

Trucs et Astuces Java

Michel CASABIANCA
casa@sweetohm.net

Ce qui fait la différence entre un programmeur novice et un vieux routard du code, ce sont les kilomètres de lignes de codes. Au cours de ses pérégrinations, on découvre inévitablement des astuces qui simplifient la vie du programmeur :

Opérateurs booléens

Il existe en Java 2 types d'opérateurs "et" : le premier agit sur des booléens (vrai ou faux) et est noté &&, alors que le second agit sur des bits et on le note &. On peut cependant faire opérer le second sur des booléens car un booléen est représenté par un bit. On peut alors se demander quelle est la différence entre ces deux opérateurs lorsqu'on les utilise sur des booléens !

- Le premier ne poursuit pas l'évaluation lorsque le résultat est certain. Par exemple, si on teste "A && B" et que A est faux, alors A && B est aussi nécessairement faux, donc l'évaluation n'est pas poursuivie, ce qui fait gagner du temps.
- Le second évalue complètement l'expression, même si le résultat est connu. Donc dans le cas ci-dessus, B sera tout de même testé, même si A est faux.

Cette différence est très utile car dans le cas où B est un appel à une méthode, il peut être nécessaire de l'appeler même si on connaît le résultat du test. Il est à noter que la même distinction existe pour les opérateurs "ou" (| et ||).

Variables d'environnement

La méthode `getenv()`, qui permettait de connaître la valeur d'une variable d'environnement, est maintenant "deprecated", ce qui veut dire que les futures versions de Java ne la supporteront pas. En effet, les concepteurs de Java ont estimé que cette manière de procéder est dépendante de la plateforme et il faut maintenant procéder autrement, mais la doc du JdK n'est pas très claire à ce sujet :

Il faut passer ces arguments sur la ligne de commande (en arguments de la VM et non de l'application) avec l'option `-D`. On retrouve alors ces valeurs dans les propriétés du système.

exemple

Supposons que l'on veuille passer à un programme la valeur de "PATH", il faut procéder comme suit pour l'appel du programme (sous Windows) :

```
java -Dpath=%path% maClasse
```

Au lancement du programme, il est chargé une nouvelle clef dans les propriétés du système : path, qui a pour valeur le path de votre machine. On peut alors le récupérer par :

```
String path=System.getProperty("path");
```

Cette méthode présente un inconvénient majeur pour les CGI programmés en Java : on est obligé, pour récupérer les variables d'environnement du serveur, de lancer le programme par un batch (pour passer en argument le path). Comme en plus il faut lancer l'interpréteur à chaque appel du CGI, il me semble que Java n'est pas le langage idéal pour écrire du CGI.