**Recommandation générale**

On permet n'importe quelle violation du guide si elle augmente la lisibilité. Le but principal de la recommandation est d'améliorer la lisibilité et de ce fait l'organisation et la qualité générale du code. Il est impossible de couvrir tous les cas spécifiques d'un guide général et le programmeur doit être flexible. Le lecteur est invité à faire montre de jugement. Si une pratique est inutile, inapplicable, voire contreproductive au regard des contraintes d'un projet ou des objectifs corporatifs, soyons assez intelligents pour y déroger.

**Conventions de nomenclature**

Nous énumérons ici les règles à suivre pour assurer la lisibilité et la cohérence pour tous les noms que vous donnerez à vos variables, classes, etc.

**Nommer une variable :**

Le premier caractère doit être une lettre ;

L'espace, le point, et les caractères spéciaux !, @, &, $ et # ne sont pas permis, mais le caractère \_ (souligné) peut être utilisé ;

Le nom doit être unique à l'intérieur de sa portée

Le nom doit être significatif et explicite;

Des abréviations peuvent être utilisées, mais elles doivent êtres explicites

Bons exemples : age, prenom, montant\_total, nombre\_lignes\_vides, nb\_lignes\_vides

Mauvais exemples : a, pnom, montanttotal, NombreLignesVides, nbr\_lg\_vds

**Nom des constantes**

Le nom doit être significatif et explicite ;

Toutes les lettres formant le nom doivent être MAJUSCULES ;

Le caractère \_ (souligné) doit être utilisé pour séparer les mots ;

Les caractères accentués (É, À, Ê, Ç, etc.) peuvent être utilisés ou non ;

Des abréviations peuvent être utilisées, mais elles doivent êtres explicites.

Bons exemple: PI, TPS, POURCENTAGE\_POURBOIRE, ADRESSE\_DEPART\_CELLULE, ADR\_DEPART\_CELL

Mauvais exemples : P, tps, PourcentagePourboire, ADRESSEDEPARTCELLULE, ADR\_DPT\_CL

**Nom des classes et des sous-classes**

Les mêmes règles au nom des classes et des sous-classes qu’au nom des variables et des constantes. Plus, nous ajoutons les règles suivantes :

Le nom doit illustrer l’utilité d’une procédure ou la valeur de retour d’une fonction ;

La première lettre de chaque mot formant le nom doit être MAJUSCULE et toutes les autres lettres, minuscules ;

Le caractère \_ (souligné) doit être utilisé pour séparer les mots ;

Les caractères accentués (é, à, ê, ç, etc.) peuvent être utilisés ou non ;

Des abréviations peuvent être utilisées, mais elles doivent êtres explicites.

Bons exemples : Afficher\_Infos Calculer\_Coordonnées Est\_Impair Est\_Une\_Lettre Sinus

Mauvais exemples : Afficher Calculer afficherinfos CalculerCoordonnées Fonction\_2

**Indentation et aération du code source**

L’indentation et l’aération du code source sont essentielles pour que le code source soit lisible par un être humain. L’indentation représente l’espace laissé entre la marge de gauche et le début de la ligne de code.

L’aération est la ligne vide laissée entre deux blocs de code. Un bloc de code regroupe les instructions d’un même « niveau » et d’une même « idée ». Le corps d’une procédure ou d’une fonction est formé par un ou plusieurs blocs de code qui s’exécutent les uns après les autres.

Chaque bloc de code commence par un commentaire et est isolé des autres blocs de code par une ligne vide. Une seule ligne vide suffit pour séparer deux blocs de code.

Lorsqu’une instruction doit être écrite sur plus d’une ligne pour faciliter la lecture ou l’impression du code source, la deuxième ligne et les suivantes doivent être indentées par rapport à la première ligne d’une ou de plusieurs tabulations.

**Commentaires dans le code source**

Les commentaires dans le code source sont essentiels pour comprendre le code source une fois qu’il a été écrit, par soi-même ou par d’autres. Un bon commentaire doit être écrit dans en français ou en anglais et doit être autant que possible exempt d’erreurs.

Tout d’abord, toutes les variables déclarées doivent être commentées, même celles qui semblent évidentes. Le commentaire est placé sur la même ligne que la déclaration et tous les commentaires sont alignés ensemble. Ensuite, tous les blocs de code doivent commencer par un commentaire indiquant, ce que réalisent les quelques lignes qui suivent. Si une logique complexe est utilisée, elle est expliquée dans ce commentaire. Les boucles doivent être décrites au début de celles-ci et non à la fin, même si la condition de boucle est placée à la fin. Les commentaires de blocs de code doivent avoir la même indentation que le bloc de code commenté.

Les commentaires ne doivent pas comporter de caractères accentués(à, é, è, ô, etc.) Il faut remplacer les accents par la lettre équivalente (à devient a, é devient e, è devient e, etc.).

Sites de Référence :

<https://cours.etsmtl.ca/seg/ethe/inf130/Norme_INF130.pdf>

<https://www.studocu.com/fr-ca/course/ecole-de-technologie-superieure/ordinateur-et-programmation/3345432>