

# Les tests et les conditions en Java

## I - Le test de base

### I - 1 - Le test simple

Un test permet de savoir si la valeur d'une variable est égale à un nombre particulier, et d'exécuter un morceau de code spécifique. Le programme suivant, par exemple, teste si la variable **ma\_variable** est égale à 30, et affiche un message si le test est positif. Essayez ce programme dans les 2 cas suivants :

- \* La variable **ma\_variable** est initialisée à la valeur 30 sur la ligne
- \* La variable **ma\_variable** est initialisée à une valeur autre que 30 sur la ligne

```
1 class test {
2     public static void main(String[] args) {
3
4         /* déclaration la variable ma_variable de type int : */
5         int ma_variable;
6
7         /* initialisation de la variable ma_variable : */
8         ma_variable=30;
9
10        /* teste si la variable ma_variable est égale à 30 */
11        if (ma_variable == 30)
12        {
13            System.out.println("ma_variable vaut trente");
14        }
15
16        System.out.println("Fin du programme");
17    }
18 }
```

Code source 1

Remarques à propos du *Code source 1* :

- \* En Java, l'instruction permettant de faire un test est le **if**
- \* Un test d'égalité est fait par l'opérateur **==** [deux signes « égale » à la suite]
- \* Dans un **if**, la condition du test doit être mise entre parenthèses : (ma\_variable == 30) sur la ligne 8
- \* Si la condition est vraie, alors le bloc d'instruction suivant le **if** est exécuté. Ce bloc doit être encadré par des accolades [commençant ligne 9 et finissant ligne 11 dans le *Code source 1*], et peut contenir plusieurs lignes à exécuter

### I - 2 - Le test complet

Dans le Code source 2 suivant, un message différent est affiché, et ce, que la variable soit égale ou différente à 30 :

```
1 class test {
2     public static void main(String[] args) {
3         int ma_variable;
4         ma_variable=30;
5
6         if (ma_variable == 30)
7         {
8             System.out.println("ma_variable vaut trente");
9         }
10        else
11        {
12            System.out.println("ma_variable ne vaut pas trente");
13        }
14    }
15 }
```

Code source 2

Remarques à propos du *Code source 2* :

- \* La condition placée dans le **if** sur la ligne 5 est « **la variable ma\_variable est égale à 30** »
- \* Si cette condition est **VRAIE**, alors le bloc placé entre accolade commençant à la ligne 6 et finissant à la ligne 8 est exécuté
- \* Si cette condition est **FAUSSE**, alors le bloc placé entre accolade commençant à la ligne 10 et finissant à la ligne 12 est exécuté
- \* L'instruction **else**, placée sur la ligne 9, veut dire « **sinon** », et désigne le bloc à exécuter si la condition du **if** est **FAUSSE**
- \* Pour bien observer le fonctionnement et le déroulement du *Code source 2*, on pourra l'exécuter pas à pas, à l'aide du débogueur de JBuilder.

## II - Les différents tests possibles

Nous venons de voir, à travers les Code source 1 et Code source 2 comment effectuer un test d'égalité : il faut utiliser l'opérateur `==`. Java permet de faire d'autres test, en utilisant les opérateurs de comparaison mentionnés dans le tableau suivant :

Les opérateurs de comparaison en Java		
Opérateur	Signification	Exemple
<code>==</code>	égal à	<code>if (i == 5)</code>
<code>!=</code>	différent de	<code>if (i != 5)</code>
<code>&lt;</code>	inférieur à	<code>if (i &lt; 5)</code>
<code>&lt;=</code>	inférieur ou égal à	<code>if (i &lt;= 5)</code>
<code>&gt;</code>	supérieur à	<code>if (i &gt; 5)</code>
<code>&gt;=</code>	supérieur ou égal à	<code>if (i &gt;= 5)</code>

Le *Code source 3* suivant teste la variable `i` par rapport à 10, en utilisant chacun des 6 opérateurs de comparaison, et affiche un message spécifique en fonction des résultats des tests :

```
1 class test {
2     public static void main(String[] args) {
3         int i;
4         i=8;
5
6         if (i == 10) { System.out.println("i égal à 10"); }
7         else { System.out.println("i pas égal à 10"); }
8
9         if (i != 10) { System.out.println("i différent de 10"); }
10        else { System.out.println("i pas différent de 10"); }
11
12        if (i < 10) { System.out.println("i inférieur à 10"); }
13        else { System.out.println("i pas inférieur à 10"); }
14
15        if (i <= 10) { System.out.println("i inf. ou égal à 10"); }
16        else { System.out.println("i pas inf. ou égal à 10"); }
17
18        if (i > 10) { System.out.println("i supérieur à 10"); }
19        else { System.out.println("i pas supérieur à 10"); }
20
21        if (i >= 10) { System.out.println("i sup. ou égal à 10"); }
22        else { System.out.println("i pas sup. ou égal à 10"); }
23    }
24 }
```

