OPL – Code Rewinder

*Le réparateur automatique de bugs par agrégation de code multi-version*

Code Rewinder est un réparateur automatique de bugs découpé en trois principales parties. Dans un premier temps, il télécharge les n dernières versions du projet sur GitHub. Dans un second temps, il construit, grâce à ces versions, une nouvelle version qui contient toutes les versions des fonctions. Et pour finir, il exécute les tests unitaires tout en faisant varier les versions des méthodes jusqu’à trouver le plus de tests « successful » possibles.

WATTEBLED Étienne

[Choisir la date]

OPL – Code Rewinder

Le réparateur automatique de bugs par agrégation de code multi-version

Table des matières

[Introduction 2](#_Toc469252558)

[Améliorations et configurations 3](#_Toc469252559)

# Introduction

Code Rewinder est un réparateur automatique de bugs utilisant les n derniers commits de SVN pour parvenir à ses fins.

Il fonctionne principalement en trois temps :

* Avant tout, il doit exécuter un batch permettant de télécharger les n dernières versions du projet à réparer sur GitHub.
* À l’aide de ces versions, il construit en utilisant une librairie appelée Spoon, une nouvelle version comprenant toutes les versions des méthodes. Un switch sur un attribut static de la classe permet de passer une méthode d’une version à l’autre.
* Et il exécute ensuite les tests unitaires tout en faisant varier les versions des méthodes jusqu’à obtenir le maximum de tests « successful » possibles.

Avant tout, quelques améliorations ont été apportées à Code Rewinder.   
Par la suite, Code Rewinder a été exécuté pour la première fois sur un projet concret et des corrections ont été apportées. Des tests unitaires ont alors été mis en place afin d’éviter qu’il y ait une quelconque régression dans un futur plus ou moins éloigné.

# Travail technique

## Améliorations

Des améliorations et configurations ont été apportées à Code Rewinder, même si celles-ci n’étaient pas toutes importantes :

1 - Code Rewinder utilise une librairie Java appelée Reflections qui permet entre-autres d’accéder et de modifier la valeur des attributs static des classes (et donc, de changer la version d’une méthode). Cependant, la dépendance n’était pas présente dans le POM Maven, elle a été ajoutée.

2 – La plupart des utilisateurs dispose d’un OS Windows. Malheureusement, le batch permettant de télécharger les n dernières versions d’un projet github était conçu pour Linux. Un batch Windows a donc été mis en place afin de satisfaire un maximum d’utilisateurs.

3 – Le réparateur était divisé en deux parties, une partie concernant le téléchargement des versions et la transformation, et une autre partie permettant d’exécuter les tests unitaires et donc, de réparer. Les deux parties ont donc été liées afin de simplifier l’exécution de Code Rewinder.  
  
4 – Lors du chargement des classes avec Reflections, le nom du package était autrefois écrit en dur dans l’application. Un argument -packages a donc été ajouté et celui-ci permet de lister les packages du projet à réparer dans Code Rewinder (chaque package doit être séparé par un point-virgule, comme le classpath). Il est même possible même de préciser le nom de la classe avec le package et, indiquer le nom du package en entier n’est pas obligatoire (si le package des classes contient un de vos mots alors la classe est prise en compte, qu’il s’agisse d’une classe de test ou d’une classe du projet).

## ANALYSE ET CORRECTIONS

### Exécution du projet sur jsoup

Code Rewinder a été pour la première fois exécuté sur une projet concret, plus précisément, Jsoup.