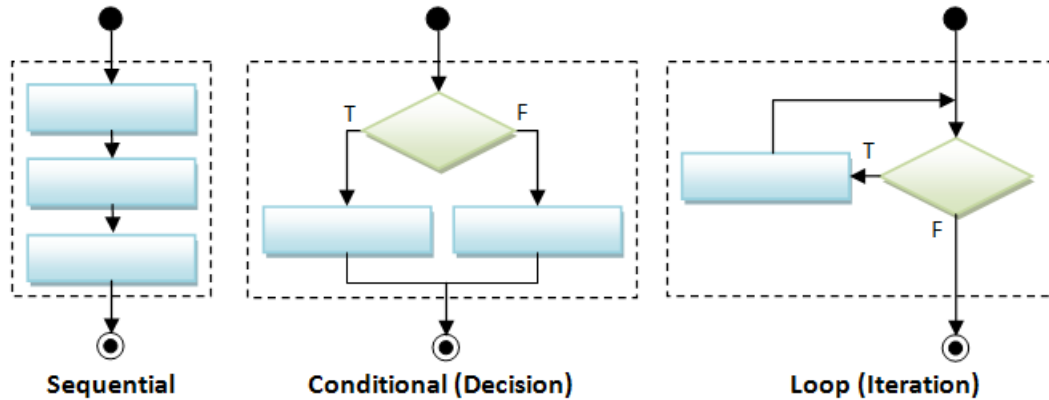


## Estructures en java



### Estructura alternativa o condicional

Una condició és una expressió que pot ser vertadera o falsa. La **sentència if** permet a un programa prendre una decisió sobre la base d'una condició de valor.

### Operadors relacionals

Les condicions en les sentències **if** es poden formar utilitzant:

- els operadors d'igualtat (`==` i `!=`)
- i els operadors relacionals (`>`, `<`, `>=` i `<=`).

Es resumeixen a la següent taula:

| Operador           | Exemple d'us           | Significat             |
|--------------------|------------------------|------------------------|
| <code>==</code>    | <code>z == y</code>    | x és igual a y         |
| <code>!=</code>    | <code>x != y</code>    | x no és igual a y      |
| <code>&gt;</code>  | <code>x &gt; y</code>  | x és major que y       |
| <code>&lt;</code>  | <code>x &lt; y</code>  | x és menor que y       |
| <code>&gt;=</code> | <code>x &gt;= y</code> | x és major o igual a y |
| <code>&lt;=</code> | <code>x &lt;= y</code> | x és menor o igual a y |

## 1. Relació simple = Sentència if

Una declaració de **if** espera un *statement* en el seu cos, però pot contenir diversos *statements* si estan tancats en un conjunt de claus (**{}**)

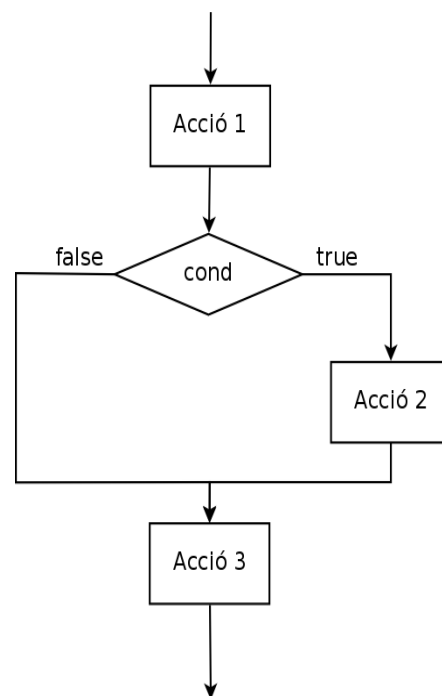
Per exemple:

```
if (number1 > 10) {  
    System.out.println("El número és més gran que 10");  
    System.out.println("-----");  
}
```

```
accio1;  
if (cond) {  
    accio2;  
}  
accio3;
```

