



Costrutto switch

Come utilizzare il costrutto switch

Costrutto switch (1/4)

- 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");  
    }  
}
```

- Nell'esempio, le diverse alternative sono selezionate in base al valore della variabile `a` tramite una successione di confronti

Costrutto switch (2/4)

- 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**.

Costrutto switch (3/4)

- 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**.

Costrutto switch (4/4)

- 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**.

Costrutto switch (5/4)

- 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");  
   }
```

- In questo caso, non ci sono istruzioni dopo le clausole case 'a', 'e', 'i', 'o', quindi viene eseguita l'istruzione che segue la clausola case 'u' (non essendo presenti istruzioni break)

Esempio “inLettere”

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.

```
import fiji.io.Lettore;

public class inLettere {
    public static void main(String[] args) {
        int a;
        System.out.println("Inserisci il valore");
        a = Lettore.in.leggiInt();

        switch(a){
            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("Numero non valido");
        }
    }
}
```


Esercizio “Data”

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

Soluzione (1/6)

```
import fiji.io.Lettore;

public class Data {

    public static void main(String[] args) {

        int giorno, mese;
        int trascorsi = 0;
        /* La variabile valido indica se la data è valida. Viene
         * inizializzata a true e posta a false nel caso la data
         * non sia valida. */
        boolean valido = true;

        /* Input dei dati */
        System.out.println("Inserisci il giorno");
        giorno = Lettore.in leggiInt();
        System.out.println("Inserisci il mese");
        mese = Lettore.in leggiInt();
```

Soluzione (2/6)

```
/* Controllo della validità e calcolo
 * dei giorni trascorsi */
switch(mese){
case 1:
    if(giorno > 0 && giorno < 32){
        trascorsi = giorno;
    }else{
        valido = false;
    }
    break;
case 2:
    if(giorno > 0 && giorno < 29){
        trascorsi = 31 + giorno;
    }else{
        valido = false;
    }
    break;
```

Soluzione (3/6)

case 3:

```
    if(giorno > 0 && giorno < 32){  
        trascorsi = 28 + 31 + giorno;  
    }else{  
        valido = false;  
    }  
    break;
```

case 4:

```
    if(giorno > 0 && giorno < 31){  
        trascorsi = 31 + 28 + 31 + giorno;  
    }else{  
        valido = false;  
    }  
    break;
```

case 5:

```
    if(giorno > 0 && giorno < 32){  
        trascorsi = 30 + 31 + 28 + 31 + giorno;  
    }else{  
        valido = false;  
    }  
    break;
```

Soluzione (4/6)

case 6:

```
if(giorno > 0 && giorno < 31){
    trascorsi = 31 + 30 + 31 + 28 + 31 + giorno;
}else{
    valido = false;
}
break;
```

case 7:

```
if(giorno > 0 && giorno < 32){
    trascorsi = 30 + 31 + 30 + 31 + 28 + 31
               + giorno;
}else{
    valido = false;
}
break;
```

case 8:

```
if(giorno > 0 && giorno < 32){
    trascorsi = 31 + 30 + 31 + 30 + 31 + 28
               + 31 + giorno;
}else{
    valido = false;
}
break;
```

Soluzione (5/6)

case 9:

```
if(giorno > 0 && giorno < 31){
    trascorsi = 31 + 31 + 30 + 31 + 30 + 31
              + 28 + 31 + giorno;
}else{
    valido = false;
}
break;
```

case 10:

```
if(giorno > 0 && giorno < 32){
    trascorsi = 30 + 31 + 31 + 30 + 31 + 30
              + 31 + 28 + 31 + giorno;
}else{
    valido = false;
}
break;
```

case 11:

```
if(giorno > 0 && giorno < 31){
    trascorsi = 31 + 30 + 31 + 31 + 30 + 31
              + 30 + 31 + 28 + 31 + giorno;
}else{
    valido = false;
}
break;
```

Soluzione (6/6)

```
case 12:
    if(giorno > 0 && giorno < 32){
        trascorsi = 30 + 31 + 30 + 31 + 31 + 30
                   + 31 + 30 + 31 + 28 + 31 + giorno;
    }else{
        valido = false;
    }
    break;
default:
    valido = false;
}

/* Output dei giorni trascorsi */
if(valido){
    System.out.println("Sono trascorsi " + trascorsi
                       + " giorni dall'inizio dell'anno");
}else{
    System.out.println("Data non valida");
}
}
}
```

