

Utilizzo dei pulsanti di opzione per la scelta del form da caricare

http://www.vbsimple.net/activity/act_06.htm

Richiesta di: [Mauro Scaini](#) - 18 Novembre 2000

Difficoltà: 2 / 5

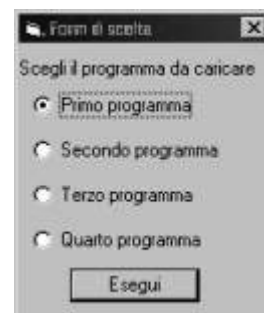
Sto sviluppando un progetto e vorrei dare all'utente la possibilità di scegliere quale form caricare utilizzando i pulsanti di opzione.

Questa è una scelta implementativa molto comune.

Abbiamo una serie di `OptionButton` e vogliamo che in base ad essi appaiano all'utente diversi forms.

Come punto di partenza definiamo il primo form che fornirà le scelte all'utente. Nel nostro caso avremo quattro programmi che l'utente può scegliere e quattro `OptionButton` sul form principale.

Dunque sul nostro `Form1` abbiamo una matrice di quattro pulsanti di opzione. Ricordiamo che per formare una matrice di controlli basta inserire almeno due pulsanti e dargli lo stesso nome. Verrà chiesto se si desidera formare una matrice di controlli. Rispondendo SI verrà impostata in automatico la proprietà `Index`, che differenzia i diversi pulsanti all'interno della matrice.



Nel nostro esempio i quattro pulsanti si chiameranno **Selezione** e le loro proprietà `Index` andranno da 0 a 3. Posizioniamo i quattro pulsanti nel modo visibile nella figura, avendo cura di posizionare il pulsante con `Index 0` nella prima posizione, quello con `Index 1` subito sotto e così via.

Inseriamo anche un pulsante standard di nome **PulsanteOK** ed assegnamogli la `Caption` *Esegui*. Al click sopra di esso verrà controllato lo stato dei quattro pulsanti e verrà caricato il programma corrispondente al pulsante la cui proprietà `Value` è `True`. Per questo scopo è fondamentale che i pulsanti abbiano valori della proprietà `Index` consecutivi (0, 1, 2, 3).

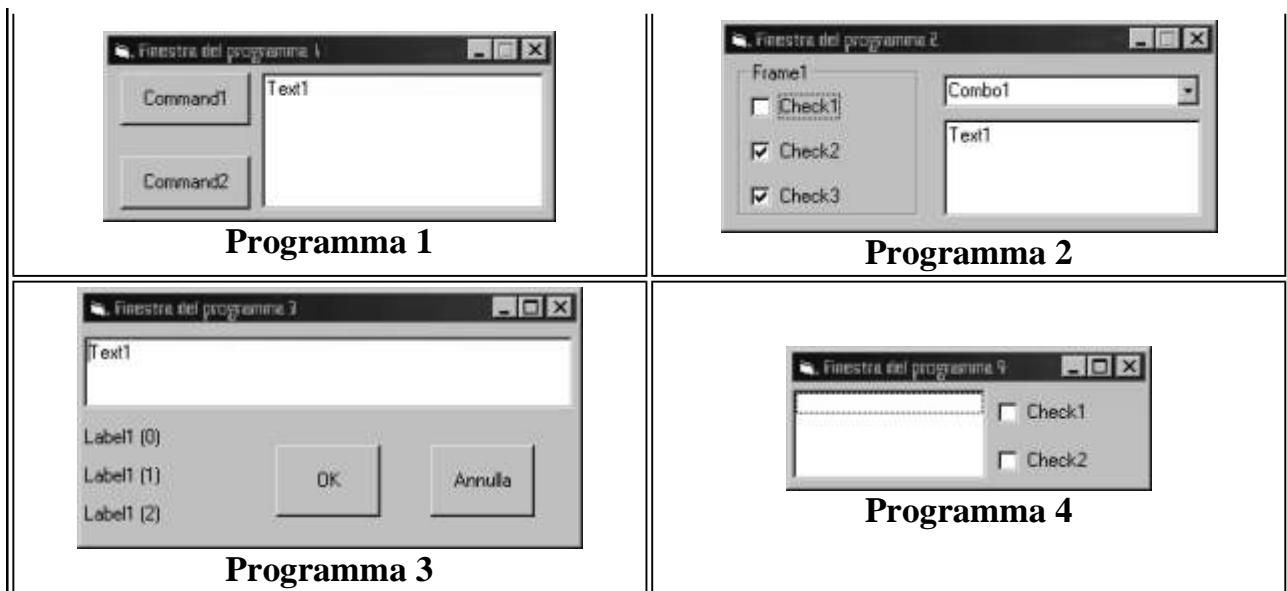
Prima di passare al codice stabiliamo quali saranno i forms che verranno caricati nel momento in cui sarà premuto il pulsante *Esegui*.

