

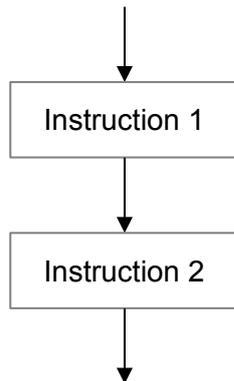
STRUTTURE DI CONTROLLO

Teorema di Jacopini-Böhm

Qualsiasi algoritmo può essere realizzato utilizzando tre sole strutture di controllo: sequenza, selezione e iterazione.

Sequenza

Le varie istruzioni sono eseguite dal calcolatore in modo sequenziale ossia nello stesso ordine con cui vengono presentate.

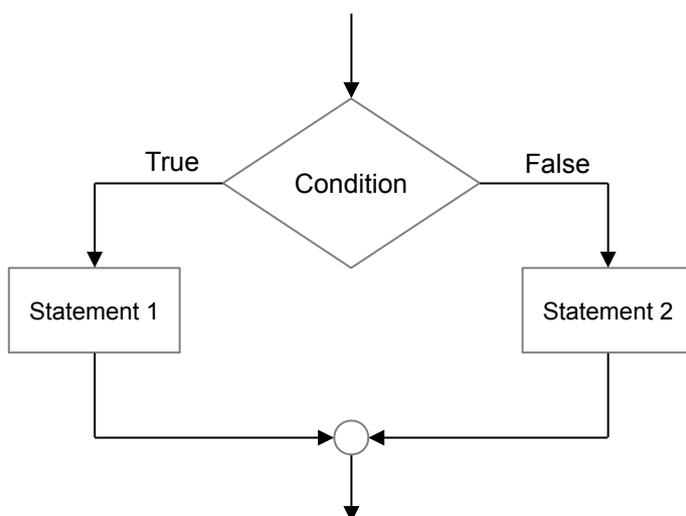


Esempio di codice in C/C++

```
cout << "Inserisci un numero";  
cin >> numero;  
numero += i;
```

Selezione

Permette di effettuare delle scelte durante l'esecuzione dell'algoritmo. In base al valore della condizione indicata (che può essere vera o falsa) il calcolatore esegue una certa istruzione piuttosto che un'altra. La condizione è rappresentata da un'operazione di confronto (per esempio ==, !=, >, <, >=, <=).



Esempio di codice in C/C++

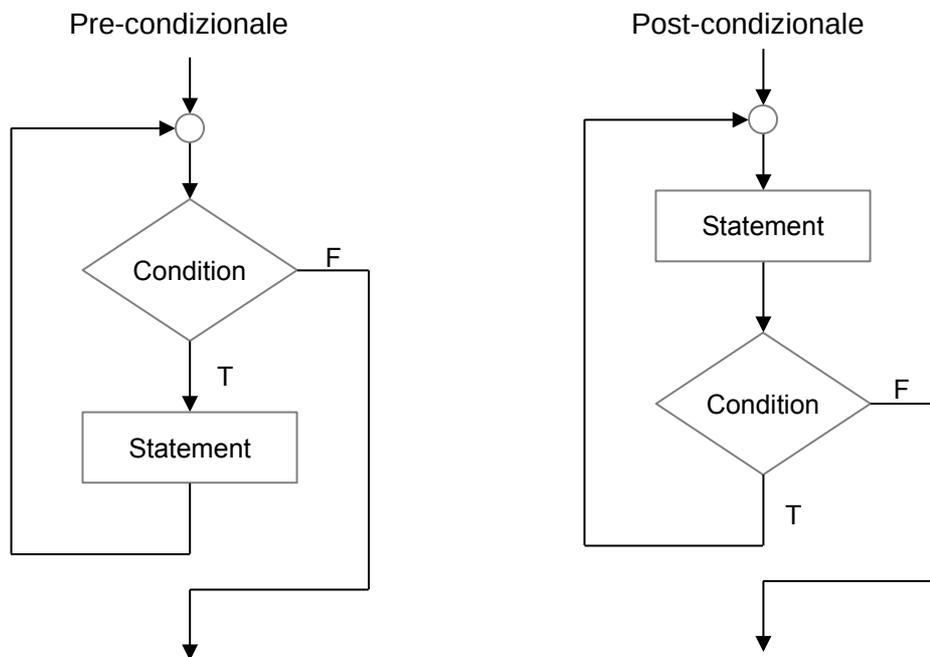
```
if (numero > 10)  
{  
    i *= 2;  
}  
else  
{  
    i ++;  
}
```

Iterazione

L'iterazione (o ripetizione) viene utilizzata quando bisogna eseguire, in modo ripetitivo, un blocco di istruzioni. Il numero di esecuzioni del blocco dipende dal verificarsi o meno di una data condizione.

Esistono tre tipi di iterazioni:

- iterazione pre-condizionale (viene verificata prima la condizione; se questa è vera viene eseguito il blocco di istruzioni e si ritorna alla verifica della condizione, altrimenti se la condizione è falsa si esce dal ciclo iterativo);
- iterazione post-condizionale (viene eseguito prima il blocco di istruzioni e poi viene verificata la condizione; se questa è vera si ritorna ad eseguire il blocco, se falsa si esce dal ciclo iterativo)^{vedi osservazione finale};
- iterazione enumerativa (il numero di esecuzioni è noto a priori e quindi la condizione consiste semplicemente nel “contare” quante volte viene eseguito il blocco).



Osservazione:

Quando si conosce a priori il numero di iterazioni da svolgere conviene utilizzare l'iterazione enumerativa, viceversa quando il numero di iterazioni dipende dal verificarsi o meno di una data condizione, il cui valore (vero o falso) cambia durante l'esecuzione dell'algoritmo, bisogna utilizzare l'iterazione pre o post condizionale.

A seconda dei casi si utilizza l'iterazione pre-condizionale o la post-condizionale, tenendo presente che nel caso della prima iterazione il blocco di istruzioni da ripetere può anche non essere mai eseguito (nel caso in cui la condizione è da subito falsa), mentre nell'iterazione post-condizionale il blocco viene eseguito almeno una volta (indipendentemente dal valore della condizione).

Esempi di codice in C/C++

Iterazione pre-condizionale (while):

```
while (x < 10)
{
    cout << x << endl;
    x ++;
}
```

Iterazione post-condizionale (do-while):

```
do
{
    cout << x << endl;
    x ++;
}
while (x < 10);
```

Iterazione enumerativa (for):

```
for (x = 1; x < 10; x ++ )
{
    cout << x << endl;
}
```

Osservazione finale

Attenzione! Per quanto riguarda l'iterazione post-condizionale, non sempre la condizione che indica l'iterazione è univoca. Ossia, in alcuni linguaggi, come ad esempio in Pascal, l'iterazione post-condizionale (*repeat-until*) itera con il falso, mentre si esce dal ciclo quando la condizione risulta vera.