

Visual Basic 程式設計

翁傳奇

P9521773@fcu.edu.tw

逢甲大學資訊工程研究所

第五章 副程式與函式 (1)

◆ 副程式

◆ 函式

◆ 遞迴

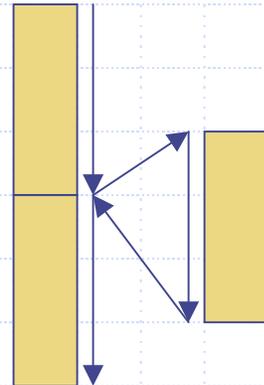
為什麼要有副程式?

◆ 程式中有許多重覆的動作

- 如下頁的例子

◆ 思考方向：

- 將某獨立、常重覆的部份發包出去
- 營養午餐與外叫便當的例子



為什麼要有副程式? (cont.)

```
*  
**  
***  
****  
****  
***  
**  
*
```

```
For i=1 to 8  
  If (i<=4) Then  
    For j=1 to i  
      Print "*";  
    Next j  
  Else  
    For j=1 to 9-i  
      Print "*";  
    Next j  
  End if  
  Print  
Next i
```

```
For i=1 to 8  
  If (i<=4) Then  
    >>畫i個*<<  
  Else  
    >>畫9-i個*<<  
  End if  
  Print  
Next i
```

化簡

副程式-Subroutine

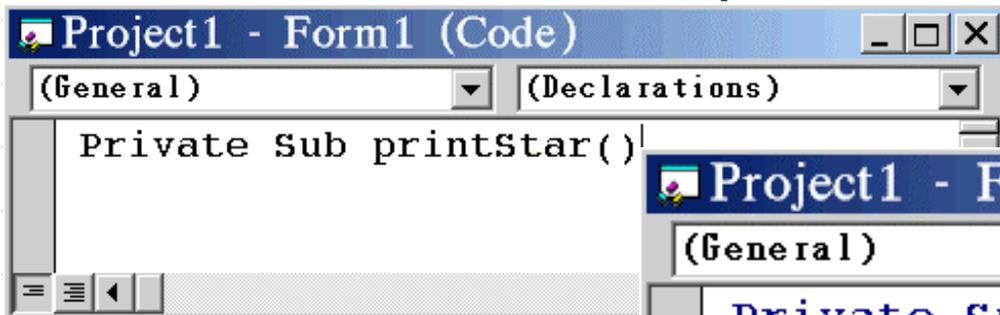
[Private | Public | Static] Sub 副程式名 ([參數])

程式碼

End Sub

◆ 在程式編輯器輸入

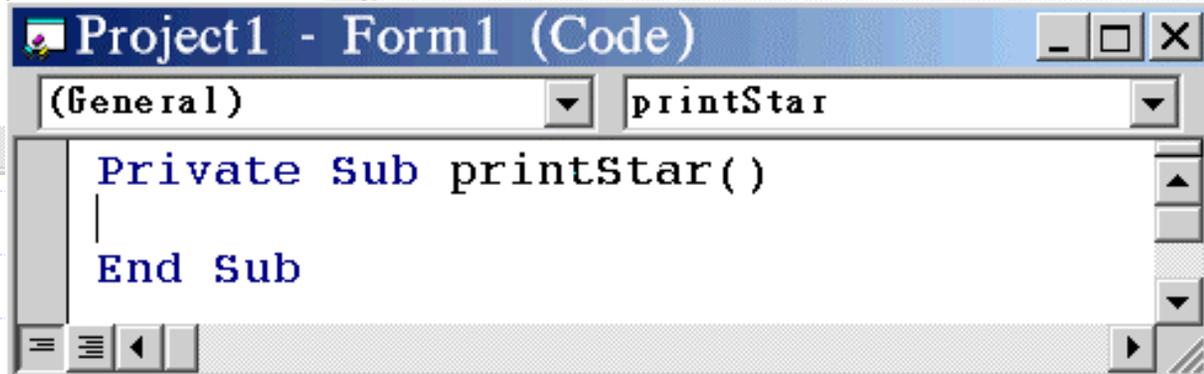
■ Private Sub printStar()



Project1 - Form1 (Code)

(General) (Declarations)

```
Private Sub printStar()
```



Project1 - Form1 (Code)

