

# Chartes de programmation Projet Java :

**IL EST ABSOLUMENT INDISPENSABLE DE FAIRE LA JAVADOC AU FUR ET A MESURE !!!!!**

## *1 \_ Nom et zones de déclaration des variables :*

Toutes les variables, que cela soit dans le programme principal ou dans les classes, sont déclarées en début de classe ou de fonctions et sont accompagnées d'un commentaire javadoc expliquant cette dernière.

Pour ce qui est du nom de la variable, choisir un nom significatif et si besoin rajouter le nom de la classe derrière.

### Exemple :

```
/**
 * Argent possédé par un humain
 */
int argentHumain;
```

***PS : Eviter dans la mesure du possible de déclarer des variables inutiles, genre "int total = a+b; return total" peut s'écrire "return(a+b);"***

## *2 \_ Fonctions :*

Toutes les fonctions doivent avoir un nom clair et précis de manière à être parfaitement identifiable. Une fonction est publique pour pouvoir être accessible dans toutes les autres classes.

Les paramètres passés en argument d'une fonction sont eux aussi précis.

Enfin une fonction est toujours précédée d'un commentaire javadoc respectant rigoureusement la forme suivante :

```
/**
 * Description de la fonction.
 * @param nomParam explicationDuParamètre (que si la fonction à des paramètres, et autant de ligne de ce type que de paramètres)
 * @author nomAuteur (pour savoir à qui demander si on comprend pas)
 * @return decriptionDeCeQuiEstRetourné (sauf pour les void)
 */
```

### Exemple :

```
/**
 * Permet à un humain de manger une certaine quantité d'un aliment.
 * @author Florian
 * @param nombre Quantité d'aliment que va manger l'humain.
 * @param aliment Aliment que va manger l'humain.
 */
public void manger(int nombre, String aliment);
```

```

/**
 * Additionne deux nombres.
 * @author Florian
 * @param nombre1 Premier élément de la somme.
 * @param nombre2 Second élément de la somme.
 * @return Somme des deux éléments passés en paramètres.
 */
public int addition(int nombre1, int nombre2);

```

### 3 \_ Classe :

Le nom d'une classe commence toujours par une majuscule.

Une classe est toujours publique.

Une classe est rangée dans un package (soit client, soit serveur, soit ihm).

Une classe est toujours accompagnée d'un petit commentaire javadoc expliquant la classe et sa fonction.

Une classe se présente toujours de la même manière :

```
package ihm;
```

```
import ... ;
```

```

/**
 * Description de la classe
 * @author Florian
 */
public class FenetrePrincipale{

    /* Liste des attributs de la classe */

    /**
     * Description attribut 1
     */
    private int attribut1; (un attribut est soit privé soit protected mais JAMAIS public)

    /**
     * Description attribut 2
     */
    private String attribut2;

    /**
     * Constructeur de la classe.
     * Explication de l'initialisation de la classe
     */
    public FenetrePrincipale(int attribut1,String attribut2){    Par convention on donne le même nom
aux paramètres que aux attribut qu'ils caractérisent
        ...
    }

```

```
/* Getters et setters avec leur commentaire javadoc */
```

```
/* Liste des fonctions de la classe avec leurs commentaire javadoc */  
}
```

**Pour faciliter la lecture du code par l'ensemble des membres de l'équipe, il est absolument indispensable de respecter cet ordre d'écriture dans une classe.**