



Gestione Eventi di foglio di lavoro con Un Modulo Di Classe

Quando si utilizzano dei controlli in un foglio di lavoro, spesso si ha la necessità di utilizzare gli eventi per gestire le azioni che possono essere eseguite. Al tempo stesso se si usano diversi controlli, può diventare problematico far interagire i vari controlli in quanto si deve aggiungere una o più sub evento per ognuno di essi. Per semplificarne la programmazione si può usare un modulo di classe con una singola routine evento per una serie di controlli uguali

Possiamo capirne meglio il significato vedendo un esempio partendo da un foglio di lavoro in cui inseriamo due serie di pulsanti di opzione tramite il percorso dal menu **Sviluppo – Inserisci** e scegliamo il pulsante di Opzione dalla sezione Controlli ActiveX

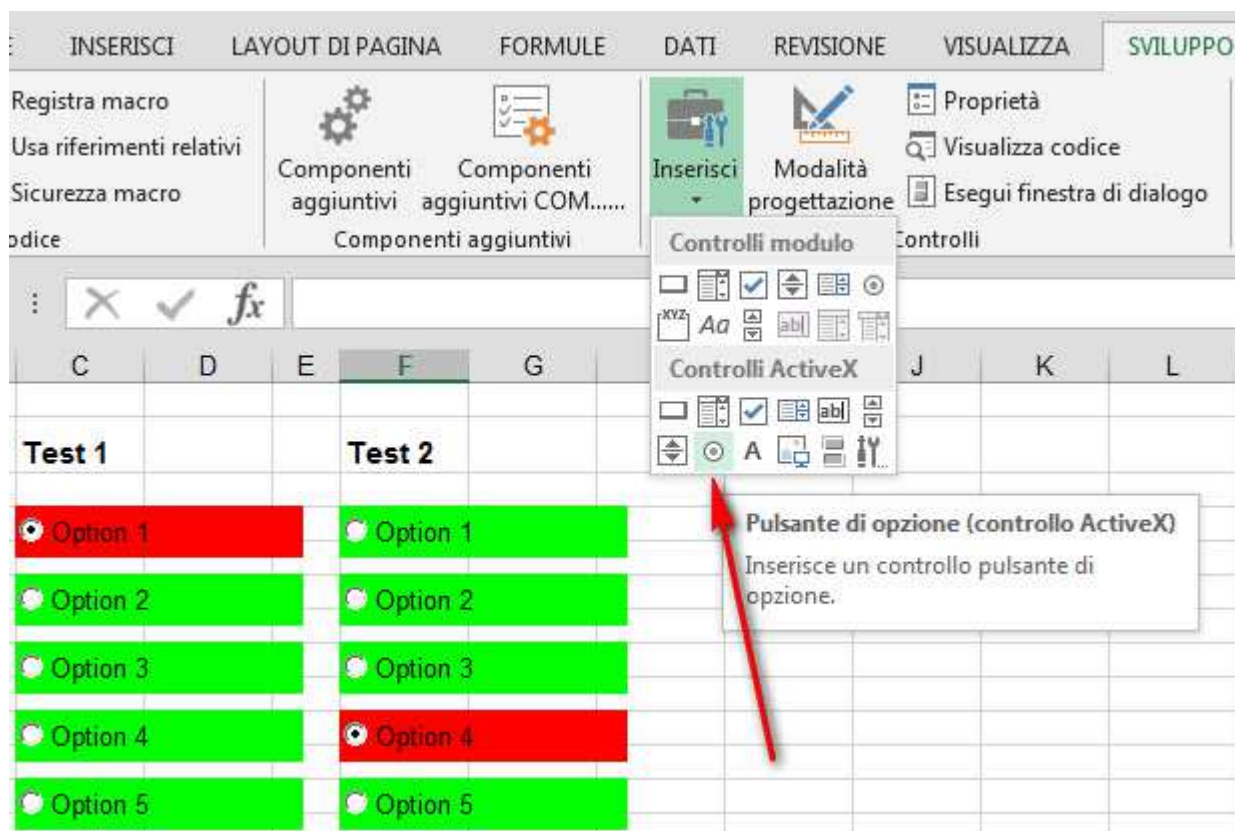


Fig. 1

Una volta inseriti e allineati i vari controlli, per poter farli lavorare come due gruppi separati di pulsanti, dovremmo assegnare due **GroupName** diversi, rispettivamente **Test1** e **Test2** tramite la modalità **Progettazione** e la finestra delle proprietà che troviamo nella scheda **Sviluppo**

