

Strutture del linguaggio

Selezione

if //Statements

```
if (condition) {  
  
    //statements;  
  
}  
  
[optional]  
else if (condition2) {  
  
    //statements;  
  
}  
  
[optional]  
else {  
  
    //statements;  
  
}
```

switch Statements

```
switch (Expression) {  
  
    case value1:  
  
        //statements;  
  
        break;  
  
    ...  
  
    case valuen:  
  
        //statements;  
  
        break;
```

```
    default:  
  
    //statements;  
  
}
```

loop Statements

```
while (condition) {  
  
    //statements;  
  
}  
  
do {  
  
    //statements;  
  
} while (condition);  
  
for (init; condition; adjustment) {  
  
    //statements;  
  
}
```

Cicli definiti

Se il numero di iterazioni è prevedibile dal contenuto delle variabili all'inizio del ciclo.

Esempio: prima di entrare nel ciclo so già che verrà ripetuto 10 volte

```
int n=10;  
for (int i=0; i<n; ++i) {  
    ...  
}
```

Cicli indefiniti

Se il numero di iterazioni non è noto all'inizio del ciclo.

Esempio: il numero di iterazioni dipende dai valori immessi dall'utente.

```
while(true) {  
    x = Integer.parseInt(JOptionPane.showInputDialog("Immetti numero  
positivo"));  
    if (x > 0) break;  
}
```

Cicli annidati

Se un ciclo appare nel corpo di un altro ciclo.

Esempio: stampa quadrato di asterischi di lato n

```
for (int i=0; i<n; i++) {  
    for (int j=0; j<n; j++) System.out.print("*");  
    System.out.println();  
}
```

Cicli con filtro

Vengono passati in rassegna un insieme di valori e per ognuno di essi viene fatto un test per verificare se il valore ha o meno una certa proprietà in base alla quale decideremo se prenderlo in considerazione o meno.

Esempio: stampa tutti i numeri pari fino a 100

```
for (int i=1; i<100; ++i) { // passa in rassegna tutti i numeri fra 1 e 100  
    if (i % 2 == 0) // filtra quelli pari  
        System.out.println(i);  
}
```

Cicli con filtro e interruzione

Se il ciclo viene interrotto dopo aver filtrato un valore con una data proprietà.

Esempio: verifica se un array contiene o meno numeri negativi

```
boolean trovato = false;  
for (int i=0; i<v.length; ++i) // passa in rassegna tutti gli indici  
dell'array v  
    if (v[i]<0) { // filtra le celle che contengono valori negativi
```

```
        trovato = true;
        break; // interrompe ciclo
    }
    // qui trovato vale true se e solo se vi sono numeri negativi in v
```

Cicli con accumulatore

Vengono passati in rassegna un insieme di valori e ne viene tenuta una traccia cumulativa usando una opportuna variabile.

Esempio: somma i primi 100 numeri interi.

```
int somma = 0; // variabile accumulatore di tipo int
for (int i=1; i<100; ++i) { // passa in rassegna tutti i numeri fra 1 e 100
    somma = somma + i; // accumula i valori nella variabile accumulatore
}
```

Esempio: data una stringa s, ottieni la stringa rovesciata

```
String rovesciata = ""; // variabile accumulatore di tipo String
for (int i=0; i<s.length(); ++i) { // passa in rassegna tutti gli indici
    dei caratteri di s
        rovesciata = s.substring(i, i+1) + rovesciata; // accumula i caratteri
    in testa all'accumulatore
}
```

Cicli misti

Esempio di ciclo definito con filtro e accumulatore: calcola la somma dei soli valori positivi di un array

```
int somma = 0;
for (int i=0; i<v.length; ++i) // passa in rassegna tutti gli indici
    dell'array v
        if (v[i]>0) // filtra le celle che contengono valori positivi
            somma = somma + v[i]; // accumula valore nella variabile
    accumulatore
    ...
```