

# Iterazione, variabili e grammatiche

## 1 Iterazione

Viene ripetuta l'esecuzione di un blocco di istruzioni in base al valore di una condizione.

### 1.1 Schema 1

```
ESEGUI
  blocco
QUANDO condizione
```

1. vengono eseguite le istruzioni contenute nel **blocco**
2. viene valutata la **condizione**
  - se è vera, si ritorna al punto 1
  - se è falsa, l'esecuzione prosegue con la prima istruzione successiva al costrutto iterativo (*unico punto di uscita*)

Caratteristiche:

- il blocco viene sempre eseguito almeno una volta
- l'esecuzione termina quando la condizione è falsa

### 1.2 Schema 2

```
QUANDO condizione ESEGUI
  blocco
RIPETI
```

1. viene valutata la **condizione**
  - se è vera, vengono eseguite le istruzioni contenute nel **blocco** e si ritorna al punto 1
  - se è falsa, l'esecuzione prosegue con la prima istruzione successiva al costrutto iterativo

Caratteristiche:

- il blocco può non essere mai eseguito
- l'esecuzione termina quando la condizione è falsa

### 1.3 Equivalenza dei due schemi

Lo schema 2 è può essere simulato combinando la selezione e lo schema 1:

```
SE condizione
  ALLORA
    ESEGUI
      blocco
    QUANDO condizione
  FINESE
```

Viceversa, lo schema 1 può essere simulato aggiungendo un'esecuzione "manuale" del blocco di istruzioni prima dello schema 2:

```
blocco
QUANDO condizione ESEGUI
  blocco
RIPETI
```

## 2 Variabile

Una **variabile** è un contenitore (identificato da un nome) preposto a contenere valori.

## 3 Assegnamento

L'istruzione di **assegnamento** assegna un valore a una variabile.

### 3.1 Sintassi e semantica operativa

variabile ← espressione

1. viene calcolato il valore dell'espressione
2. il risultato è assegnato alla variabile, *sovrascrivendo* il valore precedente

Molti linguaggi usano il simbolo = per l'assegnamento: esso non va confuso con l'uguaglianza, che invece si indica solitamente con ==.

## 4 Variabili come astrazione

- Le variabili astraggono le locazioni di memoria.
- Le operazioni sulle variabili astraggono le istruzioni di lettura e scrittura della memoria.

## 5 Tipi

Il **tipo** di una variabile specifica:

- l'**insieme dei valori** che in essa possono essere memorizzati
- l'**insieme delle operazioni** che possono essere effettuate su di essa

In particolare, l'insieme delle operazioni è di fondamentale importanza nella programmazione ad oggetti.

In memoria, tutte le variabili sono rappresentate come sequenze di bit, la cui interpretazione varia in base al tipo. Per questo, *il tipo astrae la rappresentazione effettiva dei dati*.

## 6 Dichiarazione di variabili

Molti linguaggi richiedono che ogni variabile venga **dichiarata**, specificandone il tipo.

In alcuni linguaggi, è necessario dichiarare tutte le variabili all'inizio del programma, mentre in altri è sufficiente dichiararle prima dell'utilizzo.

## 7 Sintassi e semantica

**Sintassi:** specifica *come si scrivono* le frasi di un linguaggio.

**Semantica:** specifica il *significato* di un programma.

## 8 Grammatica formale

Una **grammatica formale** (o *grammatica per linguaggi formali*) definisce la sintassi di un linguaggio.

Una grammatica  $G$  è composta da:

$$G = (T, N, P, S)$$

- un insieme finito  $T$  di *simboli terminali*, che costituiscono le *sentenze* (o *frasi*) del linguaggio (es. parole chiave, identificatori, letterali)
- un insieme finiti  $N$  di *simboli non terminali*, utilizzati nella costruzione delle sentenze del linguaggio
- un insieme finito  $P$  di *regole di produzione*, che collegano i simboli terminali e non
- un *simbolo iniziale*  $S \in N$ , il punto di partenza per la costruzione delle sentenze

Il linguaggio generato da una grammatica è l'insieme di tutte le sequenze di simboli terminali ottenibili applicando le regole di produzione  $P$  a partire dal simbolo iniziale  $S$ .

Esistono apposite notazioni per la descrizione delle regole di produzione, tra cui:

- BNF (*Backus-Naur Form*), una notazione testuale
- *carte sintattiche*, una notazione grafica

## 9 Lessico del linguaggio Java

**alfabeto:** l'insieme dei caratteri usati per scrivere i programmi (*Unicode*)

**parole riservate (o chiave):** parole con un significato predeterminato, che non possono essere utilizzate diversamente o ridefinite

**identificatori:** nomi che identificano le varie entità (es. variabili); devono iniziare con una lettera, seguita da altre lettere e/o cifre

**separatori:** separano o raggruppano parti di codice (es. ( ) [ ] { } ; , .)

**operatori:** denotano particolari operazioni (es. + = > !)

**letterali:** rappresentano valori di tipi primitivi e stringhe (es. 15, "stringa")

**commenti:** vengono ignorati dal compilatore (/ \* ... \*/ e // ...)