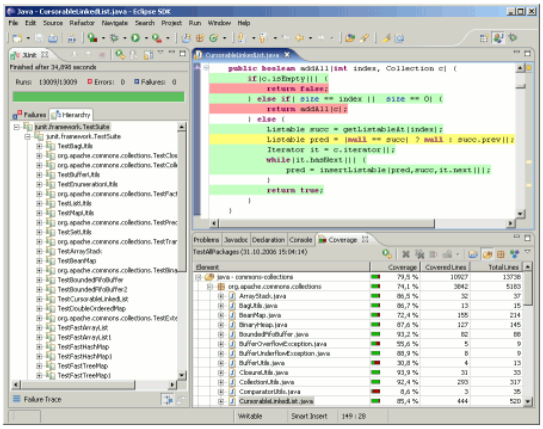
Analyse de la couverture de code avec Jacoco (Java Code Coverage)

Documentation : http://www.eclemma.org/jacoco/trunk/doc/index.html

Description

La couverture de code est une mesure importante car elle indique la qualité des tests et peut également jouer le rôle de condition d'arrêt des tests. L'utilisation d'un outil d'analyse de la couverture de code permet de savoir les parties du code qui n'ont pas encore été testées et pour lesquelles des tests additionnels devront être écrits. Il existe plusieurs outils d'analyse avec des fonctionnalités différentes : Emma, Corbetura, Jacoco.



Dans le cadre de ce lab, nous utiliserons le dernier outil Jacoco.

Jacoco est une librairie d'analyse de couverture de code pour Java. Cette librairie, gratuite et open source, est simple d'utilisation et flexible. Jacoco est developpé sous le projet eclEmma.

Jacoco permet de mesurer la couverture des lignes et des branches. Elle offre une visualisation graphique de la couverture du code et fournit des rapports détaillés de l'analyse de la couverture.

Installation

http://www.eclemma.org/installation.html

Utilisation

Jacoco peut s'utiliser de plusieurs manières. Il peut être utilisé comme un plugin intégré à Eclipse (EclEmma). Il peut également être intégré aux tâches Ant ou à d'autres programmes ou outils java ou encore être exécuté à partir de la de commande.

Utilisation de Jacoco en tant que plugin d'Eclipse

Écrire un cas de test et l'exécuter en utilisant le mode d'exécution d'eclEmma (Jacoco)

Exécution : Run/ Coverage as

Configuration : Run/Coverage…

Dans l'onglet coverage, choisir les projets devant être pris en compte lors du calcul et de l'analyse de la couverture

Analyse : Les résultats de couverture peuvent être visualisés via la vue Coverage.

Si la vue n'est pas visible ; allez dans Window/Show View/Other/Java/Coverage

Dans la vue Coverage, la couverture est affichée par projet pour les différents projets analysés (vous pouvez aller à une plus petite granularité en déroulant au fur et à mesure le projet, le package, le fichier source et la classe.

Avec un clic droit, vous pouvez exporter/importer les données de couverture en format xml, Csv, jacoco ou html.

Les classes ayant été analysées sont colorées : les lignes rouges indiquent les lignes non couvertes, les lignes jaunes, les lignes partiellement couvertes et les lignes vertes, les lignes entièrement couvertes par les tests.

Métriques de couverture (dans le fichier .xml)

Jacoco propose différentes métriques de couverture.

Couverture des instructions : **Instructions (C0 Coverage)**

(Ici, le traitement se fait sur des byte code ; une instruction peut donc être différente d'une ligne de code)

La couverture des instructions indique le nombre d'instructions exécutées (couvertes) ou non (non couvertes) par les tests.

Couverture des branches : La couverture des branches donne le nombre de branches couvertes et non couvertes. Les branches sont tous les points de décision constitués de if ou switch ou les conditions dans les boucles for, while, …

NB. : La gestion des exceptions n'est pas considérée comme une branche dans les contexte de cette métrique.

Couverture des lignes : Il s'agit des lignes dans le fichier source. Une ligne est couverte dès qu'une instruction bytecode issue de cette ligne est couverte (exécutée).

Couverture des méthodes : Une méthode est couverte si elle a au moins une ligne exécutée (couverte).

Couverture de la classe : Une classe est couverte si au moins l'une des ses méthodes est exécutée (couverte).

On peut avoir tout ou partie de ces mesures au niveau des fichiers source, de la classe et des méthodes.

Exercices

Utiliser Jacoco sur les tests écrits dans la section JUnit et analyser les rapports obtenus en faisant les exports en xml et html.