生週期 (千分之一秒)
- 常用事件
  - Tick – 每隔 interval 設定時間執行一次

# 實例探討 sample6-a3 (1)

- 程式功能
  - 倒數計時
- 程式內容

```
private void startBTN_Click(object sender,  
    System.EventArgs e)  
{  
    timer1.Enabled = true;  
    countLB.Text = countTB.Text;  
    countTB.Enabled = false;  
    startBTN.Enabled = false;  
}
```

## 實例探討 sample6-a3 (2)

- 程式內容

```
private void timer1_Tick(object sender,  
    System.EventArgs e)  
{  
    countLB.Text = (int.Parse(countLB.Text) - 1)  
        .ToString();  
    if(int.Parse(countLB.Text) == 0)  
        timer1.Enabled = false;  
}
```

# 課堂練習 sample6-b3

- 程式功能
  - 閃爍提示文字
  - 點選文字開始閃爍
  - 再點選停止
- 基本概念
  - 可利用定時更改 Label 控制項的 Visible 屬性達到閃爍效果
  - 也可定時更改 Label 控制項的文字來達成

# PictureBox 控制項

- PictureBox 控制項供計時及週期事件使用
- 常用屬性
  - Image – 設定顯示圖片
  - SizeMode – 設定載入圖片擺放方式
    - Normal：圖片放在控制項的左上角
    - StretchImage：圖片隨控制項大小伸縮
    - AutoImage：控制項隨圖片大小伸縮
    - CenterImage：圖片放在控制項正中央

# 課堂練習 sample6-b4

- 程式功能
  - 紅綠燈
  - 可設定紅燈綠燈及黃燈時間長度
- 基本概念
  - 利用 **Timer** 來決定更換圖片時間
  - 利用 **MSDN** 查詢程式執行中更換 **PictureBox** 圖片的方式
  - 可能需要一個可供所有方法存取的欄位變數

# MessageBox 類別

- **MessageBox** 類別提供錯誤或警告訊息等的對話方塊，以提醒使用者注意
- **Show** 靜態方法
  - Show (訊息文字, 標題文字, 顯示按鈕, 圖示);
  - Show (string, string, MessageBoxButtons, MessageBoxIcon)

# MessageBox 類別 - 顯示按鈕

顯示按鈕列舉常數	說明
MessageBoxButtons.OK	確定
MessageBoxButtons.OKCancel	確定、取消
MessageBoxButtons.AbortRetryIgnore	異常終止 重試、略過
MessageBoxButtons.YesNoCancel	是、否、取消
MessageBoxButtons.YesNo	是、否
MessageBoxButtons.RetryCancel	重試、取消



# MessageBox 類別 - 圖示

圖示列舉常數	圖示
MessageBoxIcon.Asterisk	
MessageBoxIcon.Information	
MessageBoxIcon.Exclamation	
MessageBoxIcon.Warning	
MessageBoxIcon.Error	
MessageBoxIcon.Hand	
MessageBoxIcon.Stop	
MessageBoxIcon.Question	
MessageBoxIcon.None	 無

# MessageBox 類別 – 回傳值

- 呼叫 `MessageBox.Show` 方法時，當使用者按下訊息方塊中其中一個按鈕時，會回傳一個 `DialogResult` 列舉型別的回傳值

列舉型別	回傳值	按鈕
<code>DialogResult.OK</code>	1	確定
<code>DialogResult.Cancel</code>	2	取消
<code>DialogResult.Abort</code>	3	異常終止
<code>DialogResult.Retry</code>	4	重試
<code>DialogResult.Ignore</code>	5	略過
<code>DialogResult.Yes</code>	6	是
<code>DialogResult.No</code>	7	否

# 實例探討 sample6-a4 (1)

- 程式功能
  - 密碼設定 (限四碼)
- 程式內容

```
private void okBTN_Click(object sender, System.EventArgs e)
{
    if(pwTB.Text.Length != 4)
        MessageBox.Show("密碼設定限四位數!!", "設定錯誤",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
    {
        MessageBox.Show("密碼設定成功!!", "設定完成",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        Application.Exit();
    }
}
```

