

## TP 4 : Analyse de mutation

Le but de ce TP est la mise en application de la technique de l'analyse de mutation. Vous aurez à évaluer les tests réalisés au cours des TP 2 et 3 sur le système de vente aux enchères, en utilisant l'outil *µjava* au sein du plugin *muclipse* (<http://muclipse.sourceforge.net/>).

### 1 µjava et muClique

#### 1.1 Présentation de µjava

µjava est un outil développé pour l'étude et l'évaluation des opérateurs de mutation, particulièrement orientés objets pour le langage *Java*. Son utilisation peut être faite au travers d'Eclipse en utilisant le plugin muClique. Le but final est de vérifier que l'analyse de mutation permet d'obtenir des tests de qualité. Vous trouverez sur la page web du cours les documents décrivant les différents opérateurs de mutation proposés par l'outil, aussi bien au niveau des classes que des méthodes.

#### 1.2 Installation de muClique

Référez vous aux procédures décrites aux adresses :

- <http://muclipse.sourceforge.net/updatesite.php> pour l'installation du plugin
- <http://muclipse.sourceforge.net/setup.php> pour la préparation de votre projet

#### 1.3 Utilisation de muClique

L'analyse de mutation est un processus en deux étapes que reprend muClique sous la forme de deux types de configuration d'exécution :

- Génération des mutants pour un programme donné et un ensemble d'opérateurs de mutation donné.  
Cf. <http://muclipse.sourceforge.net/mutants.php>
- Vérifications des tests sur l'ensemble des mutants. Cf. <http://muclipse.sourceforge.net/tests.php>

Vous pouvez également visualiser les différences de code entre chaque mutant et le programme initial. Voir

- <http://muclipse.sourceforge.net/results.php>

### 2 Première utilisation de muClique

Reprenez le processus d'utilisation de muClique décrit ci-dessus sur le projet mis à votre disposition à l'adresse <http://www.combemale.fr/vv>

Importez le projet dans votre workspace et créez :

- 1) une configuration d'exécution pour générer les mutants (*Run > Run configuration... : MuClique : Mutants*),

Rq : dans l'onglet « operators », sélectionner uniquement « Traditional operators ».

- 2) une configuration d'exécution pour exécuter les tests sur l'ensemble des mutants (*Run > Run configuration... : MuClique : Tests*). Analysez les résultats. Vous pouvez également visualiser les mutants grâce à la vue *Mutants and Results*.

### 3 Etude du système de ventes aux enchères

Évaluez ensuite (et améliorez si nécessaire) vos tests réalisés au cours des TP 2 et 3 sur le système de ventes aux enchères. Vous commencez dans un premier temps par évaluer les tests de la classe *AuctionImpl*. Pour cela, créez d'abord une configuration d'exécution pour générer les mutants (*Run > Run configuration... : MuClique : Mutants*, dans l'onglet « operators », pensez dans un premier temps à sélectionner uniquement les « Traditional operators »). Une fois les mutants générés, vous pouvez les visualiser en utilisant la vue *Mutants and Results*.

Créez ensuite une configuration d'exécution pour exécuter les tests sur l'ensemble des mutants (*Run > Run configuration... : MuClique : Tests*).

Rq : il semble préférable pour utiliser l'outil d'isoler les classes de test dans un répertoire différent (*New > Source Folder*, p.ex. *unittests*) de celui des classes sources (généralement *src*). **Les classes de tests devront être en JUnit3, et compatible Java1.4.** Vérifiez et analysez alors votre score de mutation. Améliorez si nécessaire vos tests de la classe *AuctionImpl*.

Poursuivez la séance de TP en appliquant l'analyse de mutation sur les autres classes, et en appliquant des opérateurs de mutation au niveau des classes.