

6.1 副程式(函式)一寫程式的新觀念

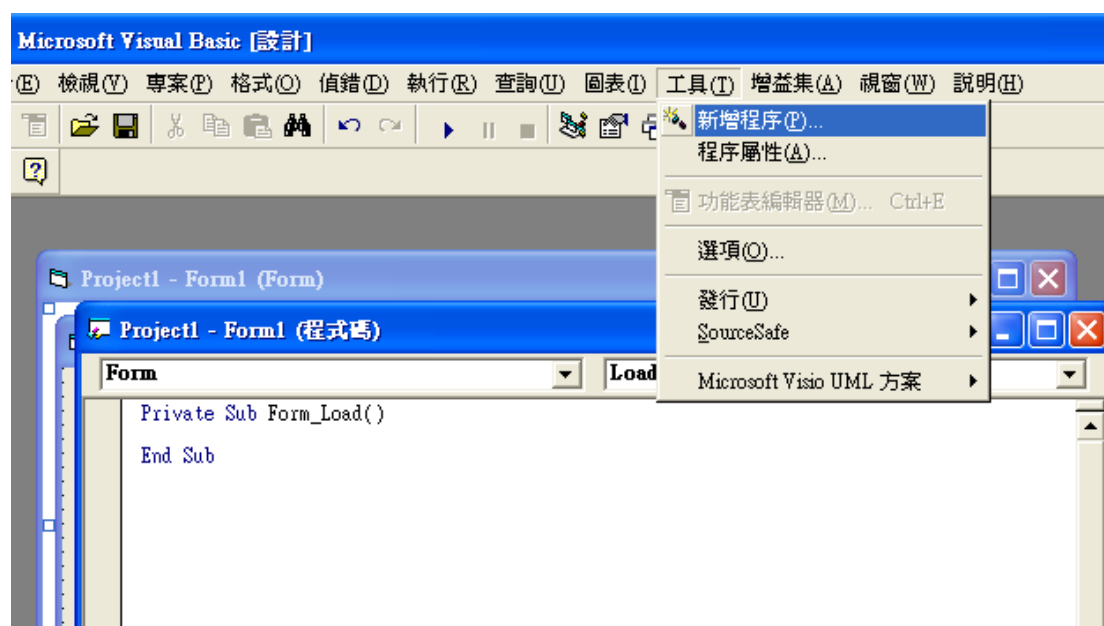
觀看上一章的賓果程式是不是會發現有許多地方都重複了，如果能夠只寫一次，那該有多好，副程式的觀念就是用以解決這個擾人的問題。

常用的副程式分為二種：Sub 及 Function，這二者的差別在於是否可以回傳一個值給主程式，Sub 的角色就像一個小兵，當班長宣佈完命令，不敬禮解散之後，Sub 就把本身內部的程式碼執行一次，之後就納涼去了。而 Function 的角色則像個學生，當老師(主程式)宣佈完作業後，Function 會將內部的程式碼執行過一次之後，回傳程式中所設定的值給主程式，就彷彿像交作業一樣。

