

RAPPORT TP4

Zeinab Taleb et Dana Seif Eddine

Tâche 1 - Boite Noire

Spécifications:

- Convertir des montants entre les devises suivantes : USD, CAD, GBP, EUR, CHF, AUD
- Il doit seulement accepter des montants entre [0, 1 000 000]

Sélection des jeux de test T

- Devise
D1 = {USD, CAD, GBP, EUR, CHF, AUD}
D2 = {d | d <> D1}
T = {CAD, AED}
- Montant
D1 = {d | $0 \leq d \leq 1\,000\,000$ }
D2 = {d | d < 0}
D3 = {d | d > 1 000 000}
T = {-1 500 000, -1, 0, 2 500, 1 000 000, 1 000 001, 1 500 000}
(analyse des valeurs frontières)

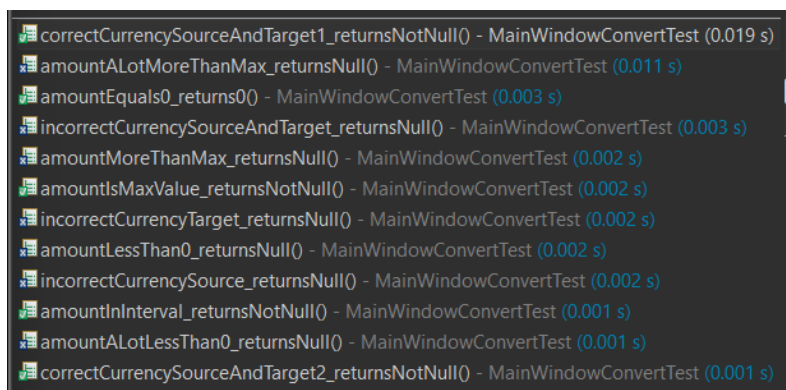
Hypothèses :

- des arguments qui ne sont pas corrects vont retourner Null.
- les arguments des devises sont des abréviations comme dans la spécification.

Notes :

On ne va pas vérifier que les valeurs de conversion retournées sont correctes puisque nous effectuons les tests sans prendre connaissance de la structure et l'implémentation du code.

Donc nous vérifions simplement si le retour est Null ou pas.



Tests des devises (le montant fait partit de l'intervalle)

- **incorrectCurrencySource_returnsNull() : fail**
retourne 0 au lieu de Null, même si la devise AED n'est pas dans la spécification
- **incorrectCurrencyTarget_returnsNull() : fail**
retourne 0 au lieu de Null, même si la devise JPY n'est pas dans la spécification

- **incorrectCurrencySourceAndTarget_returnsNull() : fail**
retourne 0 au lieu de Null, même si les devises CNY et KRW ne sont pas dans la spécification
- **correctCurrencySourceAndTarget1_returnsNotNull() : pass**
retourne bien une valeur puisque les devises EUR et GBP sont acceptées
- **correctCurrencySourceAndTarget2_returnsNotNull() : pass**
retourne bien une valeur puisque les devises CAD et GBP sont acceptées

Tests des montants (les devises sont acceptées)

- **amountALotLessThan0_returnsNull() : fail**
retourne 0 alors que -1500000.0 ne fait pas partit de l'intervalle et ne devrait pas retourner de valeur
- **amountLessThan0_returnsNull() : fail**
retourne 0 alors que -1.0 ne fait pas partit de l'intervalle et ne devrait pas retourner de valeur
- **amountEquals0_returns0() : pass**
retourne bien 0 quand le montant est 0
- **amountInInterval_returnsNotNull() : pass**
retourne une valeur quand le montant 2500.0 est dans l'intervalle
- **amountIsMaxValue_returnsNotNull() : pass**
retourne une valeur quand le montant 1000000.0 est dans l'intervalle
- **amountMoreThanMax_returnsNull() : fail**
retourne 0 alors que 1000001.0 ne fait pas partit de l'intervalle et ne devrait pas retourner de valeur
- **amountALotMoreThanMax_returnsNull() : fail**
retourne 0 alors que 1500000.0 ne fait pas partit de l'intervalle et ne devrait pas retourner de valeur

Tâche 2 - Boite Blanche

Utilisons le critère de couverture des arcs du graphe de flot de contrôle.

checkDevisesExist :

Critère de Test : Couverture des instructions et des conditions.

Raison : Vérification de la présence des devises dans la liste.

Cas de Tests : Utilisation de devises existantes et inexistantes pour couvrir différentes conditions.

Tests de Conversion entre Devises Validées :

Critère de Test : Couverture des instructions et gestion des exceptions.

Raison : Vérification de la précision des conversions entre devises validées.

Cas de Tests : Conversions entre différentes devises avec des montants variés.

Tests de Conversion avec Montants Invalides :

Critère de Test : Couverture des instructions et gestion des exceptions.

Raison : Vérification de la réaction du logiciel face à des montants invalides.

Cas de Tests : Utilisation de montants nuls, négatifs, et très élevés pour tester les limites.

Tests de Conversion de Devises Inexistantes :

Critère de Test : Couverture des i-chemins.

Raison : Vérification du comportement du programme lorsqu'on tente de convertir depuis ou vers des devises inexistantes.

Cas de Tests : Tentatives de conversion avec des devises non présentes dans la liste des devises.

Tests de Conversion entre Devises Validées :

Critère de Test : Couverture des i-chemins.

Raison : Validation des différents chemins d'exécution dans la méthode convert.

Cas de Tests : Conversions entre différentes devises avec des montants variés.

Test d'égalité de deux méthodes de convert

Critère de Test : Couverture des arcs du graphe de flot de contrôle.

Raison : les deux méthode convert retourne les mêmes résultats

Cas de Tests : l'appel de n'importe quelle méthode affiche le même résultat

Conclusion :

On remarque donc après avoir effectué tout ces tests qu'aucune des fonctions de convert ne vérifient que le montant entré est valide.

A cause de cela les fonctions nous retourne soit un montant negatif soit une valeur malgré le fait que le montant excède le maximum.

Sinon si les devises et le montant sont corrects, la valeur retournée correspond bien a celle attendue.

De plus, la fonction convert de la classe MainWindow vérifie bien que les devises entrées sont valides. Cependant, les devises valides définies ne sont pas bonnes, il en manque d'après la spécification (CAD et AUD). De plus, si une devise n'est pas trouvée au lieu de retourner une erreur c'est un montant de 0 qui est retourné.

Il faudrait donc veiller a ce que le programme lance une erreur lorsqu'une devise n'est pas acceptée, faire attention a ce que toutes les devises de la spécification soient dans la liste des devises, et verifier le montant entré puis aussi retourner une erreur s'il n'est pas dans l'intervalle.