



I MODULI DI CLASSE TUTORIAL



MS ACCESS





1. PREMESSA.....	3
2. COSA SONO LE CLASSI.....	3
3. AGGIUNGERE AL PROGETTO UN MODULO DI CLASSE	4
4. ISTANZIARE UNA CLASSE.....	4
5. PROPRIETA'	5
6. METODI	8
7. EVENTI	10
8. ESEMPI.....	12
9. CONCLUSIONI.....	13





1. PREMESSA

Questo breve articolo espone i concetti base per comprendere l'uso dei Moduli di Classe.

Sono richieste conoscenze di base per la comprensione dei concetti iniziali.

2. COSA SONO LE CLASSI

I Moduli di Classe sono sostanzialmente OGGETTI, contenitori con i quali è possibile crearsi una struttura personalizzata di Proprietà, Metodi ed Eventi.

Il vantaggio di questi oggetti è la riusabilità, vale a dire che possono essere considerati Oggetti standard dei quali è possibile creare molteplici copie in memoria (si chiamano ISTANZE) e per ognuna di esse è possibile ridefinirne Proprietà e sfruttarne Metodi ed Eventi.

Pensiamo all'Oggetto MOTOCICLETTA.

L'oggetto in sé è generico, si tratta di un veicolo a motore con 2 ruote.

Nello specifico però avremo HONDA, DUCATI ecc...

Cosa differenzia l'Oggetto MOTOCICLETTA(Honda) dall'Oggetto MOTOCICLETTA(Ducati)...

Le proprietà intrinseche del modello di cui la marca è già una di queste [Honda/Ducati...].

Detto questo se noi ci creiamo un Contenitore MOTOCICLETTA e gli associamo delle proprietà definibili, riusciremo, personalizzandole a differenziare l'Oggetto MOTOCICLETTA(Honda) da quello MOTOCICLETTA(Ducati).

In informatica il concetto di Oggetto è fondamentale da quando lo sviluppo è Programmazione ad OGGETTI.

Avendo definito un Oggetto generico e, avendolo strutturato con tutte le proprietà necessarie, non dovremo crearci

un oggetto per ogni Costruttore(Honda/Ducati...), ci basterà aprire 2(n) volte lo stesso oggetto e per ognuno di questi definire le proprietà in modo da differenziarli.

Abbiamo non solo risparmiato oggetti, ma abbiamo reso flessibile il nostro lavoro, poichè se dovessimo aggiungere [Yamaha] dovremo solo aprire un 3° oggetto...!





3. AGGIUNGERE AL PROGETTO UN MODULO DI CLASSE

La creazione vera e propria di una classe oggetti, avviene aggiungendo un nuovo modulo classe al proprio database, a tale scopo basta aprire l'editor di Visual Basic e selezionare dal menu Inserisci la voce Modulo di classe.

Ogni modulo di classe VBA incorpora la proprietà Name che è parte integrante della definizione di una classe oggetto, proprio in quanto determina il nome stesso della classe.

L'associazione di un nome alla classe, avviene specificando il nome della classe quando si salva per la prima volta il modulo, oppure specificandolo nella finestra delle proprietà di VBA.

4. ISTANZIARE UNA CLASSE

Per poter creare l'istanza di una nuova classe, bisogna dichiarare la variabile oggetto basata sulla classe che verrà utilizzata per la memorizzazione di un riferimento all'istanza di una nuova classe.

Questo verrà eseguito per semplicità da una Maschera:

```
Dim Moto As MOTOCICLETTA
Set Moto = New MOTOCICLETTA
```

Ora nella mia variabile [Moto] ho il riferimento all'istanza di MOTOCICLETTA. Per ora non ci posso fare nulla perchè non ho definito cosa questo Oggetto può fare.





5. PROPRIETA'

Come dicevamo una Classe necessita di poter essere personalizzata, questo si rende fattibile se la classe ha delle proprietà, ovvero attributi dell'Oggetto.

