

 Scelta Multipla

Select case rappresenta una versione migliorata di un normale If... Then nei casi in cui:

- occorre valutare in sequenza un numero elevato di condizioni (tante clausole Elseif se si usasse l'If normale), e
- tutte le condizioni sono da valutare rispetto alla stessa variabile (o espressione di variabili).

Select funziona dunque leggendo una prima (ed unica) volta il valore di una variabile e confrontandolo in sequenza con i valori dei vari Case, fin tanto che non ne trova uno uguale, cioè in cui il confronto restituisca Vero. La sintassi è dunque la seguente:

```
Select Case EspressioneTest  
    [Case Espressione1  
        [Blocco istruzioni-1 ]  
    [Case Espressione2  
        [Blocco istruzioni -2 ]  
    [Case Else  
        [Blocco istruzioni -n ]  
End Select
```

dove:

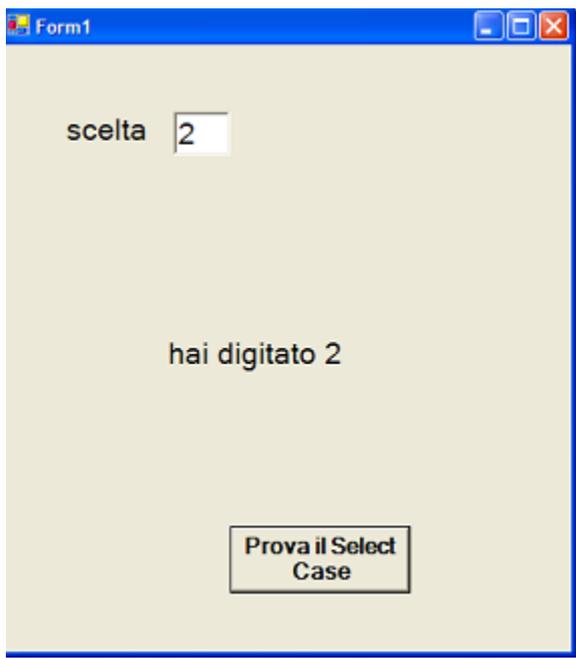
- **EspressioneTest o variabile di controllo** è l'espressione da valutare.
- **Espressione1 o valori1**, ecc. sono i valori con cui si confronta il risultato di EspressioneTest.
Possiamo indicare un solo valore, più valori separati da virgola, un intervallo di valori (se sono lettere, vale l'ordine alfabetico) separati dalla parola chiave To, e anche un valore che viene confrontato non con l'uguale, ma con un maggiore, un minore o un diverso, mediante la parola chiave Is: ad esempio Is >= 3 (Is è aggiunto automaticamente da Visual Basic ogni volta che uso >, <, >=, <=, <> [diverso])
- **BloccoIstruzioni-1**, ecc. è quello che viene eseguito se la corrispondente Espressione1 è verificata.
- L'ultimo **Case Else** (facoltativo) corrisponde a quello che si deve fare se nessuno dei Case precedenti è stato verificato.

Se volessi ottenere un risultato analogo con una struttura If dovrei ricorrere a tanti Elseif, complicati dal fatto che non potrei usare la virgola per separare argomenti (dovrei usare degli Or) né il To per indicare intervalli (dovrei trasformarli in due condizioni unite da un And: If Lettera >= "A" And Lettera <= "Z"). E' importante

ribadire però che **Select** funziona perché in tutti i confronti, quello che devo confrontare è sempre la stessa variabile: se così non fosse il ricorso a If sarebbe necessario.

Notare infine che se EspressioneTest soddisfa più di una Espressione, solo il BloccoIstruzioni corrispondente alla prima soddisfatta viene eseguito.

Vi mostro un banale esempio in cui viene usata l'istruzione Select Case per mostrare in output il la scelta digitata dall'utente.



```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Codice generato da Progettazione Windows Form

    Private Sub Button1_Click(ByVal sender As System.Object)
        Dim scelta As Integer
        scelta = Val(txtscelta.Text)
        Select Case scelta
            Case 1
                lblrisultato.Text = "hai digitato 1"
            Case 2
                lblrisultato.Text = "hai digitato 2"

            Case 3
                lblrisultato.Text = "hai digitato 3"
        End Select
    End Sub
End Class
```