



Travail pratique 4

Test du logiciel « Currency Converter »

Par Tarik Benakezouh & Kha Pham

Matricules : 20184524 & 20182335

Travail présenté à Michelis Famelis

dans le cadre du cours de

IFT 3913 – Qualité de logiciel et métriques

Remis le 8 décembre 2023

Hypothèse pour les tests boîte noire

Tout d'abord, commençons par définir les classes d'équivalence responsables de la partition du domaine des entrées et de l'analyse des valeurs frontière associées à la spécification du « Currency Converter » qui fournira les entrées non valides nécessaires à l'élaboration de l'hypothèse.

Soit le domaine D étant les devises monétaires, une inspection rapide des interfaces concernées révèle la présence de 8 devises, soit USD, GBP, EUR, CHF, AUD, CAD, CNY et JPY. Nous tournons l'attention sur les 4 dernières où les dollars canadien et australien ne figurent pas dans la base de code alors que le yuan et yen ne sont pas inclus dans la spécification. Nous obtenons ainsi 3 classes d'équivalence à partir de la première approche des tests boîte noire :

- $D_1 = \{\text{USD, GBP, EURO, CHF}\}$, soit les devises monétaires présentes dans la spécification et le programme
- $D_2 = \{\text{CNY, JPY}\}$, soit les devises monétaires présentes dans le programme exclusivement
- $D_3 = \{\text{CAD, AUD}\}$, soit les devises monétaires présentes dans la spécification exclusivement

Quant à l'analyse des valeurs frontière, concevoir les classes d'équivalence pour les montants s'avère fort simple :

- $D_4 : \{0 \leq d \leq 1\,000\,000\}$
- $D_5 : \{d < 0\}$
- $D_6 : \{d > 1\,000\,000\}$

Parmi ces 6 classes d'équivalence, 4 d'entre elles contiennent des entrées non valides pour lesquelles nous allons émettre une hypothèse individuelle concernant le comportement du code lors de leur soumission :

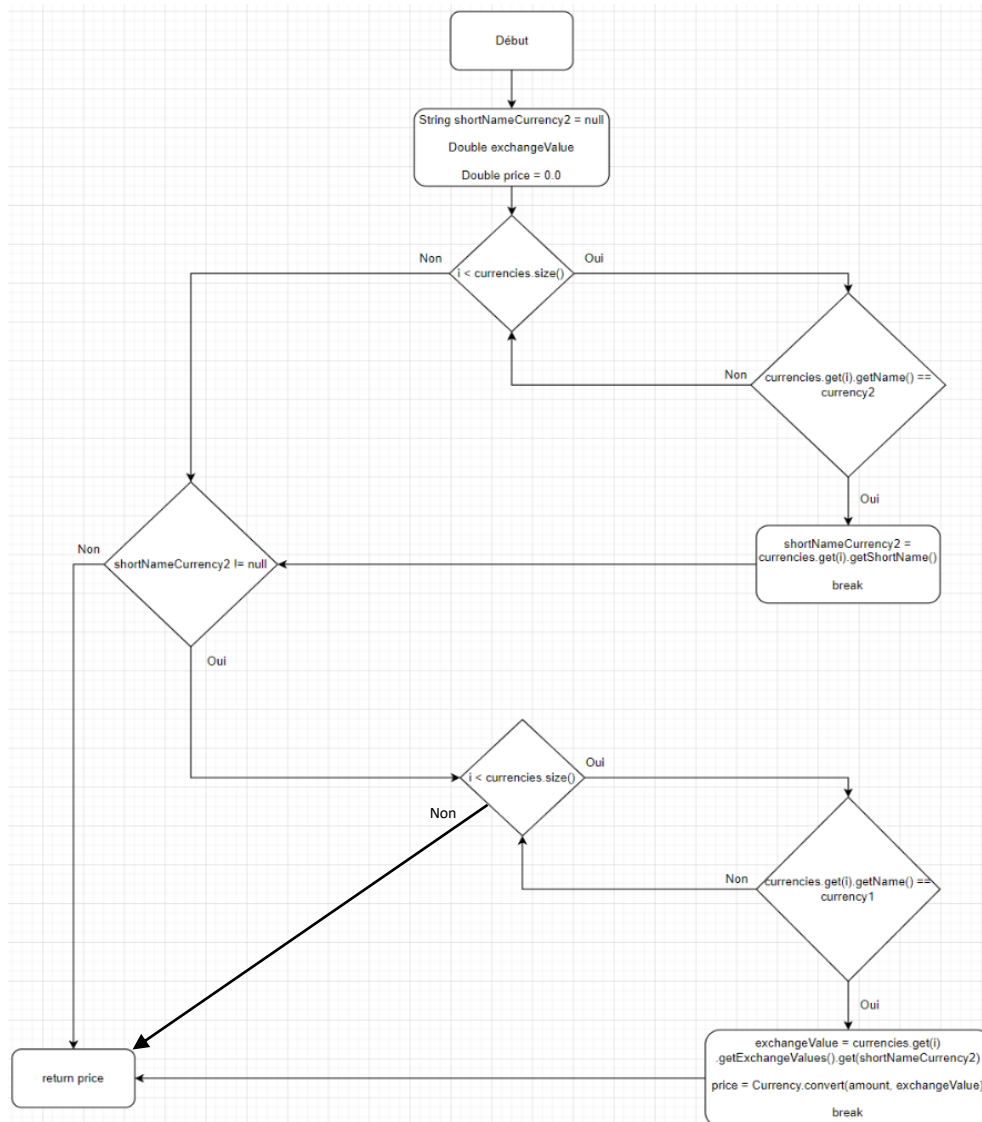
- D_2 : la spécification n'a pas impliqué que le convertisseur doit prendre en charge CNY et JPY, mais si des tests devaient être élaborés afin de vérifier leur fonctionnement, nous pensons que leur comportement serait identique aux devises de la classe D_1
- D_3 : l'implémentation des fonctions « convert » et « exchangeValues » au sein de la classe Currency nous porte à croire que le convertisseur retournera une valeur « null » vu l'inexistence de la prise en charge de CAD et AUD et donc retournera fort probablement un NullPointerException
- D_5 et D_6 : excéder les bornes des valeurs frontières établies produira un résultat égal à 0 suite à la gestion d'erreur en cas de valeur négative issue de la classe D_5 qui affectera les entrées de D_6 de la même manière

