

Checklist : Coder proprement

Utilisation

Cette liste propose un ensemble de règles que vous pouvez vérifier. Ces règles sont simples et faciles à appliquer.

- Pour démarrer, vous pouvez sélectionner une méthode à vérifier,
- Ensuite, vous pouvez utiliser la check-list de manière régulière lorsque vous codez.

Nommage

☐ J'utilise un nom de classe significatif. A partir du nom, je sais ce que réalise la classe (le contexte n'est pas nécessaire pour comprendre).

Comment améliorer ?

- Utilisez la fonction Renommer de Visual Studio pour améliorer la signification de vos variables, méthodes ou classes.
- Utilisez des noms plus longs ayant plus de sens.

Commentaires

☐ J'ai des commentaires dans le code là où c'est nécessaire : le commentaire aide à comprendre un fonctionnement particulier. En supprimant le commentaire, il faudra plusieurs minutes à un autre développeur pour comprendre le morceau de code.

Comment améliorer ?

- Vérifiez que vous n'avez pas trop de commentaire (inutile de supprimer mais pensez à ne pas trop en mettre dans le nouveau code),
- Vérifiez aussi que vous avez assez de commentaire pour les morceaux de code complexes

Une seule responsabilité pour les classes et les méthodes

☐ J'ai écrit une méthode qui réalise une seule chose

☐ J'utilise aussi cette règle pour les classes. Une classe doit s'occuper de gérer une seule responsabilité.

Comment améliorer ?

Choisissez une méthode et vérifiez si elle réalise une seule opération. Si elle s'occupe de faire deux opérations, découpez-la en deux méthodes.

Pour les classes, vous pouvez aussi appliquer ce principe. Une classe doit avoir une seule raison d'être modifiée (je vous invite à consulter l'article sur SRP :

<http://www.dotnetdojo.com/applications-solid-srp/>)

Créez des classes évolutives

☐ J'ai écrit une classe qui peut être étendue sans la modifier directement. Cela permet de modifier son comportement par exemple par héritage ou par des events.

Comment améliorer ?

Je vous invite à consulter l'article sur le sujet : <http://www.dotnetdojo.com/extensibilite-delegates-delegates-multicast-events-csharp/>

Séparer la création et l'utilisation

☐ Je sépare l'initialiser d'une classe et son utilisation

Comment améliorer ?

Mettez en place une méthode qui s'occupe d'initialiser votre application. Les classes principales doivent être créées au démarrage.

Dans votre application, vous pourrez ensuite utiliser les instances directement.

Fiche pratique [DotnetDojo](#)

Article d'origine : <http://www.dotnetdojo.com/coder-proprement/>

[Dotnetdojo.com](#) – Améliorer votre potentiel de développeur