Passaggio di argomenti alle procedure

Quando un valore esterno deve essere utilizzato da una procedura per eseguire un'azione, si passa alla procedura da variabile. Queste variabili che sono passate a una procedura sono chiamate argomenti che rappresenta il valore fornito dal codice chiamante a una procedura. Nella sintassi di dichiarazione della procedura compaiono degli elementi racchiusi tra parentesi quadre: sono i parametri o argomenti e quando il set di parentesi, dopo il nome della procedura nella dichiarazione Sub o la dichiarazione Function, sono vuoti, si tratta di un caso in cui la procedura non riceve argomenti. Tuttavia, quando gli argomenti sono passati a una procedura da altre procedure, allora questi sono elencati o dichiarati tra le parentesi.

I parametri hanno un ordine posizionale, vanno elencati separati da una virgola e devono rispettare il tipo di dato specificato nella dichiarazione, inoltre possono essere "passati" come valori fissi o tramite variabili. In sostanza occorre dare un nome ad ogni parametro e specificare il tipo di dato che conterrà (come per le variabili), se ne indichiamo più di uno occorre elencarli di seguito separati da una virgola.

Tipi di argomenti di dati

Nell'esempio che segue la dichiarazione della funzione '*voto*' contiene una variabile di tipo Integer passata come argomento e restituito un valore di tipo String. E' bene ricordare che solitamente si dichiara il tipo di dati per gli argomenti passati a una procedura, nel caso venisse omessa questa dichiarazione, il tipo di dati predefinito che VBA assegna è di tipo Variant.

```
Function voto(votazione As Integer) As String
If votazione >= 80 Then
voto = "A"
ElseIf votazione >= 60 Then
voto = "B"
ElseIf votazione >= 40 Then
voto = "C"
Else
voto = "Pessimo"
End If
End Function

Sub callvoto()
Dim i As Integer
Dim str As String
i = InputBox("Inserisci il voto per Marco")
 str = voto(i)
MsgBox "La valutazione di Marco è : " & str
End Sub
```

Nella routine *callvoto* è stata chiamata la funzione *voto* e il risultato ottenuto assegnato alla variabile str variabile. Nell'esempio sottostante la dichiarazione della funzione votiPercent contiene due variabili come argomenti, con un valore Double come tipo di dati di ritorno.

```
Function votiPercent(voti As Integer, Totalvoti As Integer) As Double
votiPercent = voti / Totalvoti * 100
votiPercent = Format(votiPercent, "#.##")
End Function

Sub callvoti_1()
Dim voti As Integer
Dim Totalvoti As Integer
Dim pcnt As Double
voti = InputBox("Inserisci Voti")
Totalvoti = InputBox("Inserisci Totale Voti")
pcnt = votiPercent(voti, Totalvoti)
MsgBox "La percentuale è del :" & " " & pcnt & "%"
End Sub
```

L'ordine posizionale è anche quello che ci servirà per riferirci ad essi quando utilizziamo la procedura. E' possibile passare dei parametri alle procedure ed alle funzioni in due modi:

- *Passaggio per riferimento (ByRef)*. Si utilizza quando all'interno della procedura (o funzione) il valore del parametro viene modificato e, alla procedura chiamante, serve il valore di ritorno modificato
- *Passaggio per valore (ByVal)*. Il valore del parametro, ritorna alla procedura chiamante con il suo valore originale, anche se viene modificato all'interno della procedura (o funzione).

Passaggio di argomenti per valore

Quando si passa un argomento per valore in una procedura, solo una copia di una variabile viene passata e qualsiasi modifica del valore della variabile nella procedura corrente non influenza o modifica la variabile stessa nella sua posizione originale perché la variabile stessa non è accessibile. Per passare un argomento per valore, si deve utilizzare la parola chiave **ByVal** . Vediamolo con un esempio

```
Sub Prova_1()
Dim x As Integer
x = 5
MsgBox x
Call Aggiungi_1 (x)
MsgBox x
End Sub

Sub Aggiungi_1(ByVal i As Integer)
i = i + 1
End Sub
```

In questo codice viene assegnato il valore '5' alla variabile x che tramite la funzione MsgBox ne riporta il valore sullo schermo, successivamente viene chiamata la procedura '*Aggiungi_1*' che prende come valore in entrata una copia del valore della variabile x e lo incrementa di 1 unità.

