22/10/2013

Informatica per la Finanza

Informatica per le Scienze Economiche

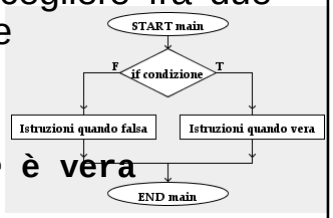
Informatica e Programmazione per il Turismo

Istruzioni di controllo del flusso

- Nei linguaggi di programmazione tradizionali (come VBA, C/C++, Pascal, …) l'ordine di esecuzione è strettamente sequenziale.
- Questo modo di eseguire i programmi non è sufficiente per gestire tutte le necessità.
  - Esistono dei modi per variare questo ordine:
    - **Istruzioni condizionali**
    - **Istruzioni cicliche**
    - **Richiamo di sottoprogrammi/funzioni**
    - istruzioni di salto (in disuso e non raccomandabile)

Istruzione di selezione

- Ogni linguaggio di programmazione possiede un'istruzione che permette di scegliere fra due strade in base ad una condizione
- L'istruzione VBA è:




```
If condizione Then
  istruzioni quando condizione è vera
Else
 istruzioni quando condizione è falsa
End If
```


Istruzione di selezione

Nella forma più generale l'istruzione è:

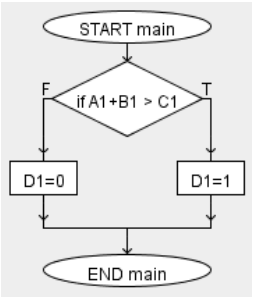
```
If condizione Then
  istruzioni
ElseIf condizione Then
  istruzioni
Else
  istruzioni
End If
```

Istruzione di selezione

- I rami ***ElseIf*** ed ***Else*** sono facoltativi.
- Ci possono essere più rami ***ElseIf*** ma <u>un sol ramo</u> ***Else***
  - Ciascun ramo può valutare una condizione diversa
  - Per evitare l'uso di molti rami ***ElseIf*** in certi casi è preferibile usare l'istruzione ***Select Case***
- Le istruzioni ***If*** possono essere nidificate

Esempio

```
Option Explicit
'scrivere nella cella D1 il valore 1 se la somma
'dei valori delle celle A1 e B1 e' maggiore del
'valore in C1 altrimenti scrivere in D1 0
Sub prv()
  If Range("A1") + Range("B1") > Range("C1") Then
    Range("D1") = 1
  Else
    Range("D1") = 0
  End If
End Sub
```



Esempio

```
' Se il valore in A1 è maggiore di quello in B1
' scrivere in C1 la somma altrimenti la differenza

Private Sub prova ()
    If Range("A1") > Range("B1") Then
      Range("c1") = Range("A1") + Range("B1")
    Else
      Range("c1") = Range("A1") - Range("B1")
    End If
End Sub
```


Esempio

- Il codice che segue viene mandato in esecuzione per stabilire il contenuto della cella D3:

```
Private Sub commento()
   If Range("D3") > 0 Then
      Range("F3").Value = "POSITIVO"
   Else
      If Range("D3") < 0 Then
         Range("F3") = "NEGATIVO"
      Else
         Range("F3") = "Nullo"
      End If
   End If
End Sub
```

Funzione Debug

– Anche VBA mette a disposizione lo strumento Debug per tracciare passo passo l'esecuzione di un programma

```
'leggere nella variabile x un valore cc
'e quindi scrive nella variabile y  il
Option Explicit
Sub prv()
  Dim y As Integer, x As Double, s As I

  x = InputBox("dammi un valore: ")
  y = CInt(x)

  If y <> x Then ' x non intero
    If y - x < 0.5 And y - x > 0 Then '
      y = y - 1
    End If
  End If

  MsgBox (y)
End Sub
```

- Punti in cui si ferma l'esecuzione
- Si ottengono portando il cursore nella posizione e schiacciando il tasto sinistro del mouse.
- L'esecuzione si ferma su questi punti;
  - l'utente deve schiacciare il tasto Esegui per avanzare

