

L'instruction conditionnelle IF du langage Java

Introduction

Dans un programme, les instructions sont exécutées séquentiellement, les unes après les autres, dans l'ordre dans lequel elles apparaissent. Cependant, il est possible pour un programme de se comporter différemment selon les circonstances, en effectuant des choix, comme par exemple en appliquant un taux de réduction variable, selon un code avantage fourni par le client. Dans le langage Java, il existe deux instructions conditionnelles : *if ... else* et *switch*.

L'instruction *if* peut s'utiliser de deux manières différentes :

<pre><u>if</u> (expression conditionnelle) { instruction_1 } <u>else</u> { instruction_2 }</pre>	<pre><u>if</u> (expression conditionnelle) { instruction_1 }</pre>
--	--

Les *instruction_1* et *instruction_2* sont soit une seule instruction, soit un bloc délimité par { et }.

Sémantique de la première forme : l'expression conditionnelle est évaluée. Si elle est vraie, c'est *instruction_1* qui est exécuté, si elle est fausse c'est *instruction_2* qui sera réalisé.

Sémantique de la deuxième forme : l'expression conditionnelle est évaluée. Si elle est vraie, *instruction_1* est exécuté, sinon aucune instruction n'est exécutée.

Exemple 1

```
double x;           // nombre réel donné par l'utilisateur  
int n;              // nombre entier donné par l'utilisateur  
  
// saisie de x et de n  
System.out.print("Entrez la valeur de x (un nombre reel) : ");  
x = entree.nextDouble();  
System.out.print("Entrez la valeur de n (un nombre entier) : ");  
n = entree.nextInt();  
  
// on affiche l'inverse de x  
if (x == 0) {  
    System.out.println("0 ne possede pas d'inverse");  
} else {  
    System.out.println("L'inverse de " + x + " est " + 1 / x);  
}  
  
// on affiche la valeur absolue de l'entier n  
System.out.print("La valeur absolue de " + n + " est ");
```

```
if (n < 0) {  
    n = - n;  
}  
System.out.println(n);
```

if imbriqué

Une instruction *if* peut elle-même contenir d'autres instructions *if*. Il faut savoir que un *else* se rapporte toujours au dernier *if* rencontré auquel un *else* n'a pas encore été attribué.

Exemple 2 : afficher une information sur le signe d'un entier

Le programme suivant indique si l'entier donné par l'utilisateur est positif (ou nul) ou bien négatif.

```
int n;          // nombre entier donné par l'utilisateur  
  
// saisie de n  
System.out.print("Entrez la valeur de n (un nombre entier) : ");  
n = entree.nextInt();  
  
// on affiche un message indiquant si n est positif ou nul,  
// ou strictement négatif  
if (n >= 0) {  
    System.out.println(n + " est positif ou nul");  
} else {  
    System.out.println(n + " est strictement negatif");  
}
```

Supposons maintenant que l'on souhaite différencier les deux cas : l'entier est strictement positif ou l'entier est nul.

```
int n;          // nombre entier donné par l'utilisateur  
  
// saisie de n  
System.out.print("Entrez la valeur de n (un nombre entier) : ");  
n = entree.nextInt();  
  
// on affiche un message indiquant si n est strictement positif ou nul,  
// ou strictement négatif  
if (n == 0) {  
    System.out.println("L'entier est nul.");  
} else {  
    if (n > 0) {  
        System.out.println(n + " est strictement positif");  
    } else {  
        System.out.println(n + " est strictement negatif");  
    }  
}
```

Le code ci-dessus peut être écrit plus clairement de la manière suivante :

```

int n;          // nombre entier donné par l'utilisateur

// saisie de n
System.out.print("Entrez la valeur de n (un nombre entier) : ");
n = entree.nextInt();

// on affiche un message indiquant si n est strictement positif ou nul,
// ou strictement négatif
if (n == 0) {
    System.out.println("L'entier est nul.");
} else if (n > 0) {
    System.out.println(n + " est strictement positif");
} else {
    System.out.println(n + " est strictement negatif");
}

```

Exemple 3 : afficher une information sur la parité de deux entiers

Il est important de présenter correctement le code pour faciliter sa compréhension. Comparer les deux versions d'un même programme.

```

if (a % 2 == 0) if (b % 2 == 0)
    System.out.println ("Les 2 entiers sont pairs ");
else
    System.out.println ("Les 2 entiers ne sont pas pairs ");

```

```

if (a % 2 == 0) {
    if (b % 2 == 0) {
        System.out.println("Les 2 entiers sont pairs");
    } else {
        System.out.println("Les 2 entiers ne sont pas pairs");
    }
}

```

Avec la deuxième présentation, on constate clairement qu'aucun message n'est affiché dans le cas où la condition $a \% 2 == 0$ est fausse. Pour corriger, il faut écrire :

```

// on affiche une information sur la parité de a et b
if (a % 2 == 0) { // a est pair
    if (b % 2 == 0) {
        System.out.println("Les 2 entiers sont pairs.");
    } else {
        System.out.println("Un seul entier est pair.");
    }
} else if (b % 2 == 0) { // nécessairement a est impair
    System.out.println("Un seul entier est pair.");
} else {
    System.out.println("Les 2 entiers sont impairs.") ;
}

```