

Test boîte noire

Spécifications :

- Convertir des montants entre les devises suivantes : USD, CAD, GBP, EUR, CHF, CNY.
- Seulement accepter des montants non-négatives.

Tests : classes d'équivalence

Le T1 teste le **montant**, en utilisant des devises valides. Le but est de vérifier que seuls les montants non négatifs sont acceptés.

D1 = montant négatifs

D2 = montant positifs

T1 = {(USD, EUR, -500), (USD, EUR, 500)}

Le test 2 teste les **devises**, le but est de vérifier si les devises du programme sont celles des spécifications. Comme ici on ne permet pas d'inscrire une valeur, on doit s'assurer que la liste contient seulement les devises de la spécification.

Si le taux de change était inclus dans la spécification, on testerait aussi chaque devise entre elle avec un montant valide. Chaque combinaison de devise serait une classe. Mais l'on ne testerait pas le yen japonais puisqu'il ne fait pas parti des spécifications et la devise canadienne qui ne s'y trouve pas. Cependant comme le taux de change n'est pas inclus dans la spécification et que l'on n'a pas accès au code pour la boîte noire, nous ne ferons que le test qui vérifie si les listes déroulantes contiennent les devises voulues.

T2 = S'assurer que les 2 listes contiennent seulement : USD, CAD, GBP, EUR, CHF, CNY.

Tests : valeurs frontières

Le test 3 teste les valeurs de **montant** autour du 0, le cas nul et le cas des décimales en utilisant des devises valides.

T3 = {(USD, EUR, nul), (USD, EUR, -1), (USD, EUR, 1), (USD, EUR, 0), (USD, EUR, 500.5)}

Pour les **devises**, on n'a pas de valeurs frontières.

Résultats :

T1 = (USD, EUR, -500) → -465.00 Erreur : doit refuser car négatif
T1 = (USD, EUR, 500) → 465.00 Ok

T2 → Erreur : manque la devise canadienne (CAD)
T2 → Erreur : devise en trop : Japanese Yen (JPY)

T3 = (USD, EUR, nul) → 0.00 Erreur : Doit refuser car aucun montant
T3 = (USD, EUR, -1) → -0.93 Erreur : Doit refuser car négatif
T3 = (USD, EUR, 1) → 0.93 Ok
T3 = (USD, EUR, 0) → 0.00 Ok
T3 = (USD, EUR, 500.5) → 465.47 Ok

Conclusion :

En effectuant un test « Boite noire » on ne teste pas en fonction du code, mais seulement des spécifications. On utilise donc l'interface. Ici on peut voir que l'on doit ajouter un message d'erreur lorsque le montant est négatif ou nul. Pour ce qui est des devises, la devise canadienne doit être ajoutée, et la devise japonaise doit être enlevée des deux listes déroulantes. Il est possible que le problème se situe au niveau du libellé de la devise. Peut-être que la devise canadienne est présente mais que son libellé est celui de la devise japonaise. Mais comme on n'a pas accès au code dans les tests de boite noire, on ne peut pas tirer de conclusion. Le test boite blanche sera en mesure d'identifier où se situe le problème.

Test boîte blanche

Complicité cyclomatique :

MainWindow.convert : **6**

5 nœuds prédicats + 1.

Currency.convert : **1**

0 nœuds prédicats + 1.

Couverture des instructions :

MainWindow.convert :

D1 : *currency1* et *currency2* existant dans *currencies*.

T1 = {"US Dollar", "Euro", *currencies*, 100.0}

Ici toutes les instructions sont exécutées si on utilise 2 devises se trouvant dans la liste. On doit donc exécuter *Currency.init* afin d'obtenir la liste et la passer en paramètres.

Currency.convert :

D1 : *amount* = x, *exchangeValue* = Y

T2 = {(100, 0.5)}

Ici, on a qu'un seul regroupement car on a aucun nœud prédicat. Toutes les instructions sont donc toujours exécutées.