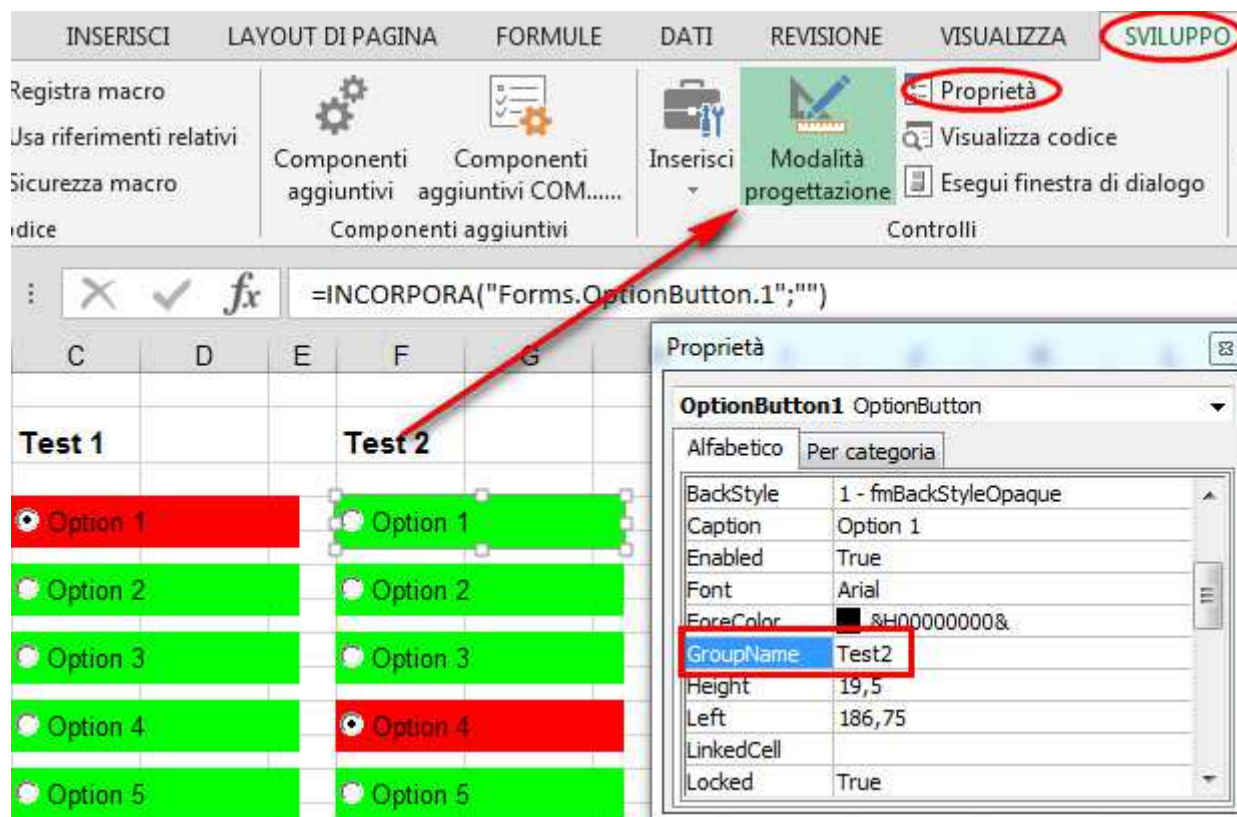


Fig. 2

In questo esempio faremo in modo che il pulsante di opzione selezionato diventi rosso mentre quelli non selezionati saranno di colore verde, inoltre mandiamo anche un avviso a video che riporti il nome del pulsante selezionato

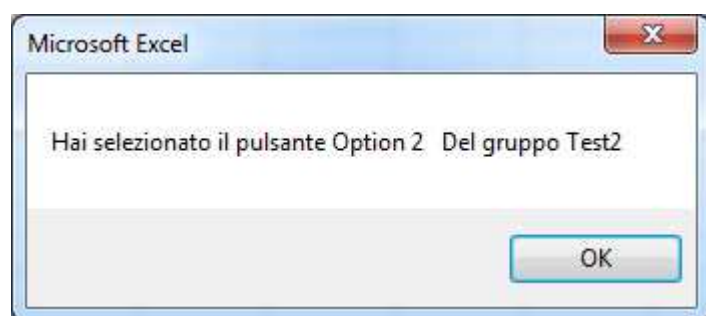


Fig. 3

Iniziamo inserendo un modulo di classe, entrando nell'editor di VBA dal menu **Inserisci – Modulo di classe**. Una volta inserita selezioniamo la classe e nella finestra delle proprietà, cambiamo il suo nome predefinito da Class1 a **clsTest**

