

1-2 Label、TextBox 與 Button

1-2-1 前言

下圖 1-2-1 是執行兩數相乘的視窗程式，表單中由左而右圈選的 3 個區域中的物件依序就是 Label、TextBox 與 Button。Label 類別最主要功能是在表單上顯示指定的字串，就如同我們將黏性的標籤貼於某一東西上一樣。TextBox 是最簡單的使用者輸入物件，鍵盤上可打出的字元都可以輸入到此一物件中。Button 是一種選擇介面，一般用來指示使用者按下此按鈕時程式會執行何種作業。本此節中我們首先要稍微深入介紹此三類物件常用的屬性與事件，然後完成一個簡單的四則運算程式。

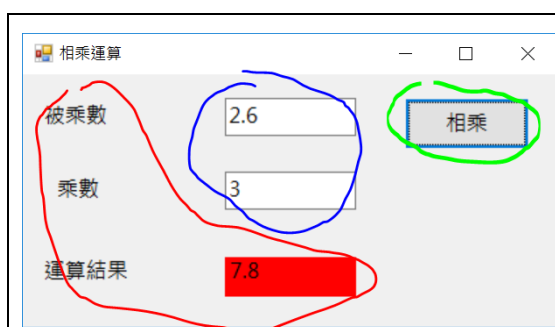


圖 1-2-1 執行兩數相乘的視窗程式

左圖 1-2-1 中的視窗程式所包含物件類別如下：

- 1.Label：共有 4 個 Label 物件，分別顯示"被乘數"、"乘數"、"運算結果"、"7.8"。其中顯示 7.8 的 Label 其 BackColor 屬性是紅色，顯示的內容是動態，端看相乘完後結果為何。
- 2.TextBox：有 2 個，作為被乘數與乘數的輸入處。
- 3.Button：1 個，顯示"相乘"，按下時會執行相乘運算並將結果顯示在紅色 Label 上。
- 4.Form：相當於視窗，作為其他物件的承載裝置。

1-2-2 二、學習目標

- 1.熟悉 Label、TextBox 與 Button 類別常用屬性、事件。
- 2.何謂建構函數及其使用時機。
- 3.能使用上述類別設計指定的程式。

1-2-3 相關知識

(一)、何謂建構函數

在 1-1 節中產生物件的方式都是從工具箱中 Double Click 或拖曳物件到表單上，此舉會讓 C# 自動為程式產生物件，且將物件擺放在表單上，這種方式適用於靜態設計上，也就是設計程式時就能確定物件個數與名稱者。但如果是程式執行期間才要動態產生物件，或是需要此物件但不想擺放到表單上時，就必須動態產生物件。要動態產生物件必須在宣告物件時使用 New 關鍵字呼叫類別的建構函數或稱為建構子(Constructor)，一般建構函數名稱會與類別名稱相同。下面範例會建立一個物件名稱為 A 的 Label 物件，並將此物件的 Text 屬性設為 KSVS。

```
Label A = new Label();    //建立名稱為A的Label
A.Text="KSVS";           //更改物件A的Text屬性
this.Controls.Add(A);     //在Form1 中新增此物件
```

從上面範例可知與宣告一般數值變數的語法差不多，只是多了 New 關鍵字，且原資料型別名稱改為建構函數名稱。至於決定物件的存取層級與變數相同，也是利用 Private(或 Public)與宣告位置來決定，細節請參閱主控台程式設計的"1-7 自訂副程式與參數傳遞"。此外所有類別都有提供建構函數，且有的類別建構函數還是多型(Polynomial)，也就是說同一函數，但有多種參數傳遞方式，例如主控台程式設計"1-8 日期與時間資料處理"中提到 TimeSpan 與 DateTime 的建構函數都是多型，下面範例顯示 TimeSpan 建構函數的三種型式。

```
Dim B As New TimeSpan(1), 等於時間長度為 0.0000001 秒
Dim B As New TimeSpan( 10, 20, 30, 40 ), 等於時間長度是 10 天 20 時 30 分 40 秒。
Dim B As New TimeSpan( 20, 30, 40 ), 等於時間長度是 20 時 30 分 40 秒。
```

(二)、Label(標籤)

功能簡述

Label 物件通常用來提供其他物件功能的描述文字或顯示指定訊息。例如，您可以如圖 1-2-1 般在 TextBox 物件旁加入 Label 物件來說明此一 TextBox 用途，也可如 1-1 節般單純顯示運算後的結果。下面是 Label 常用的屬性、方法與事件。