在 VB 中提供了四種副程式的類型，在本章中只介紹常用的 Sub 及 Function。

6.2 新增副程式(函式)的方法

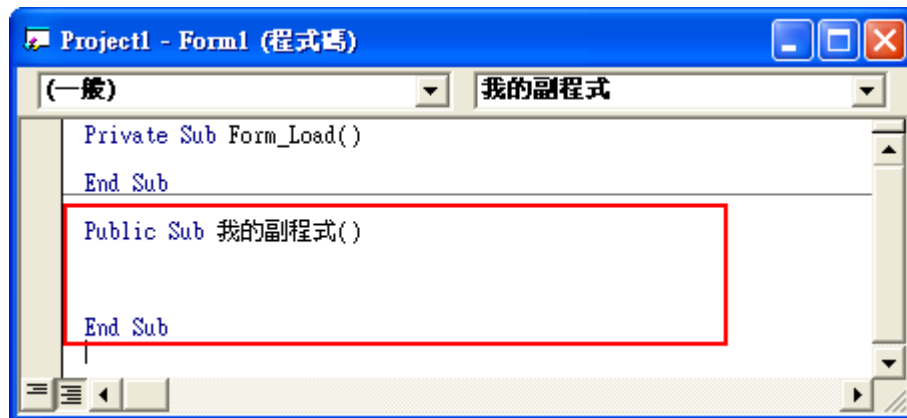
1.工具列→工具→新增程序



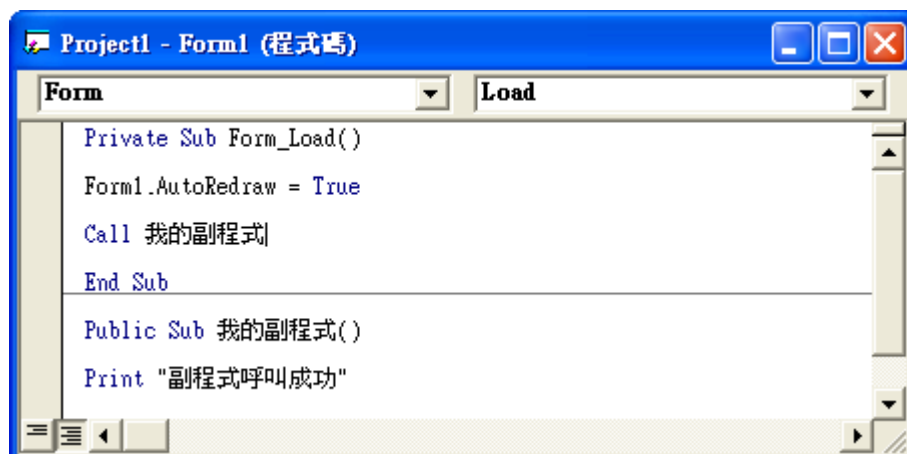
2. 副程式名稱支援中文



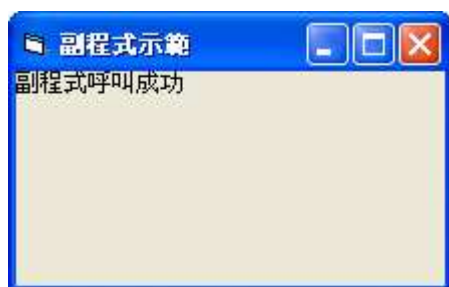
3. 紅色框框的部份就是新增的副程式



4. 接下來我們試著在副程式中寫一些簡單的程式碼，並由主程式呼叫



5. 執行結果



6.3 帶有參數的副程式

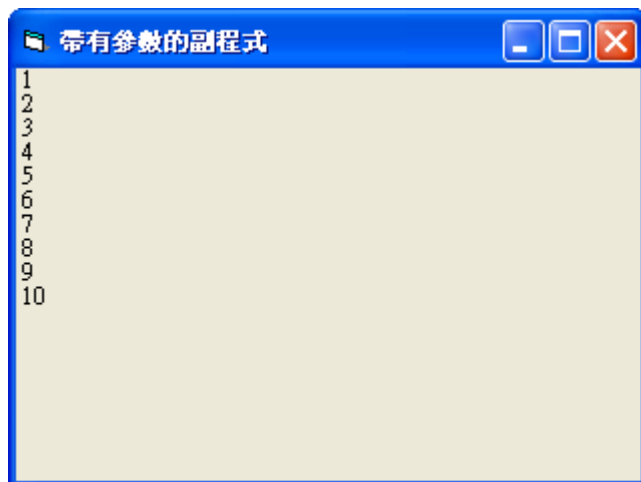
參數的意思是指，由主程式傳遞給副程式的一個值，這樣做的好處是，不需使用全域變數，節省程式佔用記憶體的空間，程式也較容易閱讀。請看以下示範

1. 如需使用參數，只需要在副程式名稱後方的 () 中宣告一個變數及型態，不必加 DIM，主程式要傳遞給副程式的值也是放在 () 中，記得一點，數量一定要相同，如果副程式只有宣告一個參數，而主程式傳遞二個是不行的。