Questo passaggio non ha effetto sul contenuto della variabile x, infatti quando viene stampato a video il valore di ritorno il valore di x è sempre 5. Modifichiamo le istruzioni in questo modo

```
Sub Prova_2()
Dim x As Integer
x = 5
MsgBox x
Call Aggiungi_2 (x)
MsgBox x
End Sub

Sub Aggiungi_2(ByRef i As Integer)
i = i + 1
End Sub
```

Questa volta la variabile è stata passata per riferimento Questo significa che la procedura va a modificare il contenuto della variabile x Infatti alla fine dell'elaborazione il valore di x è stato incrementato e vale 6. Bisogna fare attenzione che 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. Ognuna delle due modalità risulta vantaggiosa se applicata con una finalità opportuna, Il passaggio per valore permette di essere sicuri che la procedura non modificherà le variabili contenute nel programma che sta usando tali procedure, mentre Il passaggio per riferimento è più veloce perché non deve eseguire una copia. Inoltre il passaggio per riferimento permette di agire su un numero arbitrario di variabili

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, tuttavia l'uso delle funzioni permette di utilizzare il nome della funzione stessa all'interno di espressioni più complesse

```
Function calcola_comm(ByVal comm_tass As Double, ByVal vendite As Currency) As
Currency
calcola_comm = comm_tass * vendite
End Function

Sub vendite_marco()
Dim comm_marco As Currency
Dim comm_tass As Double
Dim vendite As Currency
comm_tass = InputBox("Inserisci tasso commissione")
vendite = InputBox("Inserisci Importo Vendite")
comm_marco = calcola_comm(comm_tass, vendite)
MsgBox "La commissione è di" & " " & comm_marco
End Sub
```

Passaggio di argomenti per riferimento

Quando si passa un argomento per riferimento in una procedura, la variabile stessa accede alla procedura e il valore della variabile viene modificato in modo permanente dalla procedura stessa. Per passare un argomento per riferimento, si deve utilizzare la parola chiave **ByRef** prima l'argomento da passare, inoltre il passaggio di argomenti per riferimento è anche l'impostazione predefinita in VBA, a meno che non sia esplicitamente specificato di passare un argomento per valore.

```
Function numero(ByVal i As Integer) As Long
i = 5
numero = i
End Function

Sub prova_numero()
Dim n As Integer
MsgBox numero(n)
MsgBox n
End Sub
```

In questo esempio la variabile numero restituisce il valore di 5 in quanto è la funzione numero che assegna un valore alla variabile n, ma poiché la variabile è stata passata per valore nella funzione, qualsiasi modifica dello stesso rimane nella funzione corrente e quando la funzione termina, il valore della variabile n ritornerà al valore di quando è stata dichiarata, che era pari a 0. Infatti con il secondo messaggio viene riportato il valore 0. Se la variabile fosse stata passata per riferimento nella funzione, la variabile n avrebbe definitivamente assunto il nuovo valore assegnato.

In questo esempio invece si vede il passaggio di un argomento per riferimento in una procedura utilizzando la parola chiave **ByRef**

```
Function numero_1(ByRef i As Integer) As Long
i = 5
numero_1 = i
End Function

Sub prova_numero_1()
Dim n As Integer
MsgBox numero_1(n)
MsgBox n
End Sub
```

Il valore della variabile n è impostato a 0 quando viene dichiarata e il messaggio restituito è 5 perché il richiamo della funzione numero_1, assegna un valore alla variabile n (5) e il messaggio successivo restituisce ancora 5, perché la variabile è stata passata per riferimento nella funzione numero_1, e ha definitivamente assunto il nuovo valore assegnato dalla funzione numero_1

Argomenti facoltativi

Gli argomenti possono essere specificati come facoltativi, utilizzando la parola chiave *Optional* prima dell'argomento e quando si specifica un argomento con optional, tutti gli argomenti seguenti devono essere specificati come Optional. Si noti che specificando la parola chiave Optional rende un argomento opzionale altrimenti sarà richiesto l'argomento.

L'argomento opzionale dovrebbe essere, anche se non necessario, dichiarato come tipo di dati *Variant* per consentire l'uso della funzione **IsMissing** che funziona solo quando viene utilizzato con le variabili dichiarate come Variant. La funzione IsMissing viene utilizzata per determinare se l'argomento opzionale è stata accettata nella procedura o meno, in modo da regolarsi di conseguenza nel codice senza restituire un errore. Se l'argomento opzionale non è dichiarato come Variant, la funzione IsMissing non funziona, e all'argomento opzionale verrà assegnato e il valore predefinito per il tipo di dati che è 0 per le variabili di tipo numerico (cioè Integer, Double, etc.) e Nothing per String o variabili di tipo Object.

Esempio: Dichiarazione della routine con due argomenti Optional

```
Sub dip_nome1(Optional primo As String, Optional secondo As String)
ActiveSheet.Range("A1") = primo
ActiveSheet.Range("B1") = secondo
End Sub

Sub nome_pieno1()
Dim nome1 As String
Dim nome2 As String
nome1 = InputBox("Inserisci il primo Nome")
nome2 = InputBox("Inserisci il secondo Nome")
Call dip_nome1(nome1, nome2)
End Sub
```

Esempio: Dichiarare l'argomento Optional come String, senza utilizzare la funzione IsMissing.
La dichiarazione della routine contiene due argomenti, il secondo argomento è specificato come Optional. Si noti in questo esempio che l'argomento opzionale viene dichiarato come String e non è passato nella procedura, di conseguenza il Range ("B1"), conterrà un riferimento *Null* dopo aver eseguito la sub *dip_nome2* perché all'argomento opzionale verrà assegnato il valore di default per il suo tipo di dati (String), che è Nothing cioè un riferimento Null.

