

Dal problema al programma: le basi della programmazione

L'istruzione di selezione e le condizioni logiche

prof. Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

1 L'istruzione di selezione doppia

2 Il linguaggio di progetto

- Esercizio #1
- Esercizio #2

3 La selezione semplice

- Esercizio #3
- Istruzione di selezione semplice vs. doppia

4 Esercizi

- Esercizio #4
- Esercizio #5

1 L'istruzione di selezione doppia

2 Il linguaggio di progetto

- Esercizio #1
- Esercizio #2

3 La selezione semplice

- Esercizio #3
- Istruzione di selezione semplice vs. doppia

4 Esercizi

- Esercizio #4
- Esercizio #5

L'istruzione di selezione doppia (1)

Nella descrizione di un **algoritmo** si può verificare che le operazioni da eseguire siano diverse a seconda dei dati inseriti.

Esempio

Ad esempio, se dobbiamo andare in vacanza, dobbiamo verificare di non avere “debiti scolastici” a settembre, e quindi si prospettano due soluzioni alternative:

- **Se** abbiamo tutte sufficienze, **allora** andiamo in vacanza;
- **Altrimenti** dobbiamo studiare tutta l'estate.

L'istruzione di selezione doppia (2)

Nel diagramma a blocchi questa situazione è illustrata ricorrendo a un elemento grafico specifico, detto **blocco di decisione** o **blocco condizionale**:

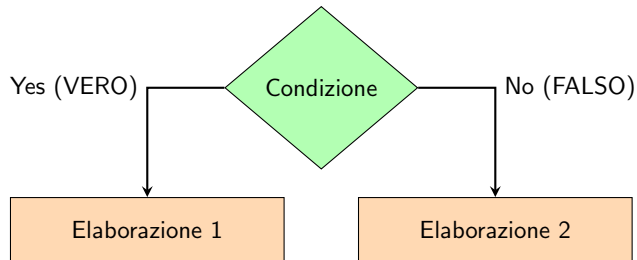


Figura 1.1: Blocco di decisione (condizionale) ramificato

L'istruzione di selezione doppia (3)

Di seguito alcuni esempi di istruzioni di confronto che possiamo scrivere nei blocchi condizionali:

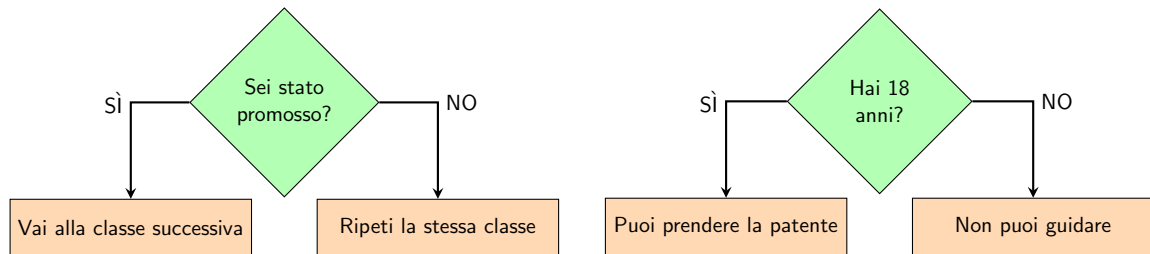
- Ho abbastanza soldi comprare la moto?
- Il serbatoio è pieno?
- Il numero 50 è maggiore di 0?
- Giochi a calcio?
- Mario è un alunno di questa classe?
- 27 è un numero pari?

L'istruzione di selezione doppia (4)

A queste domande si può solo rispondere **SÌ** oppure **NO**, esistono cioè soltanto **due alternative**: si tratta dunque di “istruzioni di confronto” che prendono il nome di **condizioni logiche** dalle quali, a seconda del valore della risposta, è sempre necessario intraprendere **percorsi alternativi**, cioè in base alla risposta è necessario eseguire operazioni diverse.

