

Le basi della programmazione

Diagrammi di flusso con Flowgorithm (parte 2)

Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

Licenze e crediti

Questo materiale è disponibile sul sito Web del docente per il corso di [Tecnologie informatiche](#) per le studentesse e gli studenti dell'anno scolastico 2025/2026.

Versione: 1.3.0 (B)
Ultima modifica: 13/04/2026 09:06
Riferimenti: [1, Cap. 13]

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#)



Indice dei contenuti

- 1 Il teorema di Böhm-Jacopini
- 2 Le strutture di controllo
 - Struttura di sequenza
 - Struttura di selezione
 - Struttura di iterazione
- 3 Struttura di iterazione
 - Iterazione con controllo della condizione
 - Iterazione enumerativa
- 4 Esercizi
 - Esercizio n. 1
 - Esercizio n. 2

Indice dei contenuti

- 1 Il teorema di Böhm-Jacopini
- 2 Le strutture di controllo
 - Struttura di sequenza
 - Struttura di selezione
 - Struttura di iterazione
- 3 Struttura di iterazione
 - Iterazione con controllo della condizione
 - Iterazione enumerativa
- 4 Esercizi
 - Esercizio n. 1
 - Esercizio n. 2

Il teorema di Böhm-Jacopini

Definizione: Teorema di Böhm-Jacopini

Qualsiasi algoritmo calcolabile può essere implementato utilizzando esclusivamente tre strutture di controllo logico fondamentali, senza la necessità di utilizzare istruzioni di salto incondizionato (`goto`).

Il teorema di Böhm-Jacopini

Le tre strutture di controllo strettamente necessarie e sufficienti per scrivere qualsiasi programma sono:

- **Sequenza:** l'esecuzione di una serie di istruzioni in ordine lineare, una dopo l'altra;
- **Selezione:** la scelta tra due percorsi di esecuzione alternativi in base alla valutazione di una condizione booleana (il tipico costrutto `if-then-else`);
- **Iterazione:** la ripetizione di un blocco di istruzioni finché una determinata condizione logica si mantiene vera (il tipico ciclo `while-do`).

Il teorema di Böhm-Jacopini

Le tre strutture di controllo strettamente necessarie e sufficienti per scrivere qualsiasi programma sono:

- **Sequenza:** l'esecuzione di una serie di istruzioni in ordine lineare, una dopo l'altra;
- **Selezione:** la scelta tra due percorsi di esecuzione alternativi in base alla valutazione di una condizione booleana (il tipico costrutto `if-then-else`);
- **Iterazione:** la ripetizione di un blocco di istruzioni finché una determinata condizione logica si mantiene vera (il tipico ciclo `while-do`).

Il teorema di Böhm-Jacopini

Le tre strutture di controllo strettamente necessarie e sufficienti per scrivere qualsiasi programma sono:

- **Sequenza:** l'esecuzione di una serie di istruzioni in ordine lineare, una dopo l'altra;
- **Selezione:** la scelta tra due percorsi di esecuzione alternativi in base alla valutazione di una condizione booleana (il tipico costrutto `if-then-else`);
- **Iterazione:** la ripetizione di un blocco di istruzioni finché una determinata condizione logica si mantiene vera (il tipico ciclo `while-do`).

Il teorema di Böhm-Jacopini

Questo enunciato ha un'importanza storica immensa perché ha fornito le basi matematiche per la nascita della [programmazione strutturata](#).

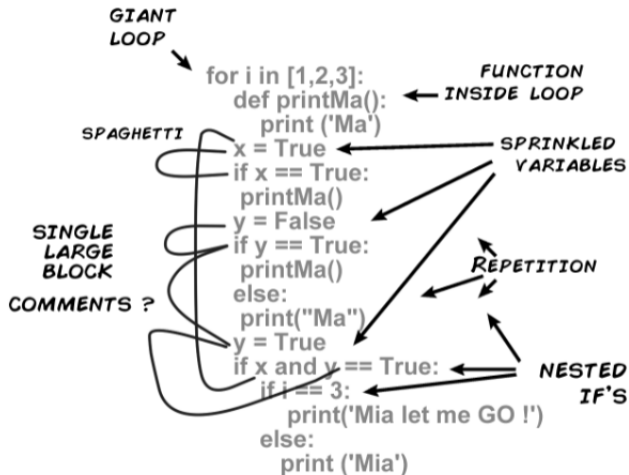
Prima di questo teorema, il flusso di esecuzione dei programmi era pesantemente basato sui salti incondizionati (`goto`), che spesso portavano a creare codice estremamente complesso e difficile da seguire (il famoso "*spaghetti code*").

Il teorema di Böhm-Jacopini

Questo enunciato ha un'importanza storica immensa perché ha fornito le basi matematiche per la nascita della [programmazione strutturata](#).

Prima di questo teorema, il flusso di esecuzione dei programmi era pesantemente basato sui salti incondizionati (`goto`), che spesso portavano a creare codice estremamente complesso e difficile da seguire (il famoso "*spaghetti code*").

