

# Normes techniques

Aurélien SVEVI

14 mars 2015

## Table des matières

<b>Normes de nommages</b>	<b>2</b>
1 Nommage des class et interfaces	2
2 Nommage des variables	2
3 Nommage des constantes	2
4 Nommage des méthodes	2
<b>Normes des fichiers</b>	<b>3</b>
1 En-tête des fichiers	3
2 Contenu des fichiers	3
<b>Normes de variables</b>	<b>3</b>
<b>Normes de méthodes</b>	<b>3</b>
<b>Normes des instructions</b>	<b>4</b>
1 L’instruction if	4
2 L’instruction for	4
3 L’instruction switch	4
<b>Exemple de code</b>	<b>5</b>

# Normes de nommages

## 1 Nommage des class et interfaces

Les noms des class et interfaces doivent respecter les conventions suivantes :

1. Première lettre en majuscule
2. Première lettre de chaque mot en majuscule
3. Donner des noms simples et descriptifs
4. Éviter les acronymes
5. N'utiliser que les caractères [a-z] et [A-Z] et [0-9]

## 2 Nommage des variables

Les noms des variables doivent respecter les conventions suivantes :

1. Première lettre en minuscule
2. Première lettre de chaque mot en majuscule
3. Donner des noms simples et descriptifs
4. N'utiliser que les caractères [a-z] et [A-Z] et [0-9]
5. Variables d'une seule lettre (pour un usage local) :
  - int : i, j, k, m, et n
  - char : c, d, et e
  - boolean : b

## 3 Nommage des constantes

Les noms des constantes doivent respecter les conventions suivantes :

1. Tout en majuscule
2. Séparer les mots par des underscore : ' \_ '
3. Donner des noms simples et descriptifs
4. N'utiliser que les lettres [A-Z], [0-9] et ' \_ '

## 4 Nommage des méthodes

Les noms des méthodes doivent respecter les conventions suivantes :

1. Première lettre en minuscule
2. Première lettre de chaque mot en majuscule
3. N'utiliser que les caractères [a-z] et [A-Z] et [0-9]

# Normes des fichiers

## 1 En-tête des fichiers

Chaque fichier devra débuter par un commentaire contenant :

1. Le nom du développeur
2. La date de création du fichier
3. Une description rapide si nécessaire

## 2 Contenu des fichiers

Un fichier ne devra pas contenir plus de 1500 lignes de codes.

Un fichier ne devra contenir qu'une seule class et le nom du fichier devra être de la forme "MaClasse.java".

Chaque ligne ne devra contenir qu'un seul traitement : Pas de ';' au milieu de la ligne sauf dans le cas des boucles for.

```
1|| res = 4 + 3;  
2|| i ++;
```

# Normes des variables

Chaque variable doit être définie sur une ligne

```
1|| String nom;  
2|| String prenom;  
3|| int age;
```

Toutes les variables utilisées dans un bloc doivent être définies au début de ce bloc (sauf les index de boucle)

Éviter de donner les mêmes noms à plusieurs variables définies dans des blocs différents (sauf variables locales).

# Normes des méthodes

Une méthode ne devra pas prendre plus de 7 paramètres

Une méthode ne doit pas contenir plus de 150 lignes

# Normes des Instructions

## 1 L’instruction if

L’instruction if devra se présenter sous cette forme :

```
1| if (condition) {  
2|     traitements;  
3| } else if (condition) {  
4|     traitements;  
5| } else {  
6|     traitements;  
7| }
```

Elle devra obligatoirement contenir les accolades '{' et '}' même s’il n’y a qu’une seule ligne de traitement.

## 2 L’instruction for

L’instruction for devra se présenter sous cette forme :

```
1| for (initialisation; condition; traitement) {  
2|     traitements;  
3| }
```

On peut également utiliser cette instruction sous la forme for each :

```
1| for(Element e : List l){  
2|     traitement;  
3| }
```

## 3 L’instruction switch

L’instruction switch devra se présenter sous cette forme :

```
1| switch (condition) {  
2| case casA :  
3|     traitements;  
4|     break;  
5| case casB :  
6|     traitements;  
7|     break;  
8| case casC :  
9|     traitements;  
10|    break;  
11| default :  
12|     traitements;  
13|     break;  
14| }
```

## Exemple de code

```
1 | class MaSuperClass{
2 |
3 |     private String nom;
4 |     private int age;
5 |     private final int ANNEE = 2015;
6 |
7 |
8 |     public MaSuperClass(String nom, int age){
9 |         this.nom = nom;
10 |        this.prenom = prenom;
11 |        this.age = age;
12 |    }
13 |
14 |
15 |    public int calculerAnneeNaissance(){
16 |        return ANNEE-age;
17 |    }
18 |
19 |
20 |    public void afficherTrancheAge(){
21 |        if(age < 18){
22 |            System.out.println("Trop_petit");
23 |        } else if (age >= 18 && age <= 30) {
24 |            System.out.println("Bienvenue");
25 |        } else {
26 |            System.out.println("Trop_vieux");
27 |        }
28 |
29 |    }
30 |
31 | }
```