

OBJECTIF

Dans ce TP, vous allez tester le logiciel « Expenses Manager », créé par Katya Kratnyk, dont le code vous pouvez télécharger en clonant le dépôt git à <https://github.com/mfamelis/ExpensesManager>. Votre bût est de créer de tests pour la méthode statique `ua.kratnyk.impl.CurrencyConvertor.convert()`.

Testez ce qui fait du sens à tester et expliquez ce qui, à votre avis, n'a pas de sens. Énoncez clairement des hypothèses, si nécessaire.

1. TESTS BOITE NOIRE (30%)

Supposez que la **spécification** du « Currency Convertor » exige qu'il doit :

- Convertir des montants uniquement entre les devises suivantes : USD, CAD, GBP, EUR, CHF, INR, AUD.
- Seulement accepter des montants entre `[0, 10000]`.

Encodez cette spécification en créant des tests pour la méthode `convert()`. Utilisez comme nécessaire les approches de partition du domaine des entrées en classes d'équivalence et d'analyse des valeurs frontières.

2. TESTS BOITE BLANC (40%)

Testez la méthode `convert()` en utilisant les 5 critères de sélection de jeux de test *quand il fait du sens* : couverture des instructions, couverture des arcs du graphe de flot de contrôle, couverture des chemins indépendants du graphe de flot de contrôle, couverture des conditions, couverture des i-chemins.

3. RAPPORT (30%)

Documentez et expliquez comment vous avez choisi les cas de tests que vous avez implémenté. Décrivez les résultats de vos tests et donnez vos observations sur les différentes approches et critères de test.

(Voir la prochaine page pour les conditions de réalisation.)

CONDITIONS DE RÉALISATION

- Travail à remettre le **vendredi 16 décembre 23h59** via StudiUM. Aucun retard ne sera accepté.
- Le livrable le plus important est le **rapport**, en format PDF **et la classe TestCurrencyConvertor.java**. Assurez-vous de communiquer clairement en français, la qualité de votre rapport est très importante. Un mauvais rapport pourrait causer une déduction très significative.
- Travaillez en groupes de 2. **Aucune soumission individuelle ne sera acceptée.** Soumettez un ZIP nommé comme suit : *prenom1_nom1_prenom2_nom2.zip*
- Le membre de l'équipe qui ne soumet pas le fichier Zip principal de la soumission doit soumettre un zip (même nom) avec le fichier readme.txt, qui devrait contenir les noms de l'équipe.
- Les rapports doivent compter au maximum 4 pages, y compris toutes les figures et références. Rapports de plus de 4 pages vont être éliminés d'office (par défaut).
- Sur votre git, vous devez mettre vos tests dans la classe :
ExpensesManager\src\test\java\ua\karatnyk\TestCurrencyConvertor.java
(vous devez créer la classe)
- Vous pouvez utiliser votre IDE préféré. Cependant, nous n'offrons pas de conseils sur l'utilisation des IDE.
- Vous devez utiliser JUnit5 ou plus récent.
- Votre fichier ZIP doit aussi contenir un **readme.txt** avec liens vers tous ressources nécessaires (y compris votre repository git). Comme en TP1 et TP2, vous devez utiliser un **repository git** pour stocker votre code. Vous pouvez utiliser n'importe quel service gratuit comme Github, Bitbucket, et autres (quelques-uns vous permettent de créer des comptes académiques avec votre courriel @umontreal.ca). Utilisez le repository pour collaborer avec votre coéquipier. Vous devez mettre dans votre repository git tout script, ou fichier de configuration que vous avez utilisé. En fait, votre repository doit contenir assez des détails pour nous convaincre que vous avez vraiment effectué le travail vous-même.
- Nous allons examiner l'historique de votre repository pour nous assurer que tous les deux coéquipiers ont travaillé sur le TP et que votre code n'est pas plagié. Un historique de commit plausible devrait contenir de nombreux petits commit, chacun avec un message approprié de commit. **Faire juste quelques commit massives proche à la date limite pourrait entraîner une déduction considérable.**