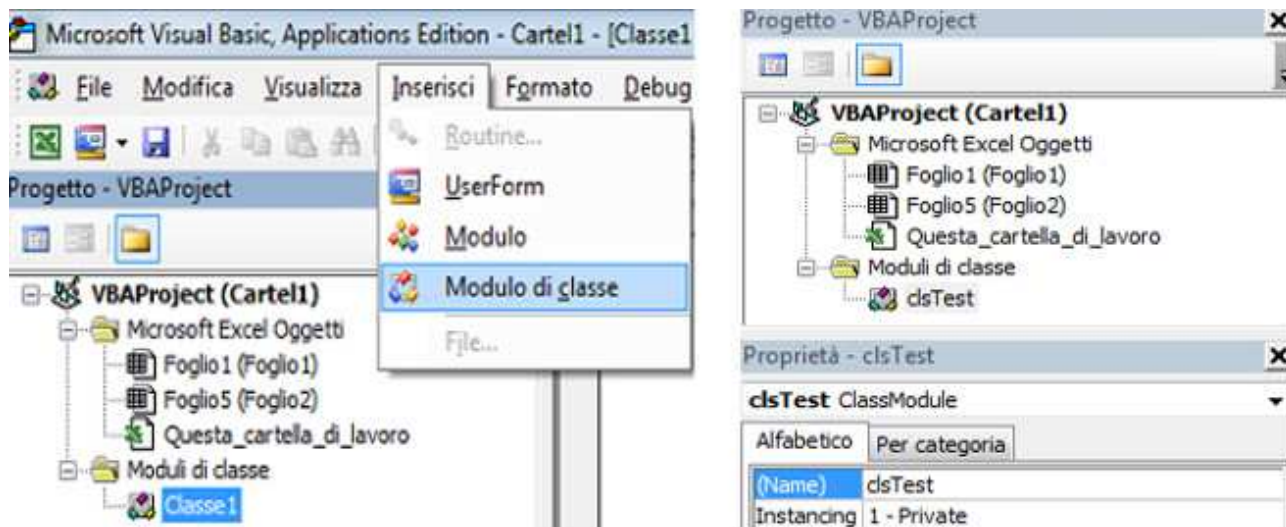


Fig. 4

Ora, nella finestra del codice digitiamo queste righe:

```
Option Explicit  
Private WithEvents testOpt As MSForms.OptionButton
```

Tramite la parola chiave **WithEvents** possiamo specificare che una variabile dichiarata (testOpt) si riferisca a un'istanza di una classe in grado di generare eventi. Infatti dopo aver fatto questo, si sarà in grado di selezionare la variabile testOpt dall'elenco a discesa nella parte superiore sinistra della finestra del codice

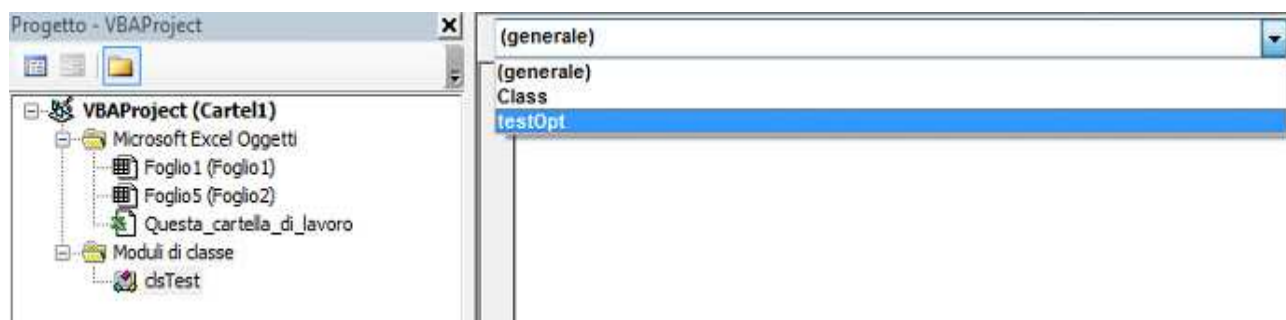


Fig. 5

Una volta selezionata, per impostazione predefinita, nella finestra del codice appare l'evento click in questo modo

```
Private Sub testOpt_Click()  
  
End Sub
```

Se ora si fa clic sul menu a discesa a destra, verranno elencati tutti gli eventi disponibili per questo tipo di controllo attraverso il modulo di classe e selezioniamo l'evento **Change** dal menu a discesa e rimuoviamo l'evento click

