

Ingénierie Logicielle 1

Mesures qualité logicielle

Arnaud Lanoix*

Tout le long du sujet de TP, il vous est demandé de compléter différents fichiers .java (voir **A faire.**). Vous rendrez ces fichiers via Madoc à la fin du TP, sous la forme d'une archive NOM-Prenom.zip.

1 Configuration d'Eclipse

1.1 Réglages du proxy

Sous linux, lancer Eclipse sans plugin.

Dans Window > Preferences > General > Network connections, choisir comme proxy (Active Provider) = "Manual" :

- Host http = proxy-etu : 3128
- Host https = proxy-etu : 3128

Valider, puis redémarrer Eclipse (File > restart).

1.2 Installation des plugins

Dans Eclipse, accéder au "Market" : Help > Eclipse Marketplace

Installer les plugins suivants :

1. Eclipse Metrics
2. Eclemma Java Code Coverage
3. Checkstyle Plug-in
4. FindBugs Eclipse Plugin
5. Eclipse-pmd

NB : vous pouvez installer tous les plugins nécessaires avant de redémarrer Eclipse.

1.3 Importation

Dans un nouveau projet Java, importer les sources Java fournies : mesures.tar.gz.

Vous devriez avoir les paquetages suivants qui contiennent des classes présentant à chaque fois un exemple à vérifier :

- checkstyle
 - Check.java
- eclemma
 - Calcul.java
 - TestCalcul.java
- findbugs
 - Find.java
- metrics
 - Metrics.java
- pmd
 - Pmd.java
- another_ruleset.xml
- ruleset.xml

*Arnaud.Lanoix@univ-nantes.fr

2 En pratique

2.1 Metrics

Plus d'info sur <http://www.stateofflow.com/projects/16/eclipsemetrics>.

Pour activer Metrics clic droit sur le projet, puis Properties > Metrics : Enable Metrics Gathering.

Dans l'onglet Problems, des warnings donnent des informations sur les "mesures" dépassées.

Le paramétrage de Metrics s'effectuent via Window > Properties > Metrics.

A faire. Quelle est la complexité cyclomatique de la méthode complexity() du paquetage metrics ? vérifier la valeur obtenue en dessinant le graphe de flot de contrôle correspondant à cette méthode, puis en comptant le nombre de chemins.

Penser à désactiver Metrics.

2.2 Eclemma

Plus d'info sur <http://eclemma.org/>.

Exécuter en mode "Coverage", la classe TestCalcul du paquetage eclemma.

Dans l'onglet Coverage, vous pouvez observer tous les détails concernant les taux de couverture.

A faire. Compléter la classe TestCalcul pour obtenir 100% de couverture pour la classe Calcul.

Penser à désactiver Eclemma.

2.3 Checkstyle

Plus d'info sur

- <http://checkstyle.sourceforge.net/>
- <http://eclipse-cs.sourceforge.net/>

Activer/configurer Checkstyle : clic droit sur l'icone du projet, puis Properties > Checkstyle.

Cocher Checkstyle active for this project.

Choisir Sun Checks (Eclipse) comme conventions de code vérifiées par Checkstyle.

Pour lancer Checkstyle sur une classe donnée, clic droit sur l'icone de la classe, puis Checkstyle > Check code with Checkstyle.

L'onglet Checkstyle violations affichent des détails sur les conventions de code non respectées.

A faire. Corriger la classe Check du paquetag checkstyle pour qu'elle soit complètement conforme aux conventions de code de Sun (au sens de Checkstyle).

Penser à désactiver Checkstyle.

2.4 Findbugs

Plus d'info sur

- <http://findbugs.sourceforge.net/>
- <http://findbugs.sourceforge.net/manual/eclipse.html>

Pour lancer Findbugs sur un projet java, clic droit sur le projet, puis Findbugs > find Bugs. Comme exemple vous pouvez utiliser la classe Find du paquetage findbugs.

Les bugs détectés sont affichés dans les onglets Bug Explorer et Bug Info.

Plus de réglages dans Properties > Findbugs.

A faire. Compléter la classe Find du paquetage findbugs pour montrer (au minimum) 5 autres exemples "intéressants" de "bugs potentiels" détectés par Findbugs. Vous pouvez également ajouter de nouvelles classes, etc. En commentaire, vous donnerez toutes les explications nécessaires.

Penser à désactiver findbugs.

2.5 PMD

Plus d'info sur

- <http://pmd.github.io/>
- <http://acanda.github.io/eclipse-pmd/>

Pour activer PMD, clic droit sur le projet, puis **Properties > PMD**. Activer PMD, puis ajouter le fichier de règles `ruleset.xml`. Dans l'onglet **Problems** vous devez voir de nouveaux warnings concernant la classe `Pmd` du paquetage `pmd`.

A faire. Quelle est la complexité cyclomatique de la méthode `complexity()` ?

A faire. Ajouter le fichier de règles `another_ruleset.xml` puis compléter la classe `Pmd` du paquetage `pmd` pour montrer (au minimum) 10 autres exemples “intéressants” de règles détectées par PMD. Vous pouvez également ajouter de nouvelles classes, etc. En commentaire, vous donnerez toutes les explications nécessaires.