Per esempio usiamo sempre la nostra MOTOCICLETTA, avrà come proprietà:

- Marca
- Modello
- Cilindrata
- Diametro Cerchi
- Immatricolazione

Come facciamo a fare in modo di dotare la nostra classe di Proprietà...? Tramite le routine [Public\Private] Property Get e Property Let.

Con [Property Get] posso LEGGERE il valore della proprietà.

Con [Property Let] posso IMPOSTARE il valore della proprietà.

C'è anche una terza opzione [Property Set], che si usa quando la proprietà è un'oggetto.

```
CLASSE
-----
Option Compare Database
Option Explicit
' Definisco un Oggetto LABEL
Private m_Label           As Access.Label

' Questo è il modo per passare un riferimento ad un'OGGETTO
Public Property Set ControlLabel(Lb As Access.Label)
    Set btn = Lb
End Property
```





Inseriamo la Classe come spiegato e salviamola chiamandola MOTOCICLETTA

```
CLASSE
-----
Option Compare Database
Option Explicit

Private m_Marca           As String
Private m_Cilindrata      As Integer
Private m_Immatricolazione As Date

Public Property Get Marca () As String
    Marca= m_Marca
End Property

Public Property Let Marca (value As String)
    m_Marca = value
End Property

Public Property Get Immatricolazione () As Date
    Immatricolazione = m_Immatricolazione
End Property

Public Property Let Immatricolazione (value As Date)
    m_Immatricolazione = value
End Property

Public Property Get Cilindrata()As Integer
    Immatricolazione = m_Cilindrata
End Property

Public Property Let Cilindrata(value As Integer)
    m_Cilindrata= value
End Property
```

Ora dalla nostra Form, nella quale abbiamo inserito l'ISTANZA possiamo vedere lavorare l'INTELLISENSE sulle proprietà dell'oggetto:

```
FORM
-----
Option Compare Database
Option Explicit

Private Sub Prova()
    Dim Motol As MOTOCICLETTA
    Set Motol = New MOTOCICLETTA
    ' Scriviamo le proprietà dell'Oggetto Motol
    Motol.Marca="HONDA"
    Motol.Immatricolazione=#01/01/2007#
    Motol.Cilindrata=1000

    ' Leggiamo le proprietà appena scritte
    MsgBox "Marca = " & Motol.Marca & vbCrLf & _
        "Immatricolazione = " & Motol.Immatricolazione & vbCrLf & _
        "Cilindrata = " & Motol.Cilindrata
End Sub
```





Se avessimo 2 moto dovremmo fare così:

```
FORM
```

```
Option Compare Database
```

```
Option Explicit
```

```
Private Sub Prova()
```

```
    Dim Moto1 As MOTOCICLETTA
```

```
    Set Moto1 = New MOTOCICLETTA
```

```
    Dim Moto2 as MOTOCICLETTA
```

```
    Set Moto2 = New MOTOCICLETTA
```

```
    ' Scriviamo le proprietà dell'Oggetto Moto1
```

```
Moto1.Marca="HONDA"
```

```
Moto1.Immatricolazione=#01/01/2007#
```

```
Moto1.Cilindrata=1000
```

```
    ' Scriviamo le proprietà dell'Oggetto Moto2
```

```
Moto2.Marca="DUCATI"
```

```
Moto2.Immatricolazione=#01/05/2006#
```

```
Moto2.Cilindrata=990
```

```
End Sub
```





6. METODI

I metodi sono azioni che l'oggetto può compiere.

Per esempio, il nostro Oggetto MOTOCICLETTA potrà avere come metodo:

- ACCELERERA
- FRENA

I metodi sono espressamente rappresentati da Sub/Function, quindi nel caso di Function avremo la possibilità di ricevere come valore di ritorno una proprietà dell'oggetto, nel nostro caso, potrebbe benissimo restituire la Velocità istantanea.

