



Ordinare i dati di una Collection

http://www.vbsimple.net/howto/ht_023.htm

Difficoltà: ►▶ 2 / 5

Vi sarà capitato spesso di dover ordinare una serie di dati.

In questo HowTo vedremo come applicare il metodo di ordinamento a bolle (Bubble Sort), studiato più approfonditamente nella sezione [Informazioni aggiuntive](#).

Partendo dal presupposto che conosciamo già il funzionamento di tale metodo di ordinamento, spiegato nell'altra sezione, trasformiamo l'algoritmo in un semplicissimo codice Visual Basic.

Ma prima vogliamo precisare che non è possibile modificare direttamente i dati contenuti in una [Collection](#). Pertanto, al fine di ordinare la Collection, utilizzeremo un array provvisorio, che utilizzeremo per effettuare l'ordinamento. Finito l'ordinamento azzeriamo e ricostruiamo la Collection in maniera ordinata.

```
1. Option Explicit
2.
3. Public Sub OrdinaCollection(ByRef COLLEDAORDINARE As Collection)
4.     Dim ARRAYDAORDINARE() As Variant
5.     Dim SCAMBIO As Variant
6.     Dim INDICE1 As Integer
7.     Dim INDICE2 As Integer
8.
9.     ReDim ARRAYDAORDINARE(COLLEDAORDINARE.Count)
10.    For INDICE1 = 1 To COLLEDAORDINARE.Count
11.        ARRAYDAORDINARE(INDICE1) = COLLEDAORDINARE.Item(INDICE1)
12.    Next INDICE1
13.
```

La funzione che effettua quest'operazione si chiamerà **OrdinaCollection** e richiede che sia passato come parametro una *Collection* di nome **COLLEDAORDINARE**. Il parametro richiesto viene passato [per riferimento](#), pertanto ogni modifica della *Collection* all'interno della Sub si rifletterà all'esterno della Sub stessa.

Alla riga 4 definiamo una matrice di [Variant](#) di nome **ARRAYDAORDINARE**, senza specificarne la dimensione, poiché essa sarà definita in maniera dinamica in base alla dimensione della Collection.

Alla riga 5 abbiamo dichiarato una variabile di nome SCAMBIO per effettuare l'inversione di due elementi dell'array. Alle righe 6 e 7 invece abbiamo le due variabili indice necessarie per effettuare i confronti tra gli elementi.

La riga 9 effettua il dimensionamento dinamico dell'array **ARRAYDAORDINARE**, definendo l'ampiezza della matrice in base al numero di elementi contenuti nella *Collection*.

Prima di entrare nel ciclo di ordinamento effettueremo la copia dei dati della Collection

COLLDAORDINARE all'interno dell'array **ARRAYDAORDINARE** (righe 10-12).

```

14.     For INDICE1 = 1 To UBound(ARRAYDAORDINARE) - 1
15.         For INDICE2 = INDICE1 + 1 To UBound(ARRAYDAORDINARE)
16.             If ARRAYDAORDINARE(INDICE2) < ARRAYDAORDINARE(INDICE1) Then
17.                 SCAMBIO = ARRAYDAORDINARE(INDICE1)
18.                 ARRAYDAORDINARE(INDICE1) = ARRAYDAORDINARE(INDICE2)
19.                 ARRAYDAORDINARE(INDICE2) = SCAMBIO
20.             End If
21.         Next INDICE2
22.     Next INDICE1
23.

```

Il ciclo che vediamo sopra è il classico algoritmo per l'ordinamento a bolle; i due indici di controllo sono **INDICE1** e **INDICE2**. Alla riga 16 viene effettuato il controllo degli elementi puntati: se il secondo elemento è minore del primo elemento, sarà necessario invertire i due valori, mediante l'utilizzo di una variabile di scambio (righe 17-19).

```

24.     For INDICE1 = 1 To UBound(ARRAYDAORDINARE)
25.         COLLDAORDINARE.Remove 1
26.         COLLDAORDINARE.Add ARRAYDAORDINARE(INDICE1)
27.     Next INDICE1
28. End Sub
29.

```

Terminato l'ordinamento effettuiamo l'azzeramento e il nuovo riempimento della *Collection* con i dati della matrice ordinata.