建構函數

Label()	初始化 Label 類別的新執行個體，例如 <code>Label A = new Label();</code> 。
---------	---

常用屬性

Text	String 型別。預設為控制項本身名稱。設定或讀取出現在 Label 中的文字。
Image	Image 類別。設定 Label 上要顯示的圖片
Enabled	Boolean 型別。是否可用(設為 False 時呈現灰階狀態)
Visible	Boolean 型別。是否隱藏(設為 False 時控制項為隱藏狀態)

常用事件

Click	在 Label 上按下滑鼠左鍵時所觸發的事件。
-------	-------------------------

(三)、TextBox(文字方塊)

功能簡述

TextBox 物件讓使用者能夠輸入字串。如果使用者的裝置支援遮罩密碼的功能，TextBox 控制項便可在輸入時遮罩密碼。

建構函數

TextBox()	建立 TextBox 類別的新執行個體。 <code>TextBox T = new TextBox();</code>
-----------	--

常用屬性

Text	String 型別。預設為控制項本身名稱。存放使用者在文字方塊中所輸入的文字。
Multiline	Boolean 型別。設為 True 時 TextBox 可以多行輸入，預設為 False。。
ReadOnly	Boolean 型別。是否唯讀。
ScrollBars	ScrollBars 的列舉型別，有 Both、Horizontal、None、Vertical 等四種值，預設為 None。當 TextBox 的 Multiline 屬性設為 True 時，經常會設定捲軸樣式來配合使用。
Enabled	Boolean 型別。文字方塊是否可用，設為 False 時按鈕為不可點選。
Visible	Boolean 型別。若設為 False 則程式執行時表單上看不到此一 TextBox。
PasswordChar	Char 型別。取得或設定在單行 TextBox 控制項中用來遮罩密碼的字元。
MaxLength	Integer 型別。文字方塊最多可輸入的字元個數。
TextAlign	HorizontalAlignment 列舉型別，控制 TextBox 上字串的對齊方式。 ■ Left：物件或文字靠左對齊控制項項目。 ■ Right：物件或文字靠右對齊控制項項目。 ■ Center：物件或文字置中對齊控制項項目。
CharacterCasing	CharacterCasing 列舉型別，取得或設定 TextBox 是否會在輸入字元時修改字元的大小寫。 ■ Normal：字元的大小寫不會變更。 ■ Upper：將所有的字元轉換成大寫。 ■ Lower：將所有的字元轉換成小寫。
WordWrap	Boolean。多行文字方塊控制項是否會在必要時自動將文字換行到下一行的開頭。

常用事件

TextChanged	當使用者在 TextBox 中鍵入每一個按鍵之後會自動觸發的事件。
Click	在 TextBox 上按下滑鼠左鍵後所觸發的事件。

常用方法

SelectAll	將控制項中的文字設為被選取的狀態
-----------	------------------

範例 1-2-1：下列程式碼範例會建立用來接受密碼的 TextBox 物件。這個範例使用 CharacterCasing 方法，以變更所輸入的所有字元為小寫，並使用 MaxLength 屬性來限制密碼長度為八個字元。這個範例也會使用 TextAlign 屬性，將 TextBox 控制項中的密碼置中對齊。

```
//Create an instance of the TextBox control.
```

```

TextBox textBox3 = new TextBox();
//Set the maximum length of text in the control to eight.
textBox3.MaxLength = 8;
//Assign the asterisk to be the password character.
textBox3.PasswordChar = Convert.ToChar("*");
// Change all text entered to be lowercase.
textBox3.CharacterCasing = CharacterCasing.Lower;
//Align the text in the center of the TextBox control.
textBox3.TextAlign = HorizontalAlignment.Center;
this.Controls.Add(textBox3);

```

(四)、Button(按鈕)

功能簡述

使用者可以按一下按鈕(Form Button)來執行特定動作。當按一下按鈕時，看起來就像是按下又放開按鈕一樣。每當使用者按一下按鈕時，就會叫用 Click 事件處理常式。您可以在 Click 事件處理常式中放入程式碼，以執行任何選擇的動作。