Modifichiamo la Classe:

```
CLASSE
-----
Option Compare Database
Option Explicit

Private m_Speed           As Integer

Public Property Get Speed() As Integer
    Speed= m_Speed
End Property

Public Property Let Speed(value As Integer)
    m_Speed = value
End Property

Public Function Accelera() As Integer
    m_Speed=m_Speed+1
    ' eseguiremo l'azione di accelerazione
    ' .....
    Accelera=m_speed
End Function

Public Function Frena() As Integer
    m_Speed=m_Speed-1
    ' eseguiremo l'azione di Frenata
    ' .....
    Accelera=m_speed
End Function
```





Stessa cosa avremo per il metodo FRENA.

Dalla nostra Form avremo:

```
FORM
```

```
Option Compare Database
```

```
Option Explicit
```

```
Private Sub Prova()
```

```
    Dim Moto1 As MOTOCICLETTA
```

```
    Set Moto1 = New MOTOCICLETTA
```

```
    ' Scriviamo le proprietà dell'Oggetto Moto1
```

```
    Moto1.Marca="HONDA"
```

```
    Moto1.Immatricolazione=#01/01/2007#
```

```
    Moto1.Cilindrata=1000
```

```
    ' Ora eseguiamo l'azione e stampiamo il risultato
```

```
    Debug.Print "Velocità = " & Moto1.Accelera
```

```
End Sub
```





7. EVENTI

I moduli di classe, definiscono solo due eventi, Initialize e Terminate, rispettivamente associati alla creazione e alla distruzione dell'istanza di una classe.

```
-----  
CLASSE  
-----  
Option Compare Database  
Option Explicit  
  
Private Sub Class_Initialize()  
' Codice di Inizializzazione  
' Normalmente si caricano i valori di DEFAULT  
End Sub  
  
Private Sub Class_Terminate()  
' Codice di distruzione  
' Normalmente si cancellano gli Oggetti interni  
End Sub
```

Ricordiamo che la Creazione ha luogo solo con la definizione di NEW [nomeClasse], e la distruzione quando impostiamo a Nothing l'oggetto:

```
Dim Moto1 As MOTOCICLETTA  
' Ora ha origine l'EVENTO Initialize  
Set Moto1 = New MOTOCICLETTA
```

Un concetto utile da ricordare è che tutti gli OGGETTI che si istanziano è bene DISTRUGGERLI per liberare area di lavoro.

Le classi possono esporre, oltre che i 2 EVENTI nativi descritti anche EVENTI personalizzati.

Pur non essendo esattamente una parte del modello di programmazione degli oggetti, gli eventi sono un utile strumento che permette a un oggetto di scambiare informazioni con il programma che ha creato l'oggetto.

In altri termini, un evento è una subroutine residente nel programma che ha creato l'oggetto, ed è chiamata dall'oggetto.

Un evento è una tecnica utile che elimina la necessità del programma di monitorare costantemente le modifiche di un oggetto.

In questo modo, l'oggetto chiama l'evento per informare il programma di una modifica allo stato dell'oggetto.





Modifichiamo la Classe:

```
CLASSE
-----
Option Compare Database
Option Explicit

Public Event Accelerazione(ByVal Speed As Integer)

Private m_Speed           As Integer

Public Property Get Speed() As Integer
    Speed= m_Speed
End Property

Public Property Let Speed(value As Integer)
    m_Speed = value
End Property

Public Function Accelera() As Integer
    m_Speed=m_Speed+1
    ' eseguiremo l'azione di accelerazione
    ' .....
    RaiseEvent Accelerazione(m_Speed)
    Accelera=m_speed
End Function
```

Come facciamo dalla nostra Maschera ad ereditare gli eventi personalizzati...?