Documentation et explication des cas de tests

D'une part, les cas de tests boîte noire ont été conçus avec toutes les classes d'équivalence en tête. Ce qui est vraiment à justifier ici est le nombre de tests émis, soit 6 sur les 36 possibles (incluant les conversions aux mêmes devises d'entrée et de sortie) qui est suffisant à nos yeux de par la simplicité de la tâche exécutée par le programme. En effet, le « Currency Converter »

effectue la conversion redondante d'un montant où les variables étant les devises d'entrée et de sortie et les taux de conversion associés et couverts par les tests où l'euro figure comme devise de sortie arbitraire tout en impliquant l'ensemble de la partition du domaine des entrées en classes d'équivalence.

D'autre part, les cas de tests boîte blanche ont été élaborés à partir des critères de sélection de jeux de test suivants : les couvertures des instructions et des arcs du graphe de flot de contrôle (A et B). La justification de ce choix s'est faite à l'aide d'un processus d'élimination nous ayant initialement permis d'éliminer assez facilement les couvertures des conditions et des i-chemins (D et E) puisque la méthode « convert » de la classe « MainWindow » n'implémente aucune boucle détentrice de conditions composées et le contexte du jeu de test ne peut tout simplement pas faire appel à des entrées capables d'effectuer des opérations de contrôle d'itération sur les boucles simples vu l'absence de boucles imbriquées et concaténées au sein du programme. Quant à la couverture des chemins indépendants du graphe de flot de contrôle (C), cette omission est celle ayant été la plus débattue à cause de sa similarité trop importante par rapport à la couverture régulière des arcs du graphe de flot de contrôle (B).



Par contre, en consultant l'organigramme de la méthode « convert » de la classe MainWindow, nous avons réalisé que le flux du code était beaucoup trop simple pour tester des concepts poussés tels que les 1-chemins, chemins indépendants et même mesurer la complexité cyclomatique de cette fonction aux boucles for redondantes et identiques. Nous avons aussi pris la décision délibérée d'omettre des cas de tests boîte blanche pour CurrencyTest faute de structure de code complexe à inspecter telle que des boucles ou un « if statement ».

Résultats et observations

Les résultats de nos tests concordent avec nos hypothèses émises pour ceux issus de la boîte noire de sorte à retourner des succès sauf les tests ayant échoué comme prévu pour les devises CAD et AUD ainsi que les entrées dépassant les valeurs frontières. Les tests boîte blanche réussis confirment aussi les couvertures adéquates des instructions et arcs du graphe de flot de contrôle de la méthode « convert » du programme « Currency Converter ». Les résultats satisfaisants des tests boîte blanche ne nous donne pas d'opportunité de conclure une quelconque observation sur les approches et critères de test. Dans le cas des tests boîte noire, nous pensons qu'une plus grande emphase sur des tests de la valeur vide ou non de la variable « getExchangeValue » aurait été plus judicieux à inclure dans le jeu de test que des devises qui ne sont pas prises en charge (CAD, AUD) puisque cette variable a un impact direct sur le prix retourné par la méthode « convert » au sein des deux classes principales du programme.