## 實例探討 sample6-a4 (2)

- 程式內容

```
private void exitBTN_Click(object sender,  
    System.EventArgs e)  
{  
    DialogResult result;  
    result = MessageBox.Show("確定離開!?", "  
    離開", MessageBoxButtons.OKCancel,  
    MessageBoxIcon.Question);  
  
    if(result == DialogResult.OK)  
        Application.Exit();  
    else  
        pwTB.Focus();  
}
```

# GroupBox 控制項

- **GroupBox** 控制項可將同性質的控制項框為群組，使得控制項可藉以分門別類並讓畫面較為整齊有條理
- 常用屬性
  - **Text** – 設定標題名稱
  - **Controls** – 加入控制項所成的集合

# RadioButton 控制項

- RadioButton 控制項提供單選選項按鈕的功能
- 同一個表單中具有多組單選選項時，可利用 GroupBox 或 Panel 來區隔
- 常用屬性
  - CheckAlign – 設定選項按鈕顯示位置
  - Checked – 按鈕是否選取
  - Text – 顯示文字
- 常用事件
  - CheckedChanged – Checked 屬性改變時發生

# 實例探討 sample6-a5 (1)

- 程式功能
  - 資料核對
  - 利用 **GroupBox** 將 **RadioButton** 區分為二組

- 程式內容

```
private void exitBTN_Click(object sender,  
    System.EventArgs e)  
{  
    Application.Exit();  
}
```

## 實例探討 sample6-a5 (2)

- 程式內容

```
private void sendBTN_Click(object sender,  
    System.EventArgs e)  
{  
    string checkStr = "您是" + (sex1RB.Checked?  
        "男生": "女生") + "，您的學歷為";  
  
    foreach(RadioButton rb in  
        educationGB.Controls)  
        if(rb.Checked) checkStr += rb.Text;  
  
    MessageBox.Show(checkStr, "資料核對");  
}
```



# CheckBox 控制項

- CheckBox 控制項提供複選核取方塊的功能
- 常用屬性
  - CheckAlign – 設定核取方塊顯示位置
  - Checked – 核取方塊是否選取
  - Text – 顯示文字
  - ThreeState – 設定為雙態或三態
  - CheckState – 取得目前勾選狀況
- 常用事件
  - CheckedChanged – Checked 屬性改變時發生
  - CheckStateChanged – CheckState 屬性改變時發生

# 實例探討 sample6-a6 (1)

- 程式功能
  - 電腦採購
  - 勾選需要配備立即更新總價
- 程式內容

## 實例探討 sample6-a6 (1)

```
private void  
    monitorCB_CheckedChanged(object  
        sender, System.EventArgs e)  
{  
    int cost = monitorCB.Checked? 12000: -  
        12000;  
    totalLB.Text = (int.Parse(totalLB.Text) +  
        cost).ToString();  
}
```

## 實例探討 sample6-a6 (2)

- 程式內容

```
private void boxCB_CheckedChanged(object  
    sender, System.EventArgs e)  
{  
    int cost = boxCB.Checked? 15000: -15000;  
    totalLB.Text = (int.Parse(totalLB.Text) +  
        cost).ToString();  
}
```

## 實例探討 sample6-a6 (2)

```
private void  
    mouseCB_CheckedChanged(object  
        sender, System.EventArgs e)  
    {  
        int cost = mouseCB.Checked? 500: -500;  
        totalLB.Text = (int.Parse(totalLB.Text) +  
            cost).ToString();  
    }
```

## 實例探討 sample6-a6 (3)

- 程式內容

```
private void  
    keyboardCB_CheckedChanged(object  
        sender, System.EventArgs e)  
{  
    int cost = keyboardCB.Checked? 500: -500;  
    totalLB.Text = (int.Parse(totalLB.Text) +  
        cost).ToString();  
}
```

# 課堂練習 sample6-b5

- 程式功能
  - 電腦採購
  - CPU 、 RAM 及週邊各有三種類可挑選
  - CPU 及 RAM 為必備元件，若沒採購跳出錯誤訊息視窗
  - 計算全部費用
- 基本概念
  - 可利用迴圈讀取 GroupBox 中的元件簡化程式碼