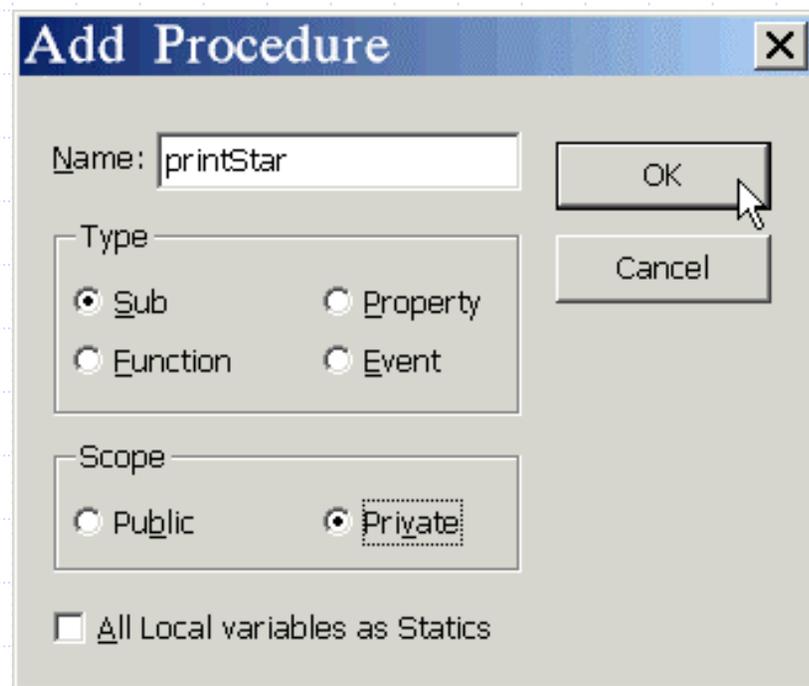
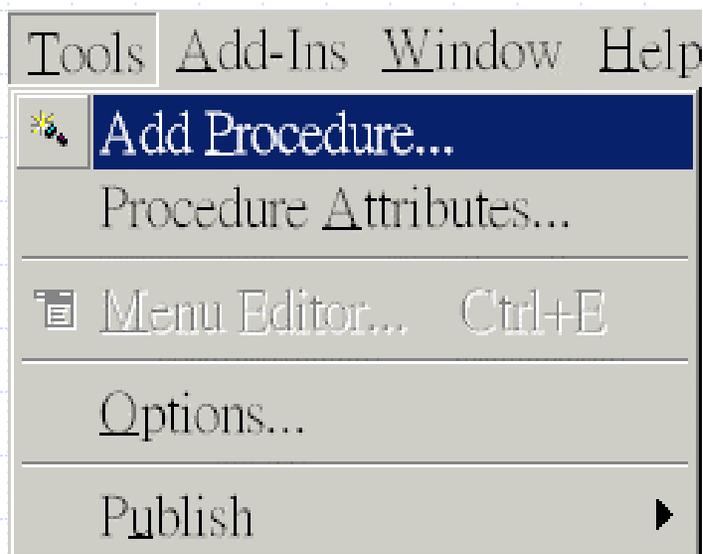
(General) printStar

```
Private Sub printStar()  
|  
End Sub
```

副程式-Subroutine (cont.)

◆ 使用工具列

- 選擇程式編輯器，按 Tool/Add Procedure



第一個副程式

VB會自動宣告num

```
Private Sub printStar (num As Integer)
    Dim i As Integer
    For i=1 to num
        Print "*";
    Next i
    Print
End Sub
```

副程式呼叫方式:

1. 副程式名 參數
2. Call 副程式名(參數)

```
For i=1 to 8
    If (i<=4) Then
        1.[printStar i]
        2.[Call printStar(i)]
    Else
        1.[printStar 9-i]
        2.[Call printStar(9-i)]
    End if
    Print
Next i
```

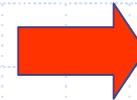
副程式之輸入參數與輸出參數

```
Call sumxy( s,   a, b)
           ↑   ↓ ↓
Sub sumxy(sum, x, y)
```

對副程式sumxy而言，
x與y是它的“輸入參數”，
而sum則是它的“輸出參數”

```
Private Sub Form_Activate()  
    a = 10 : b = 20  
    Call sumxy(s, a, b)  
    print "兩數的和為"; s  
End Sub
```

```
Sub sumxy(sum, x, y)  
    sum = x + y  
End Sub
```



函式 - Function

- ◆ 函數的定義及用法與副程式很相似，不過它是利用函數名稱來傳遞結果，而不像副程式是靠參數來傳遞。

```
Function 函數名稱(參數列) as 型別
...
函數名稱 = 運算式
...
End Function
```

函式 - Function (cont.)

