

## Concetti generali per la programmazione in Basic

Qualunque sia il linguaggio di programmazione utilizzato, ci sono una serie di concetti comuni a tutti di cui bisogna tenere conto. Quindi prima di passare alle istruzioni specifiche del Basic, e in particolare, del LetPicBasicLite, ci soffermeremo su questioni più generali, però non meno importanti.

### Parti di un programma

Le parti di cui è composto un programma, il modo di denominarle, la loro obbligatorietà o meno, compreso il loro ordine, possono variare leggermente da un linguaggio di programmazione all'altro. In ogni caso le possiamo raggruppare in base alle funzioni che realizzano, dato che queste sono sempre imprescindibili. Così, cominciamo con il dire che un programma è formato da subroutines, o subprogrammi. Ognuna di queste subroutines contiene delle istruzioni, organizzate secondo una struttura di controllo. Sia le subroutines che le istruzioni e le strutture di controllo necessitano di variabili e costanti per il loro funzionamento.

Per rappresentare visivamente come si sviluppa un programma si utilizzano gli organigrammi.

### Variabili e costanti

Una variabile è una specie di cassetto, al quale diamo un

nome, e nel quale si possono mettere diverse cose. Per contro una costante sarà un cassetto che contiene sempre la stessa cosa. Quindi una variabile potrà cambiare nel tempo e contenere diversi valori, anche se sempre uno per volta, e sempre che "ci stiano" nel cassetto, mentre una costante non può essere modificata. Sia una che l'altra facilitano la programmazione, perché evitano di dover far

caratteri, ecc. e in molti linguaggi questo deve essere specificato, dato che, a seconda del tipo, si utilizza più o meno spazio per immagazzinarle. Può essere necessario, inoltre, specificare all'inizio del programma che tipo di variabili si vogliono utilizzare, e inizializzarle con un valore, anche se questo valore in seguito sarà modificato.

TIPI DI DATI	ESEMPIO DI VALORI	ESEMPIO DI DICHIARAZIONI
Numeri esadecimali	\$0A	Dim b
Numeri binari	%00001010	
Numeri decimali	10	
Simbolo	B.2	Symbol LED=B.2
Dispositivo	16F87x	Device 16F87x

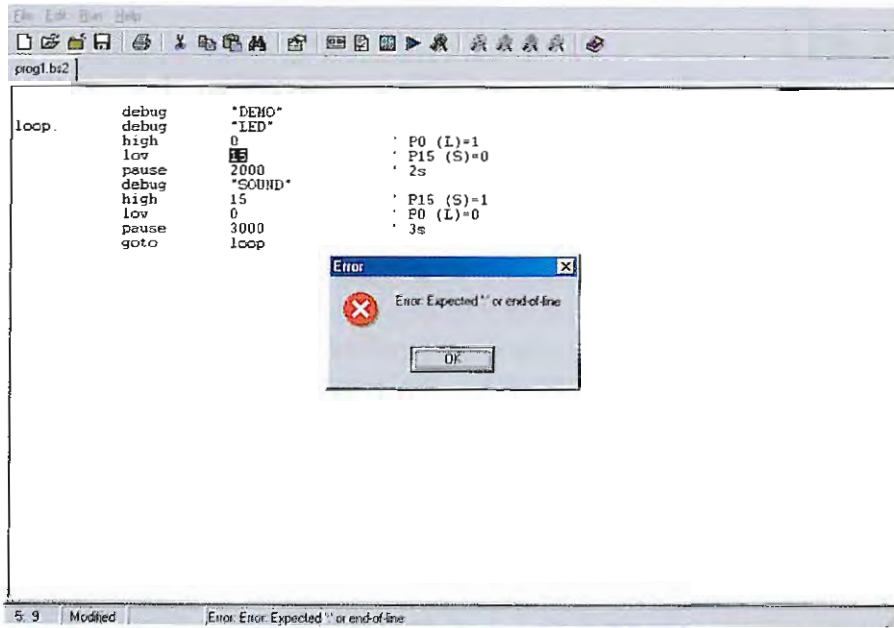
Tipi di dati in LetPicBasic ed esempi di dichiarazioni.

riferimento a valori concreti, il che potrebbe generare dei problemi, come possibili equivoci o ripetizioni di codici, ci limiteremo quindi a "fare x con quello che c'è nel cassetto y", senza dover ricordare se il contenuto del cassetto sono camicie oppure calze ecc. Le variabili possono essere di diversi tipi, secondo la classe dei valori che contengono, il che potrebbe equivalere ai diversi attaccapanni che ci sono in un armadio, per gonne, cravatte, pantaloni... Allo stesso modo le variabili potranno essere dei numeri interi, decimali,

### Istruzioni

Quando si dà un'istruzione a qualcuno, è per fargli realizzare concretamente una determinata azione. L'istruzione può essere molto dettagliata oppure risultare ambigua, dato che dalla sua corretta definizione dipenderà il conseguimento dell'obiettivo desiderato. Anche in un programma le azioni si realizzano secondo una sequenza di istruzioni, ma non c'è spazio per l'ambiguità.

Ogni istruzione ha un nome specifico, dei parametri, e svolge una determinata azione. Se il



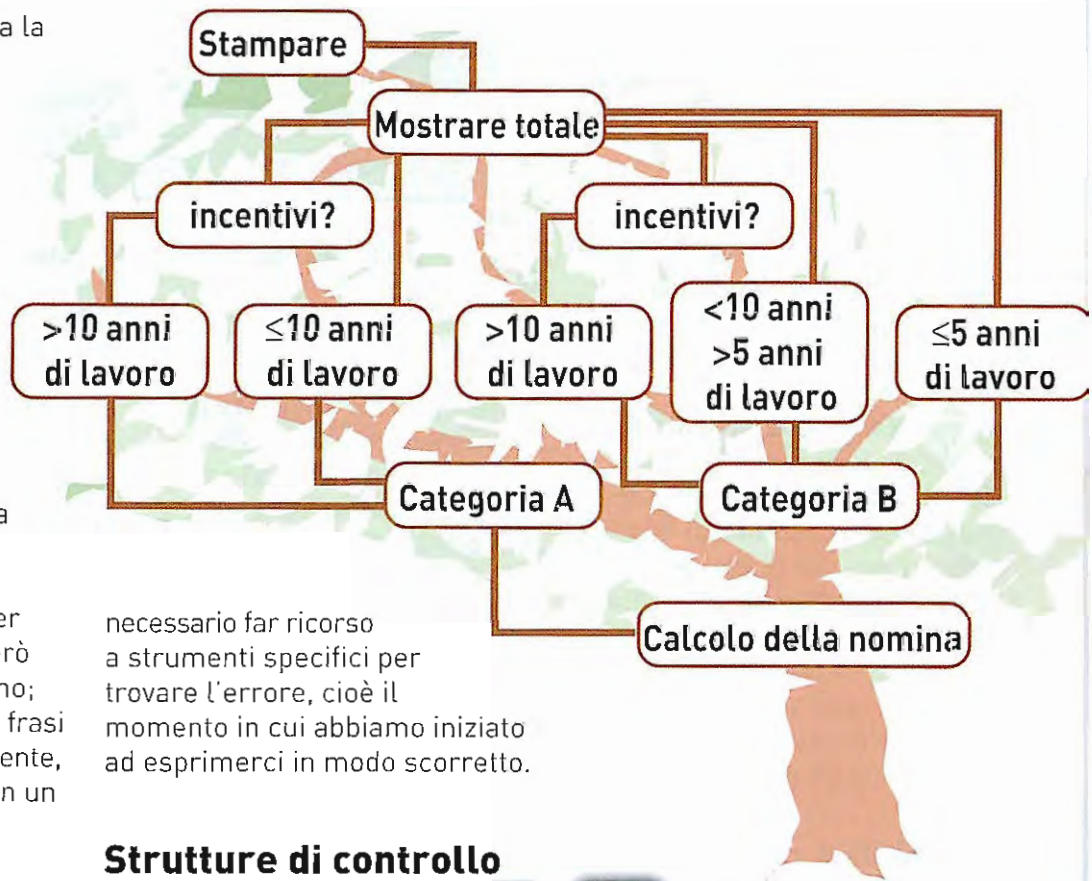
I compilatori avvisano degli errori sintattici, però non di quelli semantici.

sarebbe formato da una serie di istruzioni, collocate una dietro l'altra, in modo sequenziale. Questo significa che si eseguirebbero sempre le stesse istruzioni, ognuna di esse un'unica volta, senza tener conto delle decisioni dell'utente, dei risultati e di qualsiasi altro evento.

Questo, dal punto di vista del programmatore, è la cosa più semplice da fare, ma di solito non è la prassi più comune. Normalmente si confrontano i dati, e si decide se eseguire una serie di istruzioni o un'altra, cioè si utilizzano le strutture condizionali o alternative: un insieme di istruzioni si ripete un

programmatore non rispetta la sintassi, o la "forma" delle istruzioni si genera un errore, dato che il computer non le può capire. Anche rispettando la sintassi, può succedere che il computer confonda le istruzioni, le mescoli in un ordine permesso, ma non adeguato allo scopo, o equivochi i parametri. Non è la stessa cosa dire di salire a cavallo e andare a casa oppure andare a casa e salire a cavallo.

In questo caso il computer non genera alcun errore, però non farà ciò che ci aspettiamo; sarebbe come formare delle frasi ben fatte se prese singolarmente, ma senza relazioni fra loro in un dialogo completo. Questo è conosciuto come errore semantico. Ci rendiamo conto della sua esistenza quando vediamo il risultato, ed è



necessario far ricorso a strumenti specifici per trovare l'errore, cioè il momento in cui abbiamo iniziato ad esprimerci in modo scorretto.

## Strutture di controllo

Se non esistessero le strutture di controllo, un programma

Un programma può sembrare un albero con percorsi multipli da eseguire.