建構子

Button()	初始化 Button 類別的新執行個體。
----------	----------------------

常用屬性

Text	String 型別。設定 Button 上的顯示文字。
Image	Image 類別。取得或設定 Button 上顯示的影像。
Enabled	Boolean 型別。設為 False 時無法點選此一 Button。
Visible	Boolean 型別。若設為 False 則程式執行時表單上看不到此一 Button。

常用事件

Click	在 Label 上按下滑鼠左鍵時所觸發的事件。
-------	-------------------------

常用方法

Focus	將控制項設為目前駐點(Focus)。
-------	--------------------

(五)、圖片式按鈕的設定步驟

1.先將圖片加入專案成為一項資源，以便將來程式移植時圖片也能跟著移動，不會受限於原圖片的儲存位置。

2.欲將圖片加入專案成為一項資源，請依序點選專案、屬性、資源、影像、加入現有檔案，如下圖 1-2-2。從 PC 上挑選要匯入成為資源的圖片檔，然後按開啟。再按儲存所有檔案

的圖示。

3.這時可開始準備設定 Button 上要顯示的图片，首先視需要清除 Button 的 Text 屬性

4.其次按 Button 的 Image 屬性，即可顯示選取資源的對話方框，從專案資源檔中選取要顯示图片，如圖 1-2-3，最後按確定。

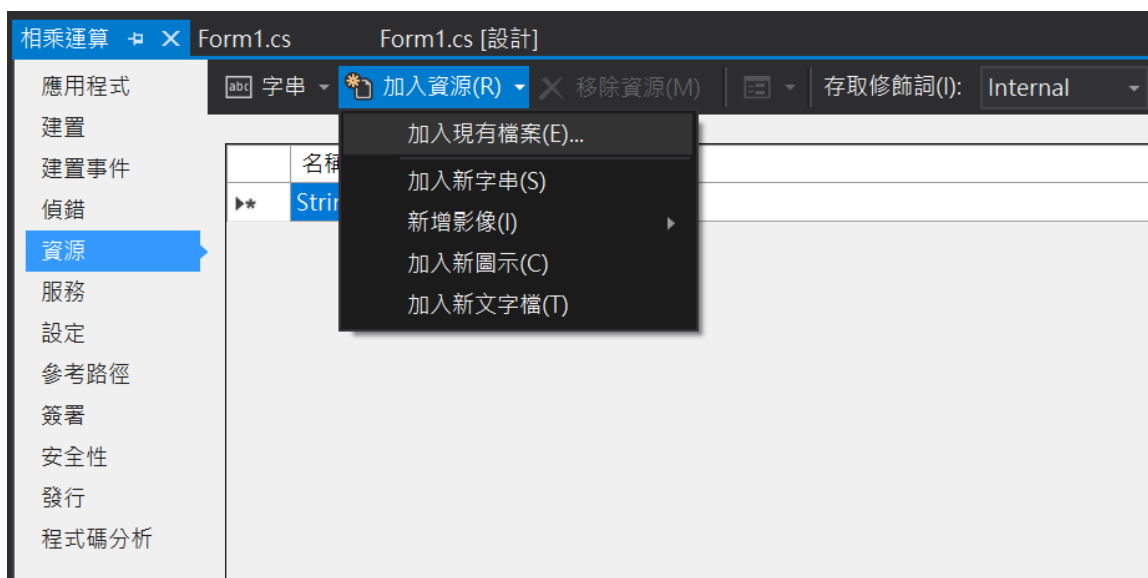


圖 1-2-2 加入現有圖片作為程式的資源

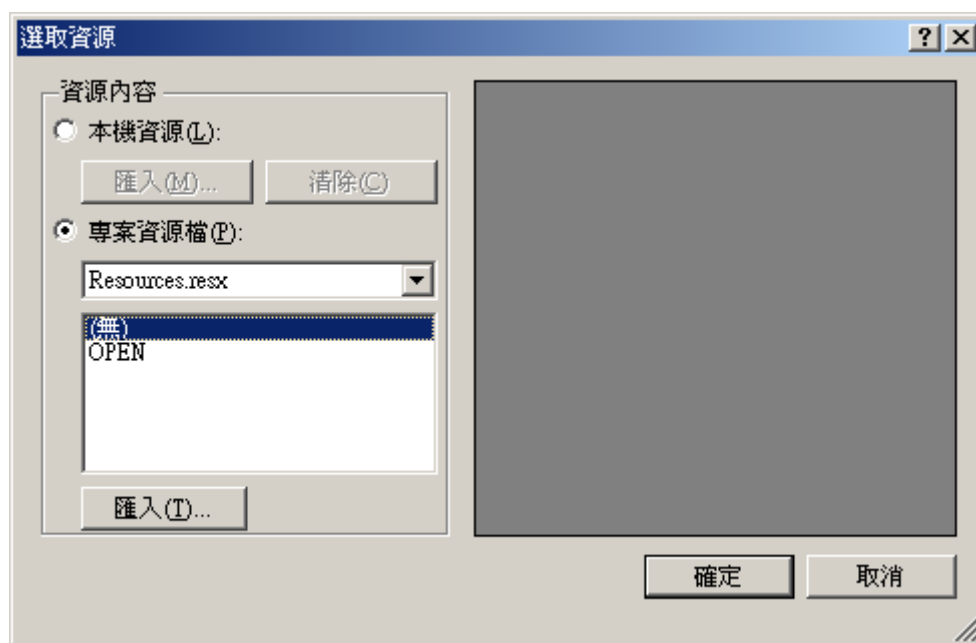


圖 1-2-3 從資源中選取 Button 的图片

1-2-4 程式範例

(一)、四則運算

1. 執行畫面

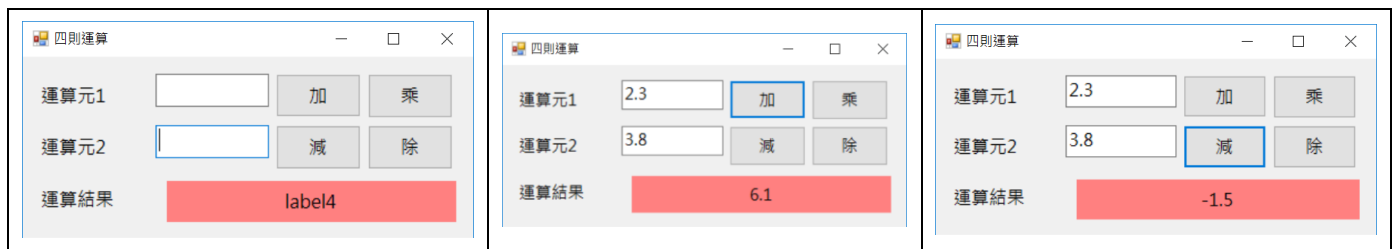


圖 1-2-4

2. 物件名稱說明

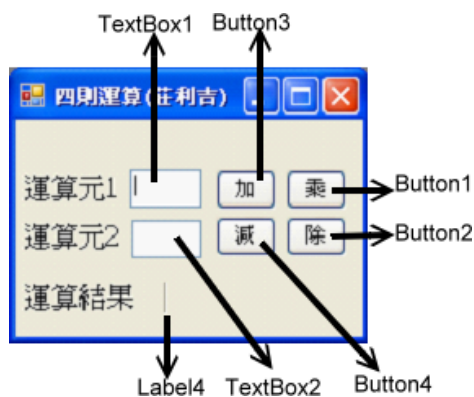


圖 1-2-5

3. 程式解析

```
private void button1_Click(object sender, EventArgs e)
{
