

MASTER MIAGE 2ÈME ANNÉE
UNIVERSITÉ PARIS NANTERRE

MÉMOIRE DE FIN D'ÉTUDES PRÉSENTÉ POUR
L'OBTENTION DU GRADE DE MASTER

**Comment les flots de contrôle
peuvent-ils nous permettre de faire du
refactoring de code en Java.**



PRÉSENTÉ PAR THIBAUT
SARTRE

Tuteur :
...

Septembre 2018 — Juillet 2019

Sommaire

1	Introduction	5
1.1	Présentation	5

Chapitre 1

Introduction

1.1 Présentation

Le refactoring est une activité d'ingénierie logiciel consistant à modifier le code source d'une application de manière à améliorer sa qualité sans altérer son comportement vis-à-vis des utilisateurs. L'objectif du refactoring est de réduire les coûts de maintenance et de pérenniser les investissements tout au long du cycle de vie du logiciel en se concentrant sur la maintenabilité et l'évolutivité.[3]

Le refactoring permet donc de passer d'un code possédant de mauvaise base à un code propre.

"With refactoring you can take a bad design, chaos even, and rework it into well-designed code." [2] Un bon refactoring doit pouvoir améliorer la qualité d'un code tout en gardant son fonctionnement du point de vue de l'utilisateur. Concernant la partie des tests, tout les tests qui fonctionnaient avant le refactoring se doivent d'être fonctionnels après.

Dans ce mémoire, nous allons analyser différentes techniques de refactoring. Puis nous allons étudier le principe des flots de contrôle.

Enfin je vous proposerais et évaluerais une solution pour faire du refactoring de code en utilisant les flots de contrôle pour le langage Java.[1][4]

Bibliographie

- [1] Jean-Michel Doudoux. Le refactoring, Janvier 2007.
<https://www.jmdoudoux.fr/java/dejae/chap009.htm>.
- [2] Martin Fowler. *Refactoring : Improving the Design of Existing Code*. Addison-Wesley Professional, 1999.
- [3] Jean-Philippe Retail . *Refactoring des applications JAVA/J2EE*. Eyrolles, 2005.
- [4] Alexander Shvets. Refactoring techniques. <https://sourcema-king.com/refactoring/refactorings>.