femto-st SCIENCES & DISC TECHNOLOGIES

UFC UNIVERSITÉ DE FRANCHE-COMTÉ

JDEV2013

Atelier T6.A5
Bien écrire les tests de son composant logiciel Java
Déroulement de la séance

Assistance au démarrage de la séance

- récupérez ce document sur http://jdev2013.lix.polytechnique.fr/Ateliers/T6/
- récupérez les sources soit sur
    - https://github.com/cnrs-jdev
        - sélectionnez la branche step1 puis Download Zip
    - http://jdev2013.lix.polytechnique.fr/Ateliers/T6/
        - récupérez unitestor.java-step1.zip
- décompressez le zip dans le répertoire de votre choix
- lancez votre IDE
- ouvrez ou créez le projet

Description du projet au début de l'étape 1

- les sources modélisent un Robot capable de se déplacer. Le robot dispose de ses coordonnées et de sa direction (points cardinaux). Il peut accomplir 2 actions modifiants ses coordonnées (déplacement avant et déplacement arrière) et 2 actions pour modifier sa direction (tourner à droite et tourner à gauche). Dans l'état actuel de l'implémentation, certaines classes n'ont pas encore d'utilité.
- la classe de test RobotUnitTest contient les premiers tests associés à la classe Robot. Ces 2 classes sont dans le même package *robot* mais dans des répertoires différents
- la classe BatteryUnitTest teste la classe Battery

Etape 1 : que faire ?

- ouvrir dans l'éditeur de l'IDE la classe BatteryUnitTest
- comprendre et exécuter les tests un à un
- exécuter tous les tests de la classe et corriger les tests en échec s'il y en a
- effectuer la même chose sur la classe RoboUnitTest
- la classe Robot n'est pas complètement testée, ajouter les tests manquants

Fin de l'étape 1 :

Nous avons abordé :
- le test unitaire de méthodes simples en utilisant la classe org.junit.Assert
- le test d'apparition d'exceptions
- le test de méthodes privée

En cas de problème avec l'état d'avancement de vos sources, vous pouvez récupérer les sources de l'étape 2.

- https://github.com/cnrs-jdev
    - sélectionnez la branche step2 puis Download Zip
- http://jdev2013.lix.polytechnique.fr/Ateliers/T6/
    - récupérez unitestor.java-step2.zip

Etape 2 :

L'objet de l'étape 2 est d'introduire la prise en compte de la consommation d'énergie qui intègre la consommation de base modulé par les aspérités du terrain. Pour ce faire, il faut reprendre le code de Robot et intégrer dans les méthodes en charge des déplacements la prise en compte de la consommation d'énergie.
Les tests sont eux aussi à mettre à niveau.

Fin de l'étape 2 :

Nous avons abordé :
- le test unitaire en isolation par l'utilisation de mock et stubs avec mockito

En cas de problème avec l'état d'avancement de vos sources, vous pouvez récupérer les sources de fin de l'étape 2.

- https://github.com/cnrs-jdev
    - sélectionnez la branche step3 puis Download Zip
- http://jdev2013.lix.polytechnique.fr/Ateliers/T6/
    - récupérez unitestor.java-step3.zip