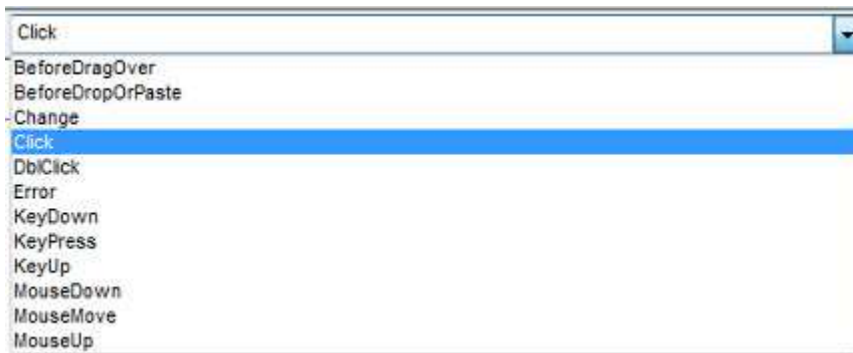


Fig. 6

E otterremo qualcosa di simile:

```
Option Explicit
Private WithEvents testOpt As MSForms.OptionButton

Private Sub testOpt_Change()

End Sub
```

Si diceva di cambiare il colore del controllo selezionato e, naturalmente, anche cambiare il colore del controllo deselezionato, inoltre si voleva anche un messaggio che avvisi quale controllo è stato selezionato. A tal proposito ho pensato di usare questo codice nel modulo di classe

```
Private Sub testOpt_Change()
    If testOpt.Value = 0 Then
        testOpt.Object.BackColor = RGB(0, 255, 0)
    Else
        MsgBox "Hai selezionato il pulsante " & testOpt.Caption & " Del gruppo " &
        testOpt.GroupName
        testOpt.Object.BackColor = RGB(255, 0, 0)
    End If
End Sub
```

Quando si clicca su un pulsante di opzione per selezionarlo, sia il pulsante selezionato e i pulsanti deselezionati eseguiranno i loro eventi di modifica, quindi questa Sub Evento sarà eseguita due volte, una volta per il controllo selezionato e una volta per il controllo deselezionato, il primo avrà valore 1 e il secondo valore 0. Tutto ciò che deve essere fatto ora è quello di collegare i controlli sul foglio di lavoro al modulo di classe. Per prima cosa si deve scrivere del codice nel modulo di classe che riceverà l'oggetto che sta per "ascoltare" da una routine di inizializzazione

```
Public Property Set Control(obtNew As MSForms.OptionButton)
    Set testOpt = obtNew
End Property
```

E infine distruggiamo la classe per liberare risorse

```
Private Sub Class_Terminate()
    Set testOpt = Nothing
End Sub
```




Ora abbiamo bisogno di creare le istanze di questo modulo di classe per quanti controlli abbiamo da collegare. Inseriamo un modulo normale nel progetto e useremo una variabile Collection per contenere le istanze dei moduli di classe:

```
Dim testC As Collection
```

Poi con un ciclo scorriamo tutti gli oggetti del foglio di lavoro e agganciamo i controlli OptionButton all'evento, ecco il codice per l'intero modulo:

```
Option Explicit
Dim testC As Collection

Sub InitializeEvents()
    Dim ctrOpt As OLEObject
    Dim Fgl As Worksheet
    Dim clsEvents As testOpt
    Set Fgl = ThisWorkbook.Worksheets(1)
    If testC Is Nothing Then
        Set testC = New Collection
    End If
    'Ciclo attraverso tutti i controlli
    For Each ctrOpt In Fgl.OLEObjects
        If TypeName(ctrOpt.Object) = "OptionButton" Then
            'Creiamo una nuova istanza della classe
            Set clsEvents = New testOpt
            Set clsEvents.Control = ctrOpt.Object
            'Aggiungiamo l'istanza ad una collezione
            testC.Add clsEvents
        End If
    Next
End Sub

Sub TerminateEvents()
    'distruggiamo la classe per liberare memoria
    Set testC = Nothing
End Sub
```

Il passo finale è quello di assicurarsi che il codice della routine **InitializeEvents** venga eseguito all'apertura della cartella, per cui useremo l'evento Workbook_Open nel modulo ThisWorkbook inserendo questo codice

```
Option Explicit

Private Sub Workbook_Open()
    InitializeEvents
End Sub
```

Infine, quando la cartella di lavoro viene chiusa o se si vuole fermare l'esecuzione della classe di rispondere agli eventi, è necessario distruggere la classe per liberare risorse con questo codice

```
Private Sub Class_Terminate()
    Set testC = Nothing
End Sub
```



Da notare che la routine **Class_Terminate** verrà eseguita per ciascuna istanza della classe creata con la routine **InitialiseEvents**. Ovviamente questa sub deve essere eseguita quando la cartella di lavoro si chiude, quindi nel modulo ThisWorkbook, aggiungere:

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    TerminateEvents
End Sub
```