

TP3 - Pratique du TDD sans dépendances

À l'issue de ce TP pratiqué en Java vous aurez :

- pratiqué le TDD sur un exemple « algorithmique » simple
- pratiqué le TDD sur un exemple « objet » simple
- surveillé notre pratique du TDD avec un outil de couverture
- révisé l'approche fonctionnelle.

Attention : le but de ce TP est de s'astreindre à tester / programmer en TDD et pas d'écrire du code compliqué. Ça n'a **aucun intérêt** d'écrire les exercices à toute vitesse avec l'approche « coder puis tester » (et encore moins « tout code puis tout tester »). Il est fortement conseillé de pratiquer le **pair programming** : l'un des membres du binôme écrit pendant que l'autre le surveille, et toutes les 30 minutes il y a changement de rôle. Vous afficherez à chaque test la couverture que vos tests font de votre code : pas 100% implique TDD mal pratiqué.

Si vous vous sentez particulièrement mal à l'aise avec le TDD, vous pouvez commencer par refaire les démos de CTD en pair programming.

Pour surveiller que couverture du code par vos tests reste à 100% vous pouvez utiliser le plugin EclEmma pour Eclipse (de préférence) ou, si vous n'utilisez pas Eclipse, l'outil Jacoco sur lequel repose EclEmma (mais il faut comprendre comment fonctionne l'outil, ce qui est masqué avec EclEmma et est hors sujet pour ce TP). Jacoco est intégré à Ant, Maven, etc mais ce n'est pas facile de l'utiliser en ligne de commande.

<http://www.jacoco.org/index.html>

<http://www.jacoco.org/jacoco/index.html>

Exercice 1 : Compte de base

Écrire et tester en TDD une classe `Compte` basique, sans découvert autorisé et sans plafond, avec deux fonctionnalités : créditer et débiter.

Exercice 2 : Sous-échantillonnage

Écrire et tester en TDD une méthode `sous_echantillonne` qui prend en paramètre une liste, une longueur de sous-échantillon et une fréquence d'échantillonnage, et qui retourne la liste des sous-échantillons concaténés.

Exemple Supposons qu'on souhaite sous-échantillonner la liste `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`, à une fréquence de 3 éléments et avec une longueur de sous-échantillons de 2. Tous les 3 éléments (fréquence de 3), donc à partir des éléments 1, 4, 7 et 10, on prélève un sous-échantillon de taille 2 : `[1, 2]`, puis `[4, 5]`, puis `[7, 8]`, puis `[10]` (ce dernier échantillon est incomplet car la liste n'est pas suffisamment longue pour le terminer). Le résultat de l'échantillonnage est donc la liste `[1, 2, 4, 5, 7, 8, 10]`.

Vous devez faire attention au comportement de votre méthode en fonction des données suivantes : longueur de la liste à échantillonner, longueur des sous-échantillons, fréquence. Les tests doivent indiquer clairement vos choix, avec des noms adaptés.