Exemple:

```
System.out.println("Inici");  
if (edat < 18) {  
    System.out.println("adolescent");  
}  
System.out.println("Final");
```



```

Compara.java
1 package exemple5;
2
3 import java.util.Scanner;
4
5 /**
6  * Programa d'exemple d'us dels diferents operadors
7  * relacionals i de les sentències if i if...else
8  *
9  */
10 public class Compara {
11
12     public static void main(String[] args) {
13         // Crea l'objecte Scanner
14         Scanner input = new Scanner(System.in);
15
16         //Defineixo les variables que faré servir
17         int number1;
18         int number2;
19
20         // Li demana a l'usuari que entri un numero
21         System.out.print("Introdueix el primer número: ");
22
23         //Es llegeix el 1r numero
24         number1 = input.nextInt() ;
25
26         // Li demana a l'usuari que entri un numero
27         System.out.print("Introdueix el segon número: ");
28
29         //Es llegeix el 2n numero
30         number2 = input.nextInt() ;
31
32         //Es fan una sèrie de comparacions
33         if ( number1 == number2 )
34             System.out.println( "Els dos números són iguals");
35
36         if ( number1 != number2 )
37             System.out.println( "Els dos números són diferents");
38
39         if ( number1 < number2 )
40             System.out.println( "Els primer número és més petit que el segon");
41
42         if ( number1 > number2 )
43             System.out.println( "Els primer número és més gran que el segon");
44
45         if ( number1 <= number2 )
46             System.out.println( "Els primer número és més petit o igual que el segon");
47
48         if ( number1 >= number2 )
49             System.out.println( "Els primer número és més gran o igual que el segon");
50     }
51 }

```

A les línies 33 i 34 es compara els valors de number1 i number2 per determinar si són iguals. Una sentència **if** sempre comença amb la paraula clau **if**, seguida d'una condició entre parèntesis.

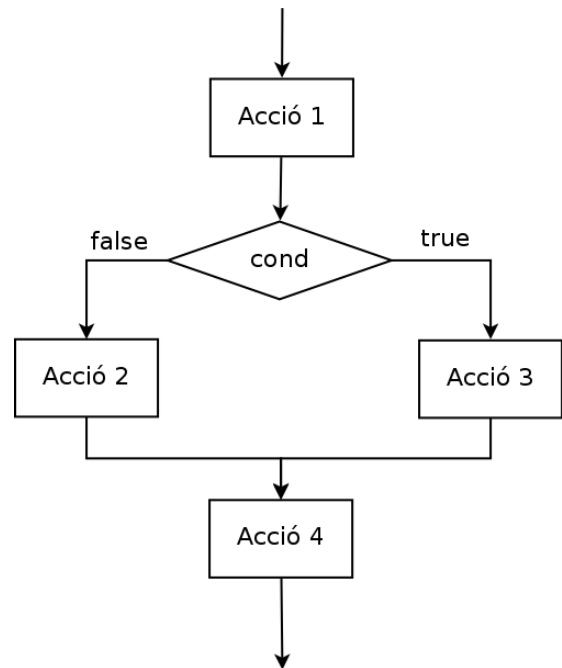
El programa conté sis sentències **if** per comparar dos nombres enters d'entrada per l'usuari. Si la condició en qualsevol d'aquestes sentències és veritat, la sentència associada a aquest comparació s'executa, en cas contrari, s'omet.

## 2. Relació doble = Sentència If...else

```
accio1;
if (cond) {
    accio3;
} else {
    accio2;
}
accio4;
```

Exemple:

```
System.out.println("Inici");
if (edat<18) {
    System.out.println("Menor d'edat");
}
else {
    System.out.println("Major d'edat");
}
System.out.println("Final");
```



### Sentència If...else

La sentència **if** realitza una (o més accions) si es compleix la condició. En cas contrari, no es fa cap de les accions. Amb una sentència del tipus **if...else** es poden especificar accions a realitzar quan es satisfà la condició i accions a realitzar quan no es satisfà.

Per exemple: No cal fer servir claus si només tenim una línia per la condició **if** o per la condició **else**.