L'istruzione di selezione doppia (5)

Li rappresentiamo graficamente nei diagrammi a blocchi con uno schema come il seguente:

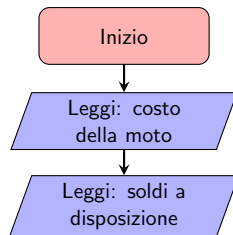


L'istruzione di selezione doppia (6)

Possiamo osservare che, a seconda della risposta data alla domanda (cioè al **test**), le strade “si dividono”, vengono eseguite operazioni diverse nei due “rami” ma, successivamente, si ricongiungono.

La presenza di istruzioni in entrambi i rami del diagramma fa sì che in questo caso si parli di **selezione doppia**.

Scriviamo il diagramma a blocchi per l'algoritmo “semplificato” che descrive la procedura per l'acquisto di una moto.



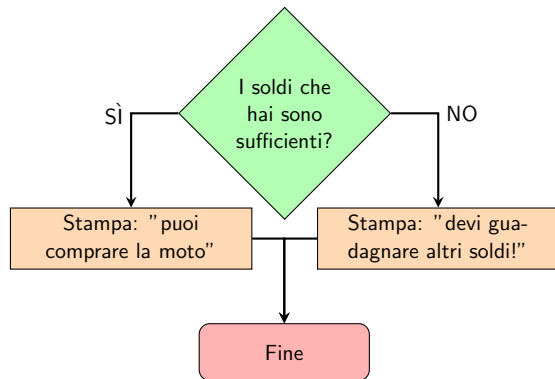
(continua)

L'istruzione di selezione doppia (7)

Possiamo osservare che, a seconda della risposta data alla domanda (cioè al **test**), le strade “si dividono”, vengono eseguite operazioni diverse nei due “rami” ma, successivamente, si ricongiungono.

La presenza di istruzioni in entrambi i rami del diagramma fa sì che in questo caso si parli di **selezione doppia**.

Scriviamo il diagramma a blocchi per l'algoritmo “semplificato” che descrive la procedura per l'acquisto di una moto.



L'istruzione di selezione doppia (8)

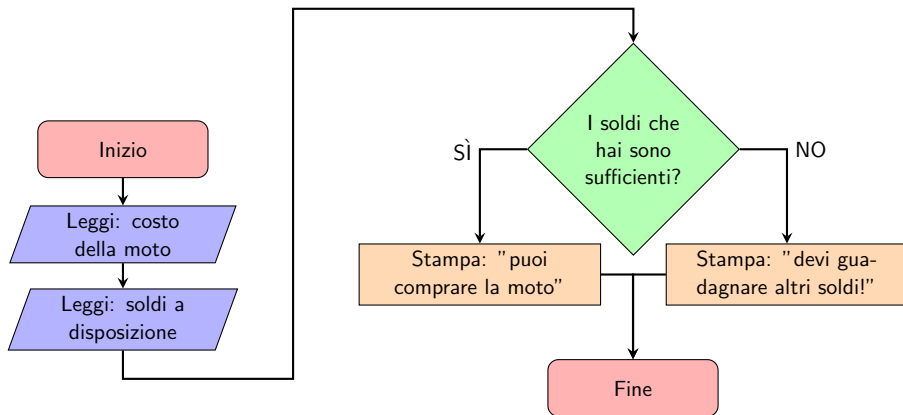


Figura 1.2: Diagramma di flusso completo

1 L'istruzione di selezione doppia

2 Il linguaggio di progetto

- Esercizio #1
- Esercizio #2

3 La selezione semplice

- Esercizio #3
- Istruzione di selezione semplice vs. doppia

4 Esercizi

- Esercizio #4
- Esercizio #5

Il linguaggio di progetto (1)

Iniziamo a definire un "nostro" **linguaggio di progetto**, ovvero un insieme di convenzioni da rispettare quando si scrive un algoritmo in pseudocodifica.

