Caratteristiche generali del linguaggio Visual Basic

Per ottenere un aiuto contestuale dall'help di Visual Basic sulla sintassi di funzioni o istruzioni, oppure su proprietà, eventi o metodi, basta selezionare la parola chiave e premere il tasto **F1**.

Gli operatori possono essere di tre tipi: aritmetici, di relazione e logici.
Gli **operatori aritmetici** sono: **+** per l'addizione, **–** per la sottrazione, ***** per la moltiplicazione, **/** per la divisione con quoziente decimale, **** per la divisione tra numeri interi e per ottenere il quoziente intero, **MOD** per il calcolo del resto della divisione tra interi, **^** per l'elevamento a potenza.

Per esempio:

- 6 \ 4 = 1; 7 \ 3 = 2; 2 \ 3 = 0.
- 6 MOD 4 = 2; 7 MOD 3 = 1; 2 MOD 3 = 2;
- dato un numero N intero qualsiasi, N è dispari se N MOD 2 = 1, pari se N MOD 2 = 0.

Gli **operatori di relazione** sono utilizzati per confrontare il contenuto di due variabili e sono indicati con i simboli: **<** minore di, **<=** minore o uguale di, **>** maggiore di, **>=** maggiore o uguale di, **<>** diverso.
Gli **operatori logici** sono: **AND** per il prodotto logico (congiunzione), **OR** per la somma logica (disgiunzione), **NOT** per la negazione, **XOR** per l'OR esclusivo.

I dati utilizzati all'interno di un programma possono essere:
- **costanti**, se non cambiano il loro valore durante l'esecuzione del programma
- **variabili**, se cambiano il valore.

Le **costanti** utilizzate nel programma vengono precedute dalla parola **CONST**, secondo frasi del tipo

CONST Nome = espressione

Per esempio:

```
CONST PiGreco = 3.14
CONST Risposta = "SI"
```

Se la costante contiene caratteri il valore della costante viene racchiuso tra virgolette, per costanti di tipo numerico la separazione tra cifre intere e decimali è indicata con il carattere . (punto).

La dichiarazione delle **variabili** utilizzate nel programma inizia con la parola **DIM**

DIM Nome **AS** tipo

I nomi delle variabili devono iniziare con una lettera e possono contenere numeri e lettere fino a un massimo di 40 caratteri.

I dati trattati in un programma possono essere:
- **numerici**, quali età, importi, stipendi, misure;
- **alfanumerici** (o **stringhe**), quali nomi, descrizioni, codici.

I tipi principali per le variabili in Visual Basic sono:

Boolean	Tipo di dati con solo due valori possibili, ovvero True (–1) o False (0). Le variabili di tipo *Boolean* vengono memorizzate come numeri a 16 bit (2 byte).
Integer	Tipo di dati contenente variabili memorizzate come numeri interi a 16 bit (2 byte) nell'intervallo da –32.768 a 32.767.
Long	Intero di 32 bit (4 byte), ovvero un numero intero compreso tra –2.147.483.648 e 2.147.483.647.
Currency	Tipo di dati compreso nell'intervallo da –922.337.203.685.477,5808 a 922.337.203.685.477,5807. Utilizzato per calcoli monetari o a virgola fissa in cui la precisione è fondamentale.
Single	Tipo di dati che contiene variabili a virgola mobile e precisione singola a 32 bit (4 byte), compresi tra –3,402823E+38 e –1,401298E–45 per valori negativi e tra 1,401298E–45 e 3,402823E+38 per valori positivi. (7 cifre significative)
Double	Tipo di dati che contiene numeri a virgola mobile e doppia precisione a 64 bit (8 byte) compresi tra –1,79769313486232E+308 e –4,94065645841247E–324 per i valori negativi, tra 4,94065645841247E–324 e 1,797693134862325E+308 per i valori positivi. (15 cifre significative)
Date	Tipo di dati utilizzato per memorizzare date e orari come numeri reali. Le variabili di tipo *Date* vengono memorizzate come numeri a 64 bit (8 byte). Il valore a sinistra del separatore decimale rappresenta una data e il valore a destra rappresenta un orario.
String	Tipo di dati utilizzato per memorizzare una sequenza di caratteri contigui. Può includere lettere, numeri, spazi e segni di punteggiatura. Il tipo *String* può contenere stringhe di lunghezza fissa con lunghezza compresa tra 0 e circa 63 KB di caratteri; è possibile stabilire la lunghezza della stringa al momento della definizione della variabile indicando dopo *String* il numero dei caratteri preceduto da un asterisco. Per esempio: *Dim Nome As String * 15* specifica che la variabile *Nome* può contenere al massimo 15 caratteri.
Variant	È il tipo di dati in cui vengono trasformate tutte le variabili se non sono dichiarate esplicitamente come tipo diverso utilizzando l'istruzione *Dim*. *Variant* è un tipo di dati speciale che può contenere qualsiasi tipo sia numerico che alfanumerico. È possibile utilizzare *Variant* al posto di qualsiasi tipo per gestire i dati in modo più flessibile.

Esempi di dichiarazione di variabili:

```
Dim Contatore As Integer
Dim Anni As Integer
Dim Statura As Single
Dim AreaCerchio As Double
Dim AreaTriang As Double
Dim Nome As String
Dim Trovato As Boolean
```

L'identificatore di una variabile o di una costante è una sequenza qualsiasi di caratteri alfabetici e cifre, che inizia comunque con una lettera; si può usare anche il carattere _ per definire nomi composti, per esempio *Area_Cerchio*. Gli identificatori rappresentati con nomi composti sono spesso scritti con tutti i caratteri di seguito e utilizzando l'iniziale maiuscola per ciascun nome, per esempio *AreaCerchio*.

All'interno del programma possono essere inserite frasi contenenti **commenti** o annotazioni del programmatore, con le quali è possibile documentare il significato delle variabili o delle costanti utilizzate, oppure la funzione svolta da una parte del programma.
Le frasi di commento sono precedute dal carattere apice '. Per esempio:

```
' dichiarazione delle variabili
Dim Eta As integer      'età di una persona
```

Nella finestra dell'editor di Visual Basic le righe di commento sono colorate in verde.

L'**istruzione di assegnazione** permette di attribuire un valore a una variabile e la sintassi è del tipo:

variabile = *espressione*

Il valore dell'espressione viene assegnato alla variabile scritta a sinistra del simbolo =.
Per esempio, il calcolo dell'area di un cerchio viene indicato con l'istruzione:

```
Area = Raggio * Raggio * 3.14
```

Per le variabili alfanumeriche, il valore da assegnare va racchiuso tra virgolette.
Per esempio:

```
Lingua = "Inglese"
```

Nelle espressioni possono poi comparire le **funzioni**, ossia sottoprogrammi predefiniti (*built-in*) del linguaggio che, ricevendo un argomento, restituiscono un valore calcolato.
Per esempio, la funzione predefinita **Sqr(X)** calcola la radice quadrata del numero X.
Quindi, per calcolare l'ipotenusa di un triangolo rettangolo, si può scrivere un'istruzione del tipo:

```
Ipot = SQR(cat1^2 + cat2^2)
```

Il valore delle espressioni logiche può essere **True** o **False** e, quindi, il risultato del calcolo delle espressioni può essere assegnato a variabili dichiarate di tipo **Boolean**.
Per esempio, data la dichiarazione:

```
Dim X, Y, Z As Boolean
```

si possono scrivere le seguenti istruzioni:

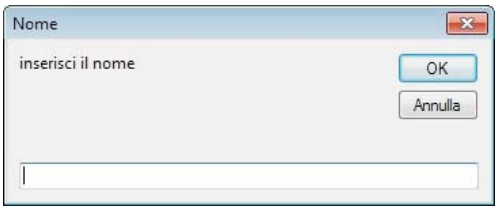
```
X = A > B
Y = NOT X
Z = A<B OR C<D
```

Si osservi che la seconda istruzione è equivalente a Y = A<=B, perché NOT (A>B) è equivalente a A <=B.
Per consentire la gestione dell'**input da tastiera** da parte dell'utente, il linguaggio Visual Basic mette a disposizione la funzione **InputBox**. Tale funzione visualizza una finestra di dialogo standard in cui viene richiesto all'utente di immettere un valore stringa. La finestra di dialogo di *InputBox* contiene una casella di testo in cui l'utente può digitare un valore alfanumerico e scegliere il pulsante *OK* o *Annulla* (in inglese, *Cancel*).
Se viene scelto il pulsante *OK* o se viene premuto il tasto *Invio*, la funzione *InputBox* restitui-sce la stringa digitata dall'utente. Se viene scelto il pulsante *Annulla*, la funzione restituisce una stringa vuota ("").

Per l'esempio l'istruzione

```
StringaNome = InputBox("inserisci il nome", "Nome")
```

assegna alla variabile *StringaNome* il valore inserito dall'utente tramite la finestra di dialogo della funzione *InputBox* rappresentata in figura.



Il primo parametro indica il messaggio per l'utente, il secondo indica la frase che deve comparire nella barra del titolo della finestra.

L'istruzione **MsgBox,** invece, permette di mandare un messaggio all'utente con una finestra di dialogo predefinita contenente anche un'icona che ricorda il tipo di messaggio (errore, avvertimento, informazione) e con uno o più tra i pulsanti standard *Si*, *No*, *Annulla* e *?*.
Per esempio, la seguente istruzione produce sul video una finestra di dialogo con la frase "*Fine lavoro*":

```
MsgBox "Fine lavoro", vbOKOnly, "Messaggio per l'utente"
```

Il secondo parametro fa comparire nella finestra il solo pulsante *OK*. La finestra di dialogo ha come titolo la frase "*Messaggio per l'utente*".

La struttura di **selezione**, come già visto nel Progetto 5, viene rappresentata in Visual Basic secondo lo schema:

IF *condizione* **THEN**
    istruzione1
ELSE
    istruzione2
END IF

Se la condizione è vera, viene eseguita l'*istruzione1*, altrimenti viene eseguita l'*istruzione2*.
Istruzione1 e *istruzione2* possono indicare, come accade nella maggior parte dei casi, non una sola istruzione, ma un gruppo di istruzioni.
La *condizione* è un'espressione booleana di cui viene valutata la verità: vengono quindi utilizzati i segni del confronto: <, >, =, >=, <=, <>, e gli operatori booleani AND, NOT, OR, XOR per costruire espressioni logiche combinando tra loro più condizioni.
La **ripetizione** si rappresenta in Visual Basic con la struttura **DO ... LOOP UNTIL** (ripetizione postcondizionale):

DO
    istruzioni
LOOP UNTIL *condizione*

La *condizione* deve essere un'espressione che rappresenta un valore *True* o *False*.

Le istruzioni comprese tra *Do* e *Loop* vengono eseguite una prima volta, dopo di che viene verificata la condizione scritta dopo *Until*: se la condizione risulta vera si prosegue con l'istruzione successiva, altrimenti si ripete l'esecuzione delle istruzioni a partire dalla prima istruzione dopo *Do*.

La struttura di **ripetizione precondizionale** viene realizzata con la struttura **DO WHILE...LOOP**

DO WHILE *condizione*
    istruzioni
LOOP

Le istruzioni comprese tra *Do* e *Loop* vengono ripetute mentre la condizione scritta vicino a *While* si mantiene vera.
La struttura di **ripetizione con contatore** è rappresentata con la struttura **FOR**...**NEXT**:

FOR *contatore* = *iniziale* **TO** *finale*
    istruzioni
NEXT *contatore*

Le istruzioni comprese tra *For* e *Next* vengono ripetute tante volte quante occorrono per passare dal valore iniziale della variabile *contatore* al valore finale, incrementando di 1 a ogni esecuzione.

Le strutture derivate

In aggiunta alle precedenti strutture di controllo, il linguaggio Visual Basic possiede altre varianti delle **strutture di ripetizione**, che possono essere considerate come strutture derivate da quelle fondamentali:

DO UNTIL *condizione*
    istruzioni
LOOP

DO
    istruzioni
LOOP WHILE *condizione*

Inoltre la struttura di **scelta multipla**, derivazione della struttura *If ... Then*, è realizzata dall'istruzione **SELECT CASE** che ha la seguente struttura sintattica:

SELECT CASE *VariabileDiControllo*
CASE *valori1*
    istruzioni1
CASE *valori2*
    istruzioni2
. . . . . .
CASE *valori*n
    *istruzioni*n
CASE ELSE
    istruzioni
END SELECT

Dopo le parole *Select Case* viene indicato il nome della variabile *VariabileDiControllo* (o variabile **selettore**) di cui si deve controllare il valore per decidere quale strada seguire tra quelle possibili. Accanto ai valori previsti devono essere scritte le istruzioni da eseguire nel caso in cui la variabile assuma quei valori.