

Refactoring

Jérémie Primas
07.04.2022

Refactorer – Calisthénie

- Plusieurs niveaux d'indentation
- Mots clés else
- Plusieurs point par ligne
- Abréviations
- Utilisation de getters/setters
- ...

Refactorer – 5 règles

- Renommer
- Extract
- Inline
- Move
- Safe delete

Refactoring – Lisibilité

Le plus important : la lisibilité
Améliore la compréhension du code

Refactoring – Lisibilité

- Formatage
- Renommage - Noms explicites, plus d'abréviations
- Suppressions – commentaires excessifs, code mort
- Extraction
- Réordonner

Refactoring – Design

- Extraire vers des méthodes privées à partir de méthodes longues
- Encapsuler dès que possible
- Supprimer les doublons
- ...

Refactoring – Code smells

Bloaters

- Méthodes trop longues
- Classes trop grandes
- Obsession primitive
- Liste de paramètres trop longue

Refactoring – Code smells

Abus Orienté-Objet

- Switch statements ou séquence de ifs à éviter
- Fields temporaires : Lisibilité
- Faux héritage
- Classes alternatives avec des interfaces différentes – Mêmes fonctions mais noms différents

Refactoring – Code smells

Empêcheurs de changement

- Changement divergent : un changement entraîne trop de changements à faire dans une classe
- Shotgun surgery : un changement entraîne plein de petits changements dans différentes classes

Refactoring – Code smells

Coupleurs

- Feature envy : Une méthode accède aux données d'un autre objet de manière excessive
- Chaîne de messages : Trop de points : `Dog.Body.Tail.Wag();`
- Middle man : Classe qui délègue tout son travail : A supprimer
- ...

Merci