

Le variabili

Anche Visual Basic, come tutti i linguaggi di programmazione, prevede l'uso delle **variabili**, queste sono dei nomi, che noi definiamo, in grado di memorizzare dei valori (ad esempio dei numeri o dei testi).

Nei nostri primi due esempi non abbiamo usato variabili, perché non era strettamente indispensabile. Tuttavia possiamo notare che abbiamo utilizzato la proprietà Text della label risultato o la proprietà text del controllo TxtPrimo. Può essere allora più elegante (in questo caso, in altri è invece non solo elegante, ma utile o indispensabile), memorizzare il testo scritto dall'utente in una variabile che potrà essere di tipo **stringa** se capace di accogliere testi (ma anche numeri trattati come fossero testi – notare che una variabile di tipo numerico non saprebbe fare viceversa, cioè accogliere testi, che non possono essere trattati come fossero numeri), o diversamente di tipo **integer** o **single** se conterrà valori numerici. Mediante le variabili è possibile memorizzare dei valori testuali o numerici in strutture a cui il programma può accedere grazie a un nome assegnato loro in fase di creazione.

Una variabile è detta **locale** quando è definita all'interno di una procedura; la sua creazione avviene quando si fa riferimento ad essa per la prima volta, oppure quando è eseguita l'istruzione **Dim**, che presenta la seguente sintassi:

Dim <nome> [As <tipo>]

in cui <nome> rappresenta il nome da assegnare alla variabile.

Esso non si sottrae alla regola valida per tutti gli identificatori, ivi compresi i nomi degli oggetti, che impone loro di essere costituiti da delle sequenze di caratteri alfabetici o numerici prive di spazi ed aventi per iniziali delle lettere dell'alfabeto.

È inoltre possibile, anche se non indispensabile (come sottolineato dalla presenza delle parentesi quadre), aggiungere all'istruzione **Dim** un'indicazione del tipo di dato da creare. Visual Basic prevede numerosi tipi standard. I più utilizzati, come vi ho anticipato, sono **integer**, usato per rappresentare i dati numerici interi compresi fra -32,768 a 32,767, **string**, che può contenere delle sequenze (stringhe) alfanumeriche composte al massimo da circa 65500 caratteri, **long**, in grado di ospitare numeri interi compresi fra -2,147,483,648 e 2,147,483,647, **single** e **double**, con cui è possibile memorizzare numeri reali anche di notevole entità.

Se invece è omessa la dichiarazione del tipo, la variabile è definita **variant**. Questo formato non è previsto dalla maggior parte dei linguaggi di programmazione tradizionali. La sua caratteristica fondamentale è l'universalità. Una variabile **variant**, infatti, può contenere dei dati aventi qualsiasi formato.

È buona norma, tuttavia, non fare largo uso di strutture di questo tipo, in quanto la loro gestione da parte dell'interprete è poco efficiente.

Una variabile locale può anche non essere dichiarata. In questo caso, la sua creazione avviene la prima volta in cui si fa riferimento ad essa.

In assenza di dichiarazione, è necessario aggiungere al nome un identificatore di tipo, costituito da un carattere. In caso di sua omissione, la variabile è creata di tipo *Variant*. I principali identificatori di tipo sono i seguenti: \$ (*String*), % (*Integer*), & (*Long*), ! (*Single*), # (*Double*).

Ad esempio, è possibile creare una variabile di tipo numerico intero ed assegnarle il valore 5 sia scrivendo

```
Dim A as Integer
```

```
A = 5
```

sia con la riga

```
A% = 5
```

Ora vediamo come modificare uno dei nostri esercizi con l'utilizzo delle variabili. Prendiamo ad esempio questo:

```
Private Sub btnCALCOLAx_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCALCOLAx.Click
```

```
Dim a, b, c as integer
```

```
Dim x as single
```

```
a = Val(txta.Text)
```

```
b = Val(txtb.Text)
```

```
c = Val(txtc.Text)
```

```
x = (c - b) / a
```

```
lblx.Text = x
```

```
End Sub
```

Form1

Realizzare un progetto che date a,b,c di un'equazione ax+b=c calcoli il valore di x.

x + =

CALCOLA il valore di X

x =

Sarà quindi sempre necessario, disegnare l'interfaccia, definire le proprietà dei controlli, dichiarare le variabili, assegnare alle variabili di input il contenuto della proprietà text delle text box utilizzando la funzione VAL, eseguire le operazioni/istruzioni richieste dal testo, assegnare il valore della variabile alla proprietà text del controllo label che abbiamo predisposto in fase di progettazione come contenitore dell'output.