# ListBox 控制項 (1)

- **ListBox** 控制項提供文字項目清單供使用者選擇
- 常用屬性
  - **Items** – 元素內容 (**ListBox.ObjectCollection** 類別)
  - **MultiColumn** – 設定多欄顯示
  - **ColumnWidth** – 多欄顯示時每一個欄位寬度



# ListBox 控制項 (1)

- **SelectionMode** – 設定清單可選取數目
  - **None** – 不能選取
  - **One** – 單選
  - **MultiSimple** – 複選 (點一下選取，再點一下取消)
  - **MultiExtended** – 複選 (可利用 **Ctrl**、**Shift** 及拖曳選取)

# ListBox 控制項 (2)

- 常用屬性
  - SelectedItem – 取得被選取項目
  - SelectedItems – 取得被選取項目集合
  - SelectedIndex – 取得選取項目索引
  - SelectedIndices – 取得所有選取項目索引
- 常用事件
  - SelectedIndexChanged – 清單中選取項目改變時發生

# 相關類別

- **ListBox.ObjectCollection** 類別
- 常用屬性
  - **Count** – 計算清單項目個數
- 常用方法
  - **Add** – 新增項目至清單
  - **Remove** – 刪除清單中某項目
  - **Clear** – 清除全部項目
- 用法
  - **Ex** : `listBox1.Items.Add("Alice");`

# 實例探討 sample6-a7 (1)

- 程式功能
  - 設定啓用功能
- 程式內容

```
private void Form1_Load(object sender,  
    System.EventArgs e)  
{  
    string[] funcStr = {"檔案","編輯","檢視","專案",  
        "","建置","偵錯","工具","視窗","說明"};  
    foreach(string str in funcStr)  
        allLB.Items.Add(str);  
}
```

## 實例探討 sample6-a7 (2)

- 程式內容

```
private void addBTN_Click(object sender, System.EventArgs e)
{
    if(allLB.SelectedItem == null)
        MessageBox.Show("尚未選取欲新增功能!!", "錯誤",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
    else
    {
        enableLB.Items.Add(allLB.SelectedItem);
        allLB.Items.Remove(allLB.SelectedItem);
    }
}
```

# 實例探討 sample6-a7 (3)

- 程式內容

```
private void removeBTN_Click(object sender,
    System.EventArgs e)
{
    if(enableLB.SelectedItem == null)
        MessageBox.Show("尚未選取欲刪除功能!!", "錯誤",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
    else
    {
        allLB.Items.Add(enableLB.SelectedItem);
        enableLB.Items.Remove(enableLB.SelectedItem);
    }
}
```

# ComboBox 控制項

- ComboBox 控制項提供下拉式選單功能，為 ListBox 控制項的延伸
- 常用屬性
  - Text – 取得設定文字
  - MaxLength – 使用者可輸入最大長度
  - MaxDropDownItems – 下拉部份一次顯示最大項目個數
  - DropDownStyle
    - DropDown – 可編輯，箭頭按鈕顯示清單
    - DropDownList – 不可編輯，箭頭按鈕顯示清單
    - Simple – 可編輯，清單永遠顯示。

# 實例探討 sample6-a8 (1)

- 程式功能
  - 地址輸入
- 程式內容

```
private void addressTB_Enter(object sender,  
    System.EventArgs e)  
{  
    if(addressTB.Text == "輸入地址")  
        addressTB.Text = "";  
}
```



## 實例探討 sample6-a8 (2)

- 程式內容

```
private void area1CB_SelectedIndexChanged(object sender,
    System.EventArgs e)
{
    area2CB.Items.Clear();
    if(area1CB.SelectedIndex == 0)
    {
        area2CB.Items.Add("信義區");
        area2CB.Items.Add("大安區");
        area2CB.Text = area2CB.Items[0].ToString();
    }
    else
    {
        area2CB.Items.Add("汐止市");
        area2CB.Items.Add("板橋市");
        area2CB.Text = area2CB.Items[0].ToString();
    }
}
```

