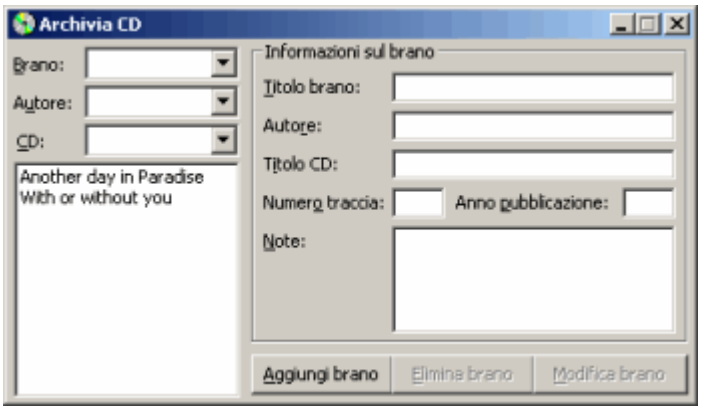
Corso Intermedio - Lezione 8

http://www.vbsimple.net/intermed/int_08.htm

- Le proprietà di base del controllo ListBox (ListIndex, List, ListCount).
- Lettura casuale da un file binario.
- Eliminazione di un brano.



Il nostro piccolo progetto inizia a prendere forma: nella lezione 6 abbiamo visto come inserire i dati nel nostro archivio, mentre nella lezione 7 abbiamo recuperato gli stessi dati all'avvio del programma. Ci occuperemo stavolta di leggere i dettagli di un singolo brano, nel momento in cui questo viene selezionato dall'elenco di sinistra.

Diviene perciò importante imparare ad utilizzare il controllo ***ListBox***. Si suppone naturalmente che vi sia almeno un brano nell'archivio. Esiste una proprietà di nome ***ListIndex*** che determina quale sia l'elemento attualmene selezionato ed il suo valore è a **base 0**; ciò significa che il primo elemento è indicato dal valore 0, il secondo dal valore 1 e così via. Invece nel momento in cui nessun elemento dell'elenco risulta selezionato, il valore di tale proprietà è **-1**.
La proprietà funziona sia in lettura (recupero dell'elemento selezionato) che in scrittura (selezione di un elemento) e può essere utilizzata come segue:

```
MsgBox "L'elemento correntemente selezionato è: " & lstBrani.ListIndex
lstBrani.ListIndex = 1
```

Ma la proprietà *ListIndex* restituisce soltanto l'indice dell'elemento selezionato e non riporta il contenuto di tale elemento. Sapremo infatti solo la sua posizione all'interno dell'elenco ma non ne conosceremo il contenuto. Possiamo quindi ovviare a questo inghippo semplicemente utilizzando la proprietà ***List***. Si tratta di una proprietà indicizzata ovvero richiede la specifica dell'indice dell'elemento di cui restituire il contenuto. Prendendo per esempio la figura accanto il contenuto dell'elemento con indice 0 è *"Another day in Paradise"*, ed il contenuto dell'elemento 1 è *"With or without you"*. Il suo utilizzo è davvero molto semplice:

```
MsgBox "Il primo brano è: " & lstBrani.List(0)
MsgBox "Il secondo brano è: " & lstBrani.List(1)
```

Anche questa proprietà è utilizzabile in operazioni di lettura e di scrittura; ciò significa che è possibile variare il contenuto di un elemento anche dopo il suo inserimento con una sintassi del genere:

```
lstBrani.List(0) = "Nuovo contenuto dell'elemento 0"
```

La combinazione delle due proprietà qui presentate ci consente di recuperare o modificare il contenuto dell'elemento selezionato:

```
MsgBox "L'elemento selezionato è: " & lstBrani.List(lstBrani.ListIndex)
```


È fondamentale ricordare che la proprietà List richiede l'indice dell'elemento da recuperare e che la proprietà ListIndex può anche riportare valore -1, ad indicare che nessun elemento dell'elenco è selezionato. In tal caso l'operazione di lettura mediante List produce un errore. Non è infatti possibile leggere il contenuto dell'elemento il cui indice è -1. Prima di utilizzare la proprietà *List* assicurarsi quindi che la proprietà ListIndex abbia un valore differente da -1:

```
If lstBrani.ListIndex <> -1 Then
  MsgBox "L'elemento selezionato è: " & lstBrani.List(lstBrani.ListIndex)
Else
  MsgBox "Nessun elemento selezionato"
End If
```

Un'altra proprietà del controllo ListBox utilizzata frequentemente è ***ListCount*** che restituisce il numero di elementi contenuti nel controllo; nel nostro esempio restituirebbe il valore 2. Viene utilizzata frequentemente nei cicli che eseguono operazioni sui valori dell'elenco:

```
For intConta = 0 To lstBrani.ListCount - 1
  If lstBrani.List(intConta) = strTestoDaRicercare Then ...
Next intConta
```

La precedente rappresenta una grezza forma di ricerca che scorre tutti gli elementi dell'elenco per trovare quello indicato dalla variabile **strTestoDaRicercare**. Il ciclo è effettuato partendo dal valore 0 (la base della proprietà *List*) fino al numero indicato dalla proprietà ListCount - 1. Questo perché *ListCount* restituisce il numero di elementi e non l'indice dell'ultimo elemento. Dati infatti 2 elementi, la proprietà ListCount restituisce valore 2, mentre l'ultimo elemento è indicato dall'indice 1.

Nelle lezioni precedenti abbiamo utilizzato le proprietà *NewIndex* per recuperare l'indice di un nuovo elemento dopo l'operazione di inserimento e *ItemData* per assegnare un valore numerico ad ogni riga dell'elenco.