◆ 函式: 有傳回值的副程式

```
[Private|Public|Static] Function 函式名 _  
([參數]) [As] 型別  
    程式碼  
End Function
```

VB會自動宣告一變數，
其名稱為函式名

```
Private Function Add _  
    (x As Integer, y As Integer) As Integer  
    Add=x+y  
End Function
```

函式 - Function (cont.)

```
Private Function Add  
(x As Integer, y As Integer) As Integer  
    Add=x+y  
End Function
```

結果是?

```
Private Sub Form_Click()  
    print Add(1,2)  
End Sub
```

把函式當副程式

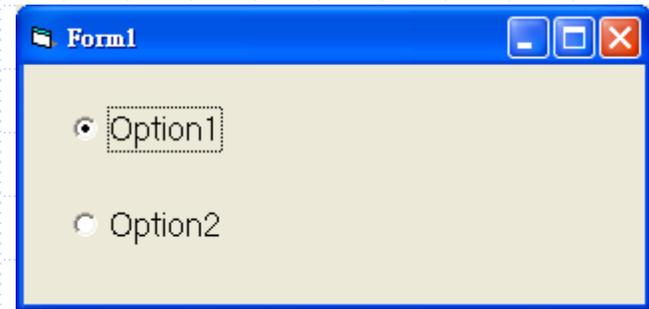
1. Add 1,2
2. Call Add(1,2)

常用元件介紹

- 選項按鈕 (Option Button)

◆ 每組元件只能有一個被勾選

屬性	
Caption	出現的文字
Enabled	是否有致能
Value	True False



◆ 若想要讓程式執行時，能夠先選擇Option2，則可以在Form_Load輸入Option2.value = true

常用元件介紹

- 選項按鈕 (Option Button) (cont.)

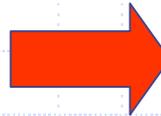
```
Private Sub Command1_Click()  
    If Option1.Value = True Then  
        Label1.Caption = "你按了Option1按鈕"  
    ElseIf Option2.Value = True Then  
        Label1.Caption = "你按了Option2按鈕"  
    End If  
End Sub  
End Sub
```

Form1

Option1

Option2

Check Option



Form1

Option1

Option2

Check Option

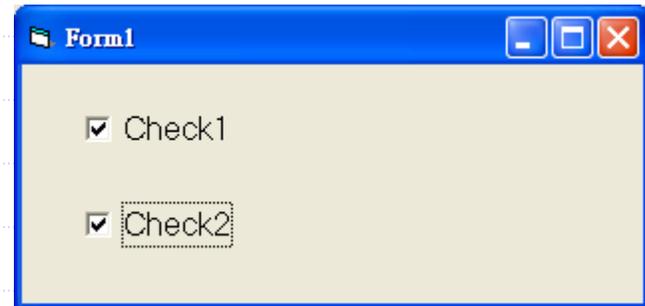
你按了Option1按鈕

常用元件介紹

- 核取方塊 (Check Box)

◆ 每個元件獨立

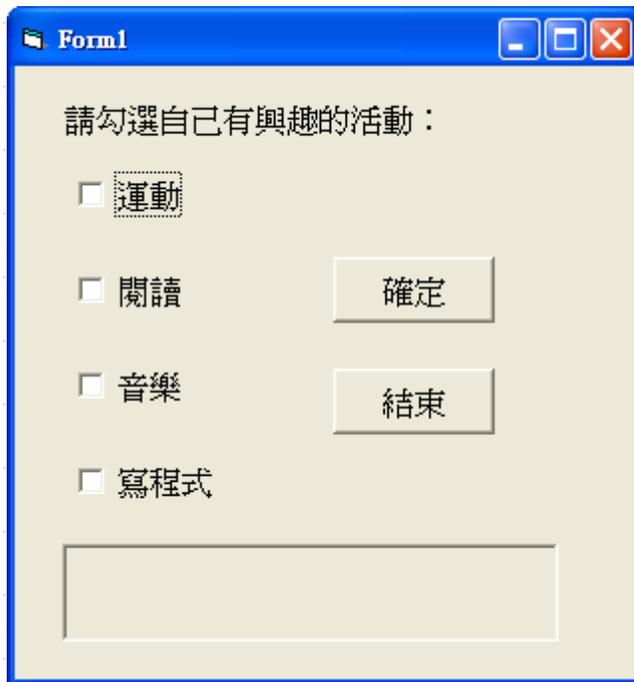
屬性	
Caption	出現的文字
Enabled	是否有致能
Value	0 未勾選 1 勾選 2



常用元件介紹

- 核取方塊 (Check Box) (cont.)

◆ Try it!