Funzione Debug

– Per ispezionare il valore delle variabili:
  - Si utilizza il menù **Debug** selezionando la voce
    Aggiungi Espressione di Controllo
    – In questo modo si aggiunge il nome della variabili di cui si vuole controllare il valore
    – Serve anche per verificare il valore di una espressione
  - Si pone il cursore sulla variabile e si vede il valore

– In ogni caso è mostrato il valore della variabile/espressione PRIMA dell'esecuzione dell'istruzione

Esempio

```
'leggere nella variabile x un valore con una Input Box
'e quindi scrive nella variabile y  il valore troncato
Option Explicit
Sub prv()
  Dim y As Integer, x As Double, s As Integer

  x = InputBox("dammi un valore: ")
  y = CInt(x)

  If y <> x Then ' x non intero
    If y - x < 0.5 And y - x > 0 Then ' arrotondato per eccesso
      y = y - 1
    End If
  End If

  MsgBox (y)
End Sub
```


Esempio

- verificare se il valore nella cella A2 è di tipo numerico e se è intero verificare se è pari.
  - Per verificare se il valore dato è un numero intero si usa la funzione di libreria **`CInt()`** se il valore restituito è uguale al suo argomento allora questo rappresenta un numero intero

Esempio

```
Option Explicit
Sub prv()
  If IsNumeric(Range("A2")) Then 'verifico se è numerico
     If CInt(Range("A2")) = Range("A2") Then 'è intero
       If Range("A2") Mod 2 = 0 Then 'è pari
          MsgBox ("il numero " & Range("A2").Value & " è pari")
       Else
          MsgBox ("il numero " & Range("A2").Value & " è dispari")
       End If
     Else
       MsgBox ("il numero " & Range("A2").Value & " non è intero")
     End If
  Else
    MsgBox (Range("A2").Value & " non sembra un numero")
  End If
End Sub
```


Esempio

Dato un valore nella cella A1 scrivere in B1

- il valore della somma di tutti i numeri compresi fra 1 se il valore in A1 è numerico, intero e positivo
- -1 se il valore in A1 non è numerico
- 0 se è numerico ma non positivo o intero

Esempio

```
If Verifica se A1 contiene un numero Then
   If Verifica se A1 è intero Then
     If Verifica se A1 è positivo Then
        Calcola la somma con la regola di Gauss:
              somma = N*(N+1)/2
        dove N è un numero intero positivo
     Else
       scrive 0 in B1
     End If
   Else
     scrive 0 in B1
   End If
 Else
     scrive -1 in B1
 End If
```


Esempio

```
Option Explicit
' Dato un valore nella cella A1 scrivere nella cella B1
' - il valore della somma di tutti i numeri compresi fra 1
'   se il valore in A1 è numerico, intero e positivo
' - -1 se il valore in A1 non è numerico
' - 0 se è negativo o non intero
Sub prv()
  If IsNumeric(Range("A1")) Then
    If Range("A1") = CInt(Range("A1")) Then
      If Range("A1") > 0 Then
        Range("B1") = Range("A1") * (Range("A1") + 1) / 2
      Else
        Range("B1") = 0
      End If
    Else
      Range("B1") = 0
    End If
  Else
      Range("B1") = -1
  End If
End Sub
```

Esempio

```
Option Explicit
Sub prv()
  If IsNumeric(Range("A1")) Then
    If Range("A1") = CInt(Range("A1")) And Range("A1") > 0
   Then
      Range("B1") = Range("A1") * (Range("A1") + 1) / 2
    Else
      Range("B1") = 0
    End If
  Else
      Range("B1") = -1
  End If
End Sub
```

Si verificano assieme le due condizioni

Esempio

- Il codice che segue calcola quoziente e resto della divisione fra il contenuto di A1 con B1. Se B1 vale 0 scrive un messaggio di errore:

```
Private Sub prova()
 If Range("B1") = 0 Then
   Range("C1") = "Impossibile"
 Else
   Range("C1") = Range("A1") / Range("B1")
   Range("D1") = Range("A1") Mod Range("B1")
 End If
End Sub
```