Couverture des arcs du graphe de flot de contrôle :

MainWindow.convert :

D1 = *Currency* 1 et 2 sont valide

D2 = *Currency* 1 et 2 sont invalides

D3 = *Currency* 1 est invalide et 2 est valide

T3 =

{"US Dollar", "Euro", *currencies*, 100.0),
("Invalide", "Invalide", *currencies*, 100.0),
("Invalide", "Euro", *currencies*, 100.0)}

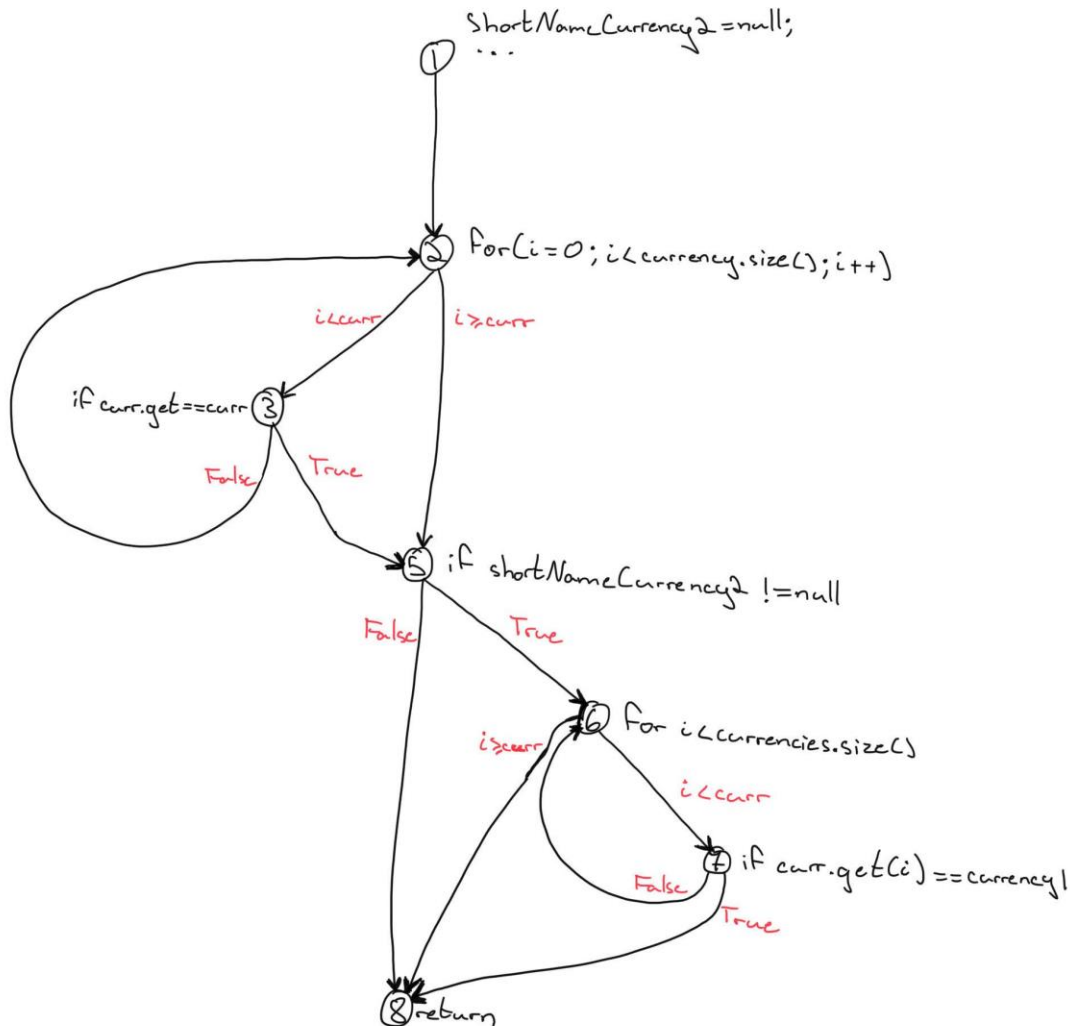
Ici le premier cas touche tous les arcs sauf 2. Le 2^e cas touche 2 autres arcs. Celui des 2 premiers « if ». Pour atteindre le 3^e et dernier arc, *currency1* doit être invalide et *currency2* doit être valide.

Currency.convert :

Comme on n'a pas de nœuds prédicat ici, donc aucune instruction conditionnelle, ce test n'est pas pertinent dans ce cas.

Couverture des chemins indépendants du graphe de flot de contrôle :

Le graphe de flux de contrôle pour la méthode `currencyConverter.MainWindow.convert` est le suivant :



Puisque la complexité cyclomatique de cette méthode est 6, il suffit de trouver 6 chemins indépendants comme base. Ces chemins sont : 1-2-5-8, 1-2-3-2-5-8, 1-2-3-2-5-6-7-6-8, 1-2-3-2-5-6-7-8, 1-2-3-5-8, 1-2-3-2-5-6-8

Chemin 1-2-5-8 : $D1 = \{currency1, currency2, currencies, amount, shortNameCurrency \mid currencies.size() = 0\}$

