



TESTS UNITAIRES EN JAVA AVEC JUNIT

Rédigé par Yannick Le Nir

Ref : *CO-UNIT-TEST*

Revu par : l'équipe pédagogique

Crée le 21/11/2012

1 Présentation :.....	2
2 Test Unitaires :.....	2
3 Implémentation des tests unitaires avec Junit 4 :.....	2
4 Assertions.....	2
5 Importation des classes Junit.....	3
6 Exécution des tests et récupération des résultats :.....	3

1 Présentation :

JUnit est un « framework » permettant l'automatisation des tests unitaires pour les classes Java, créée par Kent Beck et Erich Gamma. La version actuelle est JUnit 4, qui apporte des évolutions majeures par rapport à la précédente version JUnit 3.8. Nous nous efforcerons de présenter les différences essentielles entre ces deux versions, bien qu'il existe une compatibilité descendante entre JUnit 4 et JUnit 3.8. En effet, la plupart des entreprises utilise encore la version 3, qui n'autorise par exemple pas les annotations dont la version 4 fait largement usage. Ces annotations permettent de tirer partie des ajouts faits dans Java à partir de Java 5.0. JUnit peut être utilisé directement en ligne de commande (Il s'agit d'un jar), ou via un plugin pour la plupart des IDE tels eclipse, netbeans ou intellij.

2 Test Unitaires :

Les tests unitaires sont des programmes écrits par le développeur qui exécute des fonctionnalités spécifiques d'un code à travers certains tests. L'objectif des test unitaires est de s'assurer qu'un code fonctionne comme prévu par les différents tests choisis (cf cours précédent). Les tests unitaires permettent également de valider qu'un code fonctionne toujours correctement après certaines modifications qu'on lui aurait apporté. On parle dans ce cas de tests de non régression, sur lesquelles nous reviendront au prochain cours.

Les différentes étapes pour l'exécution de tests unitaires sont les suivantes :

1. Préparer (set up) l'environnement de test pour être exactement dans les conditions requises par le test. Cela revient à définir et appliquer des pré-conditions (par exemple initialiser des champs, activer les logs, ...)
2. Exécuter les cas de test (CT) que vous avez prévus lors de la phase d'écriture du programme (cf cours précédent).
3. Evaluer les résultats ou effets de bords engendrés par l'exécution des CT pour vérifier qu'ils correspondent à ce qui a été prévu par les oracles.
4. Nettoyer (tear down) l'environnement de test si besoin, afin que les tests suivants puissent être exécutés sans problèmes (post-conditions).

3 Implémentation des tests unitaires avec Junit 4 :

JUnit est un « framework » permettant d'implémenter et de d'exécuter des test unitaires selon les étapes vues ci-dessus. L'ensemble de ces étapes est implémenté au sein d'une classe Java test. Les méthodes de la classe test sont décorées par des annotations permettant à Junit d'effectuer les actions suivantes :

1. Set up : les actions devant être exécutées pour préparer l'environnement sont implémentées dans une ou plusieurs méthodes, annotées avec **@Before**.
2. Test : les CT sont implémentés dans des méthodes annotées avec **@Test**. L'évaluation des résultats attendus (oracle), est également implémentée dans ces méthodes en utilisant des assertions (par exemple **assertEquals(ValeurAttendue,ValeurCalculee)**).
3. Tear down : le nettoyage de l'environnement peut être implémenté dans une ou plusieurs méthodes annotées par **@After**, qui sont exécutées après chaque test.

4 Assertions

Afin de comparer les différentes valeurs du programme avec l'oracle, Junit offre un certain nombre de méthodes appelées assertions. La plus utilisée est la méthode **assertEquals(ValeurAttendue,ValeurCalculee)**.

Vous trouverez toutes les autres assertions dans la documentation.(JavaDoc)

5 Importation des classes Junit

Afin de compiler correctement votre programme doit importer les classes de Junit utilisées, par exemple :

```
import org.junit.Test;
```

```
import static org.junit.Assert.assertEquals;
```

Lors de la compilation ou de l'exécution en ligne de commande, n'oubliez pas d'inclure Jnit (jar) à votre classpath.

6 Exécution des tests et récupération des résultats :

Le « framework » JUnit 4 propose la classe **org.junit.runner.JUnitCore**, afin d'exécuter les tests et de récolter les résultats. Les tests peuvent être exécutés depuis un simple programme Java utilisant la méthode **org.junit.runner.JUnitCore.runClasses()**. Les tests peuvent être également lancés en ligne de commande avec : **java org.junit.runner.JUnitCore**

Les résultats sont affichés par défaut dans la console.