

## Travaux Pratiques Test – TP1

L'objectif de ces TP est la prise de contact avec un logiciel de test et la simulation d'une campagne de tests simplifiée.

Le logiciel utilisé est le logiciel EMMA, qui permet de mesurer la couverture de code.

### Logiciel EMMA

EMMA est un outil open-source qui permet de mesurer et visualiser la couverture de code Java.

Le site de l'outil, pour explications, exemples commentaires est le suivant :

<http://emma.sourceforge.net/>

Le même site vous permet de télécharger le logiciel.

Le logiciel dispose d'une version plug-in Eclipse et d'une version indépendante. Vous avez l'entière liberté quant à l'utilisation d'une de ces versions, cependant nous donnons ici uniquement les consignes pour l'installation de la version indépendante.

Pour utiliser le logiciel il faut que emma.jar se trouve dans votre répertoire de travail, ou qu'il soit visible à cet endroit.

Le choix de l'outil que vous utilisez dans ces TP a eu pour but de vous familiariser avec un outil que vous pourrez utiliser facilement dans tout développement Java (logiciel open-source, compact et facile à utiliser).

En principe, EMMA est utilisé en complément d'une suite de test fonctionnel (e.g. JUnit), mais le logiciel peut aussi aider pour des tests, moins formels, effectués « à la volée ».

Pour calculer la couverture du code EMMA peut prendre en compte soit une seule exécution, soit plusieurs, après une *instrumentation* du code source. Pour cela, EMMA a besoin d'une copie locale du code source à instrumenter, qu'elle va modifier en lui rajoutant des informations dont EMMA a besoin pour calculer les couvertures cumulées.

En plus de ça, EMMA garde, dans un fichier local coverage.em, des meta-informations sur le fichier à tester. Avec ces informations on peut tester en mode différé le logiciel instrumenté.

### Travail à effectuer

Nous allons considérer un des exemples les plus complexes de la distribution de java : SwingSet2.

1. Familiarisez-vous avec l'exemple pour le lancer à partir de la ligne de commande vous tapez :  
`java -jar SwingSet2.jar`
2. jouez avec l'exemple en indiquant à EMMA de suivre la couverture de code qui correspond à vos actions  
`java -cp emma.jar emmarun -jar SwingSet2.jar`

3. Effectuez la même chose qu'au point 2, tout en générant un fichier .html  
`java -cp emma.jar emmarun -r html -jar SwingSet2.jar`
4. Paramétrez EMMA pour changer la présentation du fichier .html
5. Effectuez la même chose qu'au point 3, tout en suivant la couverture sur le code source :  
`option -sp ../jdk1.4.2/demo/jfc/SwingSet2/src`
6. Initialiser l'instrumentation du code source (après avoir créé une copie locale)  
`cp ../jdk1.4.2/demo/jfc/SwingSet2/SwingSet2.jar .`
7. `java -cp emma.jar emma instr -m overwrite -cp SwingSet2.jar`
8. Lancez l'exécution de l'exemple en mode différé  
`java -cp SwingSet2.jar:emma.jar SwingSet2`
9. Générez le rapport de couverture qui utilise les données obtenues lors des exécutions antérieures :  
`java -cp emma.jar emma report -r html -in coverage.em,coverage.ec`
10. Suivez la couverture sur le code source.

### Consignes sur le déroulement de TP de Test

Il est conseillé aux étudiants :

- de se regrouper par équipes de max 3 étudiants;
  - de rédiger au fur et au mesure du déroulement des séances de TP un document résumant le résultat du travail effectué par les membres de l'équipe lors de chaque TP et exposant les solutions. Ces documents permettront de suivre le travail effectué par chaque équipe. Ils devront être déposés sous moodle lors de la dernière séance de TP.
- Le rapport de TP sera noté et sera pris en compte dans la note de contrôle continu (50% l'autre 50% étant le QCM).