# 課堂練習 sample6-b6

- 程式功能
  - 書籍分類清單

小說	射鵬英雄傳	神鵬俠侶	倚天屠龍記
電腦	C# 程式設計入門	十天學會 JAVA	資料庫程式設計
食譜	韓式料理	日式料理	台式料理

- 基本概念
  - 利用 ComboBox 及 ListBox 完成

# TreeView 控制項

- TreeView 控制項提供樹狀結構檢視功能
- 常用屬性
  - Nodes – 子節點所成的集合  
(TreeNodeCollection 類別)
  - SelectedNode – 選取節點
  - CheckBoxes – 顯示核取方塊
  - PathSeperator – 傳回路徑的字串分隔字元
- 常用事件
  - AfterSelect – 變更選取後

# 相關類別

- **TreeNodeCollection** 類別
  - **Count** 屬性 – 取得節點數
  - **Add** 方法 – 增加節點至該點子節點中
  - **Remove** 方法 – 刪除節點 (傳入 **TreeNode**)
- **TreeNode** 類別
  - **Text** – 節點顯示文字
  - **Remove** 方法 – 刪除節點
  - **Nodes** – 子節點所成的集合
  - **FullPath** – 取得從根樹狀節點通往目前樹狀節點的路徑。

# 實例探討 sample6-a9 (1)

- 程式功能
  - 簡易資料夾檢索
- 程式內容

```
private void Form1_Load(object sender,  
    System.EventArgs e)  
{  
    string[] drives = Directory.GetLogicalDrives();  
  
    foreach(string drive in drives)  
        diskCB.Items.Add(drive);  
    updateTreeView(diskCB.Text);  
}
```

## 實例探討 sample6-a9 (2)

- 程式內容

```
private void
diskCB_SelectedIndexChanged(object sender,
System.EventArgs e)
{
    updateTreeView(diskCB.Text);
}
private void updateTreeView(string disk)
{
    try
    {
        dirTV.Nodes.Clear();
        string[] dirs = Directory.GetDirectories(disk);
```

# 實例探討 sample6-a9 (3)

- 程式內容

```
int i=0;
foreach(string dir in dirs)
{
    dirTV.Nodes.Add(dir);
    try
    {
        string[] subDirs =
Directory.GetDirectories(dir);
        foreach(string subDir in subDirs)
            dirTV.Nodes[i].Nodes.Add(subDir);
    }
    catch {}
    i++;
}
}
catch{}
}
```

# 課堂練習 sample6-b7

- 程式功能
  - 班級資料表
  - 可新增班級
  - 重複新增不予動作
- 基本概念
  - 利用 **TreeView** 顯示班級資訊



# ListView 控制項

- **ListView** 控制項提供項目清單檢視功能
- 常用屬性
  - **Columns** – 設定包含欄位
  - **Items** – 清單項目內容  
(**ListView.ListViewItemCollection** 類別)
  - **View** – 清單檢視方式
  - **CheckBoxes** – 顯示核取方塊
  - **GridLines** – 顯示格線

# 相關類別 (1)

- **ListView.ListViewItemCollection** 類別
  - **Count** 屬性 – 取得項目個數
  - **Add** 方法 – 新增項目
  - **Remove** 方法 – 移除指定項目
  - **Clear** 方法 – 清除所有項目
- **ListViewItem** 類別
  - **SubItems** 屬性 – 子項目  
(**ListViewItem.ListViewSubItemCollection** 類別)
  - **Text** 屬性 – 設定取得項目內容

## 相關類別 (2)

- **ListViewItem.ListViewSubItemCollection** 類別
  - **Count** 屬性 – 取得子項目個數
  - **Add** 方法 – 新增子項目
  - **Remove** 方法 – 移除指定子項目
  - **Clear** 方法 – 清除所有子項目
- **ListViewItem.ListViewSubItem** 類別
  - **Text** 屬性 – 設定取得子項目內容

