

# Opérations de refactoring sous eclipse

## *Travaux pratiques évalués*

Dans ces travaux, vous constituerez des programmes nécessitant des *refactorings*, puis vous leur appliquerez des opérations de refactoring d'eclipse dont vous ferez une analyse.

Les refactorings d'eclipse sont documentés, par exemple ici (mais cherchez la documentation pour votre version d'eclipse si besoin) :

<https://help.eclipse.org/neon/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2Fconcepts%2Fconcept-refactoring.htm>

Vous pouvez utiliser IntelliJ, je n'ai pas testé, mais vous devriez trouver plus ou moins les mêmes fonctionnalités.

Le travail sera réalisé par binôme (*personne1*, *personne2*). Mettez vos deux noms dans tous les fichiers produits et vous inscrivez vos binômes sur la feuille à l'url : <https://docs.google.com/spreadsheets/d/1-KUfKRmfiUhCBefNpqZVquxY6MDEPd-41Ji63V97Vjo/edit#gid=0>

Format pour rendre le travail du binôme : Sur le Moodle de l'université avant le **5 octobre 2020, 23h**.

## 1 Création de programmes nécessitant un refactoring

Lors du TD/TP du 29 septembre, deux refactorings vous sont affectés (un pour la *personne1*, un pour la *personne2* du binôme). Pour cette étape, *personne1* doit inventer un programme pour le refactoring qui lui est affecté et sur lequel il sera pertinent d'appliquer chacun de ces refactorings. Vous aurez donc **deux** programmes à réaliser par binôme, des programmes qui ne sont pas très bien faits. S'il y a plusieurs classes mettez-les ensemble dans le même fichier, une seule étant publique ou bien mettez-les dans un paquetage Java. Chaque programme doit contenir une fonction `main` qui teste le code assez exhaustivement.

**Résultat attendu : Pour chacun des deux refactorings, un programme Java nécessitant son application et sa fonction de test `main`; les programmes doivent fonctionner correctement; ils doivent également contenir un commentaire expliquant le refactoring et ce que vous en attendez (quel problème pertinent vous espérez qu'il va résoudre).**

## 2 Application de refactorings sur les programmes de l'autre personne (du binôme)

*personne1* va appliquer le (ou les) refactoring(s) attendu(s) sur le programme de *personne2*, et inversement.

**Résultat attendu : Pour chacun des deux programmes à refactorer, le programme Java obtenu après application de l'opération de refactoring identifiée sous eclipse et qui fonctionne toujours; Un document expliquant (1) le refactoring réalisé, (2) la procédure appliquée sous eclipse (vous ferez des copies d'écran des différentes étapes, ce sera le plus simple), (3) quel intérêt vous lui voyez, (4) si vous trouvez que la mise en oeuvre sous eclipse est bien réalisée ou non, et pourquoi (5) si le test s'est bien déroulé et si vous avez eu besoin de modifier le `main`.**

## 3 Comparaison de 2 catalogues de refactorings

**Résultat attendu : Trouvez un refactoring proposé dans <https://refactoring.com/catalog/> que vous ne trouvez pas dans eclipse, et inversement un refactoring d'eclipse que vous ne trouvez pas dans <https://refactoring.com/catalog/>. Expliquez-les.**