

VBA

Il Visual Basic for Application

- ⇐Le funzioni
- ⇐Le procedure

Funzioni

- μ E' possibile (e consigliato) scomporre un problema in sotto-problemi e combinare poi assieme le soluzioni per ottenere la soluzione complessiva.
- μ Una funzione è un modo per indicare la soluzione di un sotto-problema una volta che è possibile individuare un insieme di dati in ingresso e un risultato in uscita

Funzioni

- μ si dice che la funzione *restituisce* un risultato
- μ il risultato restituito da una funzione deve essere un *unico* dato di tipo *primitivo* (o un oggetto)
- μ ovvero non si può restituire 2 o più valori, il valore restituito deve essere unico

Le funzioni e le procedure

- μ La differenza fra una funzione e una procedura o subroutine è che al termine dell'esecuzione una funzione restituisce un valore mentre la subroutine no
- μ tuttavia vedremo come le subroutine possono agire modificando il valore di un numero arbitrario di variabili

Dichiarare le funzioni

μ Per dichiarare una funzione si scrive:

```
Function Nome(param1 As tipo, param2 As tipo) _  
    As tipo_restituito  
    'corpo della funzione  
End Function
```

Dichiarare le funzioni

μ Si scrive una intestazione:

- ← si deve premettere la parola chiave Function
- ← assegnare un nome alla funzione
- ← indicare fra parentesi separati da virgole i parametri di ingresso specificando il nome e il tipo
- ← in ultimo si indica il tipo del risultato

μ si racchiude il corpo della funzione fra la intestazione e la parola chiave End Function

Esempio

μ Questa funzione somma 1 al valore passato

```
Function SommaUno(valore As Integer) As Integer
    SommaUno=valore+1
End Function
```

Dichiarare le funzioni

μ La restituzione di un valore avviene assegnando tale valore alla variabile che ha come nome il nome della funzione

Usare le funzioni

μ Dopo aver dichiarato una funzione questa può essere utilizzata all'interno di un altro programma o di una altra funzione (o subroutine)

μ Es:

```
Function SommaUnoXY(x As Integer, _  
    y As Integer) As integer  
    SommaUnoXY=SommaUno(x)+SommaUno(y)  
End Function
```

Dichiarare le subroutine

μ Una subroutine è come una funzione ma non ha parametro di ritorno

μ si dichiara come

```
Sub Nome(param1 As tipo, param2 As tipo)  
    `corpo subroutine  
End Sub
```

Usare le subroutine

- μ Dato che le subroutine non restituiscono un valore, non è possibile utilizzarle all'interno di una espressione
- μ si usa una subroutine premettendo la parola chiave Call come in:
`Call Nome (param1, param2)`
- μ oppure direttamente:
`Nome param1, param2`

Esempio

```
Option Explicit

Sub programma()
Dim nome As String
nome="Antonio"
stampa nome
End Sub

Sub stampa(var As String)
'...codice per stampare
End Sub
```

Parametri per valore o per riferimento

- μ Ci sono due modi per passare una variabile in ingresso ad una funzione o procedura: per valore (o copia) oppure per riferimento (o indirizzo)
- μ La differenza fra le due modalità sta nel rendere disponibile alla funzione/procedura una copia del valore della variabile o il contenuto vero e proprio della variabile

Parametri per valore o per riferimento

- μ Sintatticamente si indica la modalità per valore o per riferimento premettendo ByVal o ByRef rispettivamente prima della dichiarazione dei parametri di ingresso

```
Sub nome (ByVal id As Type)  
Sub nome (ByRef id As Type)
```

Esempio

μ Si consideri il seguente programma:

```
Sub ProcA()  
  Dim x As Integer  
  x=5  
  MsgBox x  
  Call AddOneA(x)  
  MsgBox x  
End Sub  
  
Sub AddOneA(ByVal i As Integer)  
  i=i+1  
End Sub
```

Spiegazione

- μ In questo esempio viene assegnato il valore 5 alla variabile x
- μ poi la variabile viene passata ad una funzione (che vedremo di seguito) che ne stampa il valore sul video
- μ poi viene chiamata la procedura che prende in ingresso una copia del valore di x e lo incrementa
- μ questo però non ha effetto sul contenuto della variabile x
- μ infatti quando successivamente viene stampato il valore di x questo è sempre 5

Esempio

μ Si consideri il seguente programma:

```
Sub ProcB()  
Dim x As Integer  
x=5  
MsgBox x  
Call AddOneB(x)  
MsgBox x  
End Sub  
  
Sub AddOneB(ByRef i As Integer)  
i=i+1  
End Sub
```

Spiegazione

- μ Questa volta la variabile è stata passata per riferimento
- μ Questo significa che la procedura va a modificare il contenuto della variabile x
- μ Infatti alla fine dell'elaborazione il valore di x è stato incrementato e vale 6

Nota

Attenzione!!!