# 實例探討 sample6-a10 (1)

- 程式功能
  - 學生名單
- 程式內容

```
private void clearBTN_Click(object sender,  
    System.EventArgs e)  
{  
    idTB.Text = "";  
    nameTB.Text = "";  
    studentLV.Items.Clear();  
}
```

## 實例探討 sample6-a10 (2)

- 程式內容

```
private void addBTN_Click(object sender,  
    System.EventArgs e)  
{  
    if(idTB.Text != "" && nameTB.Text != "")  
    {  
        ListViewItem item = new ListViewItem();  
        item.SubItems.Add(idTB.Text);  
        item.SubItems.Add(nameTB.Text);  
        studentLV.Items.Add(item);  
    }  
}
```

# 課堂練習 sample6-b8

- 程式功能
  - 成績表
  - 可新增科目成績資料
  - 計算總平均
- 基本概念
  - 利用 **ListView** 來完成

# 執行緒

- 我們可以把執行緒想像為一段程式碼區段
- 一個應用程式至少會在一個以上的執行緒中執行
- **.NET** 內建提供多執行緒支援，可藉由執行緒類別成員所提供的功能來控制執行緒的行為
- **Thread** 類別提供執行緒支援，可引用 **System.Threading** 命名空間來使用該類別

# Thread 類別 (1)

- 建構式
  - `public Thread( ThreadStart start );`
- 常用屬性
  - **Name** – 取得或設定執行緒名稱
  - **IsAlive** – 取得目前執行緒執行狀況
- 常用方法
  - **Start** – 執行該執行緒
  - **Suspend** – 使執行緒暫停
  - **Resume** – 使暫停的執行緒繼續執行



# Thread 類別 (2)

- 靜態方法
  - Sleep – 暫停執行緒一段時間
- 靜態屬性
  - CurrentThread – 取得目前執行緒
- 用法
  - ThreadStart myThreadStart =  
new ThreadStart(方法成員名稱);
  - Thread myThread =  
new Thread(myThreadStart);

# 實例探討 sample6-a11 (1)

- 程式功能
  - 遊戲半成品，Thread 類別測試
- 程式內容

```
class box
```

```
{
```

```
    public static bool stopFlag = false;
```

```
    private System.Windows.Forms.Label myLB;
```

```
    public box(System.Windows.Forms.Label LB)
```

```
    {
```

```
        myLB = LB;
```

```
    }
```

## 實例探討 sample6-a11 (2)

- 程式內容

```
public void action()
{
    Random rnd = new Random();
    do
    {
        myLB.Top += rnd.Next(0,2) == 0? 50: -
50;
        if(myLB.Top <= 0) myLB.Top += 100;
        else if(myLB.Top >= 400) myLB.Top -=
100;
        Thread.Sleep(500);
    }while(!stopFlag);
}
```

# 實例探討 sample6-a11 (3)

- 程式內容

```
private void startBTN_Click(object sender, System.EventArgs e)
{
    startBTN.Enabled = false;
    box myBox1 = new box(box1LB);
    box myBox2 = new box(box2LB);

    Thread box1Thread =
        new Thread(new ThreadStart(myBox1.action));
    Thread box2Thread =
        new Thread(new ThreadStart(myBox2.action));

    box1Thread.Name = "box1";
    box2Thread.Name = "box2";
```

## 實例探討 sample6-a11 (4)

- 程式內容

```
    box1Thread.Start();  
    Thread.Sleep(200);  
    box2Thread.Start();  
}  
  
private void endBTN_Click(object sender,  
    System.EventArgs e)  
{  
    box.stopFlag = true;  
    Application.Exit();  
}
```

# 實例探討 sample6-a12 (1)

- 程式功能
  - 利用單一執行緒執行高運算量計算
- 程式內容

```
private void computeBTN_Click(object sender,  
    System.EventArgs e)  
{  
    computing();  
}  
private void computing()  
{  
    try  
    {  
        resultLB.Text =  
            (garbage(int.Parse(numTB.Text))).ToString();  
    }  
}
```

