Programmazione per Bioinformatica
Flusso di Controllo in Java: i cicli

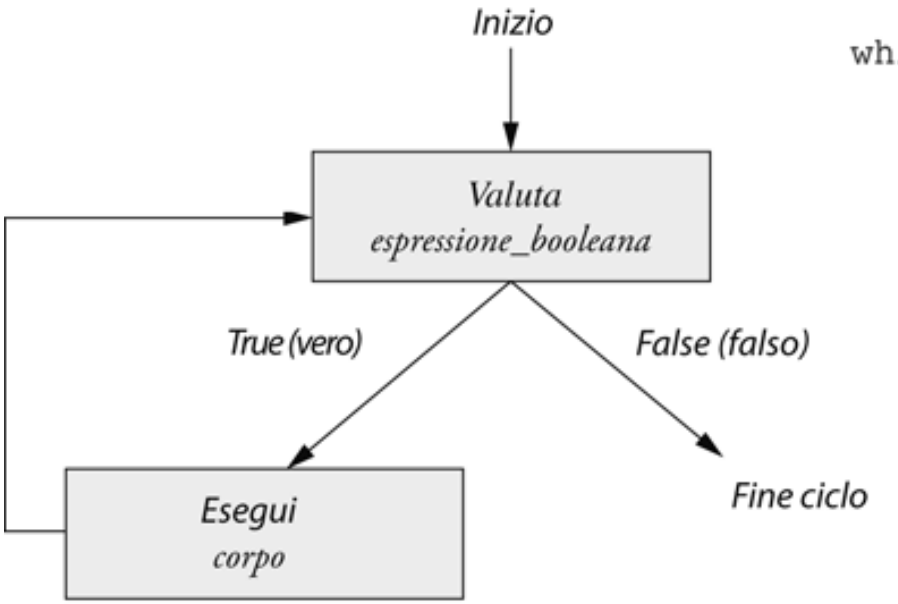
Dr Damiano Macedonio
Università di Verona

Istruzione **`while`**

```
while (espressione_booleana) {
   istruzione1;
   ...
   istruzioneN;
}
```

- Ad ogni iterazione viene valutato il valore di *espressione_booleana* :
- Se il risultato è *vero allora viene eseguito il corpo del ciclo.*
- Se il risultato è *falso allora il ciclo termina e si continua con l'istruzione immediatamente successiva al ciclo.*

Istruzione **`while`**



`while` (*espressione_booleana*)
corpo

Istruzione **`do-while`**

- Il ciclo **`while`** esegue il controllo della condizione prima che il ciclo venga eseguito.

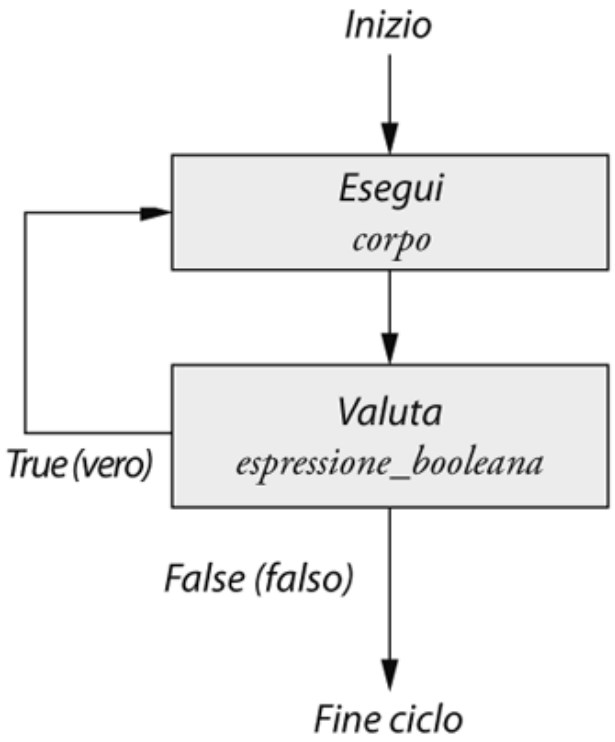
  Il corpo del ciclo potrebbe non essere mai eseguito.

- Il ciclo **`do-while`** valuta la condizione alla fine.

  Garantisce che il corpo del ciclo venga eseguito almeno una volta.

```
do {
   istruzione1;
   ...
   istruzioneN;
} while (espressione_booleana);
```

Istruzione **do-while**




```
do {
     corpo
} while (espressione_booleana);
```

Istruzione `for`

```
for (inizializzazione; espressione_booleana; aggiornamento) {
    istruzione1;
    ...
    istruzioneN;
}
```

- *inizializzazione* : utilizzata per impostare i valori iniziali *prima che cominci il ciclo (es. variabile indice).*

- *espressione_booleana* : condizioni necessarie affinché il ciclo possa continuare.
- Il ciclo continua finché *espressione_booleana* è `true`, quando *espressione_booleana* è `false` l'esecuzione continua con l'istruzione posta subito dopo il ciclo `for`.

