

Conventions d'écriture en Java

Voici les conventions écritures que nous avons fixé, il faudra les respecter pour que nous ayons un code propre et homogène, de plus elles ont été fixées pour que ce soit le plus simple pour nous (lecture rapide, propreté etc...)

Le nommage

Les attributs

Les attributs doivent **toujours** commencer avec une minuscule, pour séparer les mots, on les sépare avec une majuscule. Privilégiés les noms de variables claires et explicite, quitte à ce qu'il soit un peu long, on a l'auto complétion que diable! Donc les variables d'une lettre, à bannir! (à part *peut être* i dans le cas d'un for, s'il n'est pas réutilisé après le for

Les attributs seront préfixé par `_` afin de pouvoir les reconnaître facilement, il n'est donc pas indispensable d'utiliser `_`.

Exemples de noms d'attribut

```
int _monSuperAttribut;  
boolean _vousAvezPerdu;
```

Les variables en paramètre de méthodes

Les variables paramètre sont les variables qui ne sont que dans une méthode (et donc, elles sont détruites à la fin de la méthode). Ces variables doivent respecter la règle précédente à la différences qu'elle doivent toute commencer par un `p` (pour paramètre)

Exemples de noms de paramètre

```
int pMonSuperParametre;  
boolean pVousAvezPerdu;  
String pCacamou;
```

Les noms de constantes

Les constantes doivent être tout en majuscule, les différents mots de la constante sont séparés par des underscore (`_`). Même remarque que pour les attributs, choisissez des noms de constante clair, compréhensible par tous, pas seulement par ceux qui sont dans votre tête!

Exemple de noms de constante

```
final int MA_SUPER_CONSTANTE;  
final boolean VOUS_AVEZ_PERDU;  
final String CONST;
```

Les noms de méthodes

Les méthodes doivent commencer par une minuscule, et séparer les différents mots par une majuscule (Camel case). Les fonctions ne retournant rien (procédures) doivent toujours être à l'infinifit.

À l'opposé les fonctions retournant quelques choses doivent être au participe passé.

Il faut décomposer au maximum, n'hésitez pas à faire une méthode private si besoin est, c'est toujours plus clair d'avoir une fonction, dictant explicitement ce qu'elle fait par son nom que 10 lignes de code bizarroïdes avec 2-3 lignes de commentaire! Et donc, les noms de fonctions sont essentiels!!

Exemple de noms de fonction

```
void afficher(String pTexteAAffiche) {  
    System.out.println(pTexteAAffiche);  
}  
boolean inferieur(int entier1, int entier2) {  
    return (entier1 < entier2);  
}
```

Les noms de classes ou d'interface

Les noms de classe ou d'interface doivent tous commencer par une majuscule, les différents mots sont séparés par une majuscule, choisissez des noms de classes claires!

(oui, je me répète, mais c'est ce qui fait toute la compréhension facile, ou non, d'un programme les noms de variables, classe, paramètre, méthodes etc...)

Exemple de noms de classe

```
class MaSuperClasse { }
class UneAutreClasse extends MaSuperClasse { }
interface MaSuperInterface { }
```

Les noms de packages

Les noms de package ont la même convention que les noms d'attributs: ils commencent par une minuscule, et les différents mots sont séparés par une majuscule(Camel case), cela permettra de différencier les packages des classes.

Exemple de noms de package

```
package monPackage;
package monSuperPackage.unAutrePackage;
import monSuperPackage.unAutrePackage.MaSuperClasse;
```

L'indentation

La règle est simple, on ouvre une accolade, la ligne suivante sera décalé vers la droite(une tab = 4 caractères), on ferme une accolade, on décale l'accolade vers la gauche et tout ce qui suis.

Egalement, si une ligne est trop longue, on va à la ligne, et décalons d'une ligne vers la droite, une fois l'instruction finie, on redécale vers la gauche.

Dans le cas d'un switch, le break doit s'aligner avec le case 42: tout ce qui est entre case et break sera indenté.

Merci de mettre un espace avant chaque accolades, oui je sais je suis psychorigides, mais c'est moche sinon.

Exemple d'indentation

```
class MaSuperClasse {
    int monSuperAttribut;

    void afficherHelloWorld() {
        System.out.println(
            "Hello World");

        switch(yatta) {
            case 42:
                // ^^
                break;
            case 1337:
                // ...
                break;
            default:
                //
        }
    }
}
```

Les accolades

Les accolades ouvrante sont positionnés à la fin de la ligne demandant une accolade (switch, if, class, else, elseif, ...)

Les accolades fermantes sont positionnés une ligne après la dernière instruction. (avec une désindentation)

Les else et elseif se mettent sur la même ligne que l'accolades fermante.

Exemple d'indentation

```
int maSuperMethode() {  
    if(true) {  
        // bla bla  
    } else if(false) {  
        // bla bla  
    } else {  
        //instruction  
    }  
  
    switch(var) {  
        case 0:  
            // ^^  
            break;  
        case 1:  
            // yatta  
            break;  
        default:  
            // :-)  
    }  
}
```

Position des mots clefs

Le mot clef `abstract` doit toujours être avant le mot clef `class` ou avant le type de retour de la méthode.

Exemple de méthodes abstraites

```
abstract class Cacamou {  
    abstract int maMethode();  
}
```