## 實例探討 sample6-a12 (2)

- 程式內容

```
catch
{
    resultLB.Text = "--";
}
private int garbage(int num)
{
    if(num <= 4) return num;
    else return garbage(num-1)-garbage(num-2)
        +garbage(num-3)-garbage(num-4);
}
```

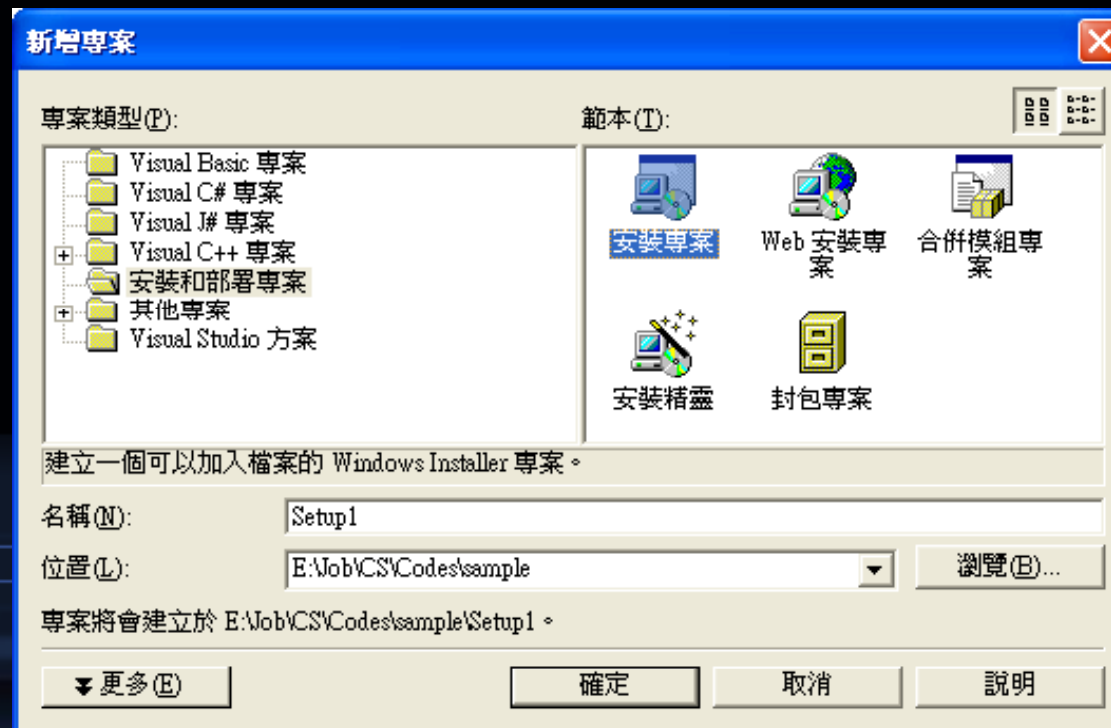
# 課堂練習 sample6-b9

- 程式功能
  - 利用多執行緒修改 **sample6-a12** 範例程式
  - 使其在運算過程中，視窗介面仍可自由操控
- 基本概念
  - 引用命名空間 **System.Threading**，利用 **Thread** 類別產生一條新的執行緒來執行運算方法內容



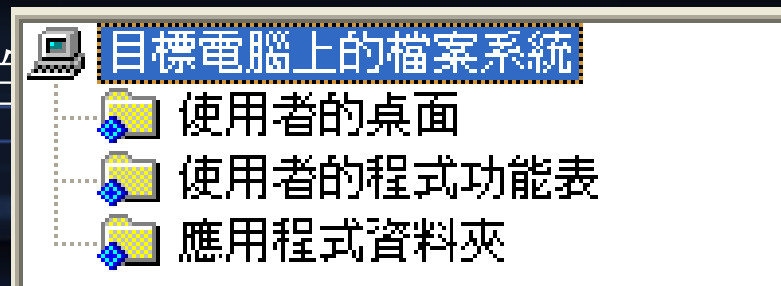
# 安裝與部署 (1)

