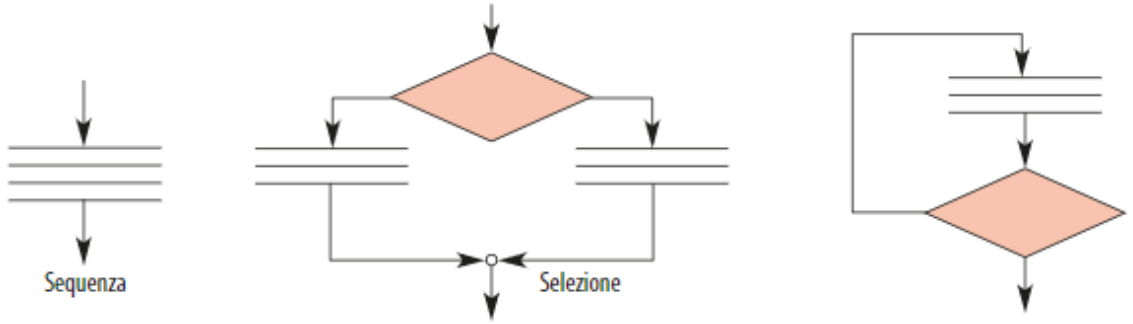
# L’istruzione ciclica o iterativa in Java

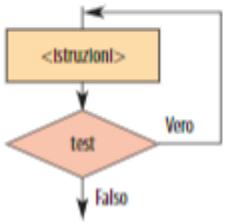
I blocchi di un diagramma a blocchi strutturato possono essere collegati secondo i seguenti schemi di flusso:
- schema di sequenza: più schemi di flusso sono eseguiti in sequenza;
- schema di selezione: un blocco di controllo subordina l’esecuzione di due possibili schemi di flusso al verificarsi di una condizione;
- schema di iterazione: si itera l’esecuzione di un dato schema di flusso.




Secondo i canoni della programmazione strutturata, la figura sopra mostra le tre strutture fondamentali con le quali è possibile codificare qualsiasi algoritmo.

## Il ciclo a condizione finale: do {...} while

L’istruzione di iterazione precondizionata, detta a controllo iniziale, esegue il test prima di effettuare il corpo del ciclo e, quindi, se è subito verificata, esso non viene eseguito neppure una volta. Tutti linguaggi di programmazione mettono a disposizione del programmatore anche una istruzione di iterazione postcondizionata, o a controllo finale, dove inizialmente viene eseguito il corpo del ciclo e solo alla sua fine viene effettuato il test di controllo.




Il funzionamento è semplice:
- si entra nel ciclo e si esegue il blocco di istruzioni (o semplicemente un’istruzione);
- al termine dell’esecuzione del blocco viene valutata la condizione di uscita, che è un’operazione di test, e quindi ha risultato VERO o FALSO;
- se l’esito è VERO si torna indietro e si ripete il ciclo un numero indefinito di volte;
- se è FALSO, si esce dall’altro ramo e si prosegue il programma con le successive istruzioni.

In linguaggio di progetto, questa istruzione viene così tradotta:

```
fai{
 <blocco di istruzioni>
}mentre <condizione di uscita>
```

In linguaggio di programmazione abbiamo invece la seguente sintassi:

**Linguaggio Java**

```
do
{
  <blocco di istruzioni>
}while <condizione di uscita>;
```

L'istruzione può essere letta nel seguente modo: "fai il blocco di istruzioni mentre la condizione è verificata (cioè il test dà esito VERO)".

Questa istruzione è particolarmente comoda per effettuare il controllo sulla correttezza dei dati inseriti in input dall'utente, in quanto, per poterli controllare, questi devono prima essere inseriti, e l'istruzione più adeguata a tale scopo è appunto quella a controllo finale.

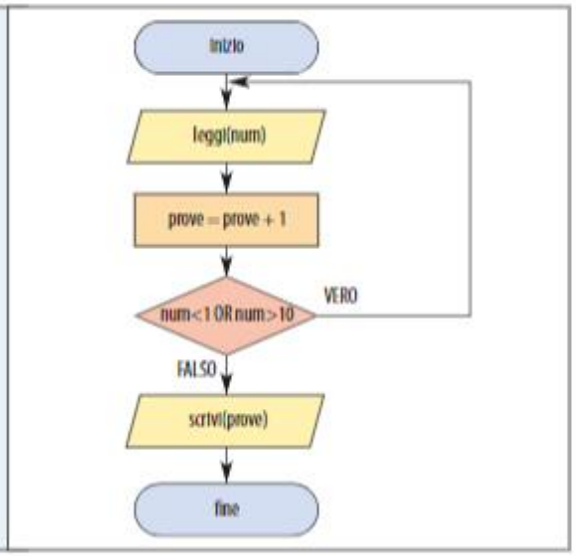
## Esempio

**III affinamento – pseudocodifica**

```
inizio
 fai
  leggi(num)
  prove<-prove+1
 mentre(num<1 OR num>10)
 scrivi(prove)
fine
```

