

Normes de développement en C#

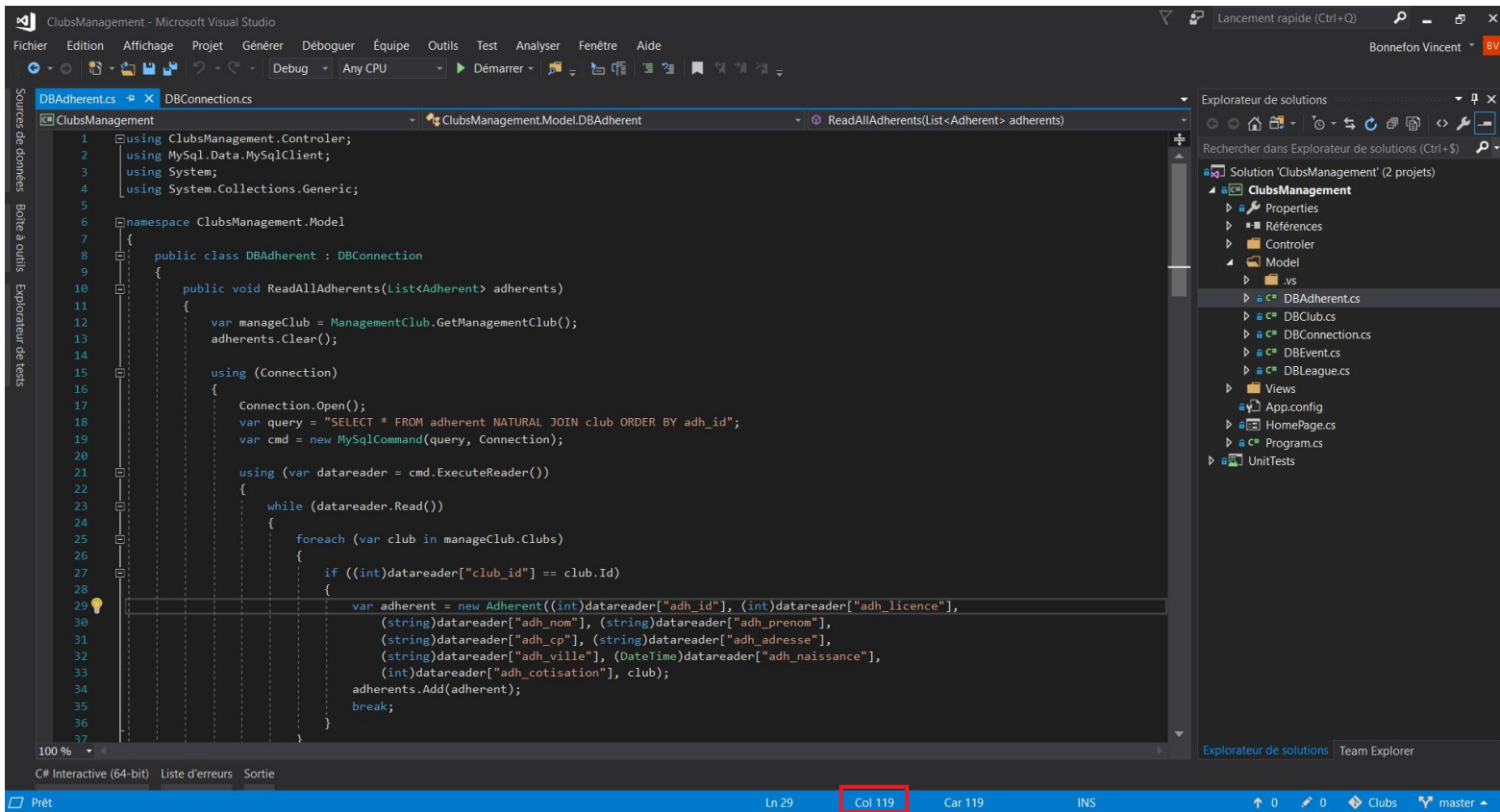
Sommaire

Normes de présentation du programme.....	2
Identificateurs.....	4
Commentaires Déclarations.....	5
Blocs logiques.....	6
Méthodes.....	6

Normes de présentation du programme

Longueur des lignes de code

Aucune ligne de code ne doit dépasser 120 caractères en longueur. Dans Visual Studio, il est possible de voir à quel caractère on est arrivé sur la ligne en cours en consultant la barre d'état, en bas à droite.



