Logo, company name

Description automatically generated

**IFT3913: TP4**

Matei Hulubas Barnia, 20161780

Thomas Duperron, 20225412

**Tests boîte noire :**

Dans les tests boîte noire, nous avons effectué deux types de tests. Premièrement nous avons réalisé des tests sur les montants et ensuite nous avons testé les devises. Lors de la rédaction de ces tests, nous avons décidé d’utiliser l’approche du try-catch.

Comme dit dans le cours, nous avons choisi une valeur de chaque classe d’équivalence pour faire les tests boîte noire.

Nous avons d’abord testé si les valeurs hors de l’intervalle de [0, 10000] donnaient des erreurs (ce qui devrait être le cas). Par la suite nous avons testé le contraire, donc si les valeurs comprises dans l’intervalle de [0, 10000] (inclusivement) ne provoquent pas d’erreur. Pour ce qui est de ces tests, pas tous ont passé. En effet, les tests qui vérifient si la fonction plante avec des valeurs inférieures à 0 ou supérieures à 10000 ne sont pas passées. Cela signifie que la fonction *“CurrencyConvertor.convert()”* ne vérifie pas que le montant à convertir est dans l’intervalle et ne lance pas d’exceptions pour les valeurs en dehors de l’intervalle acceptée. La seule fonctionnalité qui est présente est d’accepter des conversions dans l’intervalle [0,10000].

Par la suite, nous avons testé les conversions. Nous avons testé si les devises de l’énoncé sont acceptées et que la conversion se fait correctement. De plus, nous avons rajouté quelques tests pour s’assurer qu’aucune autre devise ne soit acceptée et nous avons aussi testé sur des devises qui ne sont pas dans le fichier *JSON* ou devise vide (""). Pour ce qui est de ces tests, seulement quelques-uns ont planté. Les tests qui n’ont pas passé sont les tests qui vérifient que seules les devises acceptées sont traitées. En effet, lorsque nous avons essayé de convertir des *USD* en *MXN* ou des *BOB* en *USD*, la fonction les traite quand même et ne lance pas d’exceptions. Ceci est causé par le fait que la fonctionnalité n’a pas été implémentée et la fonction ne fait que vérifier que les devises données en paramètres sont présentes dans le fichier *JSON* seulement. Les seules fonctionnalités qui sont présentes sont donc le fait de convertir les devises correctement et de ne pas accepter des devises non-existantes.

**Tests boîte blanche :**

Nous avons commencé par implémenter la spécification de la couverture des instructions. Pour couvrir toutes les instructions dans la fonction *“CurrencyConvertor.convert()”* au moins une fois, nous n’avions besoin que de faire 2 tests. Ceci est causé par le fait qu’il n’y a qu’un seul *IF* et donc il n’y a que deux chemins possibles dans cette fonction. Le premier test passe par le *ELSE* (retourne *FALSE* pour le *IF statement*) et le second test rentre dans le *IF* (retourne *TRUE* pour le *IF statement*). Tous les deux tests ont passé. Cela nous permet de conclure que la fonction se comporte comme attendu. Donc, si une devise qui n’existe pas dans le *JSON* est donnée en paramètres, la fonction ne continue pas et lance une exception.

Pour ce qui est de la couverture des arcs du graphe de flot de contrôle et de la couverture des chemins indépendants du graphe de flot de contrôle, nous n’avions pas besoin de rajouter quoi que ce soit puisque la couverture des instructions couvre déjà le tout. Ceci est causé par le fait que nous n’avons qu’un simple *IF statement* et cela simplifie grandement la tâche.

Par la suite, nous avons implémenté le critère de couverture des conditions (et des arcs). Pour ce faire, nous avons implémenté quelques tests de plus de façon à couvrir toutes les clauses élémentaires comprises dans le *IF statement*. Nous avons donc testé la fonction avec sa table de vérité pour passer par toutes les variations possibles. Tous les tests sont passés et donc nous pouvons conclure que le code fonctionne et se comporte comme attendu.

Pour ce qui est de la couverture des i-chemins, nous n’avions pas besoin d’implémenter quoi que ce soit puisque la fonction ne contient pas de boucles.