

事件驅動程式設計

- ❑ 事件驅動是Windows程式設計的精髓
- ❑ 事件驅動與真實世界的運作模式非常類似

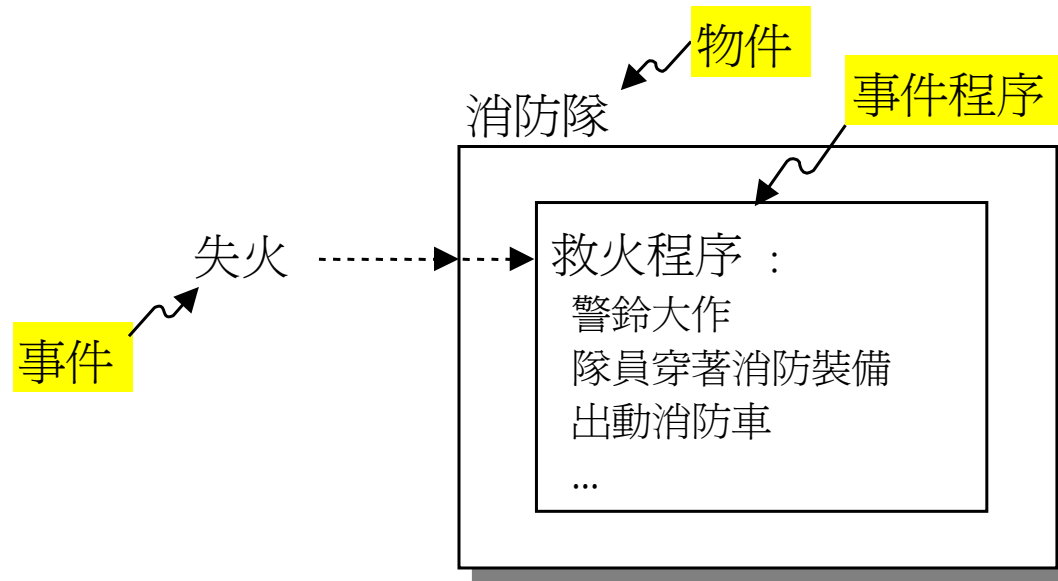
事件驅動的基本概念

事件驅動程式的要素

A. 物件

B. 事件

C. 事件程序



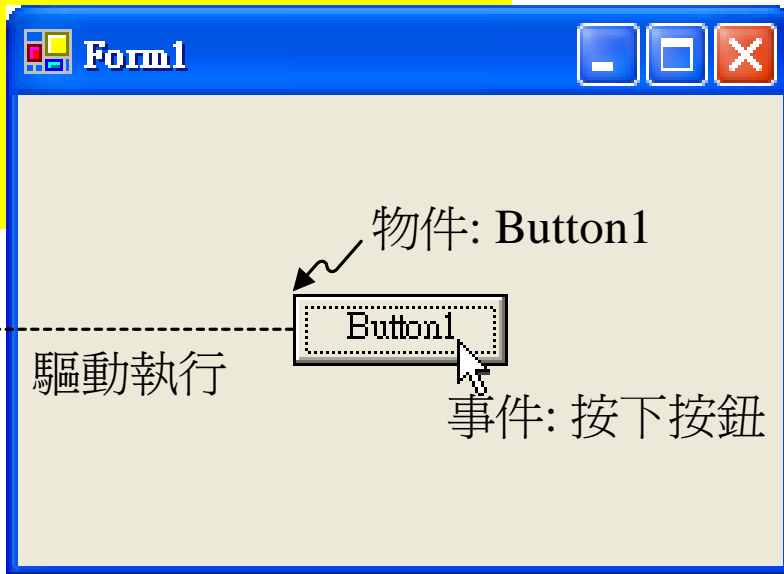
➤ 實例：

把「消防隊」視為一種物件、把「失火」視為這種物件需要處理的事件、把「救火程序」視為這種物件的一種事件程序，而「失火」會驅動「消防隊」的「救火程序」，這就構成了良好的事件驅動工作模式。