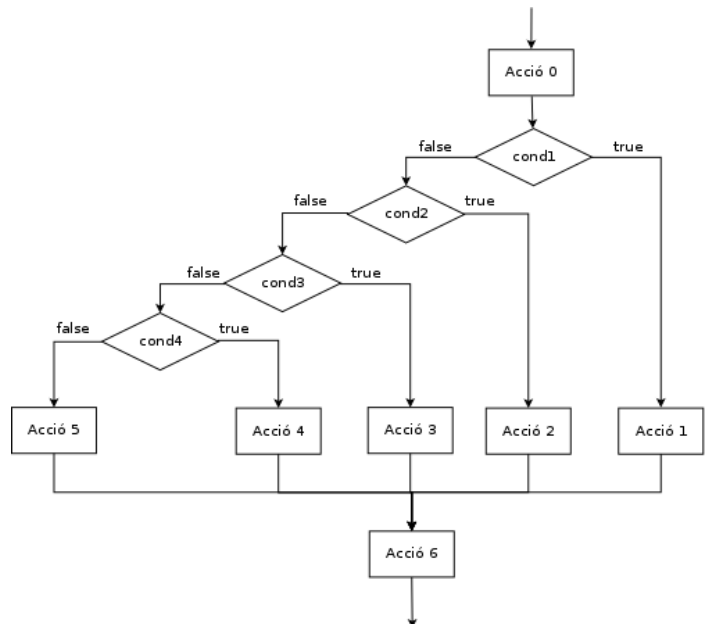
```
if (number < 0)
    System.out.println("El número és negatiu");
else
    System.out.println("El número és positiu o zero");
```

Si tenim més d'una línia s'ha de fer servir les claus:

```
if (number < 0) {
    System.out.println("El número és negatiu");
    System.out.println("-----");
} else {
    System.out.println("El número és positiu o zero");
    System.out.println("-----");
}
```

### 3. Múltiple = Sentència if ... else if ... else:

```
accio0;
if (cond1) {
    accio1;
} else if (cond2) {
    accio2;
} else if (cond3) {
    accio3;
} else if (cond4) {
    accio4;
} else {
    accio5;
}
accio6;
```

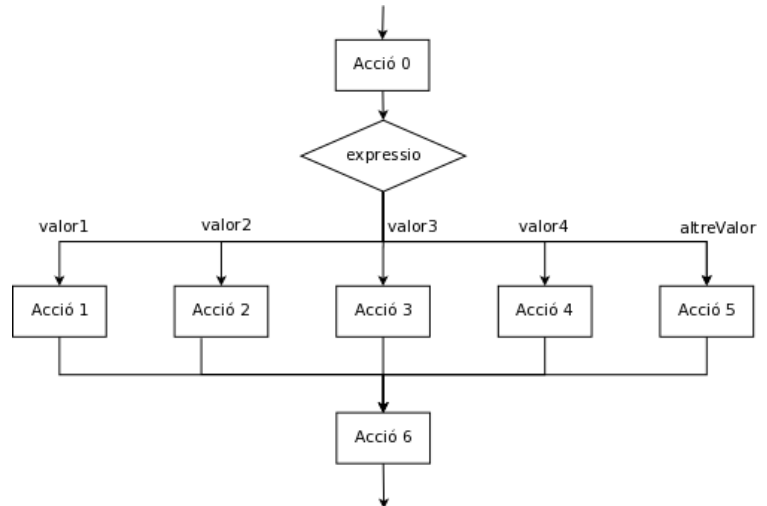


Exemple:

```
System.out.println("Inici");
if ((edat<6) {
    System.out.println("Infantil");
}
else if (edat<18) {
    System.out.println("Adolescent");
}
else if (edat<24) {
    System.out.println("Major d'edat");
}
else if (edat<30) {
    System.out.println("Adult");
}
else if (edat<40) {
    System.out.println("Madur");
}
else if (edat<65) {
    System.out.println("Gran");
}
else {
    System.out.println("Ets el millor");
}
System.out.println("Fi");
```

## 4. Múltiple amb switch

```
accio0;
switch (expressio) {
    case valor1:
        accio1;
        break;
    case valor2:
        accio2;
        break;
    case valor3:
        accio3;
        break;
    case valor4:
        accio4;
        break;
    default:
        accio5;
        break;
}
accio6;
```