Chemin 1-2-3-2-5-8 : $D2 = \{currency1, currency2, currencies, amount \mid currencies.size() = 1, currencies.get(i).getName() \neq currency2\}$

Chemin 1-2-3-2-5-6-7-6-8 : $D3 = \{currency1, currency2, currencies, amount \mid currencies.size() = 1, currencies.get(i).getName() = currency2, currencies.get(i).getName() \neq currency1\}$

Chemin 1-2-3-2-5-6-7-8 : $D4 = \{currency1, currency2, currencies, amount \mid currencies.size() = 1, currencies.get(i).getName() = currency2, currencies.get(i).getName() = currency1\}$

Chemin 1-2-3-5-8 : impossible à être exécuté car si à partir du nœud 3 on a True, alors `shortNameCurrency` a une valeur. Donc il ne sera pas possible d'aller directement au nœud 8.

Chemin 1-2-3-2-5-6-8 : impossible à être exécuté également car si on parcourt la boucle for une fois (2-3-2), ceci implique que la taille de Currency est plus que 0. Ca n'aura donc pas de sens aller du nœud 6 directement au nœud 8 (qui impliquerait que Currency est vide).

T4 =

```
{("US Dollar", "Euro", [], 100.0) ,  
("Invalide", "Invalide", currencies, 100.0),  
("Invalide", "Euro", currencies, 100.0),  
("US Dollar", "Euro", currencies, 100.0)}
```

Pour Currency.convert, le test n'est pas pertinent car on a seulement une complexité cyclomatique de 1 et alors seulement 1 chemin indépendant.

Couverture des conditions :

currencyConverter.MainWindow.convert : inutile à faire car la méthode ne possède aucune condition composée.

Pour Currency.convert : cette méthode a aucune condition, donc cette méthode n'est pas éligible pour ce critère.

Couverture des i-chemins :

currencyConverter.MainWindow.convert :

D1 : saute la boucle

D2 : une itération

D3 : deux itérations

D4 : 3 itérations

D5 : 4 itérations

D6 : 5 itérations

D7 : 6 itérations

D8 : quitte la boucle sans trouver de monnaie correspondant

```
T5 = {("US Dollar", "Euro", [], 100.0) ,  
      ("US Dollar", "US Dollar", currencies, 100.0),  
      ("US Dollar", "Euro", currencies, 100.0),  
      ("US Dollar", "British Pound", currencies, 100.0),  
      ("US Dollar", "Swiss Franc", currencies, 100.0),  
      ("US Dollar", "Chinese Yuan Renminbi", currencies, 100.0),  
      ("US Dollar", "Japanese Yen", currencies, 100.0),  
      ("US Dollar", "Invalide", currencies, 100.0)}
```

Résultats :

T1 = ("US Dollar", "Euro", currencies, 100.0) = 93.00	Ok
T2 = (100, 0.5) = 50.00	Ok
T3 = ("US Dollar", "Euro", currencies, 100.0) = 93.0	Ok
T3 = ("Invalide", "Invalide", currencies, 100.0) = 0.0	Ok
T3 = ("Invalide", "Euro", currencies, 100.0) = 0.0	Ok
T4 = ("US Dollar", "Euro", [], 100.0)	Erreur
T4 = ("Invalide", "Invalide", currencies, 100.0)	Ok
T4 = ("Invalide", "Euro", currencies, 100.0)	Ok
T4 = ("US Dollar", "Euro", currencies, 100.0)	Ok
T5 = ("US Dollar", "Euro", [], 100.0)	Erreur
T5 = ("US Dollar", "US Dollar", currencies, 100.0)	Ok
T5 = ("US Dollar", "Euro", currencies, 100.0)	Ok
T5 = ("US Dollar", "British Pound", currencies, 100.0)	Ok
T5 = ("US Dollar", "Swiss Franc", currencies, 100.0)	Ok
T5 = ("US Dollar", "Chinese Yuan Renminbi", currencies, 100.0)	Ok
T5 = ("US Dollar", "Japanese Yen", currencies, 100.0)	Ok
T5 = ("US Dollar", "Invalide", currencies, 100.0)	Ok

Conclusion :

Pour ce qui est des tests de boîte blanche, on devrait inclure une gestion d'erreur dans le cas où la liste « currencies » est vide. Soit retourner une valeur invalide telle -1, ou retourner une exception. Ce qui aurait pour effet d'augmenter la réutilisabilité de cette méthode. Ce cas n'arrive jamais dans le contexte du programme, mais si l'on voulait réutiliser cette méthode dans un autre contexte qui pourrait recevoir une liste vide, le cas de la liste vide devrait être gérée.