Form1

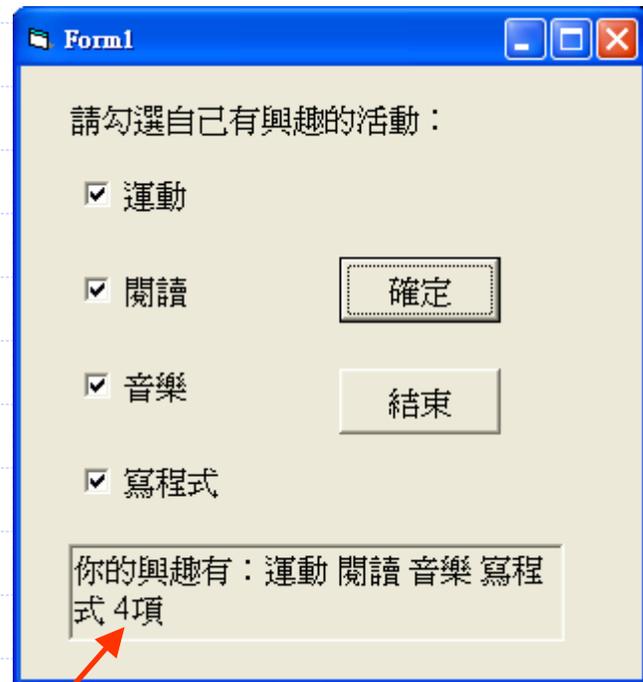
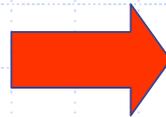
請勾選自己有興趣的活動：

- 運動
- 閱讀
- 音樂
- 寫程式

確定

結束

你的興趣有： 0 項



Form1

請勾選自己有興趣的活動：

- 運動
- 閱讀
- 音樂
- 寫程式

確定

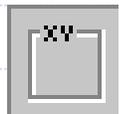
結束

你的興趣有：運動 閱讀 音樂 寫程式 4項

統計個數

常用元件介紹

- 框架 (Frame)



- ◆ 利用“框架”，可以將表單中屬於同一類的控制項圍起來，有下列的好處：
 - 讓表單的排列比較整齊，也顯得有條理。
 - 在框架內的控制項屬於同一群，“搬移”框架時，其內的各控制項也會隨著搬移，如果需要調整表單上的配置，也會比較省事。
 - 如果沒有設定框架，表單上所有的選項按鈕算是同一群，只能選取一項。如果有不同類的選項按鈕，個別要選取一項，此時可分別用框架圍起來（每個框架內的選項按鈕，可以也只能選取一項）。

常用元件介紹

- 框架 (Frame) (cont.)

◆ Try it!

Form1

請輸入姓名：

性別

男 女

學歷

博士 碩士 大專

高中職 國中小

確定 結束

男

女

Form1

請輸入姓名： xxx

性別

男 女

學歷

博士 碩士 大專

高中職 國中小

Hi!xxx先生，您的學歷：大專。

確定 結束

Form1

請輸入姓名： ooo

性別

男 女

學歷

博士 碩士 大專

高中職 國中小

Hi!ooo小姐，您的學歷：碩士。

確定 結束

母件與子件的觀念

- ◆ 如前面介紹的框架，可以在內部安置一些有關或同一類的控制項，“搬移”框架時，其內的控制項，也會隨著移動。“刪除”時也是其內的所有控制項皆會隨著消失。
- ◆ 又如表單與其上面的所有控制項，都有同樣的“主從關係”，這種關係有如母子，因此分別稱為“母件”與“子件”。
 - 母件：可背負其他物件的物件
 - 子件：被背負的物件

課堂練習 — 計算電費

◆ 假設電力公司的電費分“家庭用”與“營業用”兩類，其計費方式如下：

■ 家庭用：按累進方式，分為下列三段：

- ◆ 100度以下部份，每度2.4元
- ◆ 101~300度部份，每度3.1元
- ◆ 301度以上部份：每度4.1元

■ 營業用：按累進方式，分為下列兩段：

- ◆ 300度以下部份，每度5.9元
- ◆ 301度以上：每度6.7元

■ 註：累進收費方式以家用而言，即如果超過100度，100度部份每度為2.4元，超過部份才以較高費率計算。

◆ 請設計一個表單程式輸入用電度數與種類，計算並輸出應繳電費。

課堂練習 – 計算電費 (cont.)

