Chartre de codage

# Introduction

A quoi ça sert une convention de codage?

C'est pour que tout le monde code de la même manière plus ou moins et comme ça quand on reprend le code de quelqu'un on ne se perd pas complètement. Voilà trois raisons principales :

* 80% de la vie d'une application c'est de la maintenance
* le code ne sera pas maintenu par la même personne pour toute la durée de vie du code
* les conventions de codage permettent d'améliorer la lisibilité du code et une compréhension plus rapide d'un nouveau code

Cette convention de codage est tirée de site officiel de Sun directement : [http://java.sun.com/docs/codeconv/html [...] vTOC.doc.html](http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html).  
Le texte qui suit en est donc largement inspiré, mais est bien sûr simplifié à l'essentiel.

# Le nommage

Le nommage est très important, c'est ce qui va permette de comprendre de quoi on traite en lisant un code.

camelCase

Le camelCase c'est la manière la plus utilisée pour nommer en Java. CamelCase c'est tout simple, ça veut dire que chaque première lettre d'un mot prend une majuscule, que tous les mots sont collés les uns aux autres et petite subtilité, le premier mot ne prend pas de majuscule. Pourquoi camelCase, parce que ça ressemble aux bosses d'un chameau...

Les packages

Généralement les packages sont nommés avec un nom de domaine en commençant par la fin et en ajoutant ensuite le nom du projet.   
Nommer les packages de cette manière permet d'éviter des conflits entre des classes ayant ensuite le même nom et de bien s'y retrouver.

Les classes

Les classes sont nommées selon la méthode camelCase mais commencent pas une majuscule. Il faut aussi éviter les abréviations et essayer d'utiliser des mots qui décrivent bien la classe tout en étant pas trop long. **Exemple : FenetrePrincipale, SelecteurDeFichiers, etc.**  
Les interfaces se nomment de la même manières que les classes.

Les méthodes

Les méthodes sont nommées selon la méthode camelCase et comportent généralement un verbe d'action.

**Cela donne par exemple : fermeLaFenetre(), donnerDeLArgent(), manger(), etc.**  
En plus de cela, il faut savoir que dans une classe on retrouve souvent des méthodes de type getter et setter (accesseur, modificateur), pour récupérer une variable de classe ou la modifier sans toucher directement à la variable. Il se nomme généralement getNomDeLaVariable() et setNomDeLaVariable(Object nomDeLaVariable). Il existe également d'autres mot-clef fréquemment utilisés comme add et remove (ajout, supprimer) pour ajouter et supprimer quelque chose : addQuelqueChose(Object quelqueChose), removeTout(), etc.

Les variables

Les variables se nomment également en camelCase et en commençant par une lettre (a-z). Le nom d'une variable devrait être court et clair. Les variables à 1 caractère sont à éviter sauf pour un usage temporaire (i, j, k, l, m, n pour les entiers et c, d , e pour les caractères). **Cela donne par exemple : nombrePopcorn, tailleAppartement, i, etc.**

Les constantes

Les constantes c'est-à-dire les "static final" doivent être écrites en majuscules et pour séparer les mots les "\_" sont à utiliser. **Cela donne par exemple : LARGEUR\_MAX, ID\_COURANT, etc.**

Les commentaires

Les commentaires sont très utiles pour les programmeurs qui reprendront un code ou bien pour fournir une documentation sur le programme. Il existe la javadoc qui est un système permettant avec des tags de spécifier certains choses comme la date, l'auteur, la version, les arguments et autres.

# Conclusion

L'utilisation d'une convention de nommage peut procurer une sécurité beaucoup plus grande dans l'utilisation des programmes informatiques, du fait que le code source doit respecter des règles précises. Il sera ainsi plus facile d'atteindre des niveaux EAL(Evaluation Assurance Level) plus élevés, par exemple dans le logiciel, lorsqu'il s'agit de vérifier le code source.