Qualité du logiciel et des métriques

Devoir 3

Équipier 1 : Emily Bazile-Supplice (20200848)

Équipier 2 : Eed Flory Jean-Baptiste (20168335)

Tâche 1

**Objectif :** Concevoir et exécuter des tests boîte noire pour le système de conversion de devises (« Currency Converter ») selon les spécifications suivantes :

• 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].

Currency.convert(Double, Double)

* Test pour un cas de base

Ce test vise à confirmer que la fonction de conversion fonctionne correctement pour des valeurs attendues. Dans ce scénario, nous utilisons un taux de conversion positif ainsi qu'un montant initial positif.

Valeur initiale : 100$

Taux de conversion : 1.5

Valeur attendue : 150$

Valeur obtenue : 150$

* Test montant initial nul

L'objectif de ce test est de vérifier la conversion d'un montant de 0$, ce qui représente une limite inférieure de nos spécifications.

Valeur initiale : 0$

Taux de conversion : 1.5

Valeur attendue : 0$

Valeur obtenue : 0$

* Test montant initial 1 000 000$ .

L'objectif de ce test est de vérifier la conversion d'un montant de 1 000 000$, ce qui représente une limite supérieure de nos spécifications.

Valeur initiale : 1 000 000$

Taux de conversion : 1.5

Valeur attendue : 1 500 000$

Valeur obtenue : 1 500 000$

* Test pour un taux nul

L'objectif de ce test est de vérifier la conversion d'un certain montant valide avec un taux de conversion nul, ce qui représente une limite inférieure pour le taux de conversion.

Valeur initiale : 100$

Taux de conversion : 0

Valeur attendue : 0$

Valeur obtenue : 0$

* Test pour un montant Négatif

L'objectif de ce test est de vérifier la conversion d'un montant négatif avec un certain taux de conversion valide. Nous ne devrions pas être en mesure de convertir ce montant, car il est hors de nos spécifications.

Valeur initiale : -1$

Taux de conversion : 1.5

Badge croix avec un remplissage uniValeur attendue : ThrowException

Valeur obtenue : -1.5$

Le code ne prend pas en compte le cas où le montant entrée est négatif. Il devrait retourner un message d’erreur.

* Test pour un montant supérieur à la limite

Badge croix avec un remplissage uniCe test vise à confirmer que la fonction de conversion fonctionne correctement pour des valeurs attendues. Dans ce scénario, nous utilisons un taux de conversion positif ainsi qu'un montant initial positif.

Valeur initiale : 1 000 001$

Taux de conversion : 1.5

Valeur attendue : ThrowException

Valeur obtenue : 1 500 001.5$

Le code ne prend pas en compte le cas où le montant est supérieur à 1 millions . Il devrait retourner un message d’erreur.

* Test pour un taux de conversion négatif

Implémentation de tests pour s'assurer que la conversion est précise et fiable entre chaque paire de devises. Chaque cas de tests convertit le montant de 100$ dans une autre devise.

Valeur initiale : 100$

Taux de conversion : -1.5

Badge croix avec un remplissage uniValeur attendue : ThrowException

Valeur obtenue : -150$

Le code ne prend pas en compte le cas où le taux de conversion est invalide (négatif). Il devrait retourner un message d’erreur.

MainWindow.convert(String, String, ArrayList<Currency>, Double)

* Test pour les cas de base

Ce test vise à confirmer le bon fonctionnement de la fonction de conversion pour chaque paire de devises spécifiées. Nous examinons toutes les combinaisons possibles de devises, en excluant celles qui ne font pas partie des spécifications (comme CNY et JPY). (valeur attendu/valeur obtenu)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | USD | CAD | GBP | EUR | CHF | AUD |
| **USD** | 100/100 | 137/0 | 66/66 | 93/93 | 101/101 | 152/0 |
| **CAD** | 73/0 | 100/0 | 58/0 | 67/0 | 65/0 | 111/0 |
| **GBP** | 151/151 | 172/0 | 100/100 | 141/141 | 152/152 | 191/0 |
| **EUR** | 107.3/107.3 | 149/0 | 71/71 | 100/100 | 108/108 | 166/0 |
| **CHF** | 99/99 | 152/0 | 66/66 | 93/93 | 100/100 | 172/0 |
| **AUD** | 66/0 | 111/0 | 52/0 | 60/0 | 58/0 | 100/0 |

Les résultats montrent que la conversion se fait de manière adéquate pour la plupart des paires de devises, conformément aux spécifications. Cependant, les devises CAD et AUD, qui ne sont pas définies dans le tableau des devises, ne sont pas en mesure d’être converti. Il faudrait corriger ce bug.

* Test pour les cas invalides

L'objectif de ce test est de vérifier la conversion d’une devise invalide. Nous ne devrions pas être en mesure de convertir une devise qui n’existe pas.

Valeur initiale : 100$

USD vers une Devise inconnue

Valeur attendue : 0

Valeur obtenue : 0

Valeur initiale : 100$

Devise inconnue vers USD

Valeur attendue : 0

Valeur obtenue : 0

Tâche 2

Créer des tests boîtes blanches.

Testez les deux méthodes en utilisant les **5 critères de sélection de jeux de test** quand il fait du sens : couverture des instructions, couverture des arcs du graphe de flot de contrôle, couverture des chemins indépendants du graphe de flot de contrôle, couverture des conditions, couverture des i-chemins.

Currency.convert(Double, Double) :

**A)Critere de couverture d’instructions :**

La méthode est simple, avec seulement trois instructions principales, donc un seul ensemble de tests peut suffire à couvrir toutes les instructions.

**Instructions à couvrir** :

* L'instruction qui effectue la multiplication (amount \* exchangeValue).
* L'instruction qui arrondit le résultat (Math.round(price \* 100d) / 100d).
* L'instruction qui retourne le price.

**D1** : {(amount, exchangeValue) | Test qui couvre toutes les instructions

* (amount ≠ 0, exchangeValue ≠ 0) : couvre toutes les instructions, car la multiplication et l'arrondissement seront exécutés, et le prix sera retourné.

**Jeu de Test** : (100.25, 1.337) : Ce test assure que le montant multiplié par la valeur d'échange nécessite un arrondi, couvrant ainsi toutes les instructions de la méthode convert.

**B)Critère de couverture des arcs du graphe de flot de contrôle :**

**Arc unique** : De l'entrée de la méthode convert, passant par la multiplication (ligne 1), l'arrondi (ligne 2), jusqu'au retour du prix (ligne 3).

**Jeu de Test pour Currency.convert** : {amount = 100.25, exchangeValue = 1.337} couvre l'arc unique de la méthode.

**C) Critère de couverture des chemins indépendants du graphe de flot de contrôle :**

C’est une méthode linéaire sans décisions, la complexité cyclomatique V(G) est 1, donc un seul chemin indépendant est nécessaire, qui est déjà couvert par un seul test comme décrit précédemment.

D)Critere de couverture de conditions :  
Cette méthode ne contient pas de conditions composées. Elle réalise simplement une opération arithmétique suivie d'un arrondi. Par conséquent, il n'y a pas de conditions à tester et le critère de couverture des conditions ne s'applique pas.

E) Critère de couverture de i chemins : Ce critère ne s'applique pas car il n'y a pas de boucles.

MainWindow.convert(String, String, ArrayList<Currency>, Double)

A)