- .NET 提供安裝和部署專案，我們可以透過此專案快速產生視窗應用程式的安裝程式



## 安裝與部署 (2)

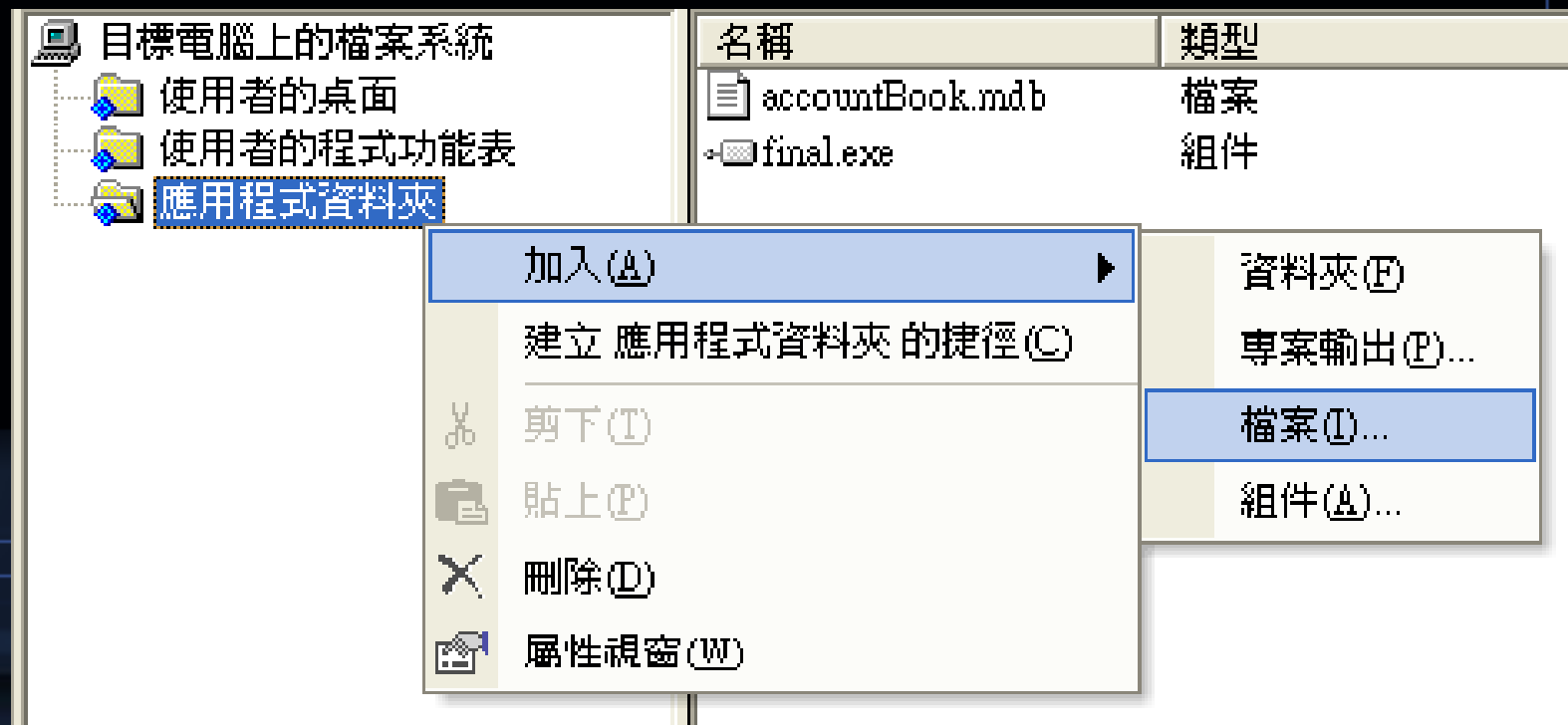
- 在檔案系統中，我們可以設定檔案與捷徑要分別安裝在那些地方
  - 使用者的桌面 – 設定欲安裝至桌面上的檔案
  - 使用者的程式功能表 – 設定欲安裝至程式集的檔案
  - 應用程式資料夾 – 設定欲安裝至 Program Files 資料夾的