事件驅動程式的要素

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim H, W
    H = InputBox("請輸入身高")
    W = H - 105
    MsgBox("標準體重 =" & W)
End Sub
```

事件程序: Button1_Click



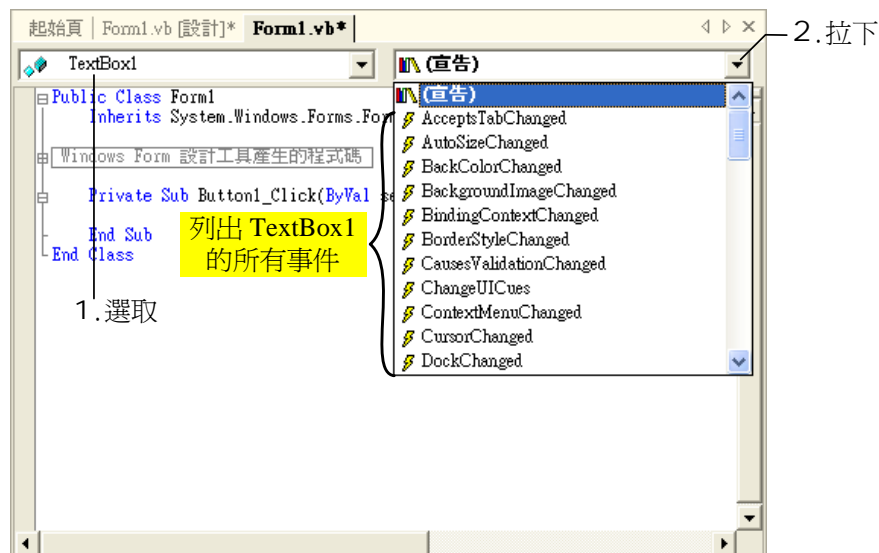
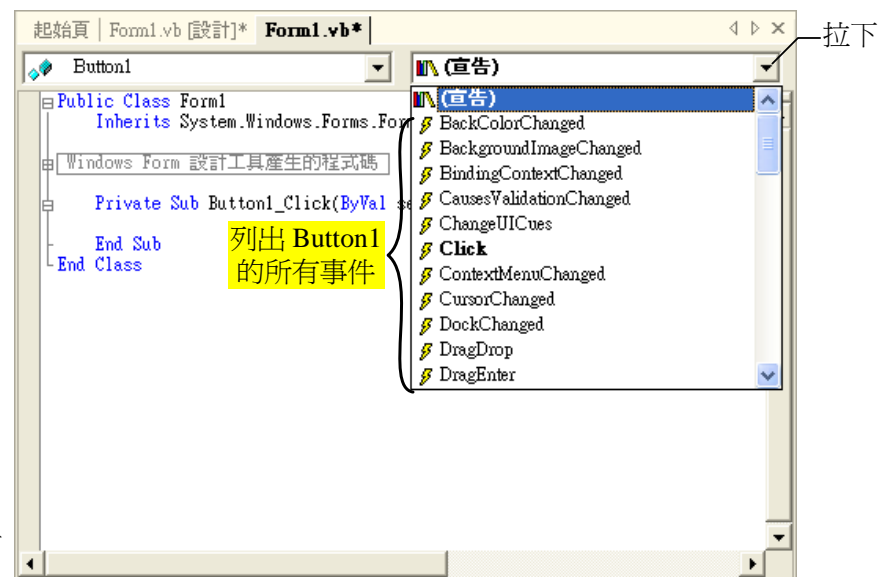
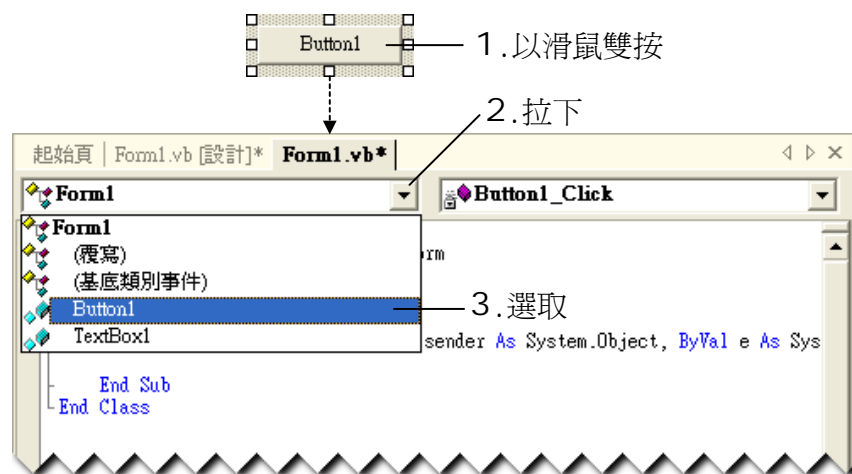
➤ 使用者按下按鈕，然後驅動執行Button1物件的Button1_Click事件程序，過程中也包含了物件、事件及事件程序三要素：

VB.NET的例子	事件驅動的要素
Button1	物件
按下按鈕(Click)	事件
Button1_Click	事件程序

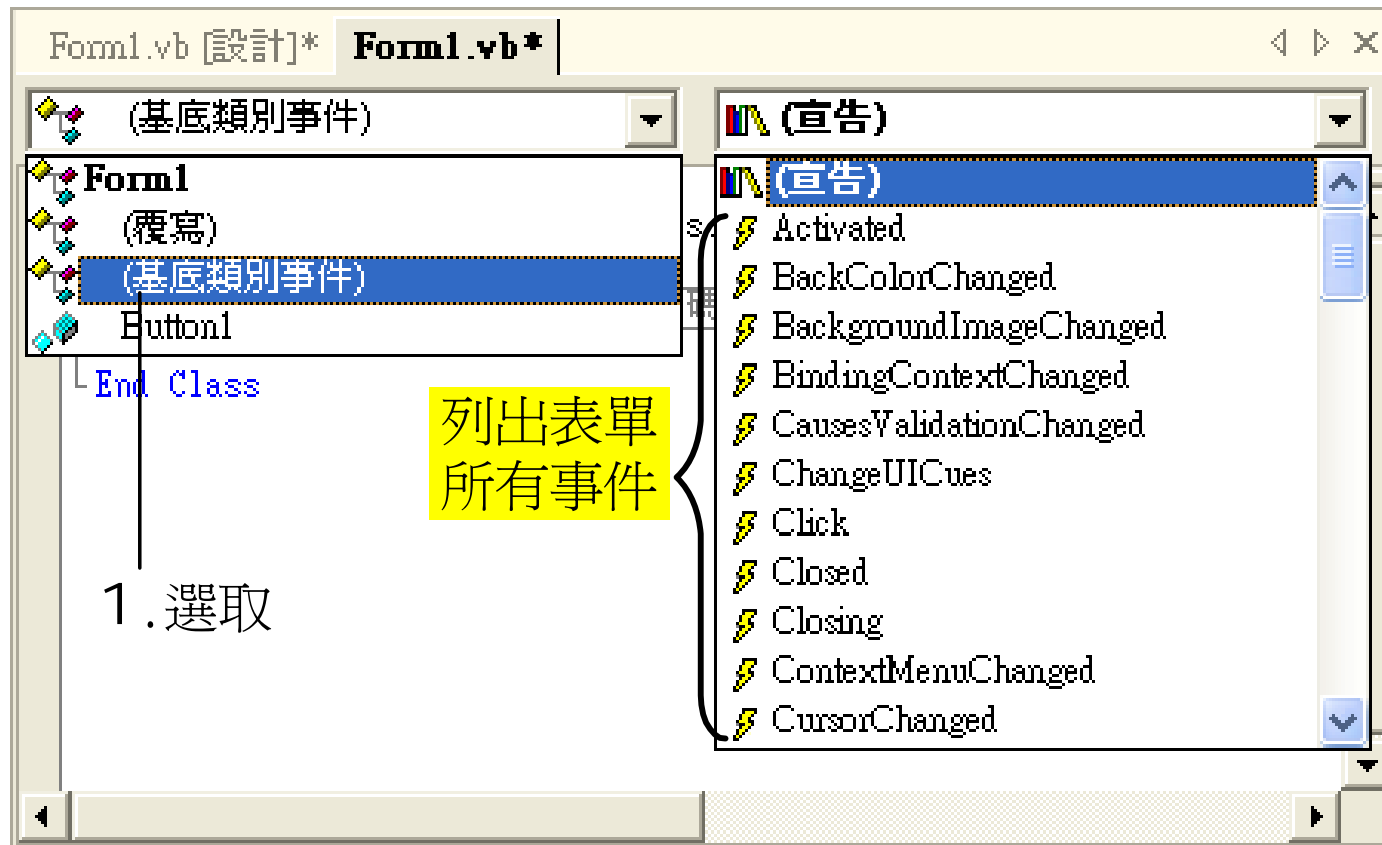
物件、事件、事件程式 與程式設計的關係

- 物件 → 來自工具箱或表單
- 事件 → VB 預先定義
- 事件程序 → 程式設計者撰寫

檢視控制元件含有哪些事件



檢視表單含有哪些事件



瞭解事件與事件程序的關係

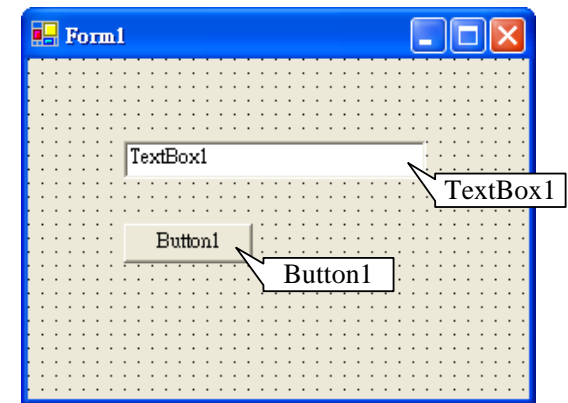
1. 建立一Windows應用程式專案。
2. 在表單上佈置一個Button及一個TextBox。

3. 以滑鼠雙按Button1物件，然後在Button1_Click之中撰寫以下程式：