Esempio

- Il codice che segue calcola il valore assoluto del numero in D3 e lo scrive in F3

```
Private Sub ValoreAssoluto ()
  If Range("D3").Value < 0 Then
     Range(“F3").Value = -Range("D3")
  Else
     Range(“F3").Value = Range("D3")
  End If
End Sub
```


Esempio

- dato il valore nella cella A1 (imponibile)
- a seconda del valore nella A2 applicare l’aliquota IVA opportuna
  - nella cella A2 i valori sono

    0 (0%)
    1 (4%)
    2 (10%)
    3 (22%)

Esempio

```
Option Explicit

Private Sub IVA_Click()
   If Range("A2") = 1 Then 'aliquota 4%
     Range("A3") = Range("A1") * 0.04
   ElseIf Range("A2") = 2 Then 'aliquota 10
     Range("A3") = Range("A1") * 0.1
   ElseIf Range("A2") = 3 Then 'aliquota 22
     Range("A3") = Range("A1") * 0.22
   Else
     Range("A3") = 0
   End If
End Sub
```


Esempio

- Il codice che segue presume che nelle celle A5, B5 e C5 ci siano i coefficienti di una equazione di secondo grado. Nella cella D5, in base al valore di delta, scrive il numero di radici dell'equazione (si usa un bottone per attivare l'elaborazione)

Esempio

```
Private Sub Eq2Grado ()
  Dim Delta As Double
     Delta = Range("B5") ^ 2
     Delta = Delta - 4 * Range("A5") * Range("C5")
     If Delta > 0 Then
        Range("D5") = "2 radici reali"
     Else
         If Delta = 0 Then
            Range("D5") = "radici reali coincidenti"
         Else
             Range("D5") = "coniugate complesse"
         End If
      End If
End Sub
```


Esempio

- Data una equazione di secondo grado stabilire usando la regola dei segni di Cartesio il segno delle radici reali
  - il cambiamento di segno fra due coefficienti consecutivi produce una radice positiva, la permanenza del segno una radice negativa

Esempio

- $5x^2 – 6x + 1$ varia varia 2 positive
- $6x^2+3x – 1$ perm varia 1 pos; 1 neg
- X^2+3x+1 perm perm 2 negative
- **+ + + 2 negative**
- **- - - 2 negative**
- **+ - + 2 positive**
- **- + - 2 positive**
- **- + + 1 positiva 1 negativa**

Esempio

- Idea di soluzione metto in tre variabili 1 se il segno del coeff è positivo 0 se è negativo
- s1 s2 s3 variaz
- $5x^2 – 6x + 1$ 1 0 1 2
- $6x^2+3x – 1$ 1 1 0 1
- X^2+3x+1 1 1 1 0

```
Sub cartesio()
   Dim s1 As Integer, s2 As Integer, s3 As Integer, variaz As Integer

   If Range("A1") > 0 Then
     s1 = 1
   Else
     s1 = 0
   End If

   If Range("B1") > 0 Then
     s2 = 1
   Else
     s2 = 0
   End If

   If Range("C1") > 0 Then
     s3 = 1
   Else
     s3 = 0
   End If

   variaz = 0
   If s1 <> s2 Then
      variaz = variaz + 1
   End If

   If s2 <> s3 Then
      variaz = variaz + 1
   End If

   Range("E1") = variaz
   Range("E2") = "rad. positive"
   Range("F1") = 2 - variaz
   Range("F2") = "rad. negative"
End Sub
```


Esempio

- Un’altra idea risolutiva si basa sul moltiplicare le coppie di coefficienti adiacenti e di verificare il segno del prodotto:
  - Negativo: variazione
  - Positivo: permanenza
- Il codice che si ottiene è più semplice