Exemple:

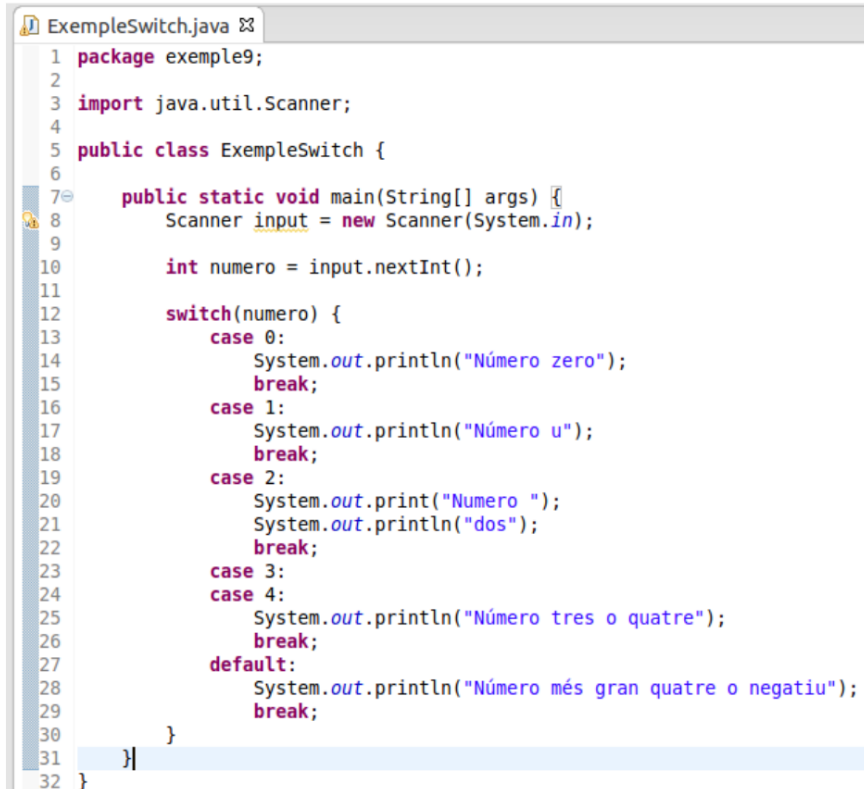
```
Scanner entrada = new Scanner(System.in);
System.out.println("Escull un número del 1 al 7");
int day=entrada.nextInt();
switch (day) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    case 3:
        System.out.println("Wednesday");
        break;
    case 4:
        System.out.println("Thursday");
        break;
    case 5:
        System.out.println("Friday");
        break;
    case 6:
        System.out.println("Saturday");
        break;
    case 7:
        System.out.println("Sunday");
        break;
    default:
        System.out.println("Opció no valida");
}
```

Com a alternativa a les sentències **if...else** nidades, tenim el **switch**.

Amb un **switch** es poden fer diferents accions basant-se en els possibles valors d'una expressió constant del tipus bàsic **byte**, **short**, **int** o **char**. També es pot fer amb un tipus especial **String** i **Enum** (aquest últim el veurem més endavant).

Nota: Des de la versió Java 7 es pot fer amb objectes **String**.

Exemple de switch:

A screenshot of a Java IDE window titled 'ExempleSwitch.java'. The code is as follows:

```
1 package exemple9;
2
3 import java.util.Scanner;
4
5 public class ExempleSwitch {
6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9
10        int numero = input.nextInt();
11
12        switch(numero) {
13            case 0:
14                System.out.println("Número zero");
15                break;
16            case 1:
17                System.out.println("Número u");
18                break;
19            case 2:
20                System.out.print("Numero ");
21                System.out.println("dos");
22                break;
23            case 3:
24            case 4:
25                System.out.println("Número tres o quatre");
26                break;
27            default:
28                System.out.println("Número més gran quatre o negatiu");
29                break;
30        }
31    }
32 }
```

Observacions:

- A sota de cada condició (**case**) pot haver-hi 0, 1 o més d'una sentència. Per exemple, al case 2 hi ha dues línies. No cal posar-hi claus.
- Amb un **break** es surt de la sentència **switch**, però no és obligatori. Per exemple, al **case 3** (línia 23) no hi ha cap **break**. Si no hi ha **break**, significarà que realitzarà també el següent **case** encara que no sigui certa la condició.
- El **break** de la línia 29 no és necessari, però és una bona pràctica de programació, per si s'afegeixen condicions.
- També és una bona pràctica posar sempre un **default**, encara que sigui per generar un error de valor no contemplat.

