

# Istruzioni di selezione in Java

Programmazione  
Corso di laurea in Informatica

## Le istruzioni del programma

- Il controllo del flusso del programma
  - Istruzioni condizionali o di selezione
  - Istruzioni di ripetizione
- Espressioni condizionali e operatori

AA2005/06 © Alberti 2 Programmazione 8. Selezione

## Flusso di controllo

- L'ordine di esecuzione delle istruzioni è sequenziale se non altrimenti specificato
- Alcune istruzioni consentono di alterare l'ordine sequenziale:
  - Decidere se eseguire o meno un'istruzione
  - Eseguire un'istruzione ripetutamente
- L'ordine di esecuzione delle istruzioni si chiama *flusso di controllo*

AA 2005/06 © Alberti 3 Programmazione 8. Selezione

## Istruzione condizionale

- Consente di stabilire quale prossima istruzione eseguire
- Detta anche *istruzione di selezione* perchè consente di scegliere e prendere decisioni
- Le istruzioni condizionali Java
  - *if statement*
  - *if-else statement*
  - *switch statement*

AA2005/06 © Alberti 4 Programmazione 8. Selezione

## Istruzione if

- Sintassi dell'istruzione *if*

La condizione deve essere *espressione booleana*. Deve essere valutata vero o falso.

*if* una parola riservata Java

```
if ( condizione )  
    istruzione;
```

Se la condizione è vera, viene eseguita l'istruzione. Se è falsa, l'istruzione è tralasciata.

AA 2005/06 © Alberti 5 Programmazione 8. Selezione

## Esempio d'istruzione if

```
if (somma > MAX)  
    delta = somma - MAX;  
    System.out.println ("La somma è " + somma);
```

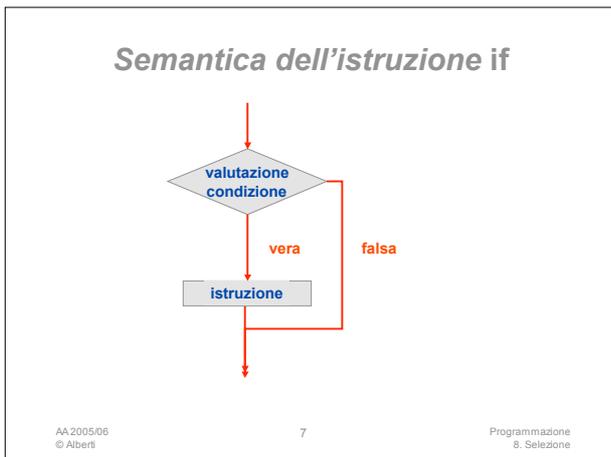
Prima si valuta la condizione: l'espressione (*somma > MAX*)

Se la condizione è vera, viene eseguita l'*istruzione di assegnamento*, altrimenti questa viene saltata.

In ogni caso viene eseguita l'istruzione *println*.

- Esempio [Age.java](#)

AA 2005/06 © Alberti 6 Programmazione 8. Selezione



### Espressioni Booleane

- Per rappresentare le condizioni si usano gli **operatori** di Java di **uguaglianza** o **relazionali**, che riportano valori **booleani**

==	uguale
!=	non uguale
<	minore
>	maggiore
<=	minore o uguale
>=	maggiore o uguale

- Si noti la differenza tra l'operatore di uguaglianza (**==**) e l'operatore di assegnamento (**=**)

AA 2005/06 © Alberti 8 Programmazione 8. Selezione

### Istruzione if-else

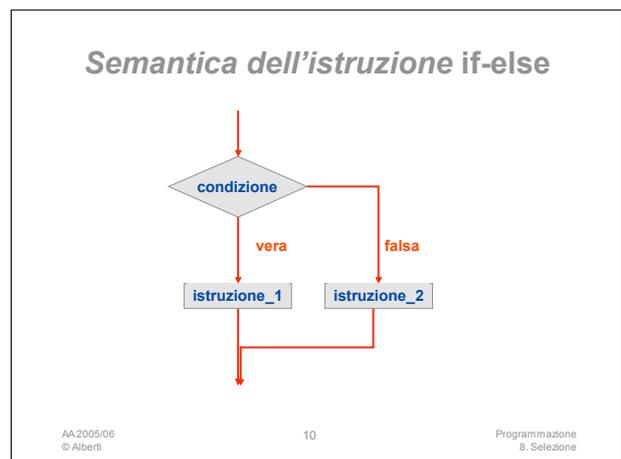
- La clausola *else* può essere aggiunta all'istruzione *if*

```

if ( condizione )
    istruzione_1;
else
    istruzione_2;
    
```

