

# JUnit4

*Rapport après amélioration du code*



**Vandroemme Téo**

Génie Logiciel

## SOMMAIRE :

- 1 - Introduction
- 2 - Code Smells
- 3 - Code Deprecated
- 4 - Correction de bugs

## INTRODUCTION :

Dans le cadre du cours de Génie Logiciel de semestre 6, j'ai amélioré le code du logiciel Junit4. Pour réaliser ce projet j'ai utilisé plusieurs outils, tels que SonarQube, le dépôt Git du logiciel mais aussi le cours de génie logiciel. Je me suis concentré sur trois points, les codes smells, le code deprecated et les corrections de bugs.

## CODE SMELLS :

Le logiciel comporte énormément de code smells( plus de 1000), il est donc impossible de tous les résoudre en temps voulu. J'ai donc décidé de me concentrer sur quelques-uns d'entre eux, au total j'en ai corrigé un peu plus de 25.

La correction de ces code smells est en général assez simple, cela consiste la plupart du temps à corriger des petites erreurs de code. Comme par exemple dans le fichier TestSuite.java, il suffisait d'inverser 'static public' pour obtenir 'public static'.

## CODE DEPRECATED :

Cette partie a été plus longue et la plus compliquée. Beaucoup de classes du logiciel sont du code deprecated, j'ai donc choisi de me concentrer sur une seule classe. J'ai choisi la classe Assert, car c'est l'une des classes les plus grandes du logiciel.

Pour réaliser cette partie j'ai commencé par supprimer la classe, puis j'ai observé les erreurs que cette suppression a provoquées. J'ai ensuite réglé les bugs un à un, pour que le code soit de nouveau fonctionnel.

Puis il a fallu adapter les tests au nouveau code de la nouvelle classe Assert. Notamment en modifiant les exceptions qui sont différentes avec cette nouvelle classe.

En supprimant le code deprecated, un certain nombre de problèmes concernant la duplication de code ont été résolus.

## CORRECTION DE BUGS :

SonarQube permet de voir les différents problèmes de code, et les classes selon leur gravité. J'ai donc corrigé un certain nombre de ces problèmes, des problèmes majeurs mais également des problèmes mineurs.

Par exemple, un des problèmes majeurs concerne la déclaration de variable volatile. J'ai donc résolu ce problème en modifiant la déclaration des variables.

J'ai également supprimé des variables qui n'étaient pas utilisées dans le code, il y en avait une dizaine.

## CONCLUSION :

En conclusion, un certain nombre de problèmes du code ont été résolus. Certaines pistes d'amélioration de code n'ont pas abouti, comme par exemple les problèmes de sécurité du code.