★ 主程式所傳遞給副程式的參數，名稱不必相同，副程式會自行將 j 的值複製一份到 i 中來使用。

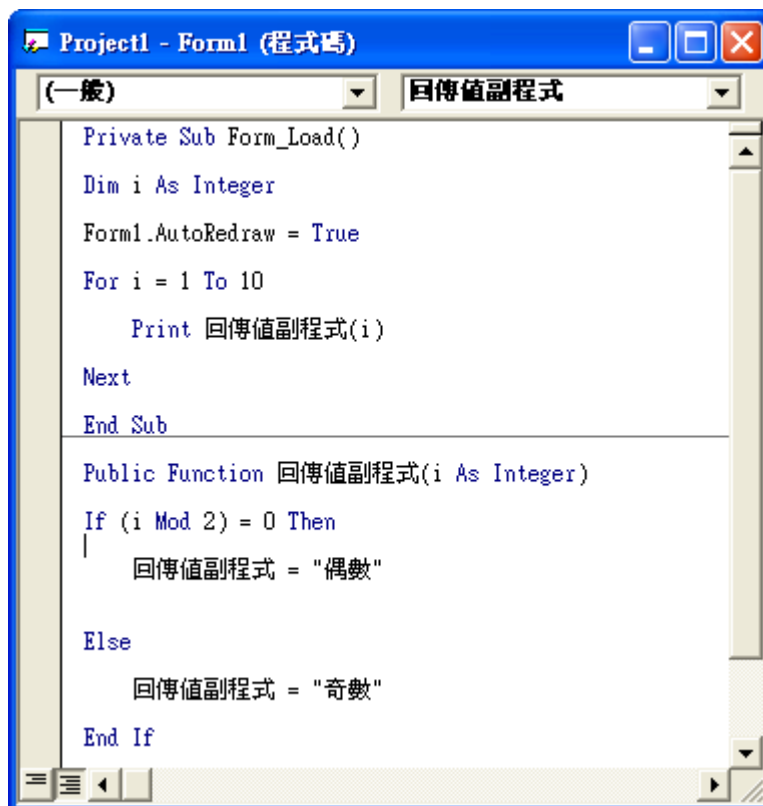
2.執行結果



6.4 具回傳值的副程式—Function

前面我們所介紹的副程式使用的都是 Sub(不具回傳值的副程式)，現在我們要示範如何使用具有回傳值的副程式，並讓讀者了解其特色。

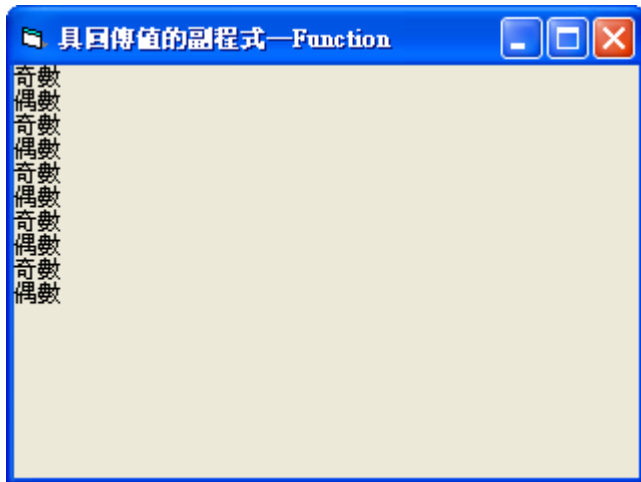
1.程式碼撰寫



★在主程式中，『 Print 回傳值副程式(i) 』這一行， 代表著將副程式傳回的值印出來

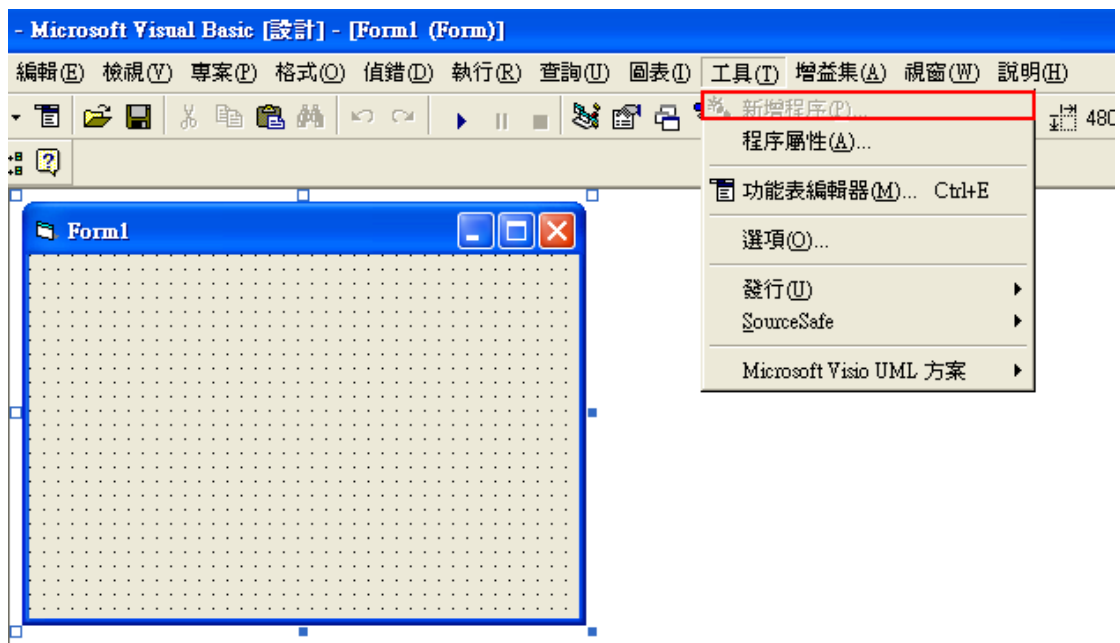
☆ 副程式中『回傳值副程式 = “ 奇數 ” 』這一行則代表將副程式的傳回值設為一字串” 奇數” 而主程式則將『奇數』印在表單上。

2.執行結果



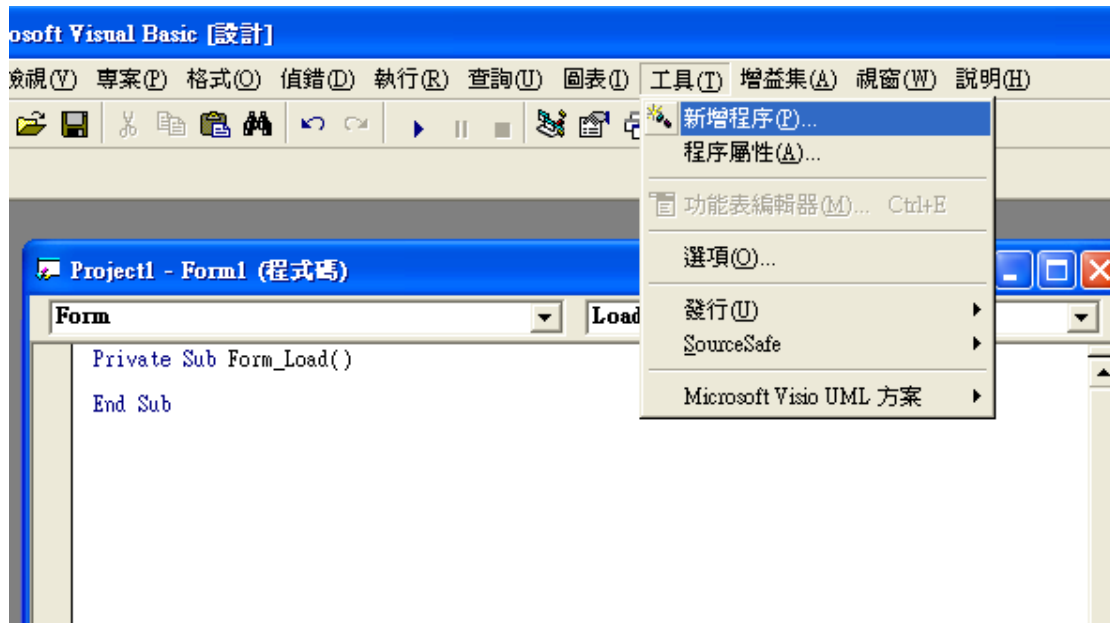
小幫手特區：新手容易犯的錯之新增副程式

1.問題



★新增程序的地方變成灰色的~~怎麼辦~~誰來救我!!!