- condizione** vera viene eseguita **istruzione\_1**; se è falsa viene eseguita **istruzione\_2**
- Ne viene eseguita una sola e non entrambe
- Esempio [Wages.java](#)

AA 2005/06 © Alberti 9 Programmazione 8. Selezione



### Istruzione blocco

- Più istruzioni possono essere raggruppate in un *blocco*
  - Il blocco è delimitato dalle parentesi graffe { ... }
- Un blocco può essere usato là dove la sintassi di Java vuole un'istruzione
- Esempio: in un'istruzione *if-else*, la porzione *if*, o la porzione *else* o entrambe, possono essere blocchi
- See [Guessing.java](#)

AA 2005/06 © Alberti 11 Programmazione 8. Selezione

### Istruzioni if innestate

- L'istruzione da eseguire come risultato della valutazione di una condizione potrebbe essere a sua volta un'istruzione *if-else*
- Queste istruzioni sono dette *istruzioni if innestate*
- esempio [MinOfThree.java](#)
- La porzione *else* è associata all'ultima istruzione *if* (non fatevi ingannare dall'indentazione)

AA 2005/06 © Alberti 12 Programmazione 8. Selezione

### Confronti tra caratteri

- Gli operatori di relazione possono essere usati sui dati di tipo carattere
- Il risultato dipende dalla posizione nella tabella Unicode

```
if ('+' < 'J')
    System.out.println ("+ è minore di J");
```
- La condizione è vera perché il car '+' viene prima del car 'J' in Unicode:
- Le maiuscole (A-Z) e le minuscole (a-z) sono in ordine alfabetico nella tabella Unicode

AA 2005/06  
© Alberti

13

Programmazione  
8. Selezione

### Confronti tra stringhe

- Una stringa in Java è un oggetto
- Non possiamo usare gli operatori relazionali ==, <, >
- Uguaglianza. Usare il metodo `equals` per confrontare il contenuto di due oggetti stringa
  - `stringa_1.equals(stringa_2)`

AA 2005/06  
© Alberti

14

Programmazione  
8. Selezione

### Confronti tra stringhe

- Confronti. Usare il metodo `compareTo` per confrontare stringhe
  - Secondo l'ordine dei caratteri nella tabella Unicode
  - `stringa_1.compareTo(stringa_2)`
    - < 0 se `stringa_1` precede `stringa_2`
    - > 0 se `stringa_1` segue `stringa_2`
    - == 0 se sono uguali

AA 2005/06  
© Alberti

15

Programmazione  
8. Selezione

### Errore comune

- Utilizzare `==` anziché `equals`
- L'operatore `==`
  - Verifica che le due stringhe si riferiscono allo stesso oggetto e non se si riferiscono a oggetti con lo stesso contenuto

```
String nome = "carlo";
if (nome == "carlo") ....vera

String soprannome = "carlo magno";
nome = soprannome.substring (0, 5);
if (nome == "carlo") .... Falso
```
- [UguaglianzaStringhe.java](#)

AA 2005/06  
© Alberti

16

Programmazione  
8. Selezione

### Confronti tra oggetti

```
Rectangle scatola = new Rectangle (5, 10, 20, 30);
Rectangle r = scatola;
Rectangle confezione = new Rectangle (5, 10, 20, 30);
```

- ```
scatola == r ... vera
```
- Perché le due variabili si riferiscono allo stesso oggetto
- ```
scatola == confezione ... falsa
```
- Perché si riferiscono a due oggetti diversi, anche se hanno contenuti identici
- ```
scatola.equals(confezione) ...vera
```
- Verifica la corrispondenza dei campi

AA 2005/06  
© Alberti

17

Programmazione  
8. Selezione

### Confronto con null

- Il riferimento a un oggetto può avere il valore speciale `null` quando l'oggetto non è stato ancora creato
- Per verificare se il riferimento è `null` si usa l'operatore di relazione `==`
  - `if (conto == null) ....`
    - Per domandarsi se l'oggetto a cui si riferisce la variabile `conto` è stato istanziato
- Alcuni metodi restituiscono `null` quando non sono in grado di restituire un oggetto valido
- Il riferimento `null` è diverso dalla stringa vuota
- La stringa vuota è una stringa a tutti gli effetti una stringa che non contiene nulla, di lunghezza 0

