

 Istruzione IF

Raramente i programmi prevedono la sola esecuzione sequenziale di istruzioni. Per di più i programmi devono eseguire operazioni diverse in risposta a condizioni diverse. Ad esempio, un programma potrebbe dover verificare il giorno della settimana ed eseguire un'operazione diversa in base al giorno. L'istruzione **If...Then** consente di valutare una condizione e di eseguire sezioni di codice diverse in base ai risultati di tale condizione.

Vi sono due forme differenti: la più semplice è l'istruzione singola, su unica linea, **If - Then**, che viene scritta come segue:

If espressione logica Then istruzione eseguibile

L'istruzione eseguibile andrà in esecuzione solo se l'espressione logica è vera, altrimenti si passerà alla successiva istruzione del codice del programma; se l'istruzione eseguibile non appare sulla medesima linea dell'espressione logica, è necessaria un'istruzione **End If**

Esempio 1

If x < 0 then y =0

Esempio 2

If reddito <= 14000 then
Tasse = 0.2*stipendio
Netto = stipendio - tasse
End If

Un blocco **If - Then -Else** invece consente di eseguire uno tra due diversi blocchi di istruzioni, a seconda del risultato del test logico. Esso permette una forma di ramificazione più ampia di quella che si ottiene con un singolo blocco If - Then:

If espressione logica Then

.....
istruzioni eseguibili

.....
Else

.....
istruzioni eseguibili

.....
End If

Se l'espressione logica è vera, viene eseguito il primo gruppo di istruzioni, altrimenti viene eseguito il secondo. Perciò viene eseguito solo uno dei due gruppi di istruzioni. In forma più generale, il blocco if – Then – Else può essere scritto come:

If espressione logica1 then

.....
istruzioni eseguibili

.....
Elseif espressione logica2 then

.....
istruzioni eseguibili

.....
Elseif espressione logica i-esima then

.....
istruzioni eseguibili

.....
Else

.....
istruzioni eseguibili

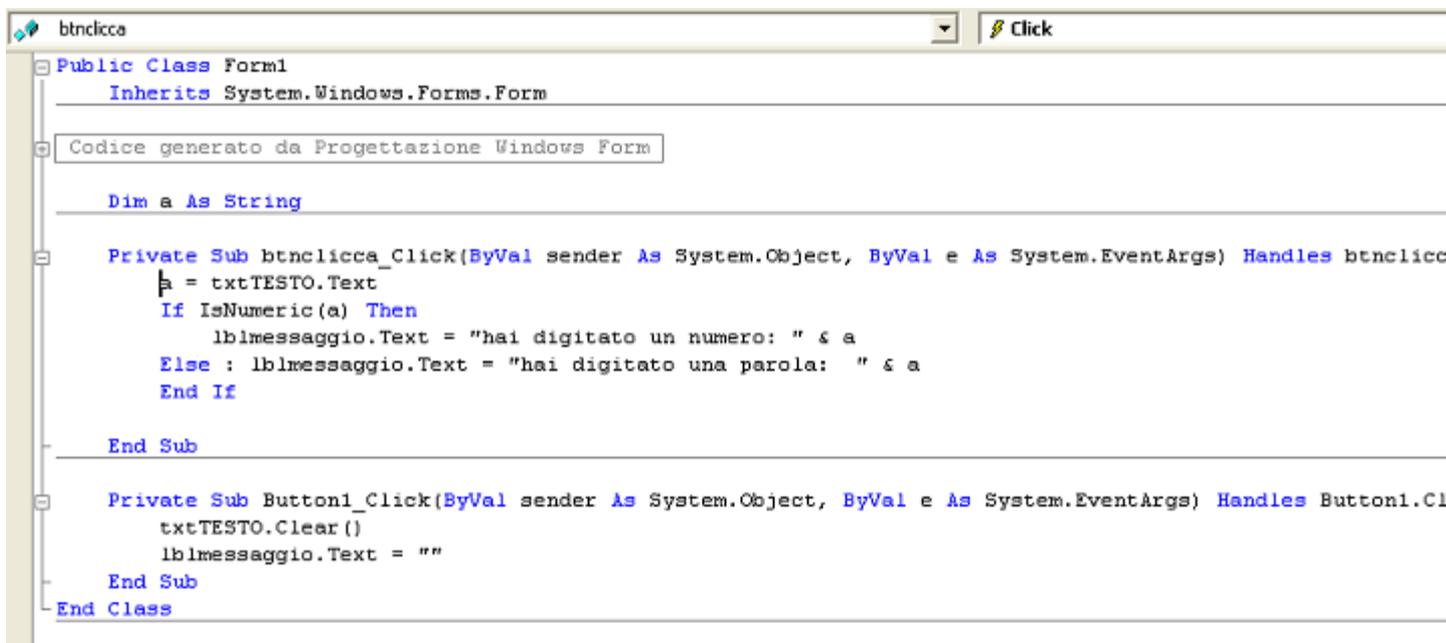
.....
End If

Ora vi propongo un esercizio svolto in cui ho utilizzato l'istruzione condizionale if ...then ...else e una funzione predefinita di Visual Basic, **IsNumeric()** che restituisce VERO se la variabile passata come argomento è un numero e restituisce falso in caso contrario.

Questa interfaccia che prevede per l'utente uno spazio per digitare quello che vuole, un numero, un testo o un numero, un bottone per avviare il codice (CLICCA) una label per visualizzare il messaggio in

risposta all'utente, e un altro bottone che permetta di riazzere le proprietà degli oggetti per poter proporre nuovi input da elaborare.

Il codice necessario è questo, potete poi ricopiarlo per provare voi stessi:



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

    Codice generato da Progettazione Windows Form

    Dim a As String

    Private Sub btnclicca_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclicca.Click
        a = txtTESTO.Text
        If IsNumeric(a) Then
            lblmessaggio.Text = "hai digitato un numero: " & a
        Else : lblmessaggio.Text = "hai digitato una parola: " & a
        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        txtTESTO.Clear()
        lblmessaggio.Text = ""
    End Sub
End Class
```

Voglio farvi notare inoltre l'utilizzo della "&" nell'assegnazione del contenuto alla proprietà text della label, questa operazione si chiama **"concatenazione"** ovvero si può attaccare ad una stringa il valore di una variabile: questa tecnica risulta essere molto utile per comunicare gli output all'utente.