

DEV1 – JAVL – Laboratoires Java**TD 3 – Alternatives**

Dans ce TD vous trouverez une introduction aux expressions booléennes et à l'instruction `if/else`.

Les codes sources et les solutions de ce TD se trouvent à l'adresse :

<https://git.esi-bru.be/dev1/labo-java/tree/master/td03-alternatives/>

Table des matières

1	Les alternatives : <code>if/else</code>	2
2	Les conditions	3
3	Exercices supplémentaires	4

1 Les alternatives : if/else

L'instruction `if` permet d'exécuter des instructions si une certaine condition est vérifiée. Le programme suivant affichera "ce nombre est positif" si `nb` est plus grand ou égal à 0 et n'affichera rien dans le cas contraire.

```
Positif.java
1 package esi.dev1.td3;
2
3 import java.util.Scanner;
4
5 public class Positif {
6
7     public static void main(String[] args) {
8         Scanner clavier = new Scanner(System.in);
9
10        System.out.print("Entrez un nombre entier: ");
11        int nb = clavier.nextInt();
12        if(nb>=0) {
13            System.out.print("ce nombre est positif.");
14        }
15    }
16 }
```

L'instruction `if/else` permet d'exécuter des instructions si une certaine condition est vérifiée et d'autres instructions si la condition n'est pas vérifiée. On peut traduire 'else' par 'sinon'.

On peut remplacer le `if` du programme précédent par le `if/else` suivant :

```
if(nb >= 0) {
    System.out.println("ce nombre est positif.");
} else {
    System.out.println("ce nombre est négatif.");
}
```

Le programme affichera "ce nombre est positif" si `nb` est plus grand ou égal à 0 et "ce nombre est négatif" sinon.

Il est aussi possible d'utiliser une succession de `if/else` :

```
if(nb > 0) {
    System.out.println("ce nombre est positif.");
} else if(nb < 0) {
    System.out.println("ce nombre est négatif.");
} else {
    System.out.println("ce nombre est nul.");
}
```

Exercice 1 Majeur - if

Écrivez un programme qui demande à l'utilisateur son âge et affiche s'il est majeur (s'il a plus de 18 ans). S'il n'est pas majeur le programme n'affiche rien.

Exemple : si l'utilisateur entre 19 le programme affiche "vous êtes majeur".

Exercice 2 Pair ou impair - if/else

Écrivez un programme qui demande à l'utilisateur un nombre entier et affiche "ce nombre est pair" ou "ce nombre est impair" selon le cas.

Exemple : si l'utilisateur entre -23 le programme affiche "ce nombre est impair".

Astuce : un nombre est pair si le reste de la division par 2 vaut 0.

Exercice 3 Maximum de 2 nombres

Écrivez un programme qui demande à l'utilisateur deux nombres et affiche le plus grand des deux.

Exemple : si l'utilisateur entre 7,5 et 2,3 le programme affiche 7,5.

2 Les conditions

En informatique, on appelle *booléens* les deux valeurs 'vrai' et 'faux'. En java les booléens se représentent par `true` (vrai) et `false` (faux).

Une condition est une expression dont la valeur s'évalue à `true` ou `false`. Voici quelques exemples de conditions :

- ▷ `age < 18` : vaut `true` si `age` a une valeur strictement inférieure à 18 et `false` sinon ;
- ▷ `nb >= 0` : vaut `true` si `nb` est supérieur à 0 ;
- ▷ `b*b - 4*a*c < 0` : vaut `true` si $b^2 - 4ac$ est strictement négatif ;
- ▷ `(nb >= 0) && (nb <= 100)` : vaut `true` si le nombre `nb` est compris entre 0 et 100 ;
- ▷ `a < b` : vaut `true` si la valeur de la variable `a` est strictement inférieure à celle de `b`.

Pour construire une condition on utilisera les opérateurs de comparaison de nombres :

signification	symbole	exemple
plus petit	<	<code>age < 18</code>
plus petit ou égal	<=	<code>age <= 10</code>
plus grand	>	<code>age > 18</code>
plus grand ou égal	>=	<code>age >= 21</code>
égal	==	<code>i == 4</code>
différent	!=	<code>nb != 42</code>

Et on combinera des conditions avec les *opérateurs booléens* :

- ▷ Le *ET* logique s'écrit `&&`.
Exemple : `(nb >= 0) && (nb <= 100)` vérifie si `nb` est compris entre 0 et 100. Cette condition sera vraie si `nb >= 0` ET si `nb <= 100`.
- ▷ Le *OU* logique s'écrit `||`.
Exemple : `a < b || a < c` sera vraie si `a < b` ou bien si `a < c` (ou les deux).
- ▷ La négation s'écrit `!`.
Exemple : `!(a < b)` sera vraie si `a < b` est faux, c'est-à-dire si `a >= b`.

Exercice 4 Conditions

Écrivez un programme qui affiche la valeur des expressions suivantes :

- ▷ `10 < 20` (écrivez simplement : `System.out.println(10 < 20);`)
- ▷ `10 > 20`
- ▷ `1 == 2`
- ▷ `20.0/2 != 10.0`

Qu'affiche votre programme pour chacune des expressions ci-dessus ?

Exercise 5 Conditions

Écrivez un programme qui demande à l'utilisateur 3 nombres entiers a, b et c et affiche la valeurs des expressions suivantes :

- ▷ `a%2 == 0` (a est divisible par 2 c'est-à-dire a est pair)
- ▷ `a%2 == 1` (a est impair)
- ▷ `a%b == 0` (a est divisible par b)
- ▷ `a < b`
- ▷ `(a <= b) && (a <= c)` (a est le minimum)
- ▷ `(a < b && b < c) || (a > b && b > c)` (b est strictement compris entre a et c)

3 Exercices supplémentaires

Exercice 6 Maximum de 3 nombres

Écrivez un programme qui demande à l'utilisateur trois nombres et affiche le maximum des trois.

Exemple : si l'utilisateur entre 7,5, 17,9 et 2,3 le programme affiche 17,9.

Exercice 7 Le type de triangle

Écrivez un programme qui demande à l'utilisateur la longueur des 3 côtés d'un triangle et affiche s'il est : équilatéral (tous égaux), isocèle (2 égaux) ou quelconque.

Exemple : si l'utilisateur entre 2,5, 5 et 5 le programme affiche "le triangle est isocèle".

Exercise 8 Conditions

Écrivez un programme qui affiche la valeur des expressions suivantes :

- [illegible]

Qu'affiche votre programme pour chacune des expressions ci-dessus?