Variante (1/3)

```
import fiji.io.Lettore;

public class Data2 {

    public static void main(String[] args) {

        int giorno, mese;
        int trascorsi = 0;
        /* La variabile valido indica se la data è valida. Viene
         * inizializzata a true e posta a false nel caso la data
         * non sia valida. */
        boolean valido = true;

        /* Input dei dati */
        System.out.println("Inserisci il giorno");
        giorno = Lettore.in.leggiInt();
        System.out.println("Inserisci il mese");
        mese = Lettore.in.leggiInt();
    }
}
```


Variante (2/3)

```
/* Controllo della validità della data */
if(giorno > 0 && giorno < 32 && mese > 0 && mese < 13 ){
    switch(mese){
        case 4:
        case 6:
        case 9:
        case 11:
            if (giorno > 30){
                valido = false;
            }
            break;
        case 2:
            if (giorno > 28){
                valido = false;
            }
            break;
    }
}else{
    valido = false;
}
```

Variante (3/3)

```
/* calcolo dei giorni trascorsi ed output */
if(valido){
    switch (mese) {
        case 12:    trascorsi += 30;
        case 11:    trascorsi += 31;
        case 10:    trascorsi += 30;
        case 9:     trascorsi += 31;
        case 8:     trascorsi += 31;
        case 7:     trascorsi += 30;
        case 6:     trascorsi += 31;
        case 5:     trascorsi += 30;
        case 4:     trascorsi += 31;
        case 3:     trascorsi += 28;
        case 2:     trascorsi += 31;
        case 1:     trascorsi += giorno;
    }
    System.out.println("Sono trascorsi " + trascorsi
        + " giorni dall'inizio dell'anno");
}else{
    System.out.println("Data non valida");
}
}
}
```

Esercizio “Calcolatrice”

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)`

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

Soluzione (1/4)

```
import fiji.io.Lettore;
public class Calcolatrice {
    public static void main(String[] args) {
        double a;    // Primo operando
        double b;    // Secondo operando
        char op;     // Operazione

        /* Input a, b, op */
        System.out.println("Inserisci il primo operando");
        a = Lettore.in.leggiDouble();
        System.out.println("Inserisci il secondo operando");
        b = Lettore.in.leggiDouble();
        System.out.println("Operazione? (+ - / * ^ L R)");
        op = Lettore.in.leggiChar();

        /* La variabile ok indica la corretta esecuzione
         * dell'operazione. Viene inizializzata a true, e posta
         * a false nel caso l'operazione non sia calcolabile
         * (es. divisione per zero) */
        boolean ok = true;
        double r = 0;    // Risultato dell'operazione
    }
}
```

Soluzione (2/4)

```
/* Elaborazione: calcolo del risultato */
switch (op) {
case '+':
    r = a + b;
    break;
case '-':
    r = a - b;
    break;
case '/':
    if(b == 0){
        ok = false;
    }else{
        r = a / b;
    }
    break;
case '*':
    r = a * b;
    break;
```

Soluzione (3/4)

```
case '^':
    if(a <= 0){
        ok = false;
    }else{
        r = Math.pow(a, b);
    }
    break;
case 'L':
    /* Per semplicità si controlla soltanto che i due
     * operandi siano positivi e che la base non sia 1 */
    if(a <= 0 || b <= 0 || b==1){
        ok = false;
    }else{
        r = Math.log(a) / Math.log(b);
    }
    break;
case 'R':
    /* Per semplicità si controlla soltanto che i due
     * operandi siano positivi */
    if(a <= 0 || b <= 0){
        ok = false;
    }else{
        r = Math.pow(b, 1.0 / a);
    }
    break;
```

Soluzione (4/4)

```
default:
    ok = false;
}
/* Output r, ok */
if(ok){
    System.out.println("Il risultato di " + a + op + b
        + " è " + r);
}else{
    System.out.println("Errore: " +
        "Impossibile eseguire l'operazione");
}
}
}
```