Pour couper une ligne trop longue, il est nécessaire d'ajouter un retour chariot là où c'est nécessaire, en général après un opérateur.

Les caractères des commentaires comptent dans la longueur d'une ligne de code.

Blancs

Un blanc (espace) doit être inséré avant et après tous les opérateurs. Aucun blanc ne doit apparaître à l'intérieur des parenthèses () et des crochets [].

Par exemple, il faut `if (a > b)`, pas `if (a > b)`. Il faut `tab[3]`, pas `tab[3]`.
Aérer le code avec des lignes blanches placées entre les blocs de logique, comme des paragraphes dans un texte. Éviter plus d'une ligne blanche d'affilée.

Accolades alignées verticalement

Les accolades ouvrantes { et fermantes } délimitent les blocs d'instructions et doivent se trouver l'une au-dessus de l'autre, comme dans le bloc suivant :

```
public Adherent GetById(int idOfAdherent)
{
    foreach (var adherent in Adherents)
    {
        if (adherent.LicenceNumber == idOfAdherent)
        {
            return adherent;
        }
    }
    return null;
}
```

Retraits de ligne (indentation)

À chaque fois qu'un nouveau bloc d'instructions est commencé, et qu'il est subordonné à une ligne précédente (if, while, etc.), alors il faut le mettre en retrait par rapport à la ligne précédente.

Par exemple, dans le if imbriqué suivant, il y a deux retraits, le second plus profond que le premier (cf: image ci-dessus).

Le retrait doit être de 4 caractères, soit la valeur par défaut que Visual Studio insèrent automatiquement.

Disposition du if et du while

La structure *if* et du *while* doit être disposée, avec les accolades alignées verticalement.

Même si une instruction *if*, *while*, *for* ou *do...while*, *while* qui ne comprend qu'une seule instruction peut ne pas contenir d'accolades on choisira d'en mettre.

Identificateurs

Les identificateurs sont, par exemple, les noms de variables ou les noms de méthodes.

Nom des identificateurs

Les identificateurs doivent être en anglais.

Nom des variables

Les noms de variables doivent être en lettres minuscules, sauf s'ils contiennent plusieurs mots. Dans ce cas, la première lettre de chaque mot de l'identificateur doit être une majuscule sauf la première lettre du premier mot (ex: `numberOfAdherent` ou `adherentToModify`).

Nom des constantes

Les noms de constantes doivent être en lettres majuscules. On utilise le caractère de soulignement (_) pour en séparer les mots (ex: PI ou TAUX_HORAIRE).

Utilisation des constantes

L'utilisation des constantes est obligatoire. Dès qu'on a une variable dont le contenu ne doit pas changer durant l'exécution du programme, on en fait une constante. Si on a besoin d'une constante physique ou mathématique, on en fait également une constante.

Les noms des méthodes doivent contenir un seul verbe précis et au moins un nom. Chaque mot de l'identificateur doit commencer par une majuscule. Les articles sont à proscrire. Le nom de la méthode doit être descriptif de son rôle.

Exemples de noms à utiliser : AddClub() ModifyClub() CountEvent()

Les déclarations des constantes doivent constituer les premières lignes de code d'une méthode. La déclaration des variables suit la déclaration des constantes. Lorsqu'une variable est déclarée, il faut initialiser sa valeur à 0 pour les types numériques et à la chaîne vide ("") pour les chaînes de caractères. Pour les booléens, on a le choix.

Commentaires Déclarations

Les variables importantes dont le nom n'est pas tout à fait clair pour le lecteur doivent être commentées lors de leur déclaration.

Blocs logiques

L'utilisation des commentaires est exigée pour chaque bloc logique (en fonction de sa complexité) important : if, while etc. Les commentaires doivent être pertinents c'est-à-dire répondre à la question « à quoi sert le bloc qui suit ? ». Si c'est nécessaire, on utilise alors des commentaires sur plusieurs lignes.

Méthodes

Chaque méthode doit être précédée d'un commentaire si sa complexité l'exige. Une ligne blanche est alors laissée avant le commentaire. Les commentaires doivent suivre le modèle suivant: