**IFT3913 – TP4 RAPPORT**

**Ithri Kendi (20222832)**

**Walid Boudehane(20206664)**

**1. TESTS BOITE NOIRE**

Nous avons que la spécification du « Currency Converter » exige que:

• Il doit convertir des montants entre les devises suivantes : USD, CAD, GBP, EUR, CHF, AUD.

• Il doit seulement accepter des montants entre [0, 1 000 000].

A partir de la documentation nous avons décidé que notre jeu de test valide serait comme suit :  
 • Pour la partition du domaine des entrées en classes d’équivalence : T = {-100, 100, 2000000}

• Pour l’analyse des valeurs frontières: T = {-1,0, 1000000, 1000001}

• Pour les devises T = {(USD,EUR) , (USD,CAD), (USD,GBP), (USD,CHF), (USD,AUD), (USD,XYZ), (XYZ,USD), (USD,EUR), (CAD,USDD), (CAD,)}

Pour la déclaration de la Arraylist<Currency> currencies qu’on doit utiliser pour la méthode MainWindow.convert, nous avons utilisé la methode Currency.init() qui est présente dans le code.

* 1. **Currency.convert**

Pour la méthode Currency.convert, comme les devises ne sont pas utilisées nous avons effectués uniquement sur le jeu de test sur les montants avec un exchange value fixe de 1.5 .

Pour les valeurs à l’intérieur de l’intervalle soit {0, 100, 1000000} les tests sont passés sans problèmes ce qui montre que la méthode fait les conversions correctement.   
Mais pour les valeurs à l’extérieur de l’intervalle soit {-100, -1, 1000001} les tests ne passent pas car par rapport à la spécification la méthode la méthode devrait retourner 0 pour ces valeurs la mais la conversion se fait alors qu’elle ne devrait pas se faire. Notre hypothèse pour ce comportement serait sur le fait qu’il n’y a probablement pas de contraintes par rapport au domaine de la spécification.

* 1. **MainWindow.convert**

Pour la méthode MainWindow.convert, Nous avons utilisés tous les jeux de tests que nous avons montré au début en plusieurs combinaisons. Nous avons utilisé (USD, EUR) comme devises valides avec les différents montants (valides ou invalides) afin de tester en premier lieu les montants. De l’autre côté, nous avons utilisé le montant 100 comme montant valide avec les différentes devises (valides ou invalides) afin de tester les devises. Pour les tests avec les valeurs valides devrait faire la conversion sans problèmes et ceux avec les valeurs invalides devrait retourner une valeur de 0 lors de la conversions comme ces valeurs-là ne font pas partie de la spécification.

Lors de résultats des tests nous avons remarqué un comportement des tests non-attendu. Les tests avec les valeurs valides ne passaient pas sauf avec la valeur 0 mais les tests avec les valeurs invalides passaient sans problèmes. Après une investigation sur les résultats des tests, nous avons remarqué que tous les tests retournaient la valeur 0 peu importe la valeur de l’entrée qu’elle soit valide ou pas.

Après avoir investigué le code des méthodes nous faisons l’hypothèse que ce comportement est relié au format des devises, nous pensons que la méthode utilise les devises sous la forme longue (US Dollar) au lieu de la forme courte (USD) or que dans la spécification nous avons uniquement accès a la forme courte. Donc toutes les entrées de devises sont considérée invalides car elles ne sont pas dans le bon format.