◆ 表單設計

The screenshot shows a Windows form titled "計算電費" (Calculate Electricity Bill). The form contains the following elements:

- Label1**: Points to the text "用電度數:" (Electricity usage).
- text1**: Points to the text input field for electricity usage.
- Frame1**: Points to the "用電種類" (Electricity type) section.
- Option1**: Points to the radio buttons for "家庭用" (Residential) and "營業用" (Commercial).
- Label2**: Points to the text "電費:" (Electricity fee).
- Label4**: Points to the text "元" (Yuan).
- Command1**: Points to the "計算" (Calculate) button.
- Command2**: Points to the "結束" (End) button.

Label3

BorderStyle : 1 – Fixed Single

課堂練習 – 計算電費 (cont.)

- ◆ 撰寫兩個副程式分別計算營業用電費以及家庭用電費。
- ◆ Form_Load：下載表單後，選家庭用電
- ◆ Command1_Click {
 - home (副程式)：計算家庭用電費
 - business (副程式)：計算營業用電費
- ◆ Command2_Click：結束程式執行

遞迴 - Recursion

◆ 在函式，呼叫自己

```
Private Function A()  
    A()  
End Function
```

◆ 要確定程式會停止

```
Private Function Sum ( x As Integer) As Integer  
    If (x=1) then  
        Sum=1  
    Else  
        Sum=Sum(x-1)+x  
    End If  
End Function
```

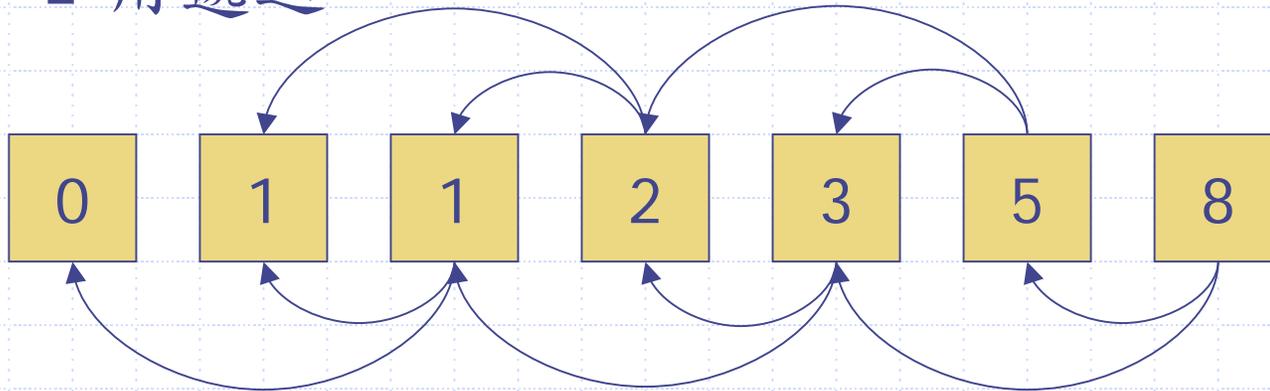
初始值的設定很重要!!

費式數列 - Fibonacci Series

◆ 0, 1, 1, 2, 3, 5, 8,

◆ 請問第100項為多少？

- 用迴圈
- 用遞迴



費式數列-使用迴圈

```
Private Function Fab(x As Integer) As Integer
    Dim a,b,i As Integer
    a=0
    b=1
    If (x=0 Or x=1) then
        Fab=x
    Else
        For i=2 to x
            Fab=a+b
            a=b
            b=Fab
        Next i
    End If
End Function
```

費式數列-使用遞迴

```
Private Function Fab(x As Integer) As Integer
    If (x=0 Or x=1) then
        Fab=x
    Else
        Fab=Fab(x-2)+Fab(x-1)
    End If
End Function
```

階乘

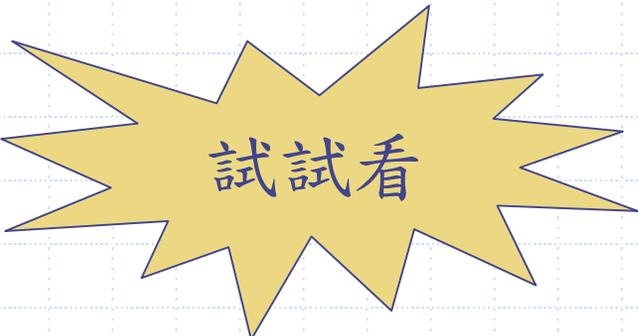
◆ 離開副程式與函式

- Exit Sub
- Exit Function

◆ $5! = 1 * 2 * 3 * 4 * 5$

◆ 求某數的階乘

- 使用迴圈
- 使用遞迴



試試看