    float a = Convert.ToSingle(textBox1.Text);
    float b = Convert.ToSingle(textBox2.Text);
    label4.Text = Convert.ToString(a * b);
}

private void button2_Click(object sender, EventArgs e)
{
    float a = Convert.ToSingle(textBox1.Text);
    float b = Convert.ToSingle(textBox2.Text);
    label4.Text = Convert.ToString(a / b);
}

private void button3_Click(object sender, EventArgs e)
{
    float a = Convert.ToSingle(textBox1.Text);
    float b = Convert.ToSingle(textBox2.Text);
```

```
        label4.Text = Convert.ToString(a + b);
    }

    private void button4_Click(object sender, EventArgs e)
    {
        float a = Convert.ToSingle(textBox1.Text);
        float b = Convert.ToSingle(textBox2.Text);
        label4.Text = Convert.ToString(a - b);
    }
```

(二)、數值相加或字串結合

1.問題

上述範例中相加部份為何不寫成 `Label4.Text = TextBox1.Text + TextBox2.Text`

2.說明

查閱"主控台程式設計(C# 2015)"的"1-2 數值資料處理"與"1-6 文字資料處理"可知"+"這個字元身兼加法與字串結合運算子符號，而 TextBox 的 Text 屬性資料型態為字串，所以 `TextBox1.Text + TextBox2.Text` 會被認為是字串結合。為了讓其成為加法運算子，必須將 Text 屬性的內容轉為數值，常見轉換方式說明如下。

