
La programmazione strutturata

Programmazione strutturata

- Metodologia introdotta agli inizi degli anni settanta.
- L'esecutore è guidato alla sequenza di esecuzione opportuna, tra tutte quelle possibili, mediante tre **strutture di controllo** fondamentali:
 - **sequenza**
permette di eseguire le istruzioni secondo l'ordine in cui sono scritte
 - **selezione**
permette di scegliere l'esecuzione di un blocco di istruzioni tra due possibili in base al valore di una condizione
 - **iterazione**
permette di ripetere l'esecuzione di una o più istruzioni in base al valore di una condizione.

Programmazione strutturata

- Tutti i programmi esprimibili tramite istruzioni di salto (*goto*) o diagrammi di flusso (*flow-chart*) possono essere riscritti utilizzando esclusivamente le tre strutture di controllo fondamentali.
- L'impiego di queste strutture migliora la leggibilità dei programmi prodotti:
 - ogni struttura di controllo ha un solo **punto di ingresso** e un solo **punto d'uscita**
 - il flusso di esecuzione è evidente dalla struttura del codice

Sequenza

- Le istruzioni sono eseguite nello stesso ordine in cui compaiono nel programma, cioè secondo la **sequenza** in cui sono scritte.

Esempio: somma di due numeri

leggi i numeri a, b

calcola $a + b$

scrivi il risultato

Selezione

- Schema (sintassi):

```
SE condizione
  ALLORA
    blocco1
  ALTRIMENTI
    blocco2
FINESE
```

- Esecuzione (semantica):

- Viene valutata *condizione*:
 - se risulta *vera*, vengono eseguite le istruzioni del *blocco1*
 - se risulta *falsa*, vengono eseguite quelle del *blocco2*
- L'esecuzione procede con l'istruzione che segue immediatamente la fine del costrutto di selezione (FINESE)

Selezione

- Nel caso in cui il *blocco2* non contenga alcuna istruzione, si può utilizzare il seguente schema semplificato:

```
SE condizione  
  ALLORA  
    blocco1  
FINESE
```

- Esecuzione (semantica):
 - Viene valutata *condizione*:
 - se risulta *vera*, vengono eseguite le istruzioni del *blocco1*, quindi l'esecuzione prosegue direttamente dalla prima istruzione che segue il costrutto di selezione
 - se risulta *falsa*, l'esecuzione prosegue direttamente dalla prima istruzione che segue il costrutto di selezione

Esempio

Calcolo della divisione tra due numeri controllando che il divisore sia diverso da zero:

leggi il dividendo e il divisore

SE il divisore è diverso da zero

ALLORA

calcola dividendo/divisore

scrivi il risultato

ALTRIMENTI

scrivi "errore: divisione per zero"

FINESE

Esempio: calcolo delle radici di $ax^2 + bx + c = 0$

leggi i valori dei parametri a, b, c

calcola il discriminante

SE il discriminante è minore di zero

ALLORA

scrivi “nessuna soluzione reale”

ALTRIMENTI

SE il discriminante è uguale a zero

ALLORA

calcola $\frac{-b}{2a}$

scrivi “Due soluzioni coincidenti: ”, il risultato

ALTRIMENTI

scrivi “Due soluzioni: ”,

calcola $\frac{-b - \sqrt{b^2 - 4ac}}{2a}$

scrivi il risultato

calcola $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$

scrivi il risultato

FINESE

FINESE

Iterazione (schema 1)

- Schema (sintassi):

ESEGUI
blocco
QUANDO *condizione*

- Esecuzione (semantica):

- Anzitutto viene eseguito il *blocco* di istruzioni. Quindi si valuta la *condizione*:
 - se risulta *vera*, si ripete eseguendo nuovamente il *blocco* e valutando ancora la *condizione*
 - se risulta *falsa*, si prosegue con l'istruzione scritta dopo il costrutto iterativo

- Osservazioni

- il *blocco* è eseguito *almeno una volta*
- l'esecuzione del costrutto iterativo termina quando la *condizione diventa falsa*.

Esempio: somma dei primi 100 numeri interi

Senza utilizzare la formula di Gauss ($\sum_{i=1}^n i = \frac{n(n+1)}{2}$).

Calcolo iterativo:

poni il valore della somma a zero

inizia a considerare il numero 1

ESEGUI

aggiungi alla somma il numero che stai considerando

considera il numero successivo

QUANDO il numero che stai considerando non supera 100

scrivi la somma

Iterazione (schema 2)

- Schema (sintassi):

QUANDO *condizione* ESEGUI
blocco
RIPETI

- Esecuzione (semantica):

- Prima di tutto viene valutata la *condizione*:

- se risulta *vera*, allora si esegue il *blocco* e si valuta nuovamente la *condizione*.

- se risulta *falsa*, si passa a eseguire l'istruzione che segue il costrutto iterativo, cioè l'istruzione che segue la parola RIPETI.

- Osservazioni

- il *blocco* può essere eseguito anche zero volte

- l'esecuzione del costrutto iterativo termina quando la *condizione diventa falsa*.

Simulazione

Il comportamento dello schema QUANDO . . . RIPETI può essere simulato combinando lo schema della selezione con lo schema ESEGUI . . . QUANDO . . . , come segue:

SE *condizione*
ALLORA
ESEGUI
blocco
QUANDO *condizione*
FINESE