Se l'argomento opzionale viene dichiarato come Integer, Range ("B1") conterrà zero dopo l'esecuzione della sub dip_nome2 in quanto all'argomento opzionale verrà assegnato il valore predefinito per il suo tipo di dati (Integer) che è pari a zero.

```
Sub dip_nome2(Optional primo As String, Optional secondo As String)
ActiveSheet.Range("A1") = primo
ActiveSheet.Range("B1") = secondo
End Sub

Sub nome_pieno2()
Dim nome1 As String
nome1 = InputBox("Inserisci il primo Nome")
Call dip_nome2(nome1)
End Sub
```

Esempio: Dichiarare l'argomento Optional come Variant, e utilizzare la funzione IsMissing.
La dichiarazione della routine contiene due argomenti, il secondo argomento è specificato come Optional. L'argomento opzionale dovrebbe essere (anche se non necessario), dichiarato come tipo di dati Variant per consentire l'uso della funzione IsMissing.

La funzione *IsMissing* viene utilizzata per determinare se l'argomento opzionale è stato approvato nella procedura o meno

```
Sub dividi1(primo_n As Integer, Optional secondo_n As Variant)
Dim result As Double
If IsMissing(secondo_n) Then
result = primo_n
Else
result = primo_n / secondo_n
End If
result = Format(result, "#.##")
MsgBox result
End Sub

Sub chiama_dividi1()
Dim numero1 As Integer
numero1 = InputBox("Inserire il primo Numero")
Call dividi1(numero1)
End Sub
```

Esempio: Dichiarare l'argomento Optional come Integer, ma la funzione IsMissing non funzionerà e il codice restituirà un errore.

La dichiarazione della routine contiene due argomenti, il secondo argomento è specificato come Optional. L'argomento opzionale viene dichiarato come Integer, ma la funzione *IsMissing* non funzionerà con una variabile non dichiarata come Variant. Quindi all'argomento opzionale verrà assegnato il valore predefinito per il tipo di dati che è 0 per i tipi di dati numerici e il codice restituirà l'errore: Errore di run-time "11": Divisione per zero.

```
Sub dividi2(primo_n As Integer, Optional secondo_n As Integer)
Dim result As Double
If IsMissing(secondo_n) Then
result = primo_n
Else
result = primo_n / secondo_n
End If
result = Format(result, "#.##")
MsgBox result
End Sub

Sub chiama_dividi2()
Dim numero_1 As Integer
numero_1 = InputBox("Inserisci il primo Numero")
Call dividi2(numero_1)
End Sub
```

Esempio: Dichiarare l'argomento Optional come String, ma la funzione IsMissing non funzionerà e il codice restituirà un errore.

La dichiarazione della routine contiene due argomenti, il secondo argomento è specificato come Optional. L'argomento opzionale viene dichiarato come stringa, ma la funzione IsMissing non funzionerà con una variabile non dichiarata come tipo di dati Variant. Quindi all'argomento opzionale verrà assegnato il valore predefinito per il tipo di dati che è Nothing (un riferimento Null) per il tipo di dati String e il codice restituirà l'errore: Errore di run-time "13": Tipo non corrispondente.

```
Sub dividi3(primo_n As Integer, Optional secondo_n As String)
Dim result As Double
If IsMissing(secondo_n) Then
result = primo_n
Else
result = primo_n / secondo_n
End If
result = Format(result, "#.##")
MsgBox result
End Sub

Sub chiama_dividi3()
Dim numero1 As Integer
numero1 = InputBox("Inserisci il primo Numero")
Call dividi3(numero1)
End Sub
```

Specificare un valore predefinito per un argomento facoltativo

È possibile specificare un valore predefinito per un argomento opzionale che sarà utilizzato se l'argomento opzionale non viene passato alla procedura. In questo modo è possibile dichiarare gli argomenti opzionali di qualsiasi tipo di dati e specificare un valore predefinito che sarà utilizzato se l'argomento opzionale viene omesso, evitando l'uso della funzione IsMissing che funziona solo con il tipo di dati Variant.

Esempio: Un argomento opzionale non viene passato alla procedura a viene utilizzato un valore predefinito.