Esempio

```
'Si basa sull'idea di moltiplicare fra loro due coefficienti
‘ consecuti e di verificare il segno del prodotto:
'  negativo: variazione
'  positivo: permanenza
Private Sub cartesio_Click()
    Dim x1 As Integer, x2 As Integer
    Dim pos As Integer

    x1 = Range("A5") * Range("B5")
    x2 = Range("B5") * Range("C5")

    If x1 < 0 Then
      pos = pos + 1
    End If
    If x2 < 0 Then
      pos = pos + 1
    End If

    Range("D5") = "ci sono " & pos & " radici positive"
    Range("D6") = "ci sono " & (2 - pos) & " radici negative"
End Sub
```



```
Option Explicit

Private Sub Radici_Click()
   Dim a As Double, b As Double
   Dim c As Double, delta As Double
   Dim permanenza As Integer

   a = Range("A5")
   b = Range("B5")
   c = Range("C5")

   delta = b ^ 2 - 4 * a * c

   If delta < 0 Then
     Range("D5") = "coniugate complesse"
   ElseIf delta = 0 Then
       Range("D5") = "coincidenti"
   Else
       Range("D5") = "2 distinte"
       'segno delle radici
       permanenza = 0
       If a * b > 0 Then
         permanenza = permanenza + 1
       End If
       If b * c > 0 Then
         permanenza = permanenza + 1
       End If

       If permanenza = 2 Then
         Range("F5") = "2 radici negative"
       ElseIf permanenza = 0 Then
         Range("F5") = "2 radici positive"
       Else
         Range("F5") = "1 radice positiva ed 1 negativa"
       End If
   End If
End Sub
```

Caso completo si verifica
anche l'esistenza delle radici reali
(i numeri complessi non hanno segno)

Istruzioni condizionali

```
Select Case espressione
Case espressioneConfronto
   Istruzioni
CaseElse
   Istruzioni
End Select
```


Istruzioni condizionali

- Le parti **`Case`** e **`CaseElse`** sono facoltative
- Ci possono essere più rami **`Case`** ma di **`Case Else`** ve ne può essere uno solo
- L'istruzione **<u>valuta una sola volta</u>** ***`espressione`*** e quindi valuta questo risultato con ogni ***`espressioneConfronto`*** fino a trovare la prima per la quale è vera
  - Al primo caso in cui **`Case`** è vero termina l'istruzione
    - Si comporta in modo differente dal C/C++
  - Il ramo **`Case Else`** si utilizza per dare un risultato quando nessuna delle condizioni si è verificata
  - Può sostituire solo parzialmente **`If`** in quanto ***`espressione`*** è valutata solo una volta

Istruzioni condizionali

- ***espressioneConfronto*** può essere
  - Un solo valore
  - Più valori separati da virgola
  - Un intervallo di valori (*inizio* **To** *fine*)
  - Operatori di confronto:
    - **IS operatoreConfronto** *valore*
      - » Operatori confronto: **>, <, >=, <=, =, <>**

Istruzioni condizionali

```
Select Case par
  Case Is < 0
      usoSelect = "negativo"
  Case 0
      usoSelect = "ZERO“
  Case 1, 2, 3, 4, 5, 6, 7, 8, 9
      usoSelect = " una cifra"
  Case 10 To 99
       usoSelect = "due cifre"
  Case Else
   usoSelect = "Più di due cifre"
End Select
```

```
If Not IsNumeric(par) Then
  Messaggi = "non è un numero"
ElseIf par < 0 Then
  Messaggi = "Negativo"
ElseIf par <= 9 Then
   Messaggi = "Una Cifra"
ElseIf par < 100 Then
   Messaggi = "Due cifre"
Else
  Messaggi = "piu' di 2 cifre"
End If
```

Istruzioni condizionali

- L'esempio con **`Select`** non rispecchia l'esempio analogo fatto con l'istruzione **`If`**
- Il primo ramo **`If`**

```
If Not IsNumeric(par) Then
     Messaggi = "non è un numero"
```

  - Non può essere reso con **`Select Case`** perché usa una espressione condizionale diversa da quella usata negli altri rami
    - Occorre usare un **`If`** con un **`Select Case`** in un ramo

Istruzioni condizionali

```
If Not IsNumeric(par) Then
     usoSelectEqv = "non è un numero"
Else
    Select Case par
      Case Is < 0
        usoSelectEqv = "negativo"
      Case 0
        usoSelectEqv = "ZERO"
      Case 1, 2, 3, 4, 5, 6, 7, 8, 9
        usoSelectEqv = "cifra"
      Case 10 To 99
         usoSelectEqv = "due cifre"
      Case Else
         usoSelectEqv = "Più di due cifre"
    End Select
End If
```