**Rapport de TP4 IFT3395**

Wen Yin (20179082) Jimmy Yassin Hassanaly (20190749)

***Automne 2023***

**Tâche 3 :**

Voici l’explication des choix des tests pour les tests boîtes blancs des 2 méthodes :

**MainWindow.convert() :**

**testNormalConversion (Couverture des Instructions) :**

Ce test est fait pour exécuter le scénario le plus normal de conversion de devise. Il s'assure que le chemin principal de la méthode convert est exécuté, ce qui inclut le calcul de la conversion de devise. C'est essentiel pour s'assurer que les instructions de base du code fonctionnent comme prévu.

***Resultat : Succès.***

**testUnsupportedCurrencyConversion (Couverture des Arcs) :**

En utilisant des devises inexistantes ("XXX" et "YYY"), ce test vise à parcourir une branche spécifique dans le code si elle existe qui gère les devises non reconnues. Ce test est essentiel pour vérifier que le code se comporte correctement quand il rencontre des devises qu'il ne peut pas traiter, couvrant ainsi les transitions de contrôle dans le code pour les entrées inattendues.

***Résultat : Echec*** car retourne 0 mais devrait retourner -1 ou un message d’erreur.

**testNegativeAmountConversion (Couverture des Chemins) :**

Ce test vérifie le comportement du code lorsque le montant de conversion est négatif, ce qui devrait normalement être traité comme une entrée invalide. Il s'assure que la méthode « convert» gère correctement ce cas particulier, couvrant un chemin d'exécution distinct qui pourrait impliquer une logique de validation ou de gestion des erreurs.

***Résultat : Echec*** car retourne une valeur négative calculée au lieu de retourner -1 ou message d’erreur.

**testZeroAmountConversion (Couverture des Conditions) :**

Ce test évalue comment le code réagit à une conversion avec un montant de zéro. Cela permet de tester une condition importante (le montant étant exactement zéro), ce qui est important pour s'assurer que les cas limites sont correctement gérés dans le code.

***Résultat : Succès*** car retourne 0.

**testAmountAboveLimitConversion (Couverture des i-Chemins) :**

Ce test explore un scénario où le montant est juste au-dessus de la limite autorisée. Il teste les interactions entre les conditions de limite de montant et de validité de la devise, couvrant des cas où différentes branches du code interagissent.

***Résultat : Echec*** car retourne le montant calculé alors que c’est au dessus de la limite autorisé au lieu de retourner -1 ou message d’erreur.

**testUnsupportedCurrenciesAndInvalidAmount (Couverture des i-Chemins) :**

En fournissant à la fois des devises non supportées et un montant invalide, ce test évalue comment le code gère la combinaison de plusieurs conditions d'erreur. Cela aide à couvrir des chemins dans le code où différentes branches de logique se croisent, et ainsi pour mieux identifier l’erreur.

***Résultat : Echec*** car retourne 0 au lieu de retourner -1 ou message d’erreur.

**testAmountAtUpperLimit (Couverture des Conditions) :**

Ce test vérifie le comportement du code pour un montant qui est exactement à la limite supérieure acceptée. Cela permet de tester une condition limite spécifique, s'assurant que la méthode « convert » traite correctement les montants qui sont à la frontière de la validité.

***Résultat : Succès*** car retourne la valeur attendu.

**Convert.convert() :**

**testNormalConversion (Couverture des Instructions) :**

Teste une conversion de devise normal avec des entrées valides. Cela garantit que les instructions de base pour calculer la conversion sont correctement exécutées, couvrant ainsi le chemin d'exécution principal de la méthode.

***Résultat : Succès*** car retourne la valeur attendue.

**testNegativeAmount (Couverture des Arcs) :**

En testant avec un montant négatif, ce test cible une section spécifique du code qui doit gérer les montants négatifs comme étant invalides. Il s'assure que le code réagit correctement à ces entrées, en testant les transitions de contrôle pour les cas d'erreur.

***Résultat : Echec*** car retourne un résultat négatif calculé au lieu de retourner -1 ou message d’erreur.

**testZeroExchangeRate (Couverture des Chemins) :**

Ce test examine un scénario spécifique où le taux de change est zéro. Cela permet de voir la réaction du code à un taux de change nul.

Résultat : Succès.

**testNegativeExchangeRate (Couverture des Conditions) :**

Teste la réaction du code à un taux de change négatif. Cela teste une condition spécifique dans le code, s'assurant que les valeurs de taux de change invalides sont correctement gérées.

***Résultat : Echec*** car retourne un résultat négatif calculé au lieu de retourner -1 ou message d’erreur.

**testZeroAmountNegativeExchangeRate (Couverture des i-Chemins) :**

Teste la conversion avec un montant nul et un taux de change négatif. Ce test couvre les interactions entre deux conditions indépendantes, ce qui est important pour vérifier le comportement du code dans des scénarios où plusieurs conditions sont impliquées.

***Résultat : Echec*** car retourne 0 calculé au lieu de retourner -1 ou message d’erreur.