

TP n°1

Introduction

L'objectif de ce premier TP était de mettre en œuvre un jeu de tests unitaires grâce à *JUnit* afin de couvrir 100 % des méthodes publiques, puis de produire la JavaDoc et un rapport de couverture de tests à l'aide de Maven et *Jacoco*. Dans un premier temps, nous allons répondre aux questions préliminaires posées, puis nous discuterons de nos tests unitaires notamment via la liste des bugs qu'ils nous ont permis d'identifier.

Questions préliminaires

Question 1

On ne sait pas si l'objet « o » a correctement été ajouté à la liste et donc le test qui vérifie que la taille de la liste est 0 ne prouve pas le bon fonctionnement de la méthode « remove ».

Question 2

Il n'y a pas forcément d'erreur dans « add », elle peut être dans la méthode « remove ».

Question 3

Il faut définir une assertion avant l'appel à la méthode « remove » et après, afin de tester la taille de la liste, puis vérifier l'ordre des objets dans la liste. Enfin, il faut vérifier l'absence de l'objet supprimé.

Question 4

L'ordre des méthodes de test n'a aucune importance, car chaque cas de test est indépendant.

Question 5

Il est possible que la méthode « add » n'utilise qu'un sous-domaine de la méthode « addBefore ». Si on considère « add » comme correcte, alors on peut considérer le sous-domaine de « addBefore » comme correct, mais il se peut que certains cas n'aient pas été testés, sur le reste du domaine.

Tests unitaires

Ci-dessous se trouve une liste de bugs que nous avons identifié et corrigé grâce à nos différents cas de test.

Méthode contains(Object o) - ligne 184

Condition « > » modifiée en « >= ». Ce bug a été trouvé par le cas de test « testContains_int ».

Méthode set(int i, Object o) - ligne 227

Incrémentation de l'index avant l'affectation. → ++index modifié par index++. Ce bug a été trouvé par le cas de test « testSet ».

Méthode remove(Object o) - ligne 259

Condition « != » remplacée par « == ». Ce bug a été trouvé par le cas de test « testRemove_null ».

Méthode removeAll() - ligne 413

Test « r < size - 1 » remplacé par « r < size ». Ce bug a été trouvé par le cas de test « testRemoveAll_filledList_atEnd ».

Conclusion (Question 6)

Ce premier TP nous a permis de nous familiariser aux tests via JUnit et à la couverture de nos cas de test via Jacoco, ce dernier nous ayant permis de suivre efficacement l'avancement de notre travail. On notera cependant qu'il peut être très difficile ou impossible de couvrir 100 % du code, par exemple si certaines parties de celui-ci sont du « code mort » ou dans des méthodes privées ou si le chemin suivi à l'exécution dépend d'une génération aléatoire. On peut donc considérer que même si la couverture du code n'est pas complète, le jeu de test peut être suffisant. Les statistiques qu'offre Jacoco nous donne quand même une bonne indication sur la qualité du jeu de test.