```
Private Sub Button1_Click(...)
```

Handles Button1.Click

```
    TextBox1.Text = "處理 Button1.Click 事件"  
End Sub
```



瞭解事件與事件程序的關係

4. 在程式模組視窗中，拉下左上方的清單方塊，選取 Button1，再拉下右上方的清單方塊，選取 **MouseEnter**，然後輸入以下程式：

```
Private Sub Button1_MouseEnter(...) Handles  
Button1.MouseEnter
```

```
    TextBox1.Text = “處理 Button1.MouseEnter 事件”  
End Sub
```

5. 在 Button1_**MouseLeave** 之中撰寫以下程式：

```
Private Sub Button1_MouseLeave(...) Handles  
Button1.MouseLeave
```

```
    TextBox1.Text = “處理 Button1.MouseLeave 事件”  
End Sub
```

瞭解事件與事件程序的關係

6. 執行程式，然後將滑鼠移到Button1之上，此時會發生MouseEnter事件，所以Button1_MouseEnter會被執行，而使得TextBox1顯示「處理 Button1.MouseEnter 事件」。

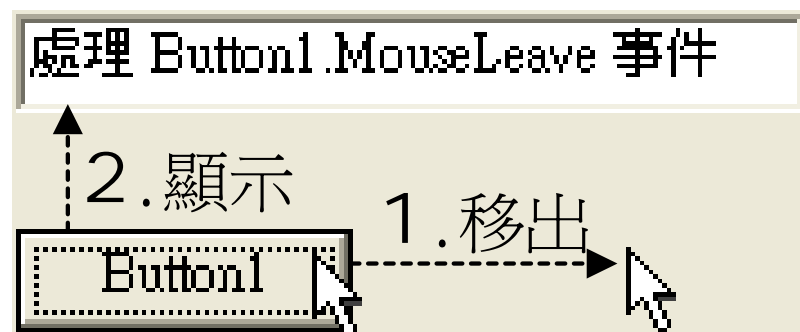


瞭解事件與事件程序的關係

7. 按下Button1，此時會發生Click事件，所以Button1_Click會被執行，而使得TextBox1顯示「處理 Button1.Click 事件」。



8. 將滑鼠移出Button1，此時會發生MouseLeave事件，所以Button1_MouseLeave會被執行，而使得TextBox1顯示「處理 Button1.MouseLeave 事件」。



瞭解事件與事件程序的關係

❖ 心得整理

1. 物件可能含有數十種事件，但只有我們關心的事件才需要撰寫對應的事件程序。
2. 當在程式模組視窗中選取物件及事件之後，VB會自動產生事件程序的程式架構，而事件程序的命名規則是「物件名_事件名」，例如Button1_Click是用來處理「Button1物件的Click事件」的事件程序。
3. 撰寫程式時還要瞭解每一種事件發生的時機，例如MouseEnter事件發生於滑鼠被移進物件、MouseLeave事件發生於滑鼠被移出物件、Click事件發生於按下按鈕...等。

選取資料與輸入資料的事件

❑CheckedChanged 事件

- ✓ 附屬於 RadioButton, CheckBox 物件
- ✓ 發生於選取狀態改變時

❑SelectedIndexChanged 事件

- ✓ 附屬於 ListBox, ComboBox 物件
- ✓ 發生於選項改變時

❑TextChanged 事件

- ✓ 附屬於 TextBox, Label 物件
- ✓ 發生於文字內容改變時

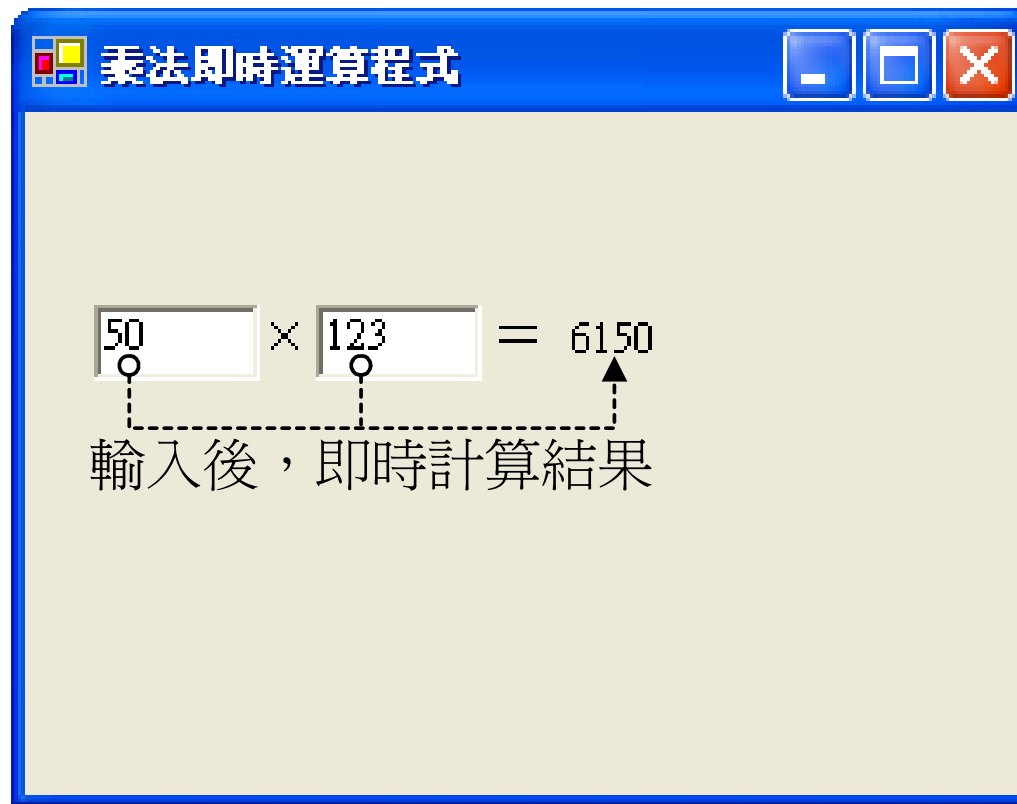
CheckedChanged事件的發生時機

1. RadioButton 被選取之後，會產生CheckedChanged事件。
2. RadioButton 被取消選取之後(因使用者選取同一群組的其他RadioButton)，也會產生CheckedChanged事件。
3. CheckBox被核取之後，會產生CheckedChanged事件。
4. CheckBox 被取消核取之後，也會產生CheckedChanged事件。

共享事件處理程序的語法： Handles 物件.事件,物件.事件

- ❖ 在CheckBox1_CheckedChanged事件程序的「Handles
CheckBox1.CheckedChanged」之後加上：「
CheckBox2.CheckedChanged,
CheckBox3.CheckedChanged」表示
CheckBox1_CheckedChanged事件程序除了用來處理
CheckBox1的CheckedChanged事件之外，也用來處理
CheckBox2及CheckBox3的CheckedChanged事件。

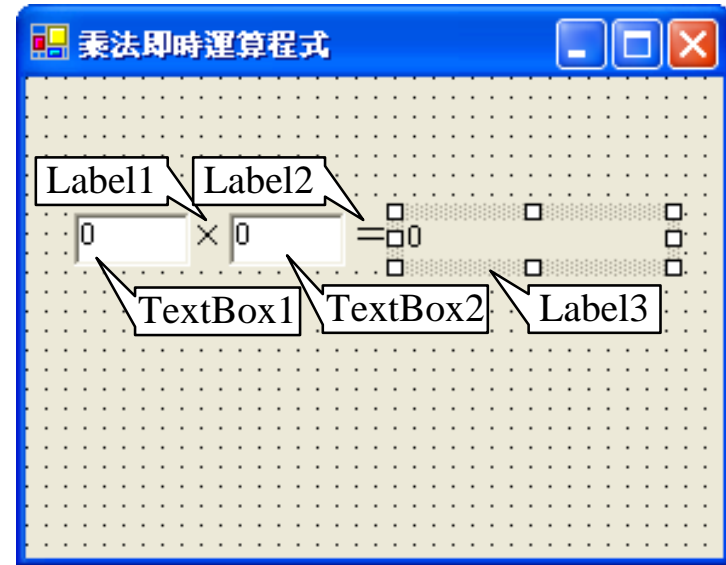
乘法即時運算程式



TextChanged事件：發生於文字內容改變時

上機：乘法即時運算程式

1. 建立一Windows應用程式專案。
2. 在表單上佈置兩個 TextBox 及三個 Label。
3. 以滑鼠雙按 TextBox1，出現 TextBox1_TextChanged 事件程序時，輸入程式：



```
Private Sub TextBox1_TextChanged(...) Handles TextBox1.TextChanged
    If IsNumeric(TextBox1.Text) And IsNumeric(TextBox2.Text) Then
        Label3.Text = Val(TextBox1.Text) * Val(TextBox2.Text)
    Else
        Label3.Text = "(請輸入數值資料)"
    End If
End Sub
```

上機：乘法即時運算程式

4. 修改第一行程式，使這個事件程序能夠同時處理TextBox1及TextBox2的TextChanged事件，修改後的敘述如下：

```
Private Sub TextBox1_TextChanged(...) Handles  
    TextBox1.TextChanged, TextBox2.TextChanged
```

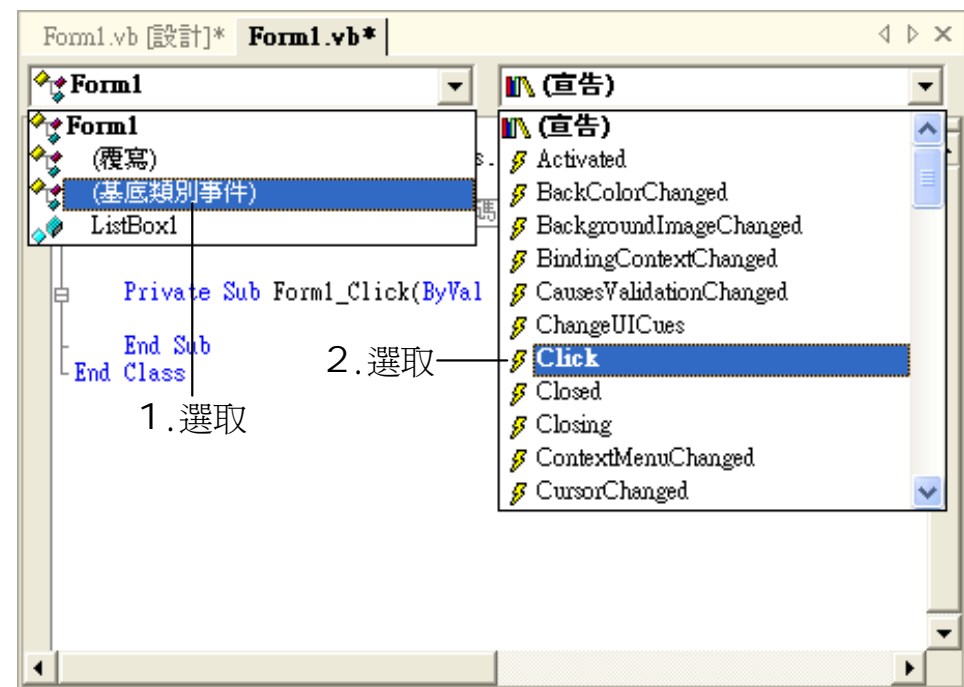
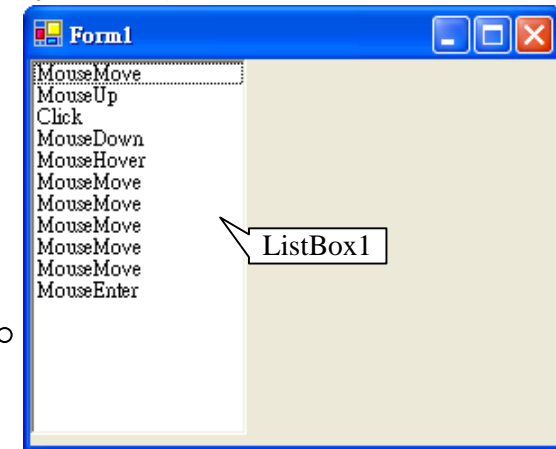
5. 執行程式，然後分別在TextBox1及TextBox2之中輸入數值資料，接著相乘的結果就會顯示在 '=' 之後。

滑鼠事件

- Click 事件：在物件上面按下滑鼠且放開。
- DoubleClick 事件：在物件上面以滑鼠快按兩下並且放開。
- MouseDown 事件：在物件上面按下滑鼠。
- MouseUp 事件：發生MouseDown事件後，放開滑鼠。
- MouseMove 事件：物件上面移動滑鼠。
- MouseEnter 事件：從物件外面移動滑鼠到物件上面。
- MouseLeave 事件：從物件上面移動滑鼠到物件外面。
- MouseHover 事件：發生MouseEnter事件後，滑鼠停止移動。

上機：滑鼠事件初體驗

1. 建立一Windows應用程式專案。
2. 在表單上左邊佈置ListBox，此一ListBox用來顯示表單的滑鼠事件。
3. 開啓程式模組視窗。
4. 在程式模組中，選取物件中的「(基底類別事件)」，選取事件中的「Click」，接著會出現Form1_Click事件程序，過程如圖。



上機：滑鼠事件初體驗

5. 在Form1_Click事件程序輸入以下程式：

```
Private Sub Form1_Click(...) Handles Me.Click  
    ListBox1.Items.Insert(0, "Click")  
End Sub
```

6. 分別在Form1_DoubleClick, Form1_MouseDown, Form1_MouseEnter, Form1_MouseHover, Form1_MouseLeave, Form1_MouseMove, Form1_MouseUp事件中輸入以下程式：

滑鼠事件初體驗

