

# COSTRUTTO SWITCH

- Il costrutto **if-else** permette di selezionare le istruzioni da eseguire tra due alternative, a seconda del valore di una condizione booleana (vero o falso).
- Nel caso si desideri, selezionare le istruzioni tra più di due alternative, si dovrebbe utilizzare una catena di **if-else**.

```
int a;
...
if (a==1) {
    System.out.println("a vale 1");
} else if (a==2) {
    System.out.println("a vale 2");
} else if (a==3) {
    System.out.println("a vale 3");
} else {
    System.out.println("caso non previsto");
}
```

- Il linguaggio java mette a disposizione un costrutto più efficiente per realizzare una selezione tra più di due alternative: il costrutto **switch**

- **switch** (<espressione>) {  
 **case** <costante-1>:  
 <statements-1>;  
 **break**;  
 **case** <costante-2>:  
 <statements-2>;  
 **break**;  
 ... // altri case...  
 **case** <costante-N>:  
 <statements-N>;  
 **break**;  
 **default**: // opzionale...  
 <statements-(N+1)>;  
} // fine switch...

- L'espressione viene valutata e si eseguono le istruzioni (statements) che seguono la clausola case associata al valore dell'espressione valutata.
- L'istruzione **break** permette di uscire dal blocco **switch**, se tale istruzione viene omessa, vengono eseguite tutte le istruzioni che seguono il **case** selezionato (anche quelle dei case successivi).
- Se per il valore dell'espressione valutata non è previsto in nessuna clausola **case**, si usa la clausola **default**.

- Il tipo della espressione valutata **DEVE** essere **INTERO**: byte, char, short oppure int.
- Il tipo della espressione valutata **NON DEVE** essere **REALE**: float oppure double.
- Le costanti nelle clausole case devono essere costanti **LETTERALI** (literals): 1, 2, 3..
- **Nota**: Gli statements nelle clausole possono essere una qualunque sequenza di istruzioni, quindi è possibile anche annidare dei blocchi if dentro al costrutto switch

Esempio di utilizzo:

```
int a; ...

switch (a) {
    case 1:
        System.out.println("a vale 1");
        break;
    case 2:
        System.out.println("a vale 2");
        break;
    case 3:
        System.out.println("a vale 3");
        break;
    default:
        System.out.println("caso non previsto");
}
```

- La variabile `a` viene valutata e si eseguono le istruzioni che seguono la clausola `case` associata al valore della variabile.
- L'istruzione `break` permette di uscire dal blocco `switch`, se tale istruzione viene omessa, vengono eseguite tutte le istruzioni che seguono il `case` desiderato (anche quelle dei case successivi).
- Se il valore di `a` non è previsto in nessuna clausola `case`, si usa la clausola `default`.

Un altro esempio di utilizzo.

```
char c; ...

switch (c) {
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u':
        System.out.println("c è una vocale");
        break;
    default:
        System.out.println("c è una consonante");
}
```

Es 1:

Programma Java che chiede un intero in ingresso e lo stampa in lettere se il valore è compreso tra 1 e 3 , altrimenti stampa un messaggio d'errore.

Es2:

Scrivere un programma che chiede in ingresso il giorno e il mese e stampa il numero dei giorni che sono trascorsi dall'inizio dell'anno fino alla data inserita. Il programma deve verificare la validità della data inserita, stampando nel caso un messaggio d'errore.

L'anno va considerato non bisestile (febbraio ha 28 giorni)

Esempio: 5 10 (5 Ottobre) → 278 giorni Esempio: 33 4 (33 Aprile) → data non valida Esempio: 3 14 (3 ?) → data non valida

Es3:

Scrivere un programma che chiede di immettere da tastiera due numeri reali a e b, ed un'operazione a scelta fra +, -, \*, /, ^, L, R visualizzando il risultato dell'operazione. L'operazione indicata con 'L' dovrà calcolare il logaritmo di a in base b, mentre quella indicata con 'R' dovrà calcolare la radice a-esima di b.

Elevamento a potenza: `Math.pow(a, b)`

Logaritmo in base e: `Math.log(a)`

$\log_b a = \log_e a / \log_e b$