ctr input compreso tra 1 e 10

condizione d'uscita




### ✔ La codifica in linguaggio di programmazione

**Linguaggio Java**

```
import java.util.Scanner;
public class CtrInput{
  public static void main(String[] args){
    Scanner in = new Scanner(System.in);
    int tanti, prove = 0;
    do                  // ciclo di controllo input
    {                   // compreso tra 1 e 10
      System.out.print("Inserisci 1 < N < 10: ");
      tanti = in.nextInt();
      prove= prove + 1;
    }while ((tanti < 1)||(tanti > 10));
    System.out.print("\nOK dopo " + prove + " !");
  }
}
```

## Il ciclo a conteggio

Nelle istruzioni di iterazione utilizzate fino a ora non conoscevamo mai in anticipo il numero di volte che doveva essere ripetuto il corpo del ciclo. L'inizio o la fine del ciclo, infatti, sono sempre stati condizionati da una variabile, il cui contenuto non era noto a priori.

Esiste tuttavia un insieme di problemi caratterizzati dal fatto di avere come dato iniziale proprio il numero di volte che un compito deve essere eseguito.

**ESEMPIO**

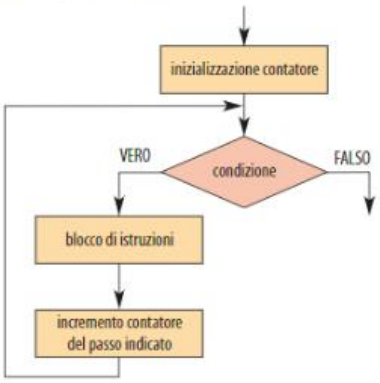
In tutte queste situazioni, un numero ci indica quante volte una azione deve essere ripetuta.

Fai **10** giorni di ferie.
Leggi **20** pagine.
Per **10** volte scrivi "devo studiare di più".
Attacca **30** figurine sull'album.
Invia **100** sms alla zia.
Per **210** giorni devi andare a scuola.

Quindi, il numero delle iterazioni è noto a priori, perciò definito.

Nel linguaggi di programmazione è presente un'istruzione che permette di codificare questo tipo di ciclo: è chiamata ciclo a conteggio, o ciclo for: il suo nome deriva dal fatto che viene realizzato mediante un contatore che, ogni volta che viene eseguito il corpo del ciclo, automaticamente viene aggiornato (incrementato o decrementato) di una certa quantità chiamata passo (step), che di default ha valore unitario (+1) fino a quando si raggiunge il valore finale (fine).

Il funzionamento è semplice:
– viene inizializzata una variabile di conteggio (detta anche variabile di controllo);
– viene eseguito il corpo del ciclo;
– al termine dell'esecuzione del blocco di istruzioni si incrementa la variabile di controllo, si torna indietro, si verifica la condizione di terminazione e si ripete il ciclo fino a giungere al valore finale di conteggio (che è predefinito).




In linguaggio di progetto il ciclo for viene così tradotto:

```
per conta<-inizio fino a fine fai
  <istruzione/i>
finePer
```

Il contatore del ciclo a ogni iterazione viene incrementato o decrementato di un valore costante che prende il nome di passo del contatore.

In linguaggio di programmazione abbiamo la seguente sintassi.

**Linguaggio Java**

```
for (inizializzazione; condizione di uscita; aggiornamento passo)
{
  < blocco di istruzioni>;
}
```

Il valore iniziale e il valore finale prendono anche il nome di estremi del ciclo e devono essere diversi tra loro, altrimenti il ciclo viene eseguito una sola volta.

ESEMPIO SVOLTO

SOMMARE TRE NUMERI

SommaconCicloFor.java

```
1
2  package sommatrenumeri;
3  import java.util.Scanner;
4
5  public class SommaconCicloFor {
6
7      public static void main(String[] args) {
8          //inserire tre numeri con ciclo FOR e sommarli
9
10         Scanner in =new Scanner(System.in);
11         int numero;
12         int somma=0;
13         int i;
14
15         for (i=0; i<3; i++)
16         {
17             System.out.println("inserisci numero da sommare");
18             numero= in.nextInt();
19             somma=somma+numero;
20
21
22         }
23
24             System.out.println("La somma è " + somma);
25
26     }
27
28 }
29
```

**Esercizio da svolgere:**

**Inserire 6 età di persone diverse e calcolare la media, utlizzando un ciclo FOR.**