Definiamo la struttura base di un programma:

```
Inizio  
...  
Fine
```

Il linguaggio di progetto (2)

Definiamo ora le istruzioni di input e output:

```
Inizio  
Leggi (distanza_km): "Inserisci la distanza in km"  
...  
Stampa (distanza_km): "La distanza in km è: [1]"  
Fine
```

Il linguaggio di progetto (3)

E l'istruzione di elaborazione:

```
Inizio  
Leggi (distanza_km): "Inserisci la distanza in km"  
Calcola: (distanza_m = distanza_km * 1000)  
Stampa (distanza_km, distanza_m): "La distanza in km è [1], mentre in metri è [2]"  
Fine
```

Il linguaggio di progetto (5)

Per finire, l'istruzione di decisione:

```
Inizio
Leggi (distanza_km): "Inserisci la distanza in km"
Calcola: (distanza_m = distanza_km * 1000)

Se (distanza_m > 1000), allora:
- Stampa (): "La distanza è più di 1000 metri!"
Altrimenti:
- Stampa (): "La distanza è meno di 1000 metri!"

Fine
```

Si consideri la Figura 1.2.

Scrivere la pseudocodifica corrispondente in linguaggio formale.

Tratto da: [1, p. 350]

Esercizio #2

Si scriva il diagramma di flusso e la relativa pseudocodifica in linguaggio formale del seguente esercizio: "*leggi due numeri e visualizza sullo schermo il maggiore di essi*".

Tratto da: [1, p. 355]

1 L'istruzione di selezione doppia

2 Il linguaggio di progetto

- Esercizio #1
- Esercizio #2

3 La selezione semplice

- Esercizio #3
- Istruzione di selezione semplice vs. doppia

4 Esercizi

- Esercizio #4
- Esercizio #5

La selezione semplice (1)

È possibile che le operazioni da eseguire siano presenti in un solo ramo, quando ad esempio si devono effettuare operazioni solo in un caso dell'esito del test.

In queste situazioni si parla di **selezione semplice**.

La selezione semplice (2)

Esempio

Scriviamo un programma che trasforma i numeri negativi in positivi: nel caso che il numero inserito sia già positivo non dobbiamo effettuare nessuna operazione mentre se è negativo lo moltiplichiamo per -1 , in modo da cambiarne il segno.

Una possibile soluzione è la seguente:

```
Inizio
Leggi (numero): "Inserisci un numero"
Se (numero < 0), allora:
-   Calcola: (numero = numero * (-1))
Stampa (numero): "Il numero è [1]"
Fine
```

Si consideri la pseudocodifica formale presentata.
Scrivere il corrispettivo diagramma di flusso.

Istruzione di selezione semplice vs. doppia

Nota bene

L'istruzione di **selezione semplice** può quindi essere considerata un caso particolare di **selezione doppia** dove viene a mancare il ramo **altrimenti**; non è invece possibile avere un'istruzione semplice senza il ramo **allora**, con cioè il solo ramo **altrimenti**.

1 L'istruzione di selezione doppia

2 Il linguaggio di progetto

- Esercizio #1
- Esercizio #2

3 La selezione semplice

- Esercizio #3
- Istruzione di selezione semplice vs. doppia

4 Esercizi

- Esercizio #4
- Esercizio #5

Esercizio #4

Si scrivano il diagramma di flusso e la pseudocodifica in linguaggio formale di un programma che, dati in input tre numeri, stabilisca se tali numeri siano o meno una terna pitagorica.

Definizione: Terna pitagorica

Tre numeri formano una terna pitagorica se il quadrato del primo è uguale alla somma dei quadrati del secondo e del terzo.

Esempio: $(3, 4, 5) \rightarrow (9, 16, 25) \Rightarrow 9 + 16 = 25$

Tratto da: [1, p. 355]

Esercizio #5

Si consideri la seguente pseudocodifica formale:

```
Inizio
Leggi (numero1, numero2): "Inserisci due numeri"
Se (numero1 < numero2), allora:
-   Stampa (numero2): "[1]"
Leggi (numero3): "Inserisci un numero"
Se (numero2 < numero3), allora:
-   Stampa (numero2): "[1]"
Altrimenti:
-   Stampa (numero3): "[1]"
Fine
```

Cosa viene visualizzato inserendo i valori (5, 2, 7)?

Tratto da: [1, p. 358]

- [1] P. Camagni e R. Nikolassy, *TeknoLab: Tecnologie informatiche E Laboratorio: Office automation, Sviluppo del Pensiero computazionale, scratch e applicazioni per arduino, app inventor per android, Algobuild e Codifica C++*. Hoepli, 2017.