

Coding Standard Java

Commentaire : commenter chaque classe et chaque méthode comme ceci :

Classe :

```
/**
 * NomClasse - description de la classe
 * explication supplémentaire si nécessaire
 *
 * @version 1.0
 *
 * @see UneAutreClasse
 * @author Jean Michel D.
 * @copyright (C) moi 2001
 * @date 01/09/2000
 * @notes notes particulières sur la classe
 *
 * @revision référence
 *      date 15/11/2000
 *      author Michel M.
 *      raison description
 *      description supplémentaire
 */
```

Méthode :

```
/**
 * nomMethode - description de la méthode
 *      explication supplémentaire si nécessaire
 *
 *      exemple d'appel de la methode
 * @return      description de la valeur de retour
 * @param      arg1 description du 1er argument
 *      :      :
 * @param      argN description du Neme argument
 * @exception   Exception1 description de la première exception
 *      :      :
 * @exception   ExceptionN description de la Neme exception
 *
 * @see UneAutreClasse#UneAutreMethode
 * @author      Jean Dupond
 * @date      12/02/2001
 * @note      notes particulières.
 */
```

Remarques :

- @return ne doit pas être utilisé avec les constructeurs et les méthodes sans valeur de retour (void)
- @param ne doit pas être utilisé s'il n'y a pas de paramètres
- @exception ne doit pas être utilisé s'il n'y pas d'exception propagée par la méthode
- @author doit être omis s'il est identique à celui du tag @author de la classe
- @note ne doit pas être utilisé s'il n'y a pas de note

Accolade :

```
return new MyClass ()
{
    @Override public void method ()
    {
        if ( condition ())
        {
            try
            {
                something ();
            }
            catch ( ProblemException e )
            {
                recover ();
            }
        }
    }
};
```

Nom des classes : Les noms de classe sont écrits en [UpperCamelCase](#).

Nom des méthodes : Les noms de méthode sont écrites dans [lowerCamelCase](#).

Nom des constantes : Les noms des constantes utilisent CONSTANT_CASE: toutes les lettres majuscules, avec des mots séparés par des underscores.

Nom des attributs : Les noms des attributs sont écrits en lowerCamelCase, ils sont toujours appelés avec le préfixe « this. »

Nom des variables locales : Noms de variables locales sont écrits dans lowerCamelCase, et peuvent être abrégées plus libérale que les autres types de noms.
Toutefois, les noms d'un seul caractère doit être évitée, sauf pour les variables temporaires et de bouclage. Même lorsque, variables locales finales et immuables ne sont pas considérés comme des constantes, et ne doivent pas être coiffés comme des constantes.

1 déclaration par ligne.

Constructeurs :

Il est préférable de définir explicitement le constructeur par défaut (le constructeur sans paramètre). Soit le constructeur par défaut est fourni par le compilateur et dans ce cas il serait préférable de le définir soit il existe d'autres constructeurs et dans ce cas le compilateur ne définit pas de constructeur par défaut.

Il est préférable de toujours initialiser les variables d'instance dans un constructeur soit avec les valeurs fournies en paramètres du constructeur soit avec des valeurs par défaut.

Préfixer tous les paramètres des méthodes par un « p » en lowerCamelCase