Il teorema di Böhm-Jacopini



Il teorema di Böhm-Jacopini

Il teorema di Böhm-Jacopini ha dimostrato che il goto non è teoricamente necessario, aprendo la strada alla creazione di linguaggi di programmazione moderni orientati alla leggibilità, al rigore e alla manutenibilità del software.

Indice dei contenuti

1 Il teorema di Böhm-Jacopini

2 Le strutture di controllo

- Struttura di sequenza
- Struttura di selezione
- Struttura di iterazione

3 Struttura di iterazione

- Iterazione con controllo della condizione
- Iterazione enumerativa

4 Esercizi

- Esercizio n. 1
- Esercizio n. 2

Le strutture di controllo

Nella scrittura di un **algoritmo** risolutivo di un dato problema, è importante seguire un insieme di regole per strutturare il lavoro in modo corretto, ma anche per rendere più facili la lettura e le successive modifiche dei procedimenti risolutivi.

DIAGRAMMA DI FLUSSO DELL'INGEGNERE



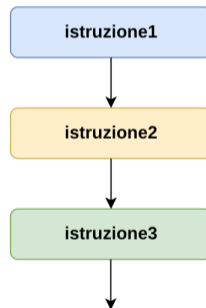
Struttura di sequenza

La **struttura di sequenza** rappresenta i blocchi di istruzioni che devono essere eseguite una dopo l'altra.

Linguaggio di pseudocodifica:

```
Inizio
istruzione1
istruzione2
istruzione3
...
Fine
```

Diagramma di flusso:



Struttura di sequenza

Q: Qualche esempio?

Struttura di sequenza

Q: Qualche esempio?

Esempio: *perimetro e area di un quadrato*

Nell'algoritmo per il calcolo del perimetro e area di un quadrato¹ le istruzioni racchiuse tra i blocchi **Inizio** e **Fine** sono una *sequenza di istruzioni*.

¹Vedasi: [2, Lezione 8] ([link](#))

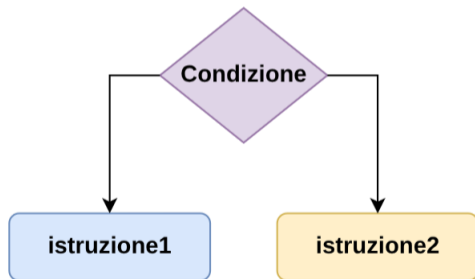
Struttura di selezione

La **struttura di selezione** rappresenta blocchi di istruzioni che devono essere eseguiti in alternativa ad altri, a seconda che una condizione sia vera o falsa.

Linguaggio di pseudocodifica:

```
Inizio  
Se (condizione), allora:  
- istruzione1  
Altrimenti:  
- istruzione2  
Fine
```

Diagramma di flusso:



Struttura di iterazione

La **struttura di iterazione** rappresenta un blocco di istruzioni che devono essere eseguite più volte.

Ci sono due modi principali di esprimere una ripetizione di istruzioni: l'**iterazione con controllo della condizione** e l'**istruzione enumerativa**.

Struttura di iterazione

La **struttura di iterazione** rappresenta un blocco di istruzioni che devono essere eseguite più volte.

Ci sono due modi principali di esprimere una ripetizione di istruzioni: l'**iterazione con controllo della condizione** e l'**istruzione enumerativa**.

Struttura di iterazione

Q: Queste due strutture in realtà le abbiamo già viste sotto altri nomi, come si chiamano?

Struttura di iterazione

Q: Queste due strutture in realtà le abbiamo già viste sotto altri nomi, come si chiamano?

A: Sono rispettivamente l'iterazione indefinita e l'iterazione definita!

Indice dei contenuti

- 1 Il teorema di Böhm-Jacopini
- 2 Le strutture di controllo
 - Struttura di sequenza
 - Struttura di selezione
 - Struttura di iterazione
- 3 Struttura di iterazione**
 - Iterazione con controllo della condizione
 - Iterazione enumerativa
- 4 Esercizi
 - Esercizio n. 1
 - Esercizio n. 2

Iterazione con controllo della condizione

Nell'**iterazione con controllo della condizione** un gruppo di istruzioni è ripetuto **mentre una condizione si mantiene vera**. Si esce quindi dalla ripetizione quando la condizione diventa falsa.

Questo significa che non si può sapere a priori quante volte vengono eseguite le istruzioni del ciclo: dipende infatti dal controllo della condizione!

Iterazione con controllo della condizione

Nell'**iterazione con controllo della condizione** un gruppo di istruzioni è ripetuto **mentre una condizione si mantiene vera**. Si esce quindi dalla ripetizione quando la condizione diventa falsa.

Questo significa che non si può sapere a priori quante volte vengono eseguite le istruzioni del ciclo: dipende infatti dal controllo della condizione!

Iterazione con controllo della condizione

Questo tipo di ripetizione può essere a sua volta realizzata in due modi:

- Con un'iterazione **pre-condizionale**;
- Con un'iterazione **post-condizionale**.