```
Private Sub Form1_DoubleClick(...) Handles Me.DoubleClick  
    ListBox1.Items.Insert(0, "DoubleClick")  
End Sub
```

```
Private Sub Form1_MouseDown(...) Handles Me.MouseDown  
    ListBox1.Items.Insert(0, "MouseDown")  
End Sub
```

```
Private Sub Form1_MouseEnter(...) Handles Me.MouseEnter  
    ListBox1.Items.Insert(0, "MouseEnter")  
End Sub
```

```
Private Sub Form1_MouseHover(...) Handles Me.MouseHover  
    ListBox1.Items.Insert(0, "MouseHover")  
End Sub
```

```
Private Sub Form1_MouseLeave(...) Handles Me.MouseLeave  
    ListBox1.Items.Insert(0, "MouseLeave")  
End Sub
```

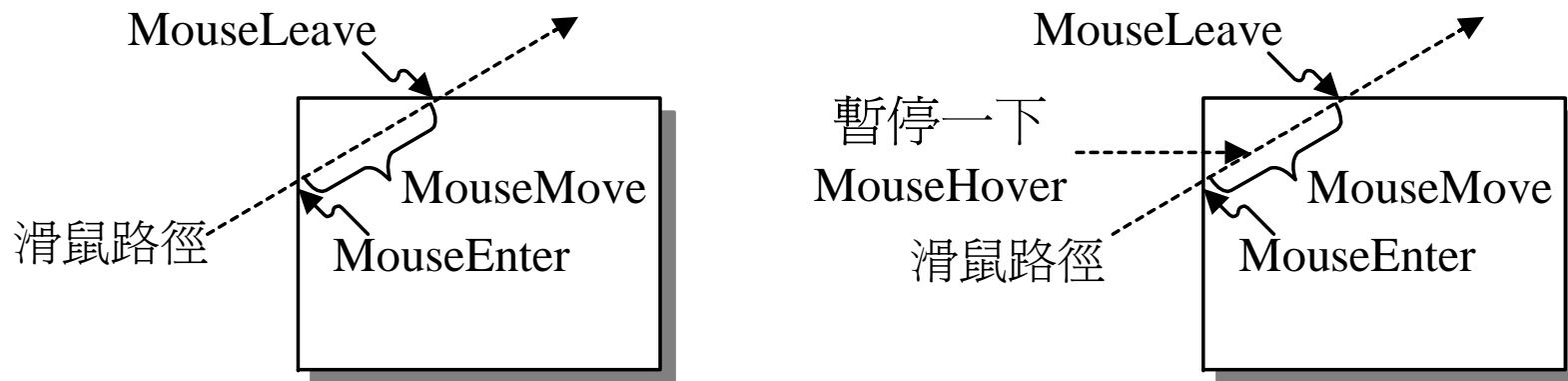
```
Private Sub Form1_MouseMove(...) Handles Me.MouseMove  
    ListBox1.Items.Insert(0, "MouseMove")  
End Sub
```

```
Private Sub Form1_MouseUp(...) Handles Me.MouseUp  
    ListBox1.Items.Insert(0, "MouseUp")  
End Sub
```

滑鼠事件初體驗

7. 執行程式，首先將滑鼠從表單之外移到表單之上，然後暫停一下，結果ListBox會新增一行MouseEnter, 多行MouseMove及一行MouseHover。
8. 在表單上按下滑鼠並且放開，結果ListBox會新增MouseDown, Click, MouseUp, MouseMove各一行。
9. 在表單上快按滑鼠兩下並且放開，結果ListBox會新增MouseDown, Click, MouseUp, MouseMove, MouseDown, DoubleClick, MouseUp, MouseMove各一行。
10. 將滑鼠移出表單，結果ListBox會新增MouseLeave一行。

MouseHover 事件



使用MouseHover的時機？
需要提供使用者訊息的時候

滑鼠事件所提供之資訊

- ❖ 在 `MouseDown`、`MouseMove`與`MouseUp` 事件中，我們可以進一步得到以下的滑鼠資訊：
 - ✓ 滑鼠座標：滑鼠所在位置的座標，其中X座標以`e.X`讀取, Y座標以`e.Y`讀取。
 - ✓ 按鈕：使用者按下哪個滑鼠鈕，以`e.Button`讀取，可能的數值如下：

e.Button的數值	意義
<code>MouseButtons.Left</code>	表示使用者按下滑鼠左鈕
<code>MouseButtons.Right</code>	表示使用者按下滑鼠右鈕
<code>MouseButtons.Middle</code>	表示使用者按下滑鼠中間鈕
<code>MouseButtons.None</code>	表示使用者沒有按下滑鼠鈕
<code>MouseButtons.XButton1</code>	表示使用者按下滑鼠第一個特殊鈕
<code>MouseButtons.XButton2</code>	表示使用者按下滑鼠第二個特殊鈕

上機：顯示滑鼠移動的軌跡

1. 建立一Windows應用程式專案。
2. 在表單的MouseMove事件程序(Form1_MouseMove)之中輸入以下程式：

