

```
revue = code->revue_de_code();
```

```
int lesConventionsDeCodage;
```

Les conventions de codage sont un moyen de standardiser des bonnes pratiques lorsque l'on code.

- améliorer la lisibilité du code
- améliorer la maintenance future
- gagner du temps

→ Ce sont rarement des règles du langage mais elles sont respectées par presque tous.

Trois types de revues de code existent :

- Revue bloquante : tout développement doit être relu avant de le mettre dans la branche principale
- Revue non-bloquante : tout code source de la branche principale doit être relu
- Revue en binôme : Deux personnes par poste échangent régulièrement les rôles et se relisent en continu

→ Permet d'identifier des bugs pendant la phase de développement.

```
if (tropComplexe) {return;}
```

La complexité d'une portion de code est proportionnelle au nombre de points de décision dans le code (condition, boucle...).

Un code trop complexe est :

- Difficile à comprendre et maintenir
- Difficile à modifier
- Difficile à tester

→ Une fonction, une mission.

```
void doublons();
```

Un doublon de code est une portion de code répétée à plusieurs endroits.

- Maintenance plus complexe
- Modifications risquées
- Risque d'oublis

→ Solution: regrouper le code dupliqué en fonctions afin de les réutiliser à différents endroits.

## Qualité\_de\_code();

Les commentaires sont des parties du code source ignorées par la machine.

- Destinés à être lus par un humain
- Donnent des détails sur le code
- Génèrent automatiquement une documentation

→ Trop de commentaires = code trop complexe.

```
assert(testUnitaire()==TRUE);
```

Les tests unitaires permettent de détecter des bugs pendant le développement.

- Nécessitent des fonctions les plus indépendantes possible
- Ajoutent des coûts de développement
- Impliquent le développement de nouveaux tests unitaires quand les fonctions changent

→ Permettent de trouver des bugs sans relecture du code mais alourdi la charge de travail.

```
class Qualité_de_code extends RobertCMartin
```

« So if you want to go fast, if you want to get done quickly, if you want your code to be easy to write, make it easy to read. »

Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship