

**Travail pratique #4 - Tests basés sur les états
Automne 2018**

1. Mise en contexte théorique

L'approche des tests basés sur les états consiste à générer un arbre des transitions à partir du diagramme d'état d'un programme ou une composante sous test. Les chemins qu'on peut trouver dans cet arbre représentent les transitions et les noeuds représentent les états. Chaque transition dans l'arbre est un chemin qui part d'un état S et revient à ce même état. Le retour à l'état S marque la fin de la branche de l'arbre. Donc, il ne peut pas y avoir des répétitions de chemins dans l'arbre. Cette méthode de test nous permet de modéliser les transitions simples, qui ne contiennent pas de boucles, afin de générer les cas de tests correspondant. Pour plus de détails et d'exemples, voir les notes de cours sur les [Tests d'états](#).

2. Objectifs

- Représenter la classe `Queue` avec un diagramme d'états.
- Générer l'arbre des transitions à partir d'un diagramme d'états.
- Identifier les cas de test et les implémenter à l'aide de JUnit.

3. Mise en contexte pratique

Pour réaliser ce travail, il faut commencer par construire le [diagramme d'états](#) de la classe `Queue` du TP3. Ensuite, il faut générer [l'arbre des transitions](#) correspondant en spécifiant les états et les conditions de garde pour chaque transition, s'il existe une. Finalement, il faut identifier les cas de test à partir de l'arbre trouvé et les implémenter avec JUnit.

4. Travail à effectuer

1. Construire le diagramme d'états de la classe `Queue`. **Indice : La classe peut être représentée avec 3 états seulement.**
2. Construire l'arbre des transitions de la classe `Queue`.
3. Identifiez tous les cas de tests (avec les conditions de garde, s'il y en a) à partir de l'arbre trouvé.

4. À l'aide de JUnit, écrire une classe de test unitaire pour tester les cas de test identifiés dans la question précédente.
5. À l'aide de l'outil JaCoCo, évaluez la couverture de la classe `Queue` selon la couverture des branches et identifiez les branches non couvertes, s'il y en a.

5. Livrable à remettre, procédure de remise et retard

1. Vos fichiers de test *.java. **NB: votre package doit être nommé TP4**
2. Le rapport JaCoCo au format HTML
3. Un rapport de quatre pages au maximum (PDF) contenant le diagramme d'états, l'arbre des transitions et les cas de tests identifiés.

Veuillez envoyer vos fichiers dans une archive de type *.zip (et seulement zip, pas de rar, 7z, etc) qui portera le nom : **LOG3430_lab4_MatriculeX_MatriculeY.zip** tel que :
MatriculeX < MatriculeY.

Date limite pour la remise :

Groupe B1 & B2 : Lundi 12 novembre à 7:00 du matin

6. Barème de correction

Rapport de 4 pages

1.5 points

- Diagramme d'états, arbre des transitions et cas de tests
- Pertinence des explications et des analyses
- Respect du nombre de pages

Tests

3.5 points

- Respect des consignes
- Cas de tests bien identifiés et commentés
- Tests exécutables
- Qualité du code

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.