

Variabili

Se pensavate di aver già visto tutto sulle variabili vi sbagliavate. Mano a mano che andremo avanti con gli esempi, scoprirete quanto possa essere complesso ciò che in apparenza è semplice. Come vedrete, con l'utilizzo delle variabili le cose non sono né bianche né nere, tutto dipende dal linguaggio di programmazione e dal punto di vista.

Definizione di variabile

Possiamo definire una variabile come una specie di cassetto all'interno del quale le "cose" si aggregano e variano con il passare del tempo. Parlando in termini più tecnici una variabile è un nome che identifica uno (o più) indirizzi di memoria, dove vengono scritti in modo temporale i diversi valori generati in un programma. In questo modo i valori potranno essere recuperati al momento del bisogno. La memoria è il luogo dove il computer, o il microcontroller, memorizza sia i programmi che le variabili utilizzate per questi programmi.

Dichiarazione di variabili

Normalmente, prima di utilizzare una variabile questa deve essere dichiarata, e alcune volte anche essere inizializzata. Per definirla bisogna indicarne il tipo e il nome,

LINGUAGGIO	DICHIARAZIONE DI VARIABILE COMMENTATA	
C	Int b	Si dichiara la variabile interna "b"
Visual Basic	Pubblico Contatore come Integer	Si dichiara la variabile pubblica "Contatore" che conterrà dati interni
Assembler PIC	CONT EQU 0A	Si dichiara la variabile "CONT", che occuperà l'indirizzo 0A della memoria dei dati
PHP	\$interno=2002;	Si definisce un interno e gli si assegna il valore 2002
FORTRAN 90	Dimensione A (3,3)	Si dichiara un array di 3x3 indirizzi

Esempi di dichiarazioni di variabili nei vari linguaggi di programmazione.

prima di iniziare a usarla. A seconda del tipo di variabile, il compilatore riserverà più o meno spazio per ognuna. Tuttavia ci sono linguaggi, come ad esempio il Visual Basic, in cui non è strettamente necessario dichiarare le variabili, anche se è conveniente, perché altrimenti viene assunto un tipo di variabile per default che occupa più spazio del necessario. Inoltre nel caso di cattivo utilizzo della variabile, ad esempio se si utilizza per cose diverse in parti diverse del programma, se viene equivocato il nome, ecc. il

compilatore non ci avviserà dell'errore. In alcuni linguaggi è possibile dichiarare diverse variabili in una stessa istruzione. Il tipo di nome valido dipende dal linguaggio di programmazione. Nel C++Builder per esempio, devono essere parole composte da lettere e/o numeri sino a 32 caratteri significativi, iniziando sempre con una lettera. Alcuni linguaggi permettono di utilizzare nomi con il carattere sottolineato. In Visual Basic la lunghezza massima del nome è 255 e non può contenere il punto. Delphi utilizza solo i primi 63

- ✓ Deve essere chiaro e con riferimento diretto al problema, cioè descrittivo per dare un'indicazione di quello che fa. Possono aiutare i nomi composti da diverse parole separate da trattino (-, _).
- ✓ Non possono far parte del nome né spazi bianchi né caratteri speciali.
- ✓ Non devono essere parole riservate del linguaggio.
- ✓ Utilizzare i nomi sempre in minuscolo o sempre in maiuscolo, dato che alcuni linguaggi sono sensibili a questa differenza.
- ✓ Non utilizzare all'interno di un programma variabili con lo stesso nome per diversi scopi, genera confusione.
- ✓ Indicare per che cosa si utilizza ogni variabile con un commento.
- ✓ Dichiarare le variabili al livello che si vogliono utilizzare: classe, metodo, procedimento, modulo...

Le regole più comuni quando si nomina una variabile.



LINGUAGGIO	NOMI DI VARIABILI VALIDE
C++ BUILDER	Nome, Età, SdoGiornaliero, IngMensile, Perimetro, Calif1
DELPHI	MiVar, INIZIO, Var3, quarta_Var, _ultima_var

Esempi di nomi validi nei due linguaggi di programmazione.

caratteri, anche se non dà errore se si dichiarano di lunghezza maggiore. Il Pascal non è sensibile all'utilizzo delle maiuscole e minuscole, quindi "Var" è la stessa variabile che "VAR". Non succede la stessa cosa nel C++ o Java in cui sarebbero variabili diverse. In generale, nominando una variabile, bisogna tenere conto di una serie di regole, sempre consigliabili, anche se solo alcune volte obbligatorie.