```
Private Sub Form1_MouseMove(...) Handles Me.MouseMove  
    Dim g As Graphics = Me.CreateGraphics()  
    g.DrawEllipse(New Pen(Color.Black), e.X, e.Y, 3, 3)  
    g.Dispose()  
End Sub
```

3. 執行程式，並且在表單之中移動滑鼠，即可看到滑鼠移動的軌跡。



繪圖的一般指令

繪圖之前需先宣告繪圖物件

→ `Dim 繪圖物件 as Graphics`

然而Graphics是抽象類別，無法直接使用，必須使用CreateGraphics方法取得繪圖區域

→ `繪圖物件=物件.CreateGraphics`

物件可為Form, GroupBox, PictureBox等，即為繪圖區域
繪圖指令：

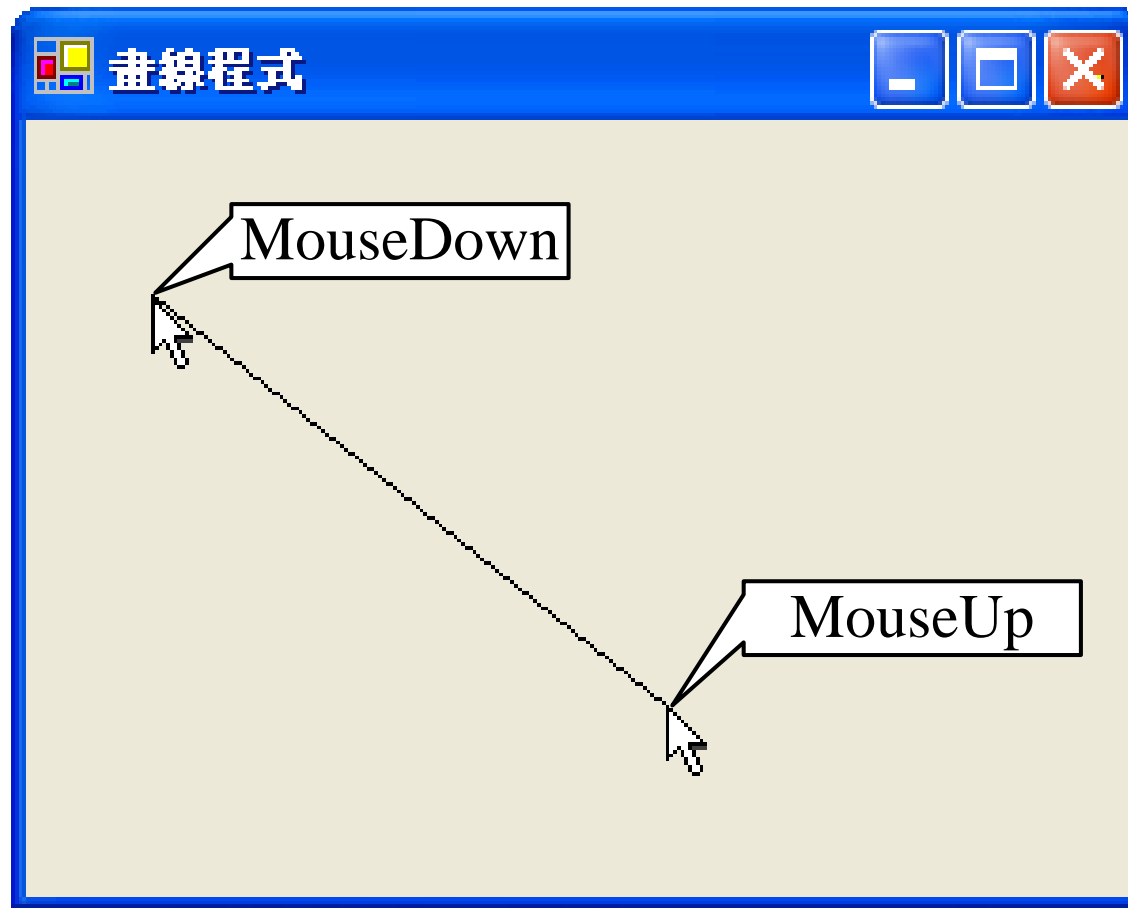
→ `繪圖物件.繪圖方法(筆物件,座標與大小)`

釋放繪圖所佔用的系統資源

→ `繪圖物件.Dispose`

繪圖方法有DrawLine, DrawRectangle, DrawEllipse等
在非表單內繪圖需特別注意相關參數

寫一個畫線的程式



寫一個畫線的程式

1. 建立一Windows應用程式專案。
2. 分別在Form1的「(宣告)區」、MouseDown及MouseUp事件程序中撰寫以下程式：

'(宣告) 區

Dim X, Y

Private Sub Form1_MouseDown(...) Handles Me.MouseDown

 X = e.X

 Y = e.Y

End Sub

Private Sub Form1_MouseUp(...) Handles Me.MouseUp

 Dim g As Graphics = Me.CreateGraphics()

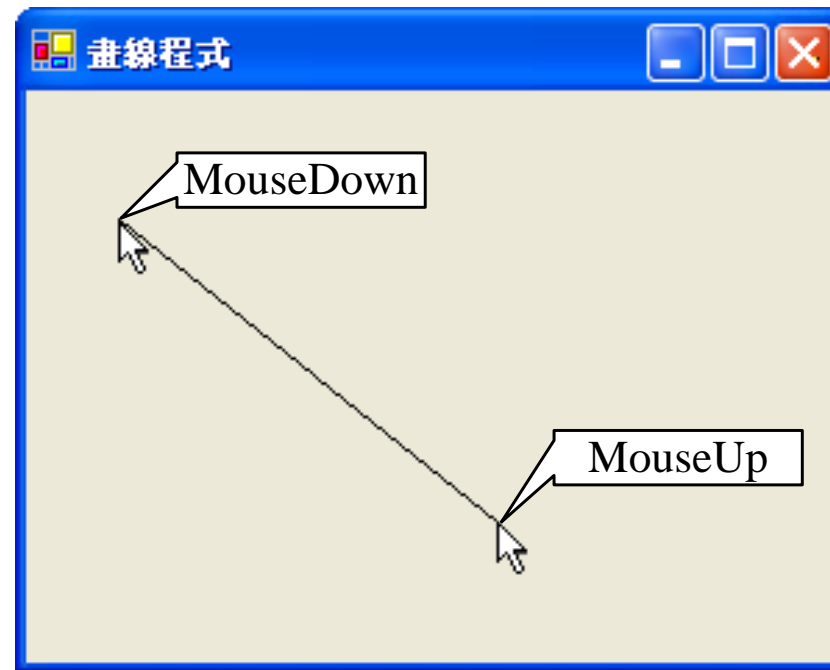
 g.DrawLine(New Pen(Color.Black), X, Y, e.X, e.Y)

 g.Dispose()

End Sub

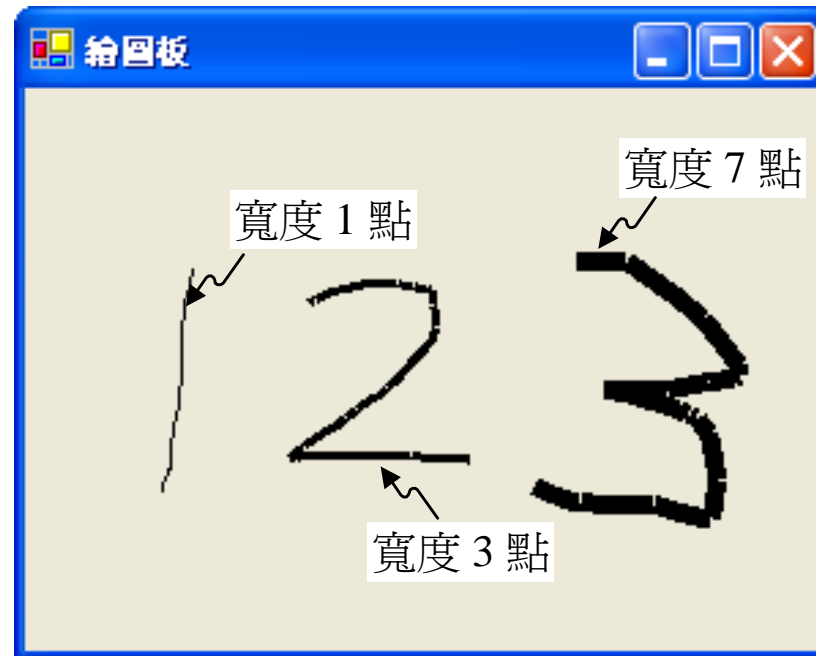
寫一個畫線的程式

3. 執行程式，然後在表單上按住滑鼠(不要放開)以決定線段的第一點，接著移動滑鼠位置，待決定線段的第二點時，放開滑鼠，即可將線段畫出。



寫一個繪圖板程式

❖ 功能: (1)以按住滑鼠及移動滑鼠的方式繪圖 (2)如果按住的滑鼠鈕為左鈕，以1點寬的線段繪圖 (3)如果按住的滑鼠鈕為右鈕，以3點寬的線段繪圖 (4)如果按住的滑鼠鈕為中間鈕，以7點寬的線段繪圖。



寫一個繪圖板程式

1. 建立一Windows應用程式專案。
2. 分別在Form1的「(宣告)區」、MouseDown及MouseMove事件程序中撰寫以下程式：

' (宣告)區

Dim X

Dim Y

Private Sub Form1_MouseDown(...) Handles Me.MouseDown

 X = e.X

 Y = e.Y

End Sub

寫一個繪圖板程式