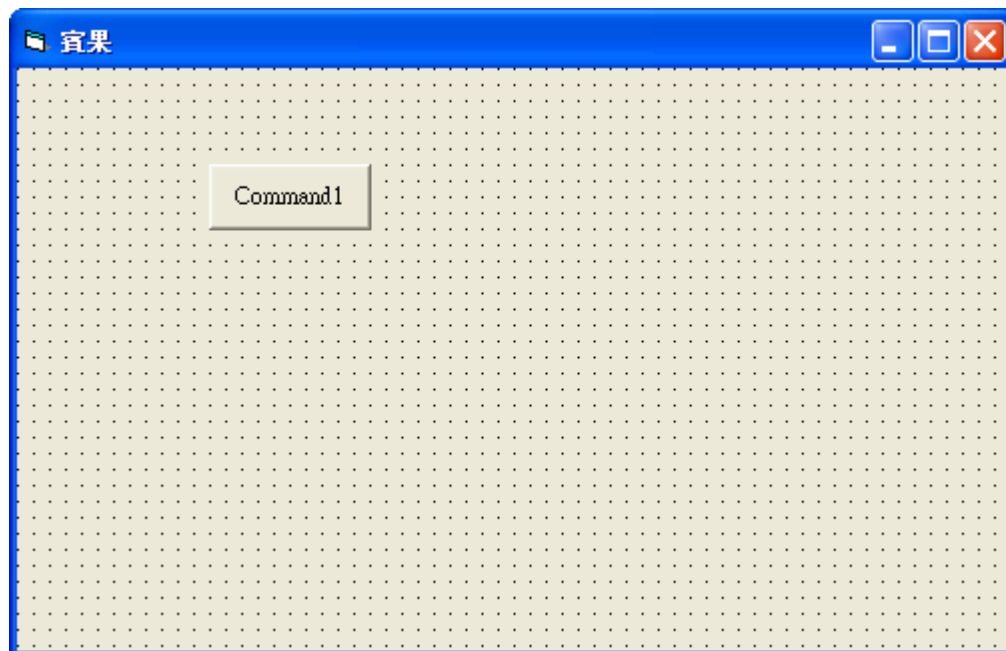
2.解法



☆原來要在程式碼的畫面下才能夠新增副程式。

6.5 以副程式方式改寫賓果程式

1.版面配置



2. 屬性設定



3. 撰寫程式碼

```
Private Sub Command1_Click(Index As Integer)
```

```
Dim i As Integer
```

```
Dim chick(11, 2) As Integer '將要檢查的位置起點、終點、遞增值存入陣列
```

```
Command1(Index).Enabled = False '將按下的數字變成灰色(不能再按)
```

```
'=====設定檢查點=====
```

```
chick(0, 0) = 0: chick(0, 1) = 20: chick(0, 2) = 5
```

```
chick(1, 0) = 1: chick(1, 1) = 21: chick(1, 2) = 5
```

```
chick(2, 0) = 2: chick(2, 1) = 22: chick(2, 2) = 5
```

```
chick(3, 0) = 3: chick(3, 1) = 23: chick(3, 2) = 5
```

```
chick(4, 0) = 4: chick(4, 1) = 24: chick(4, 2) = 5
```

```
chick(5, 0) = 0: chick(5, 1) = 4: chick(5, 2) = 1
```

```
chick(6, 0) = 5: chick(6, 1) = 9: chick(6, 2) = 1
```

chick(7, 0) = 10: chick(7, 1) = 14: chick(7, 2) = 1

chick(8, 0) = 15: chick(8, 1) = 19: chick(8, 2) = 1

chick(9, 0) = 20: chick(9, 1) = 24: chick(9, 2) = 1

chick(10, 0) = 0: chick(10, 1) = 24: chick(10, 2) = 6

chick(11, 0) = 4: chick(11, 1) = 20: chick(11, 2) = 4

'=====呼叫檢查副程式=====

For i = 0 To 11

 Call chick_Line(chick(i, 0), chick(i, 1), chick(i, 2))

Next

End Sub

Private Sub Form_Load()

Dim Number(25), R, Temp As Integer

Randomize '避免亂數重覆

Form1.ScaleWidth = 5600 '設定表單的大小

Form1.ScaleHeight = 5600

For i = 1 To 24 '載入另外 24 個按鍵

 Load Command1(i)

 Command1(i).Visible = True '並將其顯示在表單上

Next

For i = 1 To 25 '將 1~25 放入陣列中

 Number(i) = i

Next

For i = 1 To 25 '將陣列中的數字任意洗牌

 R = 1 + Int(Rnd * 25)

 Temp = Number(R): Number(R) = Number(i): Number(i) = Temp

Next

For i = 0 To 24 '設定 25 個按鍵的屬性

 Command1(i).Caption = Number(i + 1) '將 1 到 25 顯示在按鍵上

 Command1(i).Width = 800 '按鍵的寬

 Command1(i).Height = 800 '按鍵的高

 Command1(i).Left = 800 + (i Mod 5) * 800 '按鍵在橫向的排列位置

 Command1(i).Top = 800 + (i \ 5) * 800 '按鍵在垂直的排列位置

Next

End Sub

Public Sub chick_Line(i As Integer, j As Integer, k As Integer)

 Dim Bingo As Integer '計算同一條線上有幾個數字被點選

 For i = i To j Step k

If Command1(i).Enabled = False Then '假如被檢查的數字是灰色

Bingo = Bingo + 1 '被點選的數字加 1

End If

Next

If Bingo = 5 Then '假如一條線上有五個被點選就賓果

MsgBox "賓果"

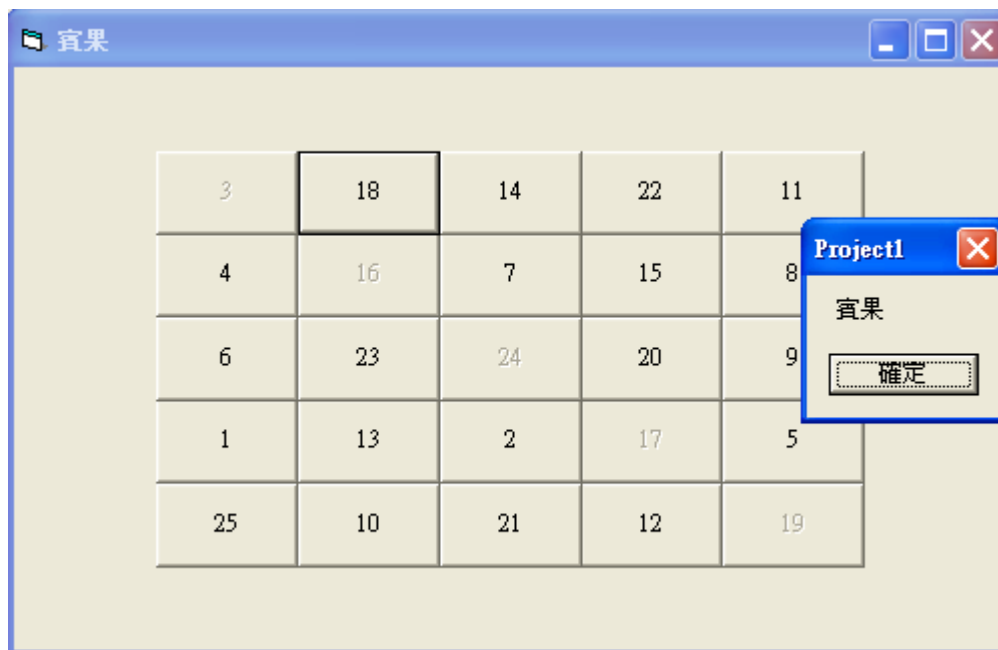
Unload Me '將表單自記憶體中移除

End If

Bingo = 0 '將 Bingo 清除內容

End Sub

4.執行結果



★ 這是不是很有趣，翻回去看看第五章的程式碼，重複的地方都比上面的程式還要長，這也是副程式與陣列及迴圈結合之後好用的地方。

6.6 模組

模組的好處在於可以將多個表單共同需要的函式，放在模組中，只需要撰寫一次即可，減少重複撰寫程式碼的困擾，也使程式更容易閱讀。在本書最後三章，將會出現 WinAPI 函數，這些是 Windows 內建的函數，用來處理一些較為低階的工作，例如讀取系統計數器等的工作，可用來補足 VB 功能上的不足，某些 API 則會被強制要求在模組中宣告。