- μ Di default il VBA considera le variabili passate per riferimento e quindi ciò che si modifica in una procedura ha effetto sulla variabile che viene passata

Pregi/difetti delle due modalità

- μ Ognuna delle due modalità risulta vantaggiosa se applicata con una finalità opportuna
- μ Il passaggio per valore permette di essere sicuri che la procedura non modificherà le variabili contenute nel programma che sta usando tali procedure
- μ Il passaggio per riferimento è più veloce perché non deve eseguire una copia
- μ Inoltre il passaggio per riferimento permette di agire su un numero arbitrario di variabili

Funzioni vs. Procedure

- μ Mentre una funzione può restituire un unico valore al termine della sua esecuzione, una procedura può prendere in ingresso per riferimento più variabili e modificarne tutti i valori
- μ tuttavia l'uso delle funzioni permette di utilizzare il nome della funzione stessa all'interno di espressioni più complesse

Esempio

```
Sub Main()  
Dim x As Integer, y As Integer, z As Integer  
x=10  
y=20  
z=quadrato(x)+doppio(y)  
End Sub  
  
Function quadrato(i As Integer)  
quadrato=i*i  
End Function  
  
Function doppio(i As Integer)  
doppio=i*2  
End Function
```

Esempio

```
Sub Main()  
Dim x As Integer, y As Integer, z As Integer  
x=10  
y=20  
Call calcolo(x,y,z) `oppure calcolo x,y,z  
End Sub  
  
Sub calcolo(ByRef i As Integer, ByRef j As _  
    Integer, ByRef k As Integer)  
k=i*i+j*2  
End Sub
```

Esempio

```
Sub Main()  
Dim x As Integer, y As Integer, z As Integer  
x=10  
Call calcolo(x,y,z) `oppure calcolo x,y,z  
End Sub  
  
Sub calcolo(ByRef i As Integer, ByRef j As _  
    Integer, ByRef k As Integer)  
k=i*i  
j=i*2  
End Sub
```

Funzioni predefinite in Word

- μ Sono a disposizione un elevato numero di funzioni predefinite
- μ una funzione è predefinita quando non è necessario scriverne il programma per l'utente
- μ Noi tratteremo solo alcune utili funzioni:
 - ←funzioni per la rappresentazione di stringhe
 - ←funzioni per l'inserimento/acquisizione di informazioni
 - ←funzioni per la manipolazione di stringhe
 - ←funzioni di conversione

Funzioni per la rappresentazione MsgBox

- μ La funzione MsgBox serve per creare una finestra sulla quale scrivere una stringa

- μ la sintassi è:

```
MsgBox stringa
```

- μ Esempio:

```
Dim frase As String  
frase="questa e' una prova"  
MsgBox frase
```

Nota

μ Per scrivere un messaggio su più linee è sufficiente inserire opportuni vbCrLf

μ per scrivere il valore di una variabile è sufficiente passare tale valore alla funzione (il numero viene automaticamente convertito in stringa)

μ Es:

```
Dim myInt As Integer  
myInt=34  
MsgBox myInt
```

Funzioni per l'acquisizione InputBox

μ Per acquisire una stringa si usa la funzione InputBox con la seguente sintassi:

```
Dim leggo As String  
leggo=InputBox("messaggio per l'utente")
```

Funzioni per stringhe

μ Ecco una lista di utili funzioni per manipolare le stringhe:

μ Len, Left, Right, Mid, Str, Val, Trim

Funzioni per stringhe

μ Len: restituisce la lunghezza (cioè il numero di caratteri) di una stringa, compresi spazi, segni di interpunzione etc.

μ Es:

← Len("Oggi piove") restituisce 10

Funzioni per stringhe

- μ Left, Right, Mid: restituiscono la porzione di una stringa rispettivamente più a sinistra, destra, nel centro
- μ Left(parola, num): restituisce un numero pari a num caratteri a partire da sinistra della stringa parola
- μ Es:
 ← Left("Oggi piove", 6) restituisce "Oggi p"
- μ Analogamente per la funzione Right

Funzioni per stringhe

- μ La funzione Mid prende in ingresso una stringa e due numeri
- μ Mid(parola, inizio, lunghezza)
- μ Es:
 ← Mid("Oggi piove", 3,4) restituisce "gi p"

Funzioni per stringhe

μ La funzione Str converte un numero in stringa

μ Es:

← Str(123) restituisce la stringa "123"

μ La funzione Val converte una stringa in un numero

μ Es:

← Val("34.5") restituisce il numero 34.5

μ ma anche:

← Val("50123 Firenze") restituisce il num 50123

Funzioni per stringa

μ La funzione Trim serve per eliminare spazi all'inizio e al termine di una parola

μ Es:

← Trim(" casa ") restituisce la stringa "casa"