```
Private Sub Form1_MouseMove(...) Handles Me.MouseMove
```

```
    If e.Button = MouseButton.None Then Exit Sub
```

```
    Dim p As Pen
```

```
    If e.Button = MouseButton.Left Then
```

```
        p = New Pen(Color.Black) '預設值為1
```

```
    ElseIf e.Button = MouseButton.Right Then
```

```
        p = New Pen(Color.Black, 3)
```

```
    ElseIf e.Button = MouseButton.Middle Then
```

```
        p = New Pen(Color.Black, 7)
```

```
    End If
```

```
    Dim g As Graphics = Me.CreateGraphics()
```

```
    g.DrawLine(p, X, Y, e.X, e.Y)
```

```
    X = e.X
```

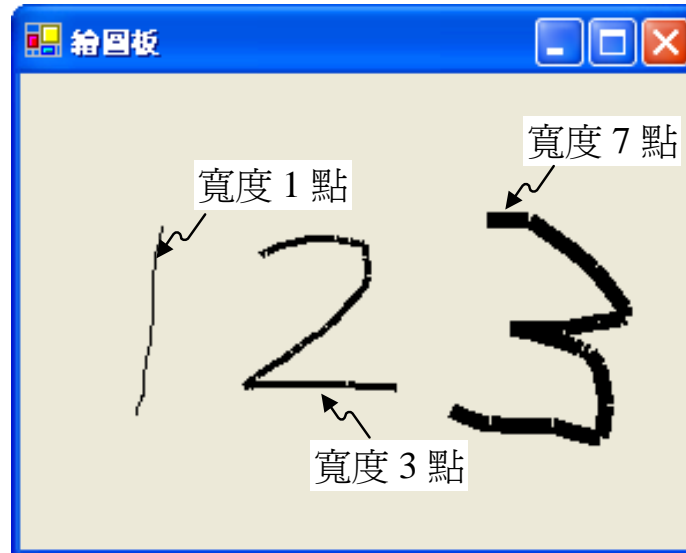
```
    Y = e.Y
```

```
    g.Dispose()
```

```
End Sub
```

寫一個繪圖板程式

3. 執行程式，先按住左鈕，移動滑鼠畫一個「1」(此時圖形的寬度為1點)，然後放開滑鼠，接著按住右鈕，移動滑鼠畫一個「2」(此時圖形的寬度為3點)，然後放開滑鼠，最後再按住中間鈕，移動滑鼠畫一個「3」(此時圖形的寬度為7點)，然後放開滑鼠。



鍵盤事件

- ❖ **KeyPress** 事件：發生於輸入字元。
- ❖ **KeyDown** 事件：發生於按下按鍵。
- ❖ **KeyUp** 事件：發生於放開按鍵。

非顯示型字元

組合鍵	的字元碼	組合鍵	的字元碼
Shift + Ctrl + @	0	Ctrl + Q	17
Ctrl + A	1	Ctrl + R	18
Ctrl + B	2	Ctrl + S	19
Ctrl + C	3	Ctrl + T	20
Ctrl + D	4	Ctrl + U	21
Ctrl + E	5	Ctrl + V	22
Ctrl + F	6	Ctrl + W	23
Ctrl + G	7	Ctrl + X	24
Ctrl + H或Backspace	8	Ctrl + Y	25
Ctrl + I或Tab	9	Ctrl + Z	26
Ctrl + J或Ctrl + Enter	10	Ctrl + [或Esc	27
Ctrl + K	11	Ctrl + \	28
Ctrl + L	12	Ctrl +]	29
Ctrl + M或Enter	13	Shift + Ctrl + ^	30
Ctrl + N	14	Shift + Ctrl + _	31
Ctrl + O	15	Ctrl + Backspace	127
Ctrl + P	16		



利用控制鍵改變表單背景顏色

按鍵組合→字元碼

Ctrl+R → 18

Ctrl+G → 7

Ctrl+B → 2

建立一Windows應用程式專案。

Private Sub Form1_KeyPress(...) Handles Me.KeyPress

Dim 字元碼 As Integer = Asc(e.KeyChar)

If 字元碼 = 18 Then

Me.BackColor = Color.Red

ElseIf 字元碼 = 7 Then

Me.BackColor = Color.Green

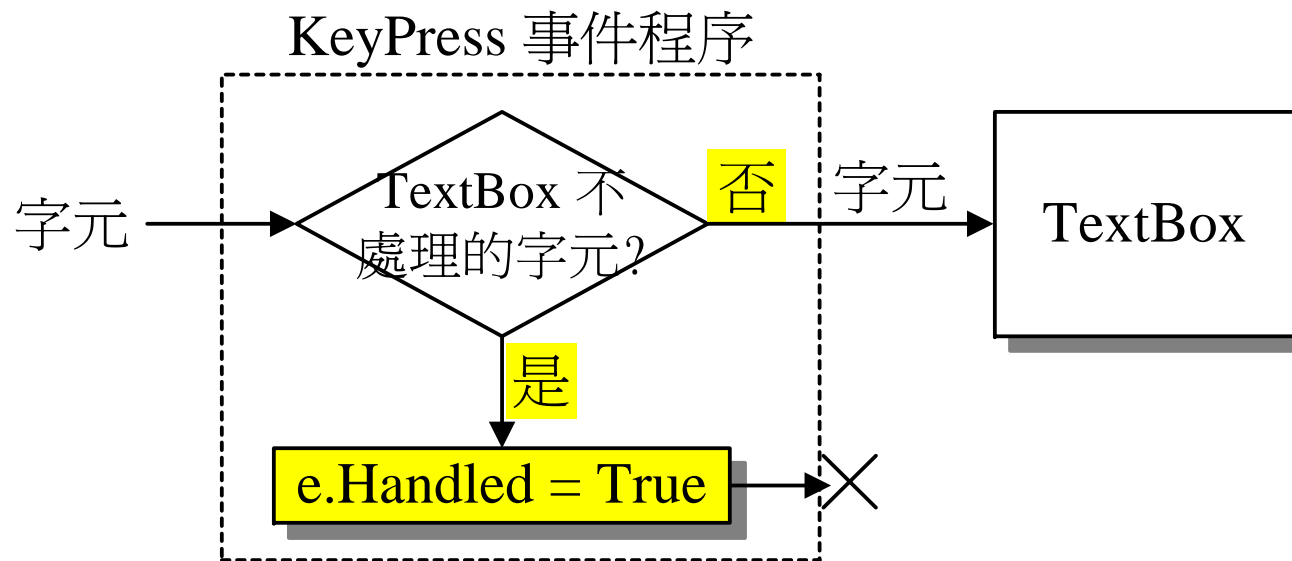
ElseIf 字元碼 = 2 Then

Me.BackColor = Color.Blue

End If

End Sub

利用e.Handle過濾字元



e.Handle設定為True，則程式不處理鍵盤的輸入事件

只能輸入數字及小數點的 TextBox

1. 建立一Windows應用程式專案。
2. 在表單上佈置一個TextBox。
3. 在TextBox1_KeyPress事件程序中輸入以下程式：

