

Règles de codage

Validées le 15/06/2020

Remarque générale :

Les normes proposées par Visual Studio pour le développement sous C#, sont globalement à respecter.

1. Commentaires

1A. Commentaires normalisés

1A1. Commentaires des modules

Ils sont obligatoires devant les modules (et le Main, qui est le module principal).

Ils doivent être de type linéaire (///) et doivent comporter :

- Une description du module
- Chaque paramètre avec sa description (dans le cas où le module a des paramètres)
- Une description de l'information retournée (dans le cas d'un retour)

Conseil : utiliser la génération automatique de commentaires sous Visual Studio.

1A2. Commentaire du fichier

Il est demandé de mettre un commentaire type "cartouche" en tête du fichier de code, sous la forme suivante :

```
/**  
 * titre :  
 * description :  
 * auteur :  
 * date création :  
 * date dernière modification :  
 */
```

1B. Commentaires informatifs

Les commentaires informatifs ne doivent pas être mis sur une même ligne qu'une ligne de code.

Chaque groupe de lignes de code ayant un rôle global précis, doit être précédé d'un commentaire informatif succinct, de préférence sur une ligne.

Cependant, les commentaires informatifs doivent rester rares car ils sont souvent difficiles à maintenir (obsolescence lors de modifications de code) : il faut favoriser une programmation modulaire avec des modules courts, n'ayant qu'une fonction et étant précédés d'un commentaire normalisé explicite. Les commentaires informatifs doivent essentiellement se limiter au Main.

2. Présentation du code

2A. Indentations

Les indentations proposées par Visual Studio sont la norme : 4 espaces ou une tabulation dans chaque bloc de code.

Conseil : laisser Visual Studio gérer les indentations (lorsque les indentations sont perdues, les faire régénérer automatiquement).

2B. Position des accolades

La position des accolades proposée par Visual Studio est la norme : accolade ouvrante au début de la ligne sous celle qui commence le bloc (aligné au niveau de la première colonne du bloc) et accolade fermante au début de la ligne qui suit la dernière ligne du bloc (aligné au niveau de la première colonne du bloc).

Exemple :

```
while (condition)  
{  
    ...  
}
```

Une ligne qui contient une accolade ne doit rien contenir d'autre.

Les accolades sont obligatoires dans tous les blocs, même lorsqu'elles sont considérées comme optionnelles (lorsque le bloc ne contient qu'une ligne de code).

2C. Espacements horizontaux

Lorsque Visual Studio ne fait pas de préconisation, voici les règles à suivre :

- Pas d'espace autour des parenthèses ouvrantes et fermantes, à l'exception de la parenthèse ouvrante qui suit une instruction et qui doit être précédée d'un espace.

Exemple :

```
if (condition)
```

- Un espace avant et après un signe (ou bloc de signes) mathématiques :
= == <= >= != + - * / += -= *= /= ++ --
- Pas d'espace avant un point-virgule.

2D. Espacements verticaux

Pas de ligne vide excepté entre chaque module.

2E. Instructions par ligne

Une seule instruction par ligne.

3. Nommage

3A. Normes générales

Ces normes concernent les noms des variables, modules et constantes donnés par le développeur.

3A1. Caractères acceptés dans un nom

Lettres minuscules et majuscules, chiffres, underscore (_).

Cas particulier de l'underscore : il ne doit être présent qu'en premier caractère, dans des cas spécifiques.

3A2. Premier caractère d'un nom

Lettre (excepté l'underscore dans des cas spécifiques).

3B. Normes spécifiques

3B1. Variables

camelCase (une majuscule au début de chaque mot, excepté le premier mot)

3B2. Modules

PascalCase (une majuscule au début de chaque mot, même le premier mot)

3B3. Constantes

Tout en majuscule