

ASTUCES

ASTUCES

Refactoring

- Commencer par la droite - il y a moins de dépendances
 - Rapprocher la déclaration des variables de leur utilisation
 - Déplacer le code vers l'objet qui contient les données
 - Extraire des méthodes et variables descriptifs
 - Faire validations en premier
-

Test legacy

- Ne pas toucher du code non testé
 - Si on doit toucher pour rendre testable: Uniquement refactorings automatiques:
 - "extract method and override"
 - "extract method"
 - "introduce parameter"
 - Commencer à tester par la gauche
 - Utiliser la couverture de test pour vérifier qu'on teste bien la branche qu'on pensait
-

Test driven development - base

- Toujours un test avant le code
 - Rapidement au vert
 - Pas de test cassé lors du refactoring
 - Pas de débogueur
 - Le nom des méthodes de test doivent être une documentation de la classe
-

TDD avec dépendances non testables

- Isoler les dépendances difficiles du reste de l'algorithme
- Définissez un contrat en testant ces dépendances en intégration (avec le vrai "truc") tant que possible
- Le reste de l'application peut se fier à ce contrat à l'aide des mocks et stubs
- Séparer les tests d'intégration ciblée des autres tests (autre job CI)