Tipi di dati

Le variabili possono essere di differenti tipi, secondo la classe del valore che contengono. Il numero di tipi disponibili dipenderà anch'esso dal linguaggio di programmazione. Nella tabella possiamo vedere differenti tipi di dati semplici che possiamo trovare, il range dei valori che possono

Alcuni tipi di dati assegnabili a variabili

TIPO DI DATO	RANGE 1	RANGE 2	DIMENSIONE
char	-128..127		1 byte
signed Char	-128..127		1 byte
unsigned Char	0..255		1 byte
byte	0..255		1 byte
Int, integer	-32768..32767		2 byte
unsigned int	0..65536		2 byte
signed short	-32768..32767		2 byte
unsigned short	0..65536		2 byte
long	-2147483648.. 2147483647		4 byte
unsigned long	0.. 4294967295		4 byte
float	1.40129E-45..3.402823E+38	-3.402823E+38..-1.40129E-45	4 byte
double	4.94065E-324..1.7976931E+308	-1.797693E+308..-4.94065E-324	8 byte
long double	1.2E-4932..	-1.2E+4932	19 byte
Boolean o logica	True-False		2 byte
date	Data/Ora		8 byte
currency	Valuta		
Object	Riferimento ad un oggetto		4 byte
String, catena	Catena di carattere		1 byte per carattere

contenere e la dimensione di questi tipi. Tuttavia alcuni valori di questi range possono variare in funzione del linguaggio utilizzato, anche se il tipo si denomina nella stessa forma. Quando due o più tipi di variabili diverse si trovano in una stessa operazione o nella stessa espressione matematica, ci sono linguaggi che realizzano automaticamente una conversione dei tipi, in cui normalmente si assume che debba essere il dato di tipo più restrittivo a dover cambiare le proprie caratteristiche. Il cambio è valido solo per queste operazioni e non comporta di dover cambiare la dimensione di memoria in cui è alloggiato in modo permanente. Oltre ai dati semplici, alcuni linguaggi possiedono altri tipi di dati, come gli arrays o le matrici, che sono insiemi di variabili dei tipi precedenti ubicati in modo sequenziale in memoria; catene che sono sequenze di caratteri; oggetti che sono degli insiemi di variabili e funzioni associate, ecc. A volte può essere conveniente creare altri tipi di variabili definite dall'utente, per esempio, un insieme di dati da utilizzare, formato da variabili di differenti tipi, però trattato in alcuni casi come un solo dato.

Ambito di una variabile

L'ambito di una variabile lo formano quelle parti del programma fra cui questa variabile è visibile e quindi utilizzabile. Si può fare una prima classificazione dell'ambito di una variabile differenziando fra variabili globali (o pubbliche) e locali (o private). Le variabili locali sono quelle che si utilizzano esclusivamente all'interno della funzione/procedimento nel quale si definiscono. Le variabili globali, invece, possono essere utilizzate da qualsiasi punto del programma incluse tutte le funzioni e i procedimenti. Finché risulta possibile è meglio evitare l'utilizzo di variabili globali, dato che una variazione realizzata in una parte del programma influenza anche il resto, e in presenza di un errore è molto difficile seguire la pista dei differenti valori; si tratta dei famigerati errori laterali che si cerca di evitare utilizzando opportuni parametri nella comunicazione con i sub programmi. Le variabili dichiarate come "pubbliche" o "private" contengono la stessa relazione di quelle "globali" e "locali" però riferite ai moduli (insiemi di procedimenti) invece che a procedimenti o funzioni semplici, caso del linguaggio Visual Basic. Normalmente una variabile di tipo locale si crea e si inizializza entrando nel procedimento e si distrugge uscendo da esso. Per fare in modo che questo non succeda, la variabile si inizializzi solo la prima volta, e dopo conservi il suo valore per la prossima chiamata al procedimento, deve essere dichiarata di tipo "statico".

