

```

1  /*
2  *  Chapitre 05
3  *  Sujet : Les conditions en Java
4  */
5
6  public class Conditions {
7      public static void main(String[] args) {
8          /*
9              Les conditions sont des tests dont le résultat est booléen et donc vaut soit
10             'true' soit 'false'
11             En fonction de la manière dont le test est fait, il peut être nécessaire de
12             mettre la condition entre parenthèses ()
13             On retrouve les conditions dans :
14             - les tests avec 'if',
15             - l'instruction switch,
16             - l'opérateur ternaire,
17             - les boucles (voir Chapitre 08 sur les Boucles)
18             */
19             // Les tests avec 'if' permettent d'exécuter des instructions en fonction du
20             // résultat du test
21
22             // 1. Les tests simples
23
24             int valeur = 3;
25
26             // Condition : est plus grand que '>'
27             if (valeur > 0) {
28                 System.out.println("Valeur est strictement positif");
29             }
30
31             // Condition : est plus grand que ou égale à '>='
32             if (valeur >= 0) {
33                 System.out.println("Valeur est plus grand que ou égal à 0");
34             }
35
36             // Condition : est plus petit que '<'
37             if (valeur < 0) {
38                 System.out.println("Valeur est strictement négatif");
39             }
40
41             // Condition : est plus petit que ou égale à '<='
42             if (valeur <= 0) {
43                 System.out.println("Valeur est plus petit ou égal à 0");
44             }
45
46             // Condition : est égale à '=='
47             if (valeur == 0) {
48                 System.out.println("Valeur est égale à 0");
49             }
50
51             // Condition : n'est pas égale à '!='
52             if (valeur != 0) {
53                 System.out.println("Valeur n'est pas égal à 0");
54             }
55
56             // 2. Les tests composés
57
58             // Test : condition 1 OU condition 2 '||'
59             // Si l'une des conditions est vraie alors le test est vrai
60             // Les conditions sont vérifiées l'une après l'autre
61             // Si la 1ère condition est vraie alors le test est vrai et la 2è condition n'est
62             // pas vérifiée
63
64             if (valeur > 0 || valeur < 0) {
65                 System.out.println("Valeur est soit positif soit négatif mais n'est pas égal
66                 à 0");
67             }
68         }
69     }

```

```

64
65 // Si la 1ère condition n'est pas vraie, alors on teste la suivante et ainsi de
suite
66 if (valeur == 3 || valeur == 2 || valeur == 1) {
67     System.out.println("Valeur est égal à 1 ou 2 ou 3");
68 }
69
70 // Test : condition 1 ET condition 2 '&&'
71 // Toutes les conditions doivent être vraies pour que le test soit vrai
72 // Si l'une des conditions est fausse alors le test s'arrête et est faux
73 if (valeur != 1 && valeur != 2 && valeur > 0) {
74     System.out.println("Valeur est supérieur à 0 mais ne vaut ni 2 ni 3");
75 }
76
77 if (valeur == 3 && valeur != 3) {
78     System.out.println("Cette ligne ne sera pas exécutée");
79 }
80
81 // L'inversion de résultat de test avec '!'
82 // Cela permet d'inverser le résultat. Si le résultat est vrai alors l'inverseur
le transforme en faux
83
84 if (! (valeur != 3)) {
85     System.out.println("Valeur est égal à 3");
86 }
87
88 // 3. Les tests avec 'sinon' = else
89
90 // Si la condition entre parenthèses n'est pas remplie, alors ...
91
92 int heure = 12;
93
94 if (heure < 12) {
95     System.out.println("C'est le matin");
96 } else if (heure > 13) {
97     System.out.println("C'est l'après midi");
98 } else {
99     System.out.println("C'est le midi");
100 }
101
102 // 4. L'opérateur ternaire
103
104 // Si la condition est vraie, la valeur assignée est celle reprise après le
caractère '?'
105 // Si elle est fausse, la valeur retournée est celle reprise après le caractère
':'
106 // L'opérateur ternaire permet donc d'assigner des valeurs distinctes à une
variable en fonction d'un test
107 var s = 5 < 3 ? 'a' : 'b';
108
109 // La ligne si dessus est donc équivalente à la suivante
110 if (5 < 3) {
111     s = 'a';
112 } else {
113     s = 'b';
114 }
115
116 // 5. L'instruction switch
117
118 // Lorsqu'il y a plusieurs tests sur la même variable, le code peut commencer à
devenir difficile à suivre
119
120 int code = 8;
121
122 if (code == 0) {
123     System.out.println("La valeur de 'code' est 0");
124 } else if (code == 1) {
125     System.out.println("La valeur de 'code' est 1");

```

```

126     } else if (code == 2) {
127         System.out.println("La valeur de 'code' est 2");
128     } else if (code == 3 || code == 4) {
129         System.out.println("La valeur de 'code' est 3 ou 4");
130     } else {
131         System.out.println("La valeur de 'code' n'est ni 0, ni 1, ni 2 ni 3, ni 4");
132     }
133
134     // Les instructions précédentes peuvent s'écrire autrement avec l'utilisation
    de switch
135
136     switch (code) { // On va tester la valeur de 'code'
137         case 0:
138             System.out.println("La valeur de 'code' est 0");
139             break;
140         case 1:
141             System.out.println("La valeur de 'code' est 1");
142             break;
143         case 2:
144             System.out.println("La valeur de 'code' est 2");
145             break;
146         case 3:
147         case 4:
148             System.out.println("La valeur de 'code' est 3 ou 4");
149             break;
150         default:
151             System.out.println("La valeur de 'code' n'est ni 0, ni 1, ni 2 ni 3, ni
152         4");
153     }
154     // La valeur de code est comparée avec celle définie juste après 'case'
155     // jusqu'à ce qu'une valeur corresponde, sinon ce sont les instructions qui se
    trouvent
156     // dans la clause 'default' qui seront exécutées
157
158     // L'utilisation de 'break' permet de dire au compilateur qu'il faut sortir de l
    'instruction 'switch',
159     // sinon, les instructions suivantes (des autres 'case' seront également
    exécutées
160
161     // Les tests se font de haut en bas et s'arrêtent dès qu'une correspondance est
    trouvée
162 }
163 }
164

```