

DECISIONI

L'ENUNCIATO IF

- Enunciato **composto**
- Sintassi:

```
if (condizione) {  
    // blocco enunciati 1  
} else {  
    // blocco enunciati 2  
}
```

CONFRONTO TRA NUMERI

OPERATORI RELAZIONALI

- `|` , `>=` , `<` , `<=` , `==` , `!=`

CONFRONTO TRA NUMERI IN VIRGOLA MOBILE

- Possibili errori di arrotondamento e troncamento
- Prevedere una tolleranza:
 - $|x - y| < \epsilon$
 - $|x - y| < \epsilon * \max(|x|, |y|)$
 - con $\epsilon = 1 * 10^{-14}$

```
final double EPSILON = 1E-14;  
if (Math.abs(x-y) <= EPSILON * Math.max(Math.abs(x), Math.abs(y))) {  
    // ...  
}
```

CONFRONTO TRA STRINGHE

- Si utilizzano dei metodi

```
if (s1.equals(s2))  
if (s1.equalsIgnoreCase(s2))
```

⚡ Attenzione

- NON si usa mai `==` perché confronta indirizzi di memoria, non oggetti
- è lecito usare `s1 == null` per verificare se si riferisce a null o meno

ORDINAMENTO LESSICOGRAFICO

- Metodo `compareTo()`

```
if (s1.compareTo(s2) < 0)
```

- Restituisce un valore int in base a ordinamento lessicografico (UNICODE)
 - negativo: s1 precede s2
 - positivo: s1 segue s2
 - zero: s1 identico a s2
- N.B. spazio bianco < numeri < lettere

CONFRONTARE OGGETTI

- Metodo `equals()` si può applicare a qualsiasi oggetto
 - affinché funzioni bene bisognerebbe ridefinire il metodo nelle proprie classi
- Metodo `compareTo()` si può applicare in altri oggetti oltre a String
 - segue sempre lo stesso schema di return type in base all'ordinamento

OPERATORI BOOLEANI O LOGICI

- Principali: `&&`, `||`, `!`
- **Valutazione pigra** (o cortocircuito)
 - la valutazione termina appena è possibile decidere il risultato
- Ordine operatori (da quelli eseguiti prima a dopo)
 - parentesi
 - `!`
 - operazioni matematiche
 - maggiore/minore
 - `&&`
 - `||`
 - `=`

LEGGI DI DE MORGAN

```
!(A && B) == !A || !B  
!(A || B) == !A && !B
```

ENUNCIATO SWITCH

```
int x;  
int y;  
  
switch (x) {  
    case 1: y = 1; break;  
    case 2: y = 4; break;  
    default: y = 0; break;  
}
```