Innanzitutto l'oggetto dovrà essere dichiarato a livello di Modulo nella Form e non di Routine, quindi inseriamo un CommandButton e su Evento Click richiamiamo il METODO Accelera.....:

```
FORM
-----
Option Compare Database
Option Explicit

Private WithEvents Moto1 As MOTOCICLETTA

' Ora magari su EVENTO LOAD generiamo l'ISTANZA
Private Sub Form_Load()
    Set Moto1 = New MOTOCICLETTA
End Sub

Private Sub cmdAccelera_Click()
    ' Appena chiamato il METODO verrà scatenato l'EVENTO
    ' e riceveremo il MSGBOX
    Moto1.Accelera
End Sub

Private Sub Moto1_Accelerazione(ByVal Speed As Integer)
    MsgBox "Attento stai andando a " & speed & " Km/h"
End Sub
```





8. ESEMPI

Per capire meglio il concetto facciamo un Esempio semplice.
Reallizziamo una Classe che, passando come riferimento una LABEL mi gestisca l'effetto RILIEVO al passaggio del MOUSE.
Ovviamente costruiamo la Classe in modo che sia usabile da N Label contemporaneamente.

Creiamo un NUOVO MODULO DI CLASSE chiamiamolo "clsLABEL"

```
CLASSE
-----
Option Compare Database
Option Explicit

' Tramite WithEvents eredito anche gli EVENTI dell'OGGETTO
Private WithEvents btn As Access.Label

Private Sub btn_MouseMove(Button As Integer, _
                          Shift As Integer, _
                          X As Single, _
                          Y As Single)

    btn.SpecialEffect = 1
    btn.FontWeight = 700
End Sub

Public Property Set ControlLabel(Lb As Label)
    Set btn = Lb
    ' Siccome ho ereditato gli EVENTI devo ABILITARE quello che
    ' vogliamo usare: MOUSEMOVE
    btn.OnMouseMove = "[Event Procedure]"
End Property

Property Get ControlLabel() As Access.Label
    Set ControlLabel = btn
End Property
```





Ora nella nostra FORM creiamo 2 LABEL chiamate `Label1` e `Label2`.
Su Evento LOAD della Form creiamo le 2 ISTANZE.

```
FORM
-----
Option Compare Database
Option Explicit

' Tramite WithEvents eredito anche gli EVENTI dell'OGGETTO

Private mc1 As clsLabelEffect
Private mc2 As clsLabelEffect

' Sfrutto l'Evento MOUSEMOVE del corpo per RESETTARE l'effetto
' sulle LABEL
Private Sub Corpo_MouseMove(Button As Integer, _
                             Shift As Integer, _
                             X As Single, _
                             Y As Single)

    ' Ripristino l'aspetto originale
    mc1.ControlLabel.SpecialEffect = 0
    mc1.ControlLabel.FontWeight = 400

    mc2.ControlLabel.SpecialEffect = 0
    mc2.ControlLabel.FontWeight = 400
End Sub

Private Sub Form_Load()
    ' Creo l'ISTANZA delle 2 Classi
    Set mc1 = New clsLabelEffect
    Set mc2 = New clsLabelEffect
    Set mc1.ControlLabel = Me!Label1
    Set mc2.ControlLabel = Me!Label2
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set mc1 = Nothing
    Set mc2 = Nothing
End Sub
```

Salviamo la FORM ed apriamola provando a passare con il MOUSE sulle LABEL. Chiaramente questo è un esempio molto semplice e banale, ma pensando alla flessibilità della tecnica probabilmente ora sarà più facile intuirne i vantaggi.

9. CONCLUSIONI

E' da sottolineare che un MODULO DI CLASSE è utile solo se può essere reale il suo utilizzo in MULTIISTANZA, altrimenti si perde del tempo.

Allego 2 riferimenti a 2 DEMO:

<http://www.alessandrobaraldi.it/DettaglioFaq.asp?IdFAQ=157>

<http://www.alessandrobaraldi.it/DettaglioFaq.asp?IdFAQ=156>