Iterazione con controllo della condizione

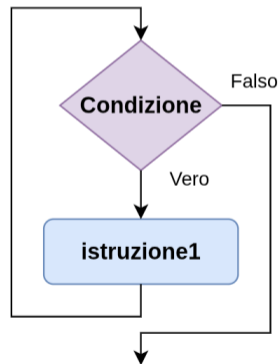
Si parla di iterazione **pre-condizionale** quando il controllo viene fatto in testa, cioè all'inizio del gruppo di istruzioni da ripetere.

Linguaggio di pseudocodifica:

```
Mentre (condizione), esegui:  
- istruzione1  
- istruzione2  
...
```

Le istruzioni comprese tra le parole **Mentre (condizione), esegui:** vengono ripetute (nella sequenza indicata) finché la condizione posta a monte è vera (TRUE).

Diagramma di flusso:



Iterazione con controllo della condizione

Si dice **iterazione post-condizionale** quando il controllo viene fatto in coda, cioè alla fine di un gruppo di istruzioni da ripetere.

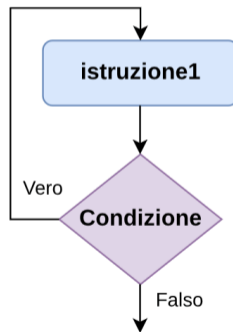
Linguaggio di pseudocodifica:

Esegui:

- istruzione1
- istruzione2

Mentre (condizione)

Diagramma di flusso:



Iterazione enumerativa

L'**istruzione enumerativa** (o con contatore) rappresenta in forma compatta la ripetizione di un gruppo di istruzioni **un numero prefissato di volte**. Di conseguenza, è utilizzata quando si conosce a priori quante volte un gruppo di istruzioni deve essere eseguito.

Linguaggio di pseudocodifica:

```
Per (variabile = A) a B con (incremento), ripeti:
```

```
- istruzione1
```

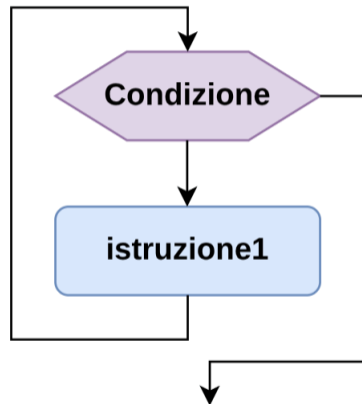
```
- istruzione2
```

```
...
```

Iterazione enumerativa

Le istruzioni comprese tra le parole **Per** (in inglese: *for*) e **ripeti** vengono ripetute tante volte quante occorrono per portare la variabile (detta **contatore**) dal valore iniziale al valore finale.

La variabile viene incrementata automaticamente in base a quanto definito in **incremento** dopo ogni esecuzione delle istruzioni da ripetere.



Iterazione enumerativa

Esempio: *stampare i numeri pari*

Si vuole scrivere un programma in pseudocodifica formale per stampare i numeri da 2 a 10 utilizzando solamente l'iterazione enumerativa.

Q: Come possiamo risolvere l'esercizio?

Iterazione enumerativa

Esempio: *stampare i numeri pari*

Si vuole scrivere un programma in pseudocodifica formale per stampare i numeri da 2 a 10 utilizzando solamente l'iterazione enumerativa.

Una possibile soluzione è la seguente:

```
Per (numero = 2) a 10 con (numero = numero + 2), ripeti:  
- Stampa (numero): "[1]"
```

Iterazione enumerativa

Una possibile soluzione è la seguente:

```
Per (numero = 2) a 10 con (numero = numero + 2), ripeti:  
- Stampa (numero): "[1]"
```

Il cui output sarà:

```
2  
4  
6  
8  
10
```

Indice dei contenuti

- 1 Il teorema di Böhm-Jacopini
- 2 Le strutture di controllo
 - Struttura di sequenza
 - Struttura di selezione
 - Struttura di iterazione
- 3 Struttura di iterazione
 - Iterazione con controllo della condizione
 - Iterazione enumerativa
- 4 **Esercizi**
 - **Esercizio n. 1**
 - **Esercizio n. 2**

Esercizio n. 1

Si consideri l'esempio "stampare i numeri pari".

Scrivere la tabella di traccia per ogni passo di esecuzione del programma.

Esercizio n. 2

Si vuole realizzare un programma per calcolare il totale della spesa al supermercato, sommando il prezzo di ciascun prodotto acquistato.

Alcuni suggerimenti:

- **Dati di input:** per ciascun prodotto, il prezzo unitario e il numero di pezzi acquistati;
- **Dato di output:** il totale da pagare;
- **Risoluzione:** ogni prodotto acquistato ha un prezzo unitario che deve essere moltiplicato per il numero di pezzi acquistati per quel prodotto. Mano a mano il risultato intermedio deve essere sommato al totale generale.

Q: Che tipo di iterazione si può utilizzare?

Riferimenti e approfondimenti

- [1] A. Lorenzi, M. Govoni e F. De Vincenzi, Informatica e reti di comunicazione. Atlas, 2026.
- [2] L. E. Dei Rossi, Corso di Laboratorio di Tecnologie Informatiche, 1, 2025.