3.字串(String)轉型為數值

將字串型別轉為數值在"主控台程式設計(C# 2015)"的"1-2 數值資料處理"中提到明確式轉換方式，簡述如下：

(1)明確式轉換：使用 Convert 類別

轉換方式是使用 Convert 類別傳回和指定型別之值相等的資料，支援的基底型別(Base Type)有 Boolean、Char、SByte、Byte、Int16、Int32、Int64、UInt16、UInt32、UInt64、Single、Double、Decimal、DateTime 和 String。例如下面指令將 TextBox1 中使用者輸入的資料轉為 Single 型別存於變數 a 中，及轉為 Integer 型別存於變數 b 中。

```
a = Convert.ToSingle(TextBox1.Text)
b = Convert.ToInt32(TextBox1.Text)
```

使用 Convert 類別中的方法時字串格式必須正確，否則會傳回例外錯誤。例如若 TextBox1 的 Text 屬性為"123.5"，則使用 `Single.Parse` 轉換沒有問題，但使用 `Integer.Parse` 會出現錯誤，因為字串中包含小數，格式不正確。

1-2-5 習題

(一)、求三角形面積

1. 程式執行畫面

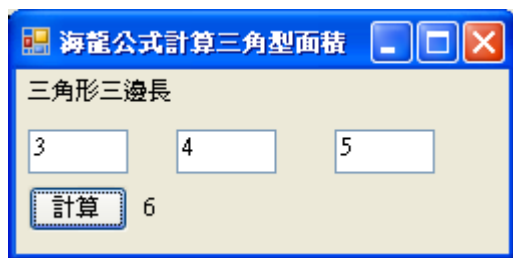


圖 1-2-6

2. 說明

海龍公式：若 A、B、C 可為三角形的三邊長，且 $S = \frac{A+B+C}{2}$ ，則三角形的面積等於

$$\sqrt{S(S-A)(S-B)(S-C)}$$

(二)、事件觸發順序

1. 請測試下列事件觸發的觸發時機及順序

- (1). FORM 的 Load 事件
- (2). FORM 的MouseDown 事件
- (3). FORM 的 MouseEnter 事件
- (4). FORM 的 MouseLeave 事件
- (5). FORM 的 MouseUp 事件
- (6). FORM 的 Click 事件
- (1). Button 的 DoubleClick 事件
- (2). Button 的 MouseDown 事件
- (3). Button 的 MouseEnter 事件
- (4). Button 的 MouseLeave 事件
- (5). Button 的 MouseUp 事件
- (6). Button 的 Click 事件

2. 提示

查詢線上文件了解各事件觸發時機，或用測試方式，先在在表單上放置一個 TextBox 物件，取名為 t，將其 Multiline 屬性設為 True、ScrollBar 屬性設為 Both。然後在各事件中以類似 `t.Text = t.Text + "Button Click" + vbCrLf` 的指令將訊息加入到 TextBox 中，來確認各事件觸發的先後順序。例如下圖 1-2-7。

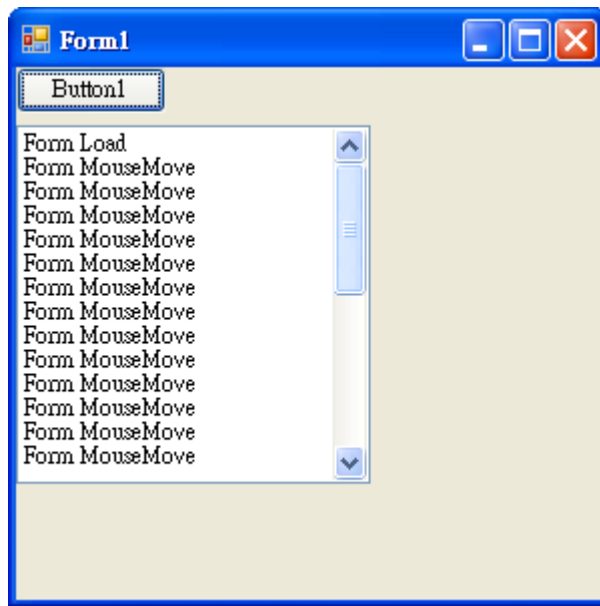


圖 1-2-7