*Code source 3*

Remarques à propos du *Code source 3* :

- \* Dans le *Code source 3*, les bloc associés aux différents **if** et aux différents **else** ont été écrits sur une seule ligne : ces blocs sont toujours encadrés par des accolades
- \* Cette manière d'écrire le code source est pratique pour diminuer le nombre de lignes du programme, mais réduit considérablement sa lisibilité

### III - Utilisation des opérateurs logiques dans les conditions de test

Pour formuler des conditions de test complètes, précises, et de complexité quelconques, Java propose l'emploi des opérateurs logiques. Ces opérateurs sont :

- \* Le **ET** logique
- \* Le **OU** logique
- \* Le **NON** logique

Les opérateurs logiques en Java		
Opérateur	Signification	Exemple
&&	ET logique	if ((i==5) && (j==8))
	OU logique	if ((i>10)    (i<2))
!	NON logique	if !(i==9)

Rappel des tables de vérité des opérateurs logiques de base :

Opérateur ET		
a	b	a && b
faux	faux	<b>FAUX</b>
faux	vrai	<b>FAUX</b>
vrai	faux	<b>FAUX</b>
vrai	vrai	<b>VRAI</b>

Opérateur OU		
a	b	a    b
faux	faux	<b>FAUX</b>
faux	vrai	<b>VRAI</b>
vrai	faux	<b>VRAI</b>
vrai	vrai	<b>VRAI</b>

Opérateur NON	
a	!a
faux	<b>VRAI</b>
vrai	<b>FAUX</b>

A retenir :

- \* Le résultat d'un **ET** est **VRAI** si et seulement si les deux conditions liant le **ET** sont **VRAIES** toutes les deux
- \* Le résultat d'un **OU** est **FAUX** si et seulement si les deux conditions liant le **OU** sont **FAUSSES** toutes les deux

Exemple de programme utilisant les opérateurs logiques pour formuler des conditions dans les tests :

```
1 class logique {
2     public static void main(String[] args) {
3         int a,b;
4         a=5;
5         b=8;
6
7         if ((a == 10) && (b == 10))
8             {
9                 System.out.println("a et b sont tous les deux égaux à dix");
10            }
11
12        if ((a == 5) && !(b == 5))
13            {
14                System.out.println("a est égal à 5 et b est différent de 5");
15            }
16
17        if ((a > 2) && (a < 14))
18            {
19                System.out.println("a est compris dans l'intervalle ]2;14[");
20            }
21
22        if (!((b >= 10) && (b <= 20)))
23            {
24                System.out.println("b n'est pas dans l'intervalle [10;20]");
25            }
26    }
27 }
```

Code source 4

Pour exécuter le *Code source 4*, donnez des valeurs aux variable a et b afin que chacun des 4 tests soit vrai à tour de rôle.

#### **IV - Propriétés logiques des conditions**

Pour tester une condition, il y a souvent plusieurs manière d'écrire ce test. Par exemple, pour tester si une variable est différentes d'une valeur, on peut soit utilisé l'opérateur de comparaison « différent de », soit utiliser l'opérateur de comparaison « égal à » associé à l'opérateur logique NON, ce qui donne les deux tests suivants qui sont strictement équivalents :

```

/* teste la différence : */
if (a != 5)
{
    System.out.println("a est différent de 5");
}

/* teste la non égalité : */
if !(a == 5)
{
    System.out.println("a n'est pas égal à 5");
}
    
```

Le tableau suivant montre des exemples de condition égales deux à deux. Remarquez que le théorème de De Morgan permet d'expliquer les 4 dernières lignes :

<b>Conditions équivalentes</b>	
<b>La condition</b>	<b>est strictement équivalente à</b>
a==5	!(a!=5)
a!=5	!(a==5)
(a!=5) && (b!=8)	!((a==5)    (b==8))
(a!=5)    (b!=8)	!((a==5) && (b==8))
!((a!=5) && (b!=8))	(a==5)    (b==8)
!((a!=5)    (b!=8))	(a==5) && (b==8)

Le tableau suivant montre des exemples de condition inverses :

<b>Conditions inverses</b>	
<b>Le contraire de</b>	<b>est</b>
a==5	a!=5
a!=5	a==5
a > 5	a <= 5
a >= 5	a < 5
a < 5	a >= 5
a <= 5	a >5
(a==5) && (b==8)	(a!=5)    (b!=8)
(a==5)    (b==8)	(a!=5) && (b!=8)

La maîtrise de l'algèbre de Boole permet généralement de simplifier l'écriture des conditions logiques, rendant ainsi le programme bien plus lisible, et sans en changer son fonctionnement.