La dichiarazione della routine contiene due argomenti, il secondo argomento è specificato come Optional. Se il secondo argomento, che è opzionale, non è passato nella procedura un valore predefinito viene utilizzato:

```
Sub dividi4(primoN As Integer, Optional secondoN As Integer = 3)
Dim result As Double
result = primoN / secondoN
result = Format(result, "#.##")
MsgBox result
End Sub

Sub calcula_2()
Dim numero1 As Integer
numero1 = InputBox("Inserisci il primo Numero")
Call dividi4(numero1)
End Sub
```

Passare un numero indefinito di argomenti - parametro Array (ParamArray)

Abbiamo visto come dichiarare procedure passando argomenti, tra cui argomenti opzionali, ma comunque sempre limitati al numero di argomenti dichiarati nella procedura. Usando la parola chiave **ParamArray** sarà consentito passare un numero arbitrario di argomenti alla procedura in modo che vengano accettati un numero indefinito di argomenti. Utilizzare *ParamArray* quando non si è sicuri del numero preciso di argomenti da passare a una procedura, e potrebbe anche essere conveniente creare un array opzionale (cioè un ParamArray) che passare attraverso il fastidio di dichiarare un gran numero di argomenti opzionali, e quindi utilizzando la funzione IsMissing con ciascuno di essi.

Una procedura utilizza le informazioni sotto forma di variabili, costanti ed espressioni per effettuare azioni ogni volta che viene chiamata e tramite la dichiarazione del procedimento si definisce un parametro che consente al codice chiamante (il codice che chiama la procedura) di passare un argomento, o il valore di tale parametro, in modo che ogni volta che la routine viene chiamata il codice chiamante può passare un argomento diverso per lo stesso parametro. Un parametro viene dichiarato come una variabile specificando il nome e il tipo di dati. Dichiarando una matrice di parametri, la procedura può accettare una matrice di valori per un parametro. Una matrice di parametri è così definita utilizzando la parola chiave ParamArray.

ParamArray può essere definita in una procedura ed è sempre l'ultimo parametro nell'elenco dei parametri. Un ParamArray è un parametro opzionale e può essere l'unico parametro facoltativo in una procedura e deve essere dichiarato come un array di tipo Variant. Indipendentemente dall'impostazione Option Base per il modulo, LBound di un ParamArray sarà sempre 0 e il valore indice per l'array partirà da 0. *ByVal*, *ByRef* o *keywords* sono opzionali e non possono essere utilizzate con ParamArray.

Per accedere al valore di una matrice di parametri si utilizza la funzione *UBound* per determinare la lunghezza dell'array che vi darà il numero di elementi o valori di indice nella matrice. Nel codice della procedura si può accedere al valore di una matrice di parametri digitando il nome dell'array seguito da un valore di indice (che dovrebbe essere compreso tra 0 e il valore UBound)

Esempio: Passare un numero arbitrario di argomenti utilizzando un parametro ParamArray.

```
Sub aggiungiN(ParamArray numeri() As Variant)
Dim somma As Long
Dim i As Long
For i = LBound(numeri) To UBound(numeri)
somma = somma + numeri(i)
Next i
MsgBox somma

End Sub
Sub chiama_aggiungiN()
Call aggiungiN(22, 25, 30, 40, 55)
End Sub
```

Esempio: Un argomento obbligatorio e un numero arbitrario di argomenti utilizzando ParamArray.
Una routine con un solo argomento richiesto (comm) e quindi permette di passare un numero arbitrario di argomenti utilizzando il parametro ParamArray.

```
Sub calcComm(comm As Double, ParamArray venD() As Variant)
Dim somma As Double
Dim n As Variant
For Each n In venD
MsgBox n
totalComm = totalComm + comm * n
Next n
MsgBox totalComm
End Sub

Sub chiama_Comm()
Dim c As Double
c = 0.3
Call calcComm(c, 100, 200, 300)
End Sub
```

Esempio: Due argomenti richiesti e un numero arbitrario di argomenti utilizzando ParamArray.
Una dichiarazione di routine con due argomenti richiesti (studente e media) e quindi permette di passare un numero arbitrario di argomenti utilizzando il parametro ParamArray.

```
Sub media_V(studente As String, media As Double, ParamArray voti() As Variant)
Dim n As Variant
Dim voto As String
Dim T_voti As String
For Each n In voti
If n >= 80 Then
voto = "Eccellente"
ElseIf n >= 60 Then
voto = "Buono"
ElseIf n >= 40 Then
voto = "Medio"
Else
voto = "Bocciato"
End If
If Len(T_voti) = 0 Then
T_voti = voto
Else
T_voti = T_voti & ", " & voto
End If
Next n
For Each n In voti
i = i + 1
somma = somma + n
media = somma / i
Next n
T_voti = """" & T_voti & """"
media = Format(media, "#.##")
MsgBox studente & " " & "ha voti di tipo " & T_voti & " " & "e una media voti di" & " " &
media
End Sub
```