Per provare questa funzione utilizzeremo una *ListBox* di nome **ListaNumeri** per contenere i numeri da ordinare e due *CommandButton* di nome **GeneraNumeri** e **OrdinaDati**: il primo genererà 10 numeri in maniera casuale e li inserirà nella *ListBox*, mentre il secondo effettua l'ordinamento dei numeri e rigenera la *ListBox* con i numeri ordinati.



Ricordiamo che l'oggetto *ListBox* possiede una [proprietà](#) di nome **Sorted** che permette l'auto-ordinamento dei valori all'interno della *ListBox*, ma per i nostri scopi terremo questa proprietà disattivata, impostandola a **False**.

Oltre al codice della funzione di ordinamento, abbiamo anche il codice degli [eventi](#) dei due pulsanti:

```

30. Private Sub GeneraNumeri_Click()
31.     Dim CONTA As Integer
32.     ListaNumeri.Clear
33.     Randomize
34.     For CONTA = 1 To 10
35.         ListaNumeri.AddItem Int(Rnd * 10)
36.     Next CONTA
37. End Sub
38.

```

Il click sopra il pulsante **GeneraNumeri** riempie la *ListBox* **ListaNumeri** con 10 numeri casuali, compresi tra 0 e 10.

La routine è semplicissima: viene azzerata la *ListBox*, viene inizializzato il generatore di

numeri casuali e vengono generati ed inseriti nella ListBox 10 numeri casuali.

```
39. Private Sub OrdinaDati_Click()  
40.     Dim NUMERI As New Collection  
41.     Dim CONTA As Integer  
42.     For CONTA = 0 To ListaNumeri.ListCount - 1  
43.         NUMERI.Add ListaNumeri.List(CONTA)  
44.     Next CONTA  
45.     OrdinaCollection NUMERI  
46.     ListaNumeri.Clear  
47.     For CONTA = 1 To NUMERI.Count  
48.         ListaNumeri.AddItem NUMERI.Item(CONTA)  
49.     Next CONTA  
50.     Set NUMERI = Nothing  
51. End Sub
```

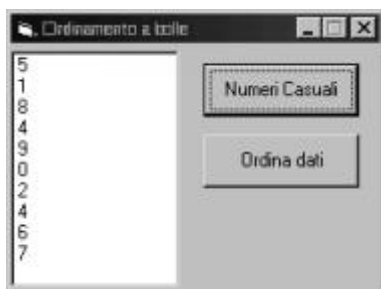
Mentre, il click sopra il pulsante **OrdinaDati**, inizialmente [alloca](#) una *Collection* di nome **NUMERI** e la riempie con i valori della *ListBox* **ListaNumeri** (righe 40-44).

Alla riga 45 viene richiamata la funzione *OrdinaCollection* che effettua l'ordinamento.

Ordinata la Collection, possiamo azzerare la ListBox e riempirla con i nuovi valori già ordinati (righe 46-49).

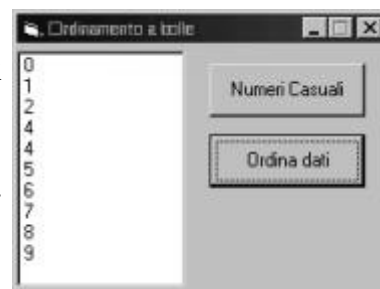
Terminato questo possiamo tranquillamente [deallocare](#) la *Collection* **NUMERI** (riga 50).

Il codice termina qui e possiamo passare alla sua prova.



Il click sopra il pulsante "**Numeri Casuali**" riempie la nostra ListBox con dei valori casuali.

Il click sul pulsante "**Ordina dati**" effettua l'ordinamento di tali valori.



L'algoritmo [Bubble sort](#) è uno dei più semplici metodi di ordinamento e non presenta rischi particolari.

Naturalmente, dove possibile, cercare di sfruttare le proprietà built-in, già esistenti, dei controlli, invece che effettuare l'ordinamento tramite questo algoritmo. Altresì, nel trattamento di dati da un database si consiglia di sfruttare le istruzioni SQL provvedute dai motori di database.

In caso di ordinamento di altro tipo dati, la funzione che abbiamo visto può essere utilizzata tranquillamente, senza presentare problemi di alcuna sorta.

[Fibia FBI](#)
31 Gennaio 2001



[Torna all'indice degli HowTo](#)