

## Convention d'écriture de code

### 1. Pourquoi une convention d'écriture de code

- 80% du coût d'un logiciel concerne sa maintenance et non son écriture initiale,
- la plupart du temps, les personnes qui maintiennent un logiciel ne sont pas celles qui l'ont écrit,
- les conventions de code augmentent la lisibilité et améliorent la compréhension du code,
- c'est un des critères de qualité du code dans l'industrie du logiciel,
- etc.

### 2. Organisation d'une application Java

- Le fichier porte le nom de la classe
- Une classe par fichier (sauf **inner** classes)
- Les packages permettent d'organiser le code

### 3. Convention de noms d'identificateurs

#### 3.1 Classes

Les classes commencent TOUJOURS par une majuscule. S'ils sont composés de plusieurs mots, chaque mot commence par une majuscule et il n'y a pas de séparateurs.  
Exemples : MonType, PileDEntiers, GrapheNonCyclique.

#### 3.2 Variables et méthodes

Les noms de variables et de méthodes s'écrivent de façon identique aux noms de classes SAUF qu'ils commencent TOUJOURS par une minuscule.

Exemples : maVariable, rechercheDuMinimum, indexDuMinimum, poidsFort.

#### 3.3 Constantes

Les constantes sont toutes en majuscules et les différents mots sont séparés par des "soulignés" ('\_').  
Exemples : MAX\_INT, NOIR, BLEU\_CLAIR, MATRICE\_IDENTITE.

### 4. Blocs

L'accolade ouvrante { se place généralement à la fin de la ligne qui est à l'origine du bloc.

Une accolade fermante } est toujours seule sur sa ligne, ou alors suivie d'un commentaire, par exemple le nom du sous-programme qui se termine.

```
/* if sans partie else */
if (condition) {
    instruction;
    instruction;
}
/* if avec partie else */
if (condition) {
    instruction;
} else {
    instruction;
}
```

Table 4.1. if et if-else

```
while (condition) {
    instruction;
    instruction;
}
```

Table 4.2. while

```
public int methode(int par1, float par2) {
    int valeur;
    ...

    instructions;
    ...
    return valeur;
} /* commentaire facultatif : fin de methode */
```

Table 4.3. Méthode

### 5. Indentation

Le code situé à l'intérieur d'un bloc, signalé par un mot clé et/ou une accolade ouvrante, doit être indenté, c'est à dire décalé de 2 à 4 caractères vers la droite.

```
while (condition) {
    instruction1;
    instruction2;
    while (condition) {
        instructio3;
        instruction4;
    }
    instruction5;
}
```

Table 5.1. Exemple d'indentation

### 6. Longueur de lignes

Limiter la longueur des lignes à 80 caractères (pour faciliter la lecture et l'impression).

Si des instructions sont trop longues, replier le code de façon à faciliter la compréhension.

```

if ((condition1 && condition2)
    || (condition3 && condition4)
    || (condition5 && condition6)) { /* test replié sur 3 lignes */
    unAppelDeSousProgramme();
}

```

Table 6.1.Exemple de replis de ligne trop longue

## 7. Déclarations et instructions

Règles :

- on écrit une déclaration de variable par ligne, accompagnée d'un commentaire donnant son rôle,
- on écrit une instruction par ligne,
- on laisse un blanc :

```

x entre les mots-clefs if, while et la parenthèse ouvrante qui suit,
if (condition1) {
x après une virgule,
public int methode(int par1, float par2) {
x de part et d'autre d'un opérateur binaire,
if (par1 >= 0) {
    par1 = par1 * 10 ;
}

```

- on ne met pas de blanc entre un opérateur unaire et son opérande.  
par1 = -1;

## 8. Cartouche de sous-programmes (javadoc)

Un cartouche est un commentaire, placé juste avant chaque sous-programme, qui contient une brève description :

- du rôle du sous-programme,
- du rôle des paramètres entrant,
- du rôle de la valeur (fonction) ou des valeurs (procédure) retournées.

```

/** Rôle : Elève x à la puissance n
 * Entrées : x scalaire et n exposant entier
 * Sortie : retourne x^n
 */
public int elevPuissance (int x, int n) {
    ...
}

```

Table 8.1.Exemple de cartouche de sous-programme

## Sommaire

1. Pourquoi une convention d'écriture de code.....	1
2. Organisation d'une application Java.....	1
3. Convention de noms d'identificateurs.....	1
3.1 Classes.....	1
3.2 Variables et méthodes.....	1
3.3 Constantes.....	1
4. Blocs.....	1
5. Indentation.....	2
6. Longueur de lignes.....	2
7. Déclarations et instructions.....	3
8. Cartouche de sous-programmes (javadoc).....	3