資

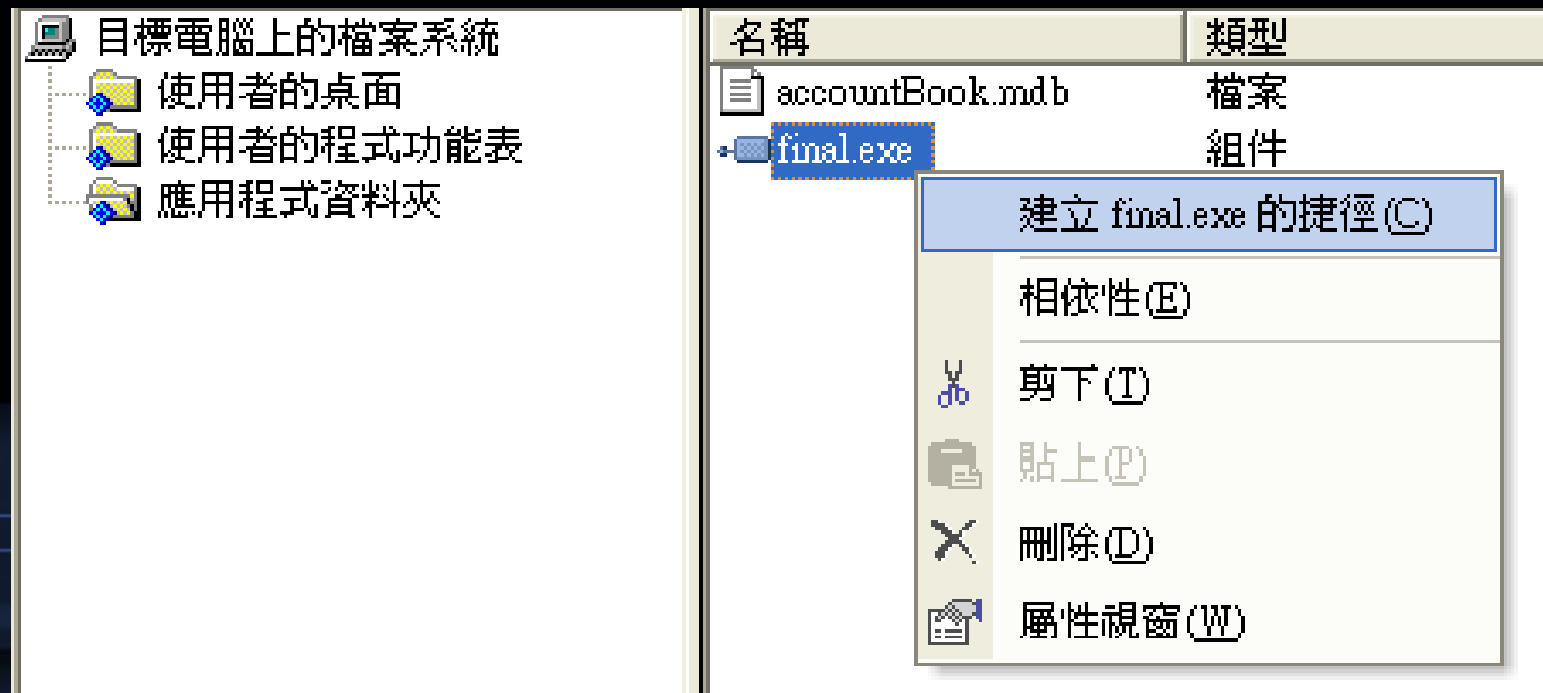
## 安裝與部署 (3)

- 在資料夾上點選右鍵，選擇 [加入] [檔案] 後，即可選擇欲加入檔案



## 安裝與部署 (4)

- 在已加入的檔案名稱上點選右鍵，即可為該檔案建立捷徑



## 安裝與部署 (5)

- 我們可以將專案的執行檔及欲連結的資料庫檔案加入至應用程式資料夾中
- 建立主要執行檔捷徑，將捷徑加入至使用者桌面及使用者的程式功能表資料夾中
- 建置專案後，即會產生 **Setup.Exe** 安裝檔案