Riassumiamo quindi in breve le principali proprietà sin qui presentate:

- **ListIndex**
  Restituisce e imposta l'indice dell'elemento selezionato.
- **NewIndex**
  Restituisce l'indice dell'elemento dopo un'operazione di inserimento mediante **AddItem**.
- **ListCount**
  Restituisce il numero di elementi contenuti nell'elenco.
- **List**
  Restituisce ed imposta il contenuto di ciascun elemento dell'elenco. Richiede la specifica dell'indice dell'elemento.
- **ItemData**

Restituisce ed imposta un valore numerico a ciascun elemento. Richiede la specifica dell'indice dell'elemento.

Con questi concetti possiamo quindi affontare l'operazione di recupero di un singolo brano scelto dall'elenco a sinistra. In occasione dell'evento *Click* sul controllo ListBox sarà possibile effettuare la lettura dal file del brano selezionato. Abbiamo detto nella lezione precedente che l'istruzione Get accetta tre argomenti di cui il secondo specifica il record da leggere. Questa operazione è detta lettura casuale da un file poiché la posizione di lettura non si presenta sequenziale né segue alcuno schema fisso. In ogni momento la lettura può riferirsi ad una posizione differente all'interno del file.

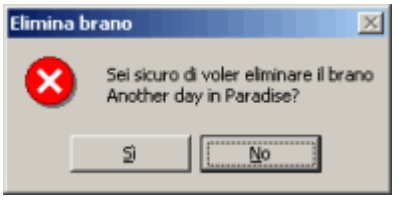
```
 1. Private Sub lstBrani_Click()
 2.   If lstBrani.ListIndex >= 0 Then
 3.     Get #intFileDati, lstBrani.ItemData(lstBrani.ListIndex), udtDiscoCorrente
 4.     With udtDiscoCorrente
 5.       txtTitoloBrano.Text = Trim$(.strTitoloBrano)
 6.       txtAutoreBrano.Text = Trim$(.strAutore)
 7.       txtTitoloCD.Text = Trim$(.strTitoloCD)
 8.       txtNumeroTraccia.Text = CStr(.intTraccia)
 9.       txtAnnoPubblicazione.Text = CStr(.intAnnoPubblicazione)
10.       txtNote.Text = Trim$(.strNote)
11.     End With
12.   End If
13. End Sub
14.
```

Alla riga 2 effettuiamo quel controllo d'obbligo che vi sia almeno un elemento dell'elenco selezionato; appurato questo provvederemo ad effettuare un'operazione di lettura casuale, mirando al record specificato nella proprietà *ItemData* dell'elemento selezionato. Abbiamo accennato nella lezione precedente la scelta di utilizzare questo tipo di valore, che diverrà fondamentale a seguito di una cancellazione o riordinamento degli elementi. Il record letto dal file sarà contenuto nella variabile **udtDiscoCorrente** i cui dati saranno in seguito riportati nelle singole caselle di testo (righe 5-10).

Prima di occuparci dell'operazione di modifica di un brano già inserito vedremo come eliminare il brano selezionato. L'operazione di cancellazione in realtà semplicemente celerà il record segnato come eliminato ma non sarà fisicamente rimosso dall'archivio. La ragione di questa scelta è molto semplice: ad ogni operazione di cancellazione si renderebbe necessario redisporre tutti i brani successivi a quello eliminato e si tratta invece di uno sforzo eccessivo. L'operazione di cancellazione segnerà il campo *blnEliminato* con il valore ***False***:

```
15. Private Sub cmdEliminaBrano_Click()
16.     If lstBrani.ListIndex >= 0 Then
17.         If MsgBox("Sei sicuro di voler eliminare il brano" _
18.             & vbNewLine & Trim$(udtDiscoCorrente.strTitoloBrano) _
19.             & "?", vbCritical Or vbYesNo Or vbDefaultButton2, _
20.             "Elimina brano") = vbYes Then
21.             udtDiscoCorrente.blnEliminato = True
22.             Put #intFileDati, udtDiscoCorrente.intIndice, udtDiscoCorrente
23.             lstBrani.RemoveItem lstBrani.ListIndex
24.         End If
25.     End If
26. End Sub
```

```
27.
```



Alla riga 16 è effettuato il consueto controllo sulla riga selezionata; se almeno un elemento è stato selezionato sarà presentata un messaggio di avviso dell'operazione di eliminazione. Se l'utente dovesse rispondere ***Sì*** (*vbYes*) all'avviso presentato, il campo **blnEliminato** del record corrente, precaricato mediante l'evento Click sull'elenco, come visto nel codice precedente, sarà segnato con il valore ***True***. All'avvio del programma, infatti, i record segnati come eliminati non saranno presentati nell'elenco dei brani. Sarà rimossa anche la riga dall'elenco dei brani (riga 23).

Affinché l'evento Click sul pulsante **cmdElimina** scatti è ovviamente necessario che il pulsante stesso sia abilitato; rimuoveremo quindi la riga `cmdEliminaBrano.Enabled = False` dal Form_Load perché al momento non esiste una piena gestione dello stato dei pulsanti.

Abbiamo fin qui presentato le operazioni più semplici: inserimento, elencazione ed eliminazione di un brano dall'archivio. Resta quindi da implementare l'operazione di modifica che si rivela un po' più complessa delle precedenti. Nella prossima lezione introdurremo la gestione dello stato dei pulsanti.

Fibia FBI
19 Ottobre 2003

Torna alla settima lezione Vai alla nona lezione