```
Private Sub TextBox1_KeyPress(...) Handles TextBox1.KeyPress
    Dim 字元碼 As Integer = Asc(e.KeyChar)
    If 字元碼 >= 32 And 字元碼 <= 126 Then
        If Not (字元碼 >= 48 And 字元碼 <= 57) And 字元碼 <> 46 Then
            e.Handled = True '不屬於數字及小數點, 將其過濾掉
        End If
    End If
End Sub
```

只能輸入數字及小數點的 TextBox

4. 執行程式，然後試著在TextBox之中輸入資料，結果只能輸入數字及小數點。

□ 程式解說：

字元碼 ≥ 32 And 字元碼 ≤ 126 ：判斷字元是否為可顯示型字元。

Not (字元碼 ≥ 48 And 字元碼 ≤ 57) And 字元碼 $\neq 46$ ：判斷字元是否不屬於數字及小數點。

ASCII 字元碼的分佈

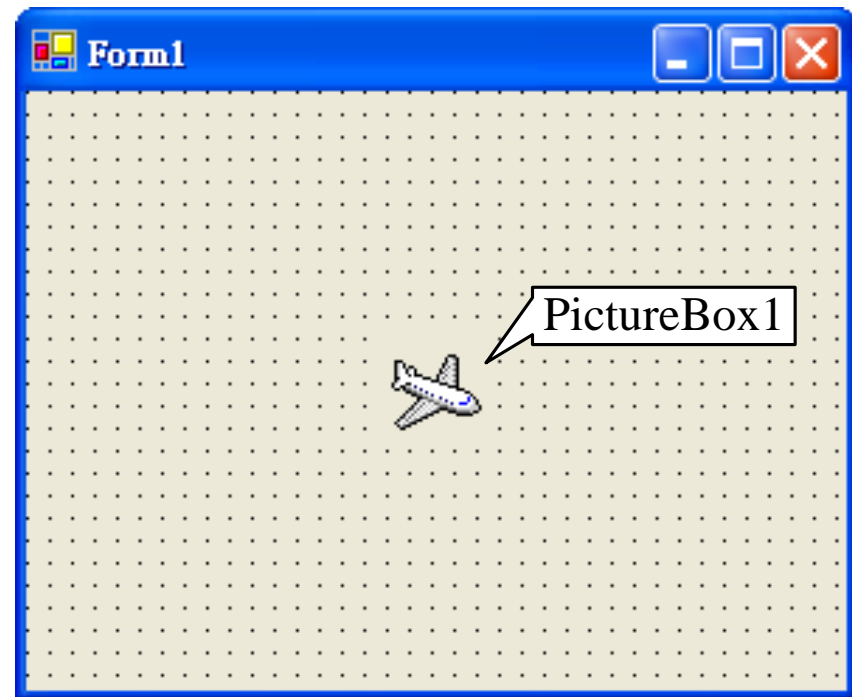
字元	字元碼 10進位	字元碼 16進位		字元	字元碼 10進位	字元碼 16進位		字元	字元碼 10進位	字元碼 16進位
	32	20		@	64	40		`	96	60
!	33	21		A	65	41		a	97	61
"	34	22		B	66	42		b	98	62
#	35	23		C	67	43		c	99	63
\$	36	24		D	68	44		d	100	64
%	37	25		E	69	45		e	101	65
&	38	26		F	70	46		f	102	66
'	39	27		G	71	47		g	103	67
(40	28		H	72	48		h	104	68
)	41	29		I	73	49		i	105	69
*	42	2A		J	74	4A		j	106	6A
+	43	2B		K	75	4B		k	107	6B
,	44	2C		L	76	4C		l	108	6C
-	45	2D		M	77	4D		m	109	6D
.	46	2E		N	78	4E		n	110	6E
/	47	2F		O	79	4F		o	111	6F
0	48	30		P	80	50		p	112	70
1	49	31		Q	81	51		q	113	71
2	50	32		R	82	52		r	114	72
3	51	33		S	83	53		s	115	73
4	52	34		T	84	54		t	116	74
5	53	35		U	85	55		u	117	75
6	54	36		V	86	56		v	118	76
7	55	37		W	87	57		w	119	77
8	56	38		X	88	58		x	120	78
9	57	39		Y	89	59		y	121	79
:	58	3A		Z	90	5A		z	122	7A
;	59	3B		[91	5B		{	123	7B
<	60	3C		\	92	5C			124	7C
=	61	3D]	93	5D		}	125	7D
>	62	3E		^	94	5E		~	126	7E
?	63	3F		_	95	5F				



移動飛機程式

❖ 目的：寫一個可以移動飛機圖片的程式：每按「上下左右鍵」一次就移動飛機一點；但如果先按住Shift鍵，則每按「上下左右鍵」一次就移動飛機十點。

1. 建立一Windows應用程式專案。
2. 在表單上佈置一個PictureBox，並匯入圖像
SizeMode 屬性 = AutoSize



上機：移動飛機程式

3. 在Form1_KeyDown事件程序中輸入以下程式：

```
Private Sub Form1_KeyDown(...) Handles Me.KeyDown
    Dim X As Integer = PictureBox1.Location.X
    Dim Y As Integer = PictureBox1.Location.Y
    Dim 點數 As Integer = 1
    If e.Shift Then 點數 = 10
    If e.KeyCode = Keys.Left Then
        X -= 點數
    ElseIf e.KeyCode = Keys.Right Then
        X += 點數
    ElseIf e.KeyCode = Keys.Up Then
        Y -= 點數
    ElseIf e.KeyCode = Keys.Down Then
        Y += 點數
    End If
    PictureBox1.Location = New Point(X, Y)
End Sub
```

上機：移動飛機程式

4. 執行程式，然後按上下左右鍵移動飛機圖片，或是先按住Shift鍵，再按上下左右鍵快速移動飛機圖片。
- 注意事項：在以上程式中，我們已知上下左右鍵對應的「按鍵碼」是Up, Down, Left, Right，但是程式中要寫成Keys.Up, Keys.Down, Keys.Left, Keys.Right，不可以省略Keys。除了上下左右鍵之外，在程式中表示按鍵一概要加上Keys.。