

La sintassi di Java

I tipi di dati statici

In Java, come in Pascal, esistono tipi di dati statici predefiniti e sono i seguenti:

byte	8 bit da -128 a 127
short	16 bit coincide con l'integer di Pascal da -32768 a 32767
int	32 bit coincide con il longint del Pascal
long	64 bit
float	32 bit e un real in semplice precisione
double	64 bit è un real in doppia precisione
char	16 bit è esteso rispetto il Pascal: deve contenere i car. UNICODE
boolean	true o false (è oscurata all'utente, la dimensione dipende dal sistema)

Le stringhe

Le stringhe in java sono oggetti e sono allocate in memoria in modo dinamico come segue :

```
String s = new String ("banana");
```

oppure anche in una forma apparentemente statica come:

```
String s = "banana";
```

Le due dichiarazioni sono equivalenti: generano in memoria lo stesso oggetto.

Per ora la trattazione si limiterà a questo breve accenno: la conoscenza della classe String sarà approfondita in un paragrafo successivo.

Gli array

In java gli array, mono e bidimensionali, sono allocati in memoria in modo dinamico.

Ad esempio, la dichiarazione di un array d'interi di 10 elementi può essere effettuata in modo esplicito secondo una delle seguenti sintassi:

```
int v[] = new int[10];  
int[] v = new int[10];
```

oppure in modo implicito tramite l'assegnazione dei valori:

```
int v[] = { 1,5,6,7,8,11,12,13,14, -1};  
int[] v = { 1,5,6,7,8,11,12,13,14, -1};
```

Analogamente per un array bidimensionale di caratteri:

```
char a[ ][ ] = new char[2][2];
```

N.B. Gli elementi sono numerati a partire da 0, perciò per il vettore **v** l'indice va da 0 a 9

Istruzioni di sequenza, selezione e ciclo

Java è un linguaggio imperativo e, come tutti i linguaggi imperativi, è dotato delle strutture di controllo: SEQUENZA, SELEZIONE, ITERAZIONE.

Istruzioni sequenziali:

```
int a,c; // dichiarazione di variabili intere di tipo int
a = 2; // assegnazione di valore
c = a*a+7; // assegnazione di valore tramite espressione
System.out.println("a vale "+a+" c vale "+c); // istruzione di stampa
```

Istruzioni di selezione:

```
if (<condizione>) {
    <blocco>
}
if (<condizione>) {
    <blocco>
} else {
    <blocco>
}
```

Istruzioni iterative:

il ciclo while:

```
while (<condizione>) |
    <blocco>

```

il repeat:

```
do {
    <blocco>
} while (<condizione>)
```

il ciclo for:

```
for (tipo i=v_iniz; i<v_fin; i=i+step) |
    <blocco>

```

N.B. Si osservi la dichiarazione della variabile **i**. Una variabile può essere definita in qualsiasi punto del programma; se è definita in un blocco la sua definizione è locale al blocco, cioè non è definita fuori dal blocco stesso.

Operatori aritmetici e logici

Operatori aritmetici

Operatore	Significato	Esempio
+	somma	3 + 5
-	differenza	
*	moltiplicazione	
/	divisione	
%	modulo (resto della div)	resto= a % b;
++	incremento	i++; equivale a i=i+1;
--	decremento	i--; equivale a i=i-1;

Operatori di confronto e logici

Operatore	Significato	Esempio
!	negazione	if (!a==true)
==	uguale a	if (a==b)
!=	diverso da	if (a !=b)
<	minore di	(a < b)
>	maggiore di	
<=	minore o uguale a	
>=	maggiore o uguale a	
&&	AND Logico	if (a && b)
	OR Logico	if (a b)