- *aggiornamento*: eseguito *dopo che tutto il corpo del ciclo è stato eseguito.*
- Generalmente utilizzato per modificare il valore della variabile indice.

- *Istruzione1; . . . istruzioneN;* *corpo del ciclo.*
- non servono le parentesi `{ }` se c'è solo una *istruzione nel corpo del ciclo.*

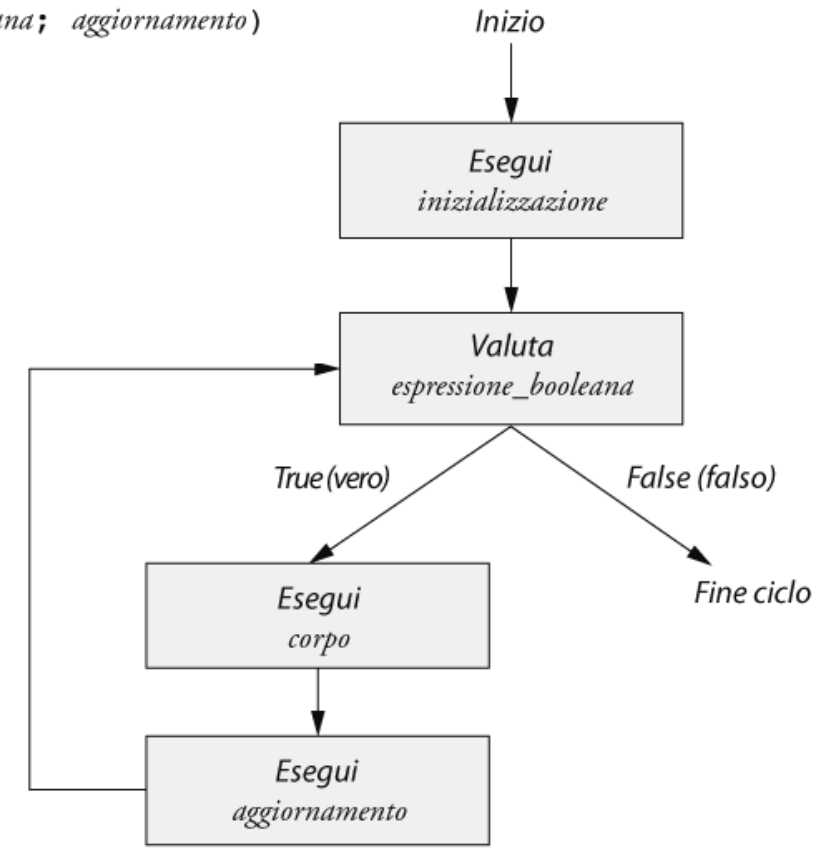
Istruzione `for`

Esecuzione ciclo **`for`**:

1. Viene eseguita l'*inizializzazione*.
2. Viene valutata *espressione_booleana*.
3. Se la condizione è `false` il ciclo termina e si procede con l'istruzione immediatamente successiva al ciclo.
4. Se la condizione è `true` viene eseguito una volta il corpo del ciclo.
5. Viene eseguito *aggiornamento*
6. Si ritorna al punto 2.

Istruzione `for`

```
for (inizializzazione; espressione_booleana; aggiornamento)
    corpo
```



Ciclo **`for`** o ciclo **`while`**?

Un ciclo **`for`** può essere tradotto in un ciclo **`while`** equivalente:

```
for (inizializzazione; espressione; aggiornamento) {
    istruzione1;
    ...
    istruzioneN;
}
```

```
inizializzazione;
while (espressione) {
    istruzione1;
    ...
    istruzioneN;
    aggiornamento;
}
```

Ciclo **`for`** o ciclo **`while`**?

E viceversa…

Un ciclo **`while`** può essere tradotto in un ciclo **`for`** equivalente:

```
while (espressione){
    istruzione1;
    ...
    istruzioneN;
}
```

```
for ( ; espressione; ){
    istruzione1;
    ...
    istruzioneN;
}
```

Ciclo `for` o ciclo `while`?

- La scelta tra ciclo **`for`** o ciclo **`while`** dipende dalle caratteristiche dell'iterazione che si deve eseguire.

- Il ciclo **`for`** è preferibile quando:
- Il ciclo deve essere eseguito un numero fisso di volte che é noto prima di iniziare il ciclo.
- Se l'inizializzazione, l'espressione e l'aggiornamento del ciclo riguardano tutte la stessa variabile.