IFT 3913 - Qualité du logiciel et métriques
Rapport du TP3

Présenté à

Prof. Michalis Famelis

Par

Geneviève Paul-Hus (20037331)

et

Jean-Claude Desrosiers (20150403)

Pour le

8 Avril 2022

Mise en Contexte

Nous devons tester :

1. `currencyConverter.MainWindow.convert(String, String, ArrayList<Currency>, Double)`
2. `currencyConverter.Currency.convert(Double, Double)`

Avec les spécifications suivantes :

- DevisesValides = { "USD" , "CAD" , "GBP" , "EUR" , "CHF" , "INR" , "AUD" }
- MontantsValides = [0 , 10’000]

Nous assumons que lorsqu’une fonction obtient un paramètre invalide, elle devrait lancer une exception (n’importe quelle sous-classe de Exception)

Tests boîte noire

Nous utilisons l’approche de partition du domaine des entrées.

Pour le domaine des listes de devises, nous avons choisi deux partitions : les listes des devises selon les spécifications et les listes incorrectes. Notre jeu de test est donc :

{ ["USD" , "CAD" , "GBP" , "EUR" , "CHF" , "INR" , "AUD"] } ∪
{ ["CAD" , "GBP" , "EUR" , "CHF" , "INR" , "AUD"] }

Pour le domaine des devises, nous avons choisi trois partitions : devises spécifiées, devises existantes hors spécifications et devises non-existantes. Notre jeu de test est donc :

{ "CAD" } ∪ { "JPY" } ∪ { "ABC" }

Pour le domaine des montants, nous avons choisi cinq partitions : montants trop petits, montants valides, montants trop grands et deux partitions pour les montants aux frontières. Notre jeu de test est donc :

{ -42 } ∪ { 42 } ∪ { 42’000 } ∪ { -0.01 , 0 } ∪ { 10’000 , 10’000.01 }

Les cas de tests sont construits de manière à tester plusieurs permutations des paramètres valides et invalides, mais seulement un paramètre invalide à la fois. Par exemple, on ne testera pas la conversion d’un montant invalide vers une devise invalide, seulement un montant valide vers une devise invalide, ou vice-versa. Ceci permet de réduire le nombre de test boîte noire (de ~800 à ~150), en faisant l’hypothèse que plusieurs paramètres invalides ne résultent pas en un résultat valide.

Première Fonction

Le résultat observé est un échec de 12 des 22 tests pour cette fonction. La plupart des échecs sont dus au manque de validation des entrées qui ne lancent pas d'exception, mais qui devraient. Aussi, certaines erreurs semblent se produire par rapport à la devise "CAD" qui devrait fonctionner, mais donne plutôt une erreur.

Seconde Fonction

Pour tester la seconde fonction, nous supposons que la spécification est stricte. Ainsi, on ne devrait pouvoir convertir un montant avec un taux de change qui n'existe pour aucune paire de devise supporté (e.g. on ne devrait pas pouvoir utiliser un taux de change de 0.008 car c'est pour le JPY -> USD).

Pour déterminer les valeurs de `exchangeValue` à tester, nous avons pris chaque paire de devise dans nos cas de test (excluant "ABC", car ce n'est pas une devise) et avons calculé le taux de change pour la conversion entre ces deux devises.

Le résultat observé est un échec de 10 des 13 tests pour cette fonction. Tous ces échecs sont des appels de fonction avec des paramètres qui sont invalides, mais un résultat qui n'est pas une exception. Il n'y a pas de validation des entrées assez strict par rapport aux spécifications.

Tests boîte blanche

Première Fonction

Étant donné que la première fonction est très courte et ne contient qu'un seul chemin d'exécution. Nous ne pensons pas qu'il vaille la peine de faire une test boîte blanche sur celle-ci.

Seconde Fonction

Nous avons choisi une approche de couverture des chemins linéairement indépendants.
La borne supérieure du nombre de chemins nécessaires est V(G) = 6 (voir Figure 1 pour le graphe de flux de contrôle). Les vecteurs suivent le format suivant :

(0-1 , 0-2 , 1-0 , 1-2 , 2-3 , 2-fin , 3-4 , 3-fin , 4-3 , 4-fin)

Et voici les chemins choisis :

1. (0, 1, 0, 0, 0, 1, 0, 0, 0, 0)
2. (2, 0, 0, 1, 1, 0, 1, 0, 0, 1)
3. (1, 0, 0, 1, 0, 0, 1, 1, 1, 0)
4. (1, 1, 1, 0, 1, 0, 2, 0, 1, 1)

Ceux-ci sont linéairement indépendants et leur somme donne au moins 1 dans chaque composante (donc on atteint tous les arcs).

Si on décrit les trois cas de tests en français, on a :

1. Conversion avec une liste de devise vide
2. Conversion entre deux devises valides, d'un montant valide, de sorte que la seconde devise est trouvée immédiatement dans la liste des devises.
3. Conversion entre deux devises valides, d'un montant valide. Mais la liste des devises ne contient que la seconde devise.
4. Conversion entre deux devises valides, d'un montant valide, de sorte que la première devise est trouvée immédiatement dans la liste des devises.

Le résultat observé est un échec des quatre tests pour cette fonction. Les erreurs semblent se produire par rapport à la devise "CAD" qui devrait fonctionner, mais donne plutôt une erreur. Aussi, il semble encore y avoir un problème de validation parce que les tests un et trois échouent. Ces derniers devraient lancer une exception, mais le montant 0 est retourné.

Annexe

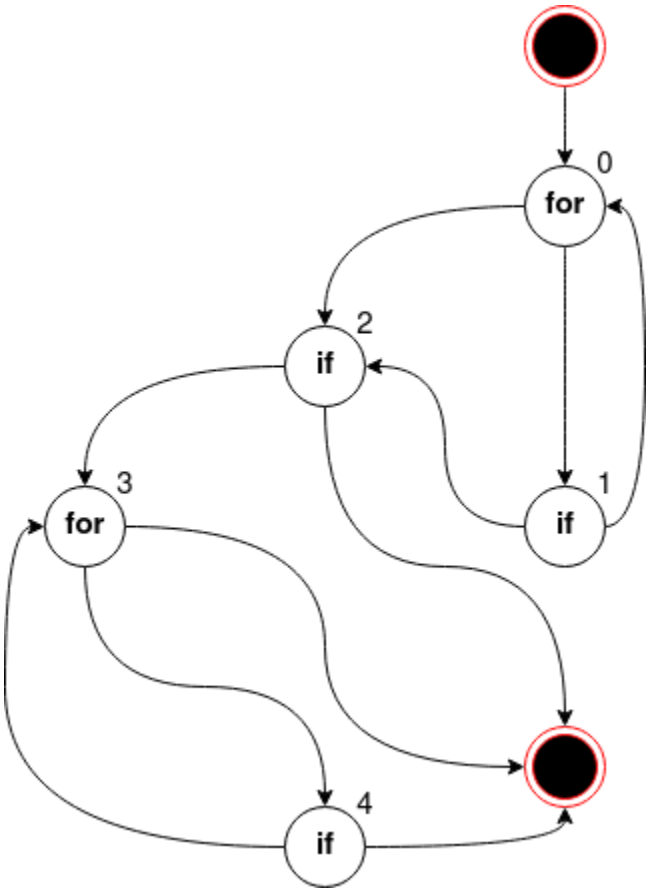

Figure 1 : Graphe de flux de contrôle de la première fonction