

Travail pratique #5 - Tests d'intégration OO
Automne 2018

1. Mise en contexte théorique

Ce travail pratique est l'application des notions théoriques vues dans le [cours](#) sur les tests d'intégration orienté objet. Ce type de test vise à tester les interactions entre les classes tout en déterminant un ordre optimal d'intégration. Ce TP se concentre sur les Diagrammes de Relation Objet (**ORD**), classes coupe-feu (**CFW**), diagrammes d'[ordre topologiques](#) et les niveaux de tests.

2. Objectifs

- Implémentation d'un test d'intégration orienté objet selon le Diagrammes de Relation Objet (**ORD**).

3. Mise en contexte pratique

Durant ce travail pratique, vous allez tester un mini projet écrit en Java qui implémente des listes et des listes chaînées personnalisées. Le projet permet de créer des listes à partir du résultat d'une opération sur deux ensembles d'éléments spécifiés par l'utilisateur. Les opérations disponibles sont listées dans le fichier «`Operator.java`» et leurs implémentations se trouvent dans la classe `SetCalculator`. Pour vous familiariser avec le projet, commencez par l'exécution des exemples fournis dans «`Main.java`», ensuite vous pouvez essayer vos propres exemples pour mieux comprendre le fonctionnement et les liens entre les différentes classes.

4. Travail à effectuer

- Établir un Diagramme Relation Objet (ORD) pour toutes les classes du projet. (Référez-vous aux diapositives 29 et 30 du cours [Tests d'intégration OO](#). Veuillez considérer les interfaces comme classes abstraites, et la relation entre une classe abstraite et son implémentation comme I).
- Pour chaque classe (C), retrouver CFW(C), soit l'ensemble des classes qui peuvent être affectées par un changement dans la classe C. (Référez-vous aux diapositives 31 et 32 du cours [Tests d'intégration OO](#))
- Établir un diagramme d'ordre topologique.
- Établir un tableau de niveau de test.

- Selon l'ordre établi précédemment, identifiez et implémentez les cas de test pour chaque classe (il faut un cas de test pour chaque méthode). Il n'y a pas de critère particulier à respecter dans les cas de test. Vous êtes donc libres de choisir un critère adéquat.

Exigences :

- Pour chaque cas de test, la méthode de test devra être commentée en identifiant le cas de test en question.
- Spécifiez l'ordre explicitement pour les méthodes de test selon les niveaux de tests. Consultez la documentation de JUnit pour voir comment définir un ordre de test.

5. Livrable à remettre, procédure de remise et retard

1. **Une** classe de test unitaire pour toutes les classes. **NB: votre package doit être nommé TP5**
2. Un document **PDF** qui contient :
 - a. Le diagramme Relation Objet (ORD)
 - b. Tableau CFW pour chaque classe
 - c. Diagramme d'ordre topologique
 - d. Tableau de niveau de test

Veuillez envoyer vos fichiers dans une archive de type ***.zip** (et seulement zip, pas de rar, 7z, etc) qui portera le nom : **LOG3430_lab5_MatriculeX_MatriculeY.zip** tel que :
MatriculeX < MatriculeY.

Date limite pour la remise :

Groupe B1 & B2 : Lundi 26 novembre à 7:00 du matin

6. Barème de correction

Diagramme Relation Objet (ORD)	2 points
CFW	2 points
Diagramme d'ordre topologique	2 points
Tableau de niveau de test	2 points
Test avec JUnit : <ul style="list-style-type: none"> - Tests exécutables - Cas de test bien identifiés - Niveaux de tests bien spécifiés 	12 points

<ul style="list-style-type: none"> - Commentaires - Qualité 	
---	--

Les travaux en retard seront pénalisés de 20% par jour de retard. Aucun travail ne sera accepté après 4 jours de retard. Si votre dépôt ne respecte pas la nomenclature définie ci-dessus, 0.5 point de pénalité sera appliqué.