6.6.1 小算盤 of Windows

我們常常會用到 Windows 內建的小算盤，他有二種模式，1 是『標準型』，2 是『工程型』，這二種模式中有許多相同的功能，例如加、減、乘、除。那我們可以將這四種功能獨立寫成函式放在模組中，這樣就可以節省相同的函式重複寫二遍。

↓ 普通型



↓ 工程型



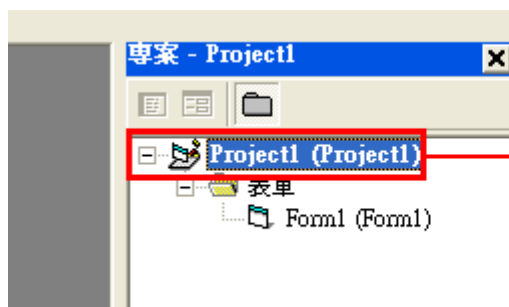
可共用的部份

6.7 多表單的觀念

我們以模組的觀念來寫一個小算盤，一樣有工程型和普通型，在這裡我們將要用到多表單的觀念，以及設定那一個為啟動表單，以下有簡單的圖片介紹。

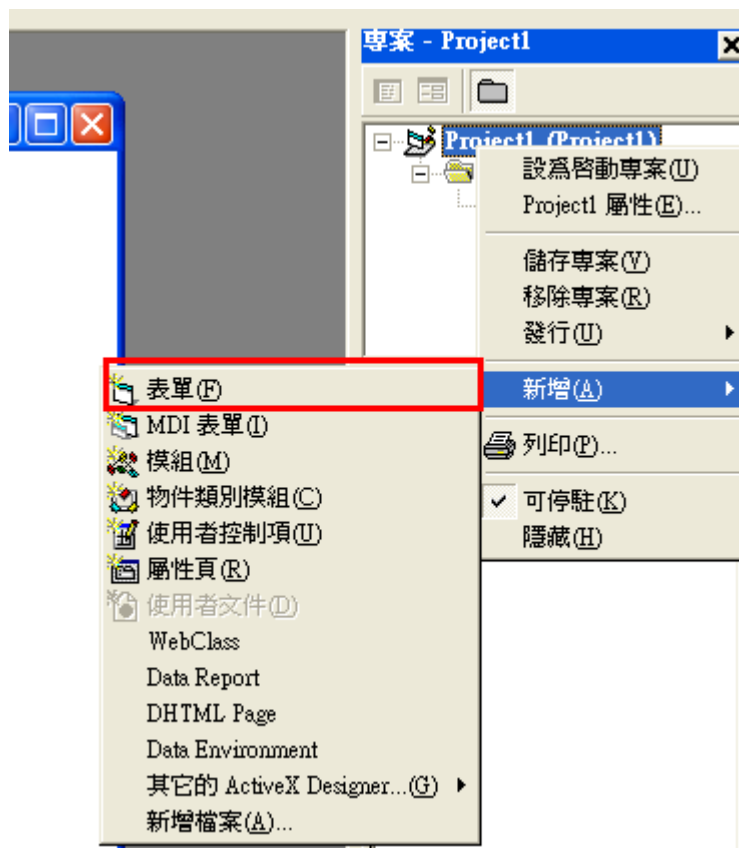
6.7.1 新增一個表單

1. 在專案視窗中的 Project1(Project1)按右鍵



在此按滑鼠右鍵

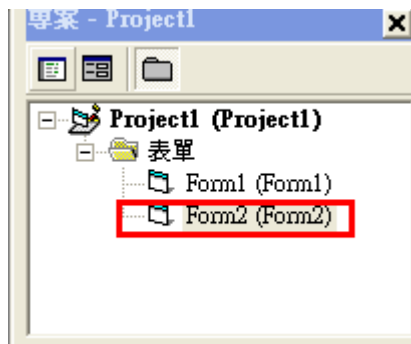
2.在新增中選取『表單(F)』



3.選取『表單』並按下『開啓(O)』



4. Form2 已新增完成

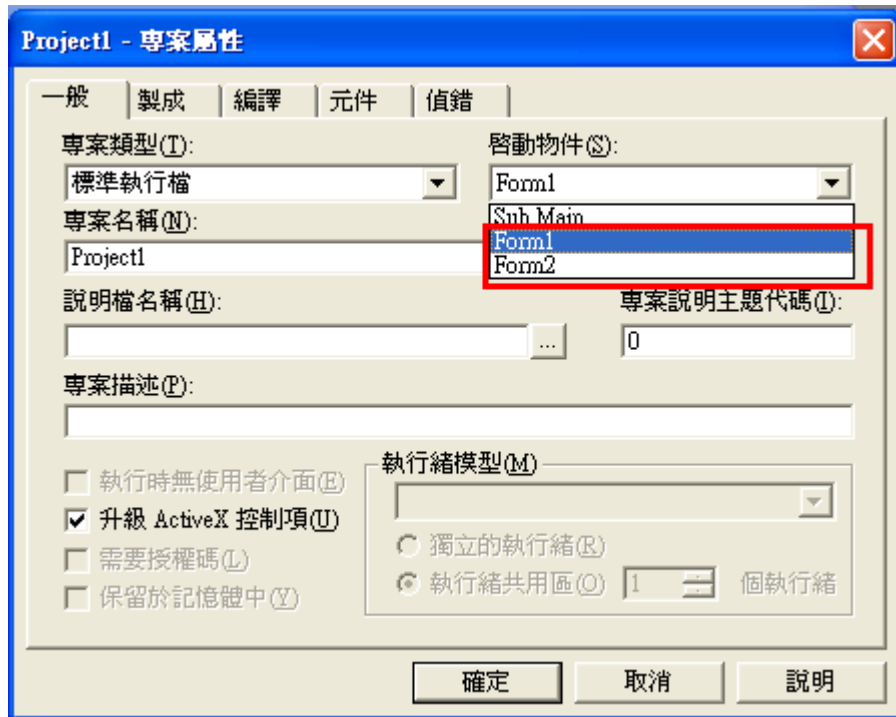


6.7.2 專案屬性設定

1. Project1 屬性設定



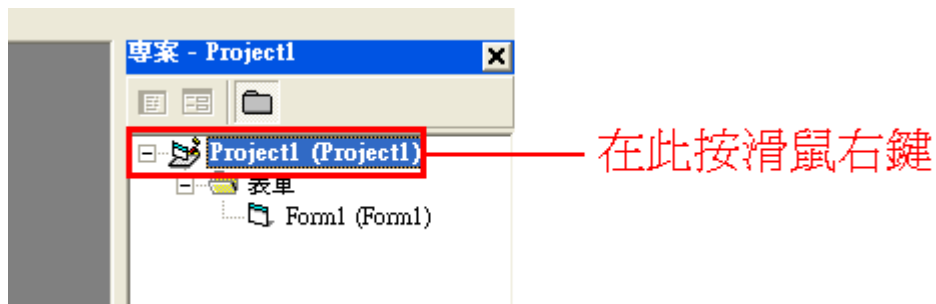
2. 設定起動表單，一般預設為 Form1 為啟動表單，這表式程式一執行會先出現 Form1，Form2 已載入，但是只存在記憶體中，沒有顯示出來。如需要顯示 Form2 時，須在程式碼中輸入『Form2.show』這樣 Form2 就會顯現出來了，如果要將 Form2 隱藏起來，則須在程式碼中輸入『Form2.Hide』即可。



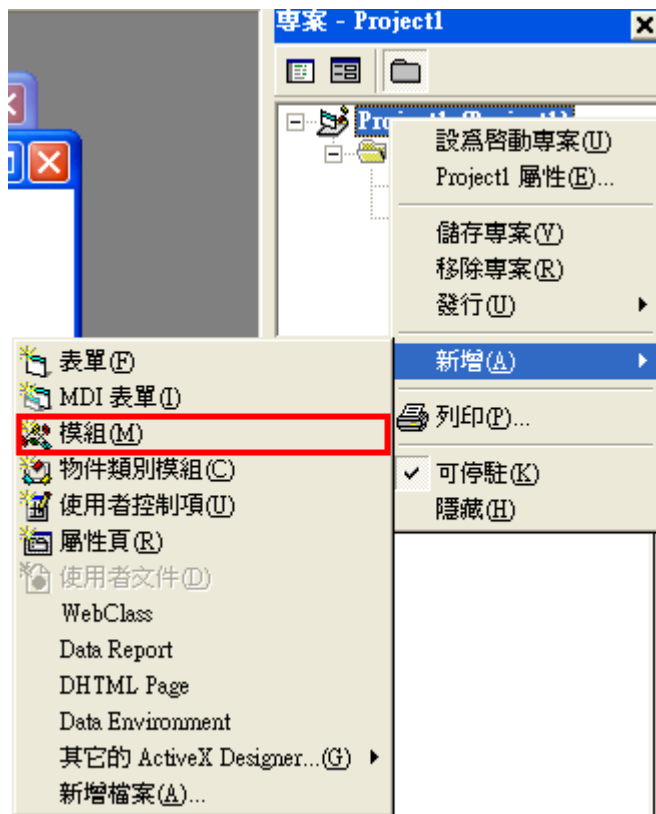
6.8 新增模組

新增模組的方式和新增表單極為相像，請參考下面圖解。

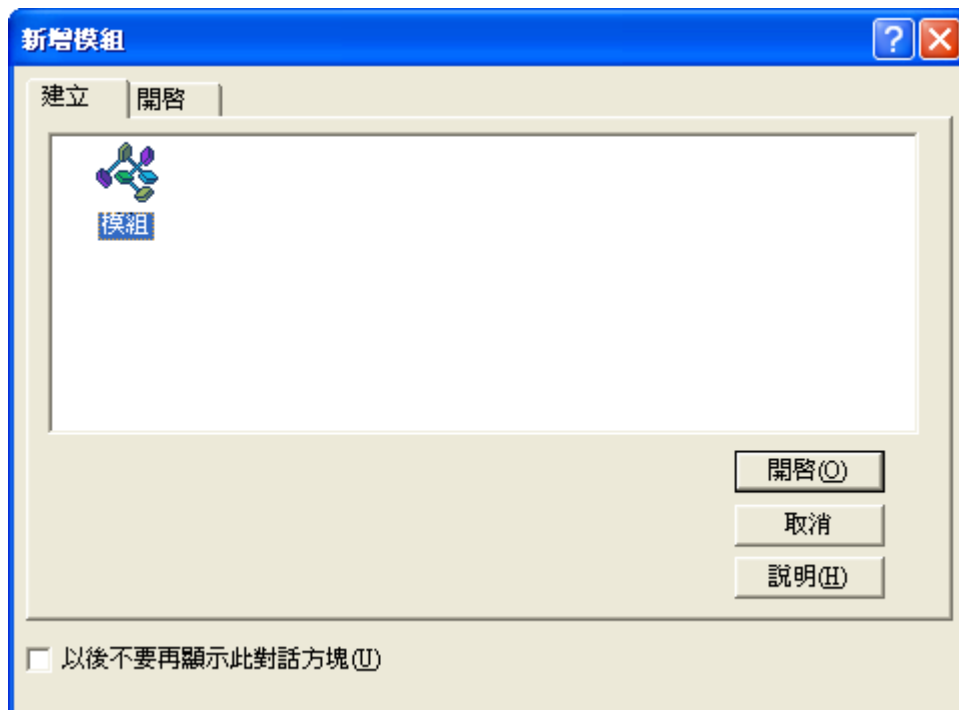
1. 在專案的地方按滑鼠右鍵



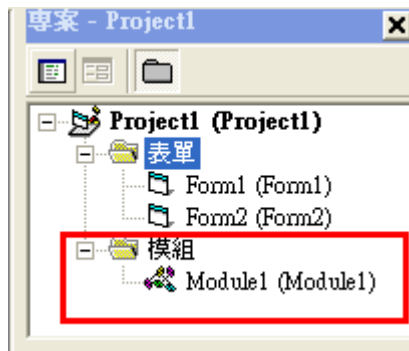
2.在新增中選擇『模組(M)』



3.選擇『模組』，按下『開啓(O)』

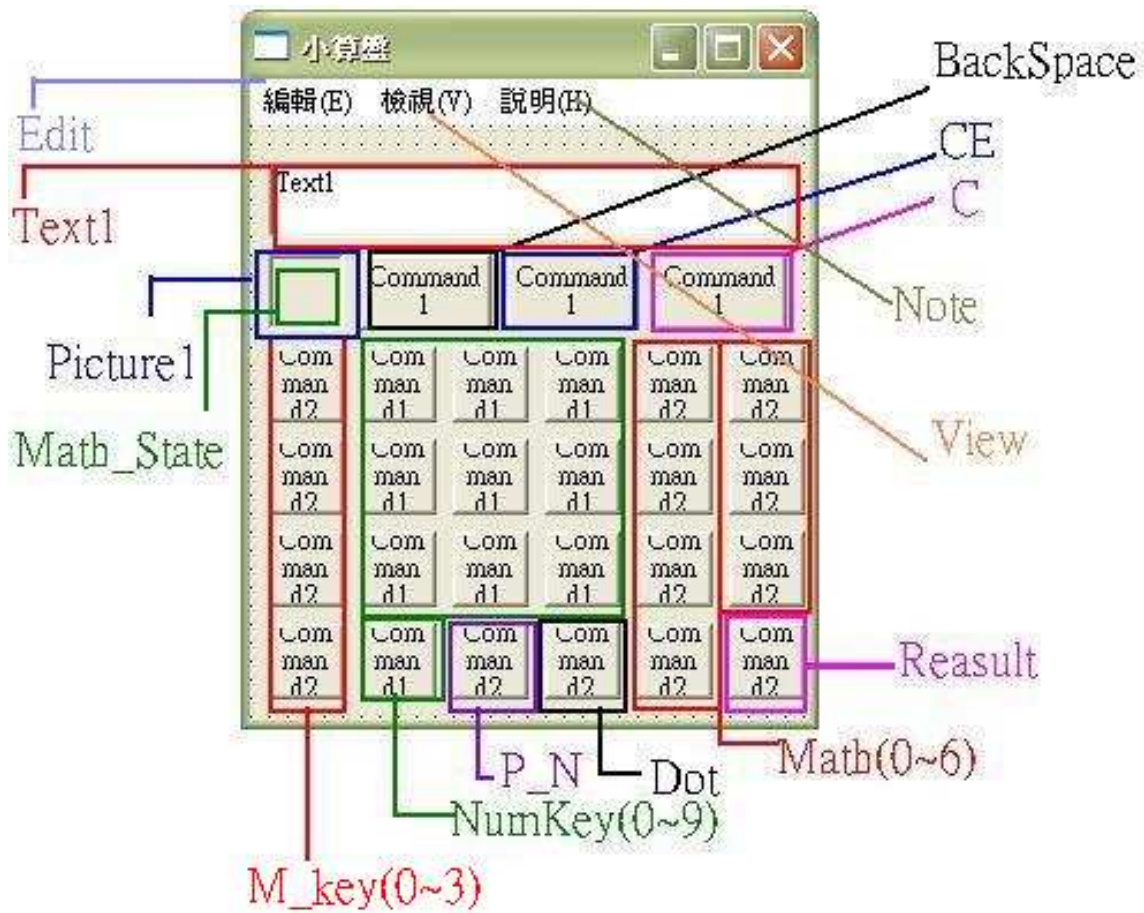


4. 模組已新增完成



6.9 建立自己的小算盤

1. 請照上述方法建立二個表單以及一個模組。
2. Form1 表單配置



3. 程式碼

由於 Fom1 以及 Fom2 大部份的程式碼皆相同，以藍色粗體的部分為 Fom2『特有』，其餘皆相同，為節省不必要的篇幅浪費，請多包涵。

-----以下為 Fom2 程式碼-----

Const PI As Double = 3.14159265358979 '圓周率

Dim Math_Index As Byte '做何種運算

Dim M As String '記憶

Dim xy As Boolean '用以辨視計算 X^Y

Dim P_N_State As Boolean '正負號辨別

Dim a, b As Double '宣告被運算的二個數

Private Sub About_Click() '設定說明

MsgBox "本小算盤乃為教學用，許多功能尚未周全，請自行於程式碼新增您所需之功能，謝謝。", , "KUAS CSIE VB 教材，第六章_模組"

End Sub

Private Sub BackSpace_Click() '消去最右邊的一位數

If Len(Text1.Text) = 0 Then '假如長度為零，則不執行此區塊程式碼

Exit Sub

End If

Text1.Text = Left(Text1.Text, Len(Text1.Text) - 1)

End Sub

Private Sub C_Click() '只清掉目前文字框中的數字

Text1.Text = ""

End Sub

Private Sub CE_Click() '清掉已準備做運算的所有數

a = 0

```
b = 0
```

```
Text1.Text = ""
```

```
Math_Index = 88 '清掉運算符號(只有 0~6 才做爲運算)
```

```
End Sub
```

```
Private Sub copy_Click() '將 TEXT1 中的資料複製到 windows 的剪貼簿
```

```
Clipboard.SetText (Text1.Text)
```

```
End Sub
```

```
Private Sub Dot_Click() '加上小數點
```

```
If Text1.Text = "" Then '如果還沒按任何數字則不執行本區塊中的程式碼
```

```
Exit Sub
```

```
End If
```

```
Text1.Text = Text1.Text + "."
```

```
Dot.Enabled = False '避免按二次
```

```
End Sub
```

```
Private Sub easy_type_Click() '變更爲簡易型
```

```
Form1.Show
```

```
Form2.Hide
```

```
End Sub
```

```
Private Sub engineer_type_Click() '變更爲工程型
```

```
Form1.Hide
```

```
Form2.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim i As Byte
```

```
設定 0~9 按鈕起始值
```

```
For i = 0 To 9
```

```
    NumKey(i).Caption = i
```

```
Next
```

```
設定正負號以及小數點
```

```
P_N.Caption = "+/-"
```

```
Dot.Caption = "."
```

```
設定四則運算、百分比、分數、平方根
```

```
Math(0).Caption = "+"
```

```
Math(1).Caption = "-"
```

```
Math(2).Caption = "*"
```

```
Math(3).Caption = "/"
```

```
Math(4).Caption = "Sqrt"
```

```
Math(5).Caption = "%"
```

```
Math(6).Caption = "1/x"
```

