Alexis Boucher 20217120
Marguerite Mireille Camara 20143122

Lien GitHub:
https://github.com/AlexisBoucher/IFT3913

1- Boite Noire

A- Partition du domaine des entrées en classes d'équivalence:

Partitionnement sur la spécification il doit accepter des montants dans l'intervalle suivants [0, 1 000 000]:

Classes d'équivalence:

- Montant < 0 ;
- Montant = 250 000 ;
- Montant > 1 000 000;

Jeu de test : T= {-10,250000,1 000 060}

Partitionnement sur la spécification le Converter doit accepter des montants entre les devises suivantes :

USD, CAD, GBP, EUR, CHF, AUD:

Classes d'équivalence:

- Devise différente de [USD, CAD, GBP, EUR, CHF, AUD] ;
- Devise faisant partir de [USD, CAD, GBP, EUR, CHF, AUD] ;

Jeu de test : T= {USD, GNF}

Hypothèses:

Pour les montants en dehors de l'intervalle [0, 1 000 000] et pour des devises différentes de celles spécifiées, le code peut lever une exception ou en renvoyer un résultat non défini. Il peut aussi spécifier à l'utilisateur par des logs que les entrées ne respectent pas la spécification.

B- Analyse des valeurs frontières

Spécification il doit accepter des montants dans l'intervalle suivants [0, 1 000 000]

Jeu de test :

T= {(typique:10, frontière:1), (typique:250000, frontiere:500000), (typique:1 000 060, frontiere:1000001)}

Pour la méthode : **currencyConverter.Currency. convert (Double, Double)**

Nous effectuons des vérifications sur des taux de change positif et négatif, sur des montants égaux à 0, des taux de change de zéro et sur des valeurs limites et extrêmes.

Nous pouvons remarquer en analysant en boite noire que cette méthode n'exclut pas les éléments qui ne font pas partir de la spécification et elle ne fait pas vraiment de gestion d'erreur.

2- Boite blanche

A- Test currencyConverter.Currency.convert(Double, Double)

Couvrir cette méthode était plutôt simple, en utilisant le critère de couverture des instructions un seul test est nécessaire puisque peu importe les entrées de la fonction toutes les instructions sont exécutées. Pour des raisons similaires, les autres critères de sélection de jeux de tests ne demandent qu'eux aussi qu'un test pour couvrir l'ensemble du jeu de tests. Le test currency.convert(30.00, 0.80) = 24 est suffisant.

B- Test currencyConverter.MainWindow.convert(String, String, ArrayList, Double)

Pour couvrir cette méthode en utilisant le critère de couverture des instructions nous trouvons 3 tests pour exécute l'ensemble des instructions. (US Dollar, Euro, currencies, 300.00), qui test une utilisation normale du code, (US Dollar, Not a Money, currencies, 300.00), qui finit la première boucle, mais ne trouve jamais la deuxième devise et (Not a Money, Euro, currencies, 300.00), qui trouve la deuxième devise, mais lors de la deuxième boucle ne trouve pas la première devise. En utilisant les autres critères de sélection de jeux de tests, nous trouvons aussi que ces mêmes tests suffisent à couvrir l'ensemble de la méthode.

3- Conclusion

En exécutant nos tests, nous avons observé que nous ne trouvons aucune erreur dans le code avec les tests de la boite blanche par contre certains tests de la boite noire nous trouve des erreurs. En observant le code nous avons trouvé que certaines monnaie trouvé dans la spécification du currency converter ne sont pas listé dans la classe Currency.java (CAD et AUD) et que les taux d'échange des devises sont pour la plupart obsolète dans le programme. Les méthodes testées ne sont donc pas fautives et il suffit de mettre à jour l'information contenue dans Currency.java pour corrigé le bug.