Nel nostro esempio inseriamo altri quattro forms di nome **Programma1**, **Programma2**, **Programma3** e **Programma4**.

Per il nostro scopo non ci interessa cosa faranno i diversi programmi; tuttavia inseriamo qualche controllo per riconoscere ogni programma da un altro.

Eccì i nostri 4 forms:





Faremo in modo che non sia possibile aprire più di un programma per volta e la chiusura di un programma provochi la ricomparsa del form di scelta. Quando l'utente ha deciso di terminare l'uso del programma dovrà chiudere il form di scelta.

Cominciamo a lavorare nel codice dei diversi programmi.

Apriamo la finestra del codice dei quattro forms Programma1, Programma2, Programma3 e Programma4 e, nella routine legata all'evento *Unload* dell'oggetto **Form**, scriviamo:

```
Form1.Show 0
```

Per ognuno dei quattro forms dovremmo avere quindi:

```
1. Private Sub Form_Unload(Cancel As Integer)
2.     Form1.Show 0
3. End Sub
4.
```

Fatto questo apriamo la finestra del codice di **Form1** ovvero del form di scelta programma e scriviamo nella routine dell'evento *Click* di **PulsanteOK**:

```
5. Private Sub PulsanteOK_Click()
6.     Dim PulsantePremuto As Integer
7.     Dim Contatore As Integer
8.     For Contatore = 0 To 3
9.         If Selezione(Contatore).Value = True Then
10.            PulsantePremuto = Contatore
11.        Exit For
12.    End If
13. Next Contatore
14. Select Case PulsantePremuto
15.     Case 0
16.         Load Programma1
17.         Programma1.Show 0
18.     Case 1
19.         Load Programma2
20.         Programma2.Show 0
21.     Case 2
22.         Load Programma3
23.         Programma3.Show 0
24.     Case 3
25.         Load Programma4
26.         Programma4.Show 0
27. End Select
```

```
28.      Me.Hide  
29. End Sub
```

In questa routine dichiareremo due variabili: **PulsantePremuto** che indicherà l'indice (ovvero il contenuto della proprietà Index) del pulsante premuto e **Contatore** che ci servirà come appoggio per controllare qual'è il pulsante premuto.

All'interno del codice i singoli elementi della matrice sono indicati con Selezione(0), Selezione(1) a seconda del loro indice.

Subito sotto le dichiarazioni abbiamo un ciclo che controlla tutti i pulsanti **Selezione**. Se durante il ciclo un pulsante è attivato, ergo ha la proprietà *Value* impostata a True, viene memorizzato il valore di Contatore all'interno della variabile **PulsantePremuto** ed avviene l'uscita forzata dal ciclo con `Exit For`.

All'uscita del ciclo viene effettuato un controllo del valore di PulsantePremuto mediante un'istruzione `Select Case`.

Viene fatto un controllo per i singoli valori ammissibili (dallo 0 al 3) e di conseguenza viene caricato (con `Load`) e mostrato (con `Show`) il form relativo.

Prima di uscire dalla routine viene nascosto il form di scelta, ma non verrà scaricato dalla memoria. Il Form1 verrà mostrato nuovamente all'uscita dei singoli programmi, attraverso l'istruzione `Form1.Show 0`.

È estremamente non sottovalutare questo: un form nascosto mediante l'istruzione `Hide` non viene né chiuso né scaricato dalla memoria. È un errore molto comune nascondere un form, mostrarne un altro e, quando terminato l'uso, chiudere il secondo senza chiudere il primo.

Visibilmente potrebbe sembrare tutto a posto perché non ci sono sullo schermo finestre visibili, ma in questa situazione di errore il form nascosto occupa aree di memoria e può pregiudicare il corretto funzionamento del computer fintanto che questa situazione dura.

Si è preferito in quest'esempio la chiarezza del codice e la sua semplicità al risparmio di istruzioni. Lo stesso codice poteva, infatti, essere rappresentato con circa 11 righe di codice anziché le 25 viste sopra.

[Fibia FBI](#)

18 Novembre 2000



[Torna all'introduzione delle Richieste dei lettori](#)