AA 2005/06  
© Alberti

18

Programmazione  
8. Selezione

### Confronti tra valori in virgola mobile

- Attenzione va posta anche per il confronto di uguaglianza tra valori in virgola mobile (**float** o **double**)
- Raramente si usa l'operatore di uguaglianza (**==**) per confrontare due numeri di tipo **float**
- È meglio considerare se i due valori sono sufficientemente vicini, anche se non identici, a causa delle approssimazioni nella rappresentazione
- Es: [Precisione.java](#)  

```
final double EPSILON = 1E-14;  
if (Math.abs (f1 - f2) < EPSILON)  
    System.out.println ("Praticamente uguali.");
```

AA 2005/06 © Alberti 19 Programmazione 8. Selezione

### Istruzione switch

- L'istruzione **switch** consente di decidere qual'è l'istruzione successiva da eseguire tra diverse
- L'istruzione **switch** valuta un'espressione, quindi ne confronta il risultato con i diversi *cas*i elencati
- Ogni caso contiene un valore e una lista di istruzioni da eseguire
- Il flusso di controllo è trasferito alla lista associata al primo valore uguale all'espressione

AA 2005/06 © Alberti 20 Programmazione 8. Selezione

### Istruzione switch - 2

- La sintassi dell'istruzione **switch**:

```
switch ( espressione )  
{  
    case valore_1 :  
        lista_istruzioni_1  
    case valore_2 :  
        lista_istruzioni_2  
    case valore_3 :  
        lista_istruzioni_3  
    case ...  
}
```

Le parole riservate **switch** e **case**

se espressione uguaglia **valore\_2**, il controllo passa qui

AA 2005/06 © Alberti 21 Programmazione 8. Selezione

### Istruzione switch - 3

- Spesso si usa un'istruzione **break** come ultima istruzione in ogni lista
- L'istruzione **break** passa il controllo alla fine dell'istruzione **switch**
- Se non si usa l'istruzione **break**, il flusso di controllo continua ai casi successivi senza controllare l'espressione ulteriormente
- Qualche volta questo è utile, ma di solito i valori dei diversi casi sono mutualmente esclusivi e al più un caso corrisponde al valore dell'espressione

AA 2005/06 © Alberti 22 Programmazione 8. Selezione

### Istruzione switch - 4

- Un'istruzione **switch** può avere un caso di default
- Il caso di default non ha un valore associato ma usa semplicemente la parola riservata **default**
- Se è presente il caso di default, il controllo è trasferito all'istruzione associata
- Solitamente il caso di default è messo alla fine dell'istruzione **switch**
- Se non è presente il caso di **default** e nessun valore corrisponde, il controllo passa all'istruzione successiva all'istruzione **switch**

AA 2005/06 © Alberti 23 Programmazione 8. Selezione

### Istruzione switch - 5

- L'espressione di un'istruzione **switch** deve produrre un valore di tipo *intero*, cioè un **int** o **character**; non può essere un valore in virgola mobile
- Notare che la condizione implicita nell'istruzione **switch** è l'uguaglianza – si cerca di uguagliare il valore di un caso
- Non si possono eseguire condizioni di relazione

AA 2005/06 © Alberti 24 Programmazione 8. Selezione

### Istruzione switch - 6

```
int digit; ...
switch (digit) {
    case 1: System.out.println ("uno");
            break;
    case 2: System.out.println ("due");
            break;
    case 3: System.out.println ("tre");
            break;
    default: System.out.println ("errore");
}

```

AA2005/06  
© Alberti

25

Programmazione  
8. Selezione

### Istruzione switch - 7

Ma non è possibile:

```
String parola; ...
switch (parola) {
    case "uno": System.out.println
                ("1"); break;
    case "due": System.out.println
                ("2"); break;
    ...
}

```

AA2005/06  
© Alberti

26

Programmazione  
8. Selezione

### Istruzione switch – 8

L'enunciato `switch` è equivalente a diversi `if` innestati

```
int digit; ...
if (digit == 1)
    System.out.println ("uno");
else if (digit == 2)
    System.out.println ("due");
else if (digit == 3)
    System.out.println ("tre");
else System.out.println ("errore");

```

AA2005/06  
© Alberti

27

Programmazione  
8. Selezione

### Esempi con istruzione switch

- [Enunciato\\_switch.java](#)
- [GradeReport.java](#)
  - Esempi con uso dell'istruzione `break` in ogni caso
- [Conta\\_giorni.java](#)
  - Esempio senza l'uso dell'istruzione `break` in ciascun caso

AA2005/06  
© Alberti

28

Programmazione  
8. Selezione