El **break** serveix perquè després d'executar una acció (**case**), salti fora del **switch**, sinó s'utilitza el break, es realitzen totes les accions que hi ha a sota (només les inferiors). Per exemple si entrem *opcio* = 2, es realitzaran els casos 2, 3 i **default**:

```
import java.util.Scanner;
public class A11b_Switch_case_no_break {
    public static void main(String args[]) {
        Scanner entrada=new Scanner(System.in);
        System.out.println("Escull una opció: 1, 2 o 3");
        int opcio=entrada.nextInt();

        switch (opcio) {
            case 1:
                opcio++;
                System.out.println("Case1: Value is: " + opcio);
            case 2:
                opcio++;
                System.out.println("Case2: Value is: " + opcio);
            case 3:
                opcio++;
                System.out.println("Case3: Value is: " + opcio);
            default:
                opcio++;
                System.out.println("Default: Value is: " + opcio);
        }
    }
}
```

Output:

```
Case2: Value is: 3
Case3: Value is: 4
Default: Value is: 5
```



## 4.1 Expressions vàlides per switch:

El **switch** funciona amb tipus de dades primitius **byte**, **short**, **char**, i **int**. També funciona amb dades de tipus **enum** i **String**,

```
String month = "Janauary";  
switch (month) {  
    ...  
}
```

```
char ch='b';  
switch(ch) {  
    ...  
}
```

```
switch(1+2+23){  
    ...  
}
```

```
switch(1 * 2 + 3 % 4){  
    ...  
}
```

```
short a = 10;  
short b = 15;  
switch(a + b){  
    ...  
}
```

```
byte x = 3;  
byte y = 19;  
byte z = -1;  
switch(x + y + z){  
    ...  
}
```

```
switch(x * y + z % y){  
    ...  
}
```

## 5. Operadors lògics

Amb els operadors lògics es poden fer definir condicions més complexes combinant condicions simples.

El operadors lògics són:

- **&&** (operador AND)
- **||** (operador OR)
- **!** (operador NOT)

Hi ha altres operadors que operen a nivell de bits però que, per ara, no veurem.

### 5.1 Operador &&

Controla que les dues condicions són veritat. Per exemple:

```
if(sexe == 'D' && edat >= 65)
    donesJubilades = donesJubilades + 1;
```

Es poden posar parèntesis i/o claus per ajudar a la lectura del codi:

```
if ((sexe == 'D') && (edat >= 65)) {
    donesJubilades = donesJubilades + 1;
}
```

Taula de la veritat de &&:

| expression1 | expression2 | expression1 && expression2 |
|-------------|-------------|----------------------------|
| false       | false       | false                      |
| false       | true        | false                      |
| true        | false       | false                      |
| true        | true        | true                       |

## 5.2 Operador ||

Controla que les una o totes dues condicions són veritat. Per exemple:

```
if(convocatoriaOrdinaria >= 5 || convocatoriaExtraordinaria >= 5)
    System.out.println("L'alumne ha aprovat!");
```

Taula de la veritat de ||:

| expression1 | expression2 | expression1    expression2 |
|-------------|-------------|----------------------------|
| false       | false       | false                      |
| false       | true        | true                       |
| true        | false       | true                       |
| true        | true        | true                       |

## 5.3 Operador !

Serveix per capgirar el significat de la condició.

```
if(!(edat >= 18))
    System.out.println("Ets menor d'edat!");
}
```

Taula de la veritat de !:

| expression | !expression |
|------------|-------------|
| false      | true        |
| true       | false       |

## Llicència

Copyright (C) Joan Sèculi (joan.seculi@gmail.com)

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is in <http://www.gnu.org/licenses/fdl.txt>