設定 =

```
Reasult.Caption = "="
```

設定記憶功能鍵

```
M_key(0).Caption = "MC"
```

```
M_key(1).Caption = "MR"
```

```
M_key(2).Caption = "MS"
```

```
M_key(3).Caption = "M+"
```

設定清除鍵

```
BackSpace.Caption = "←"
```

```
CE.Caption = "CE"
```

```
C.Caption = "C"
```

設定 TEXT1

```
Text1.Text = ""
```

```
Text1.Font.Size = 12
```

設定正負號(預設為正)

```
P_N_State = True
```

設定工程型功能鍵

```
other_math(0).Caption = "PI"
```

```
other_math(1).Caption = "n!"
```

```
other_math(2).Caption = "log"
```

```
other_math(3).Caption = "ln"
```

```
other_math(4).Caption = "X^2"
```

```
other_math(5).Caption = "X^3"
```

```
other_math(6).Caption = "X^Y"
```

```
other_math(7).Caption = "sin"
```

```
other_math(8).Caption = "cos"
```

```
other_math(9).Caption = "tan"
```

```
xy = False '關閉 x^ y 計算
```

```
Form2.Caption="小算盤_工程型"
```

```
End Sub
```

```
Private Sub M_key_Click(Index As Integer) '設定記憶鍵動作
```

```
Select Case Index
```

```
Case 0 'MC: 清除記憶
```

```
M = ""
```

```
M_State.Caption = "" '清除記憶顯示
```

```
Case 1
```

```
Text1.Text = M 'MR :讀取記憶
```

```
Case 2
```

```
M = Text1.Text 'MS: 設定記憶
```

M_State.Caption = "M" '設定記憶顯示

Case 3

M = Val(M) + Val(Text1.Text) 'M+ : 將記憶中的值與 TEXT1 的值相加並存入記憶

End Select

End Sub

Private Sub Math_Click(Index As Integer)

a = Text1.Text

Dot.Enabled = True

Select Case Index '需要二個數的四則運算

Case 0 To 3

Math_Index = Index

Text1.Text = ""

Case 4 '只需要一個數的運算

Text1.Text = Str(a ^ 0.5)

Case 5

Text1.Text = Str(a ^ 2 / 100)

Case 6

Text1.Text = Str(1 / a)

End Select

End Sub

Private Sub NumKey_Click(Index As Integer)

'按下的數字會接在既有的數字右邊

Text1.Text = Text1.Text & Index

End Sub

Private Sub other_math_Click(Index As Integer)

Dim i As Byte '迴圈用變數

Dim n As Long '用以計算階層

Dot.Enabled = True

a = Val(Text1.Text)

Select Case Index

Case 0 '輸出圓周率

Text1.Text = PI

Case 1 '計算階層

Form2.Caption = a

n = 1

For i = 1 To a

n = n * i

Next

Text1.Text = n

Case 2 計算 Log

Text1.Text = Log(a) / Log(10)

Case 3 計算 ln

Text1.Text = Log(a)

Case 4 計算平方

Text1.Text = a ^ 2

Case 5 計算立方

Text1.Text = a ^ 3

Case 6 計算 X^Y

Text1.Text = ""

xy = True

Case 7 計算 Sin

Text1.Text = Sin(a * PI / 180)

Case 8 計算 Cos

Text1.Text = Cos(a * PI / 180)

Case 9 計算 Tan

Text1.Text = Tan(a * PI / 180)

End Select

End Sub

Private Sub P_N_Click() '將正負號加在數字前

P_N_State = Not P_N_State

If P_N_State = False Then

 Text1.Text = "-" + Text1.Text

Else

 If Left(Text1.Text, 1) = "-" Then

 Text1.Text = Right(Text1.Text, Len(Text1.Text) - 1)

 End If

End If

End Sub

Private Sub past_Click() '從 windows 的剪貼簿中將資料貼上

Text1.Text = Clipboard.GetText

End Sub

Private Sub Reasult_Click() '輸出四則運算結果

b = Text1.Text

Dot.Enabled = True

Select Case Math_Index

 Case 0

```
Text1.Text = Str(Add(a, b))
```

Case 1

```
Text1.Text = Str(Subtract(a, b))
```

Case 2

```
Text1.Text = Str(Mul(a, b))
```

Case 3

```
Text1.Text = Str(Divisional(a, b))
```

End Select

If xy = True Then 'x^y 運算為特殊情況，需二個運算元

```
Text1.Text = a ^ b
```

```
xy = False
```

End If

End Sub

-----以下為模組 Module 中的程式碼-----

```
Public Function Add(ByVal a As Double, ByVal b As Double) '加法
```

```
Add = a + b
```

End Function

```
Public Function Subtract(ByVal a As Double, ByVal b As Double) '減法
```

Subtract = a - b

End Function

Public Function Mul(ByVal a As Double, ByVal b As Double) '乘法

Mul = a * b

End Function

Public Function Divisional(ByVal a As Double, ByVal b As Double) '除法

Divisional = a / b

End Function

4. 執行結果

(1).簡易型



(2)工程型



小幫手特區：Sin、Cos、Tan 函數的計算怎麼有些看不懂?還有 Log 函數的地方也有些模糊?

1. 在 VB 中使用三角函數運算必須使用『徑度』為計算的基準，徑度沒有單位，我們一般慣用的角度稱為『度度』，度度有單位（單位為度）。

徑度的定義：單位圓上角度所對的弧長為該角度的徑度。

因為單位圓周長 2π ，圓心角 360 度，所以 $2\pi = 360$ 度，所以 $\pi = 180$ 度

$$1 \text{ 徑度} = \frac{180}{\pi} \text{ 度度}, \text{ 同理可證, } 1 \text{ 度度} = \frac{\pi}{180} \text{ 徑度}。$$

$$\text{假如我們要將 } 30 \text{ 度換成徑度} \rightarrow 30 \times \frac{\pi}{180}。$$

2. 在 VB 中，Log 函數預設是以 E(自然對數為底)，等同於 ln 函數，例如我們要求一個以 10 為底的 Log 函數，則輸入 $\text{Log}(x) / \text{Log}(10)$ 即可。