

Devoir 4 – IFT 3913

Tache 1: Boite noir

Explications du choix des tests (TestMainWindow):

Il faut noter que l'ensemble des tests de cette classe ont pour but de tester la méthode **currencyConverter.MainWindow.convert**.

- **Tests de conversion valide (testValidConversion, testValidConversionUSD et autres tests similaires) :**
 - Ces tests vérifient si la méthode MainWindow.convert fonctionne correctement pour un cas normalement valide, ainsi que pour toutes les paires de devises que le logiciel "Currency Converter" doit traiter.
 - **Critères:** Fonctionnalité de la méthode MainWindow.convert et exactitude des calculs de conversion.
 - **Observation:** Ces tests permettent de recouvrir l'ensemble des cas de conversion pour assurer la couverture fonctionnelle du système. Néanmoins, nous avons remarqué que les devises "Canadian Dollar" (CAD) et "Australian Dollar" (AUD) n'ont pas encore été implémentées dans la classe Currency. C'est pourquoi aucun des tests dont la devise source est USD, CAD, GBP, EUR, CHF ou AUD, ne peut être entièrement validé tant que ces deux devises ne sont pas implémentées dans Currency. Les tests des autres paires de devises sont bien validés comme le montre testValidConversion. Nous avons donc estimé les valeurs attendues que les tests impliquant les CAD ou AUD devaient retourner. Ainsi, nous pourrions facilement valider le système une fois que l'implémentation de ces devises sera complétée.
- **Test de devise invalide (testInvalidCurrency) :**
 - Vérifie si l'entrée d'une 'currency' non reconnu par le système retourne bien une erreur 'IllegalArgumentException'.
 - **Critères:** Gestion des erreurs et validation des paramètres donnés
 - **Observation:** Ce test assure que le système gère correctement les erreurs de paramètres inconnus et renforce la fiabilité du logiciel.
- **Test de montant hors plage (testAmountOutOfRange) :**
 - Vérifie si le montant entré en paramètre est bien compris entre 0 et 1 million, et s'assure que le système renvoie une erreur 'IllegalArgumentException', sinon.
 - **Critères:** Contraintes de la spécification.

- **Observation:** Ce test assure que le paramètre du montant (amount) est bien conforme aux spécifications du système pour renforcer sa fiabilité.

Analyse des résultats (TestMainWindow):

Les tests de Conversion Valide (testValidConversion, testValidConversionUSD et autres tests similaires) ont montré qu'il manquait l'implémentation des devises "Canadian Dollar" (CAD) et "Australian Dollar" (AUD). Ces tests devraient passer lorsque l'implémentation du "Currency Converter" sera complétée.

Le test de Devise Invalide (testInvalidCurrency) et celui du montant Hors Plage (testAmountOutOfRange) permettent d'assurer que les paramètres sont valides et vérifient que les contraintes de la spécification sont bien respectées.

Explications du choix des tests (TestCurrency):

- **Tests de conversion valide (testValidConversion, testValidConversionUSD et autres tests similaires) :**
 - Vérifie si la méthode Currency.convert est fonctionnelle et calcul correctement pour un cas valide et les paires de devises qui seront utilisées par le système.
 - **Critère :** Fonctionnalité de la méthode et exactitude des calculs de conversion.
 - **Observation:** Ces tests couvrent un large éventail de paires de devises, assurant ainsi une bonne couverture fonctionnelle.
- **Test de montant négatif (testNegativeAmount) et test de montant excessif (testExcessiveAmount) :**
 - Vérifie la gestion des montants négatifs et excessifs.
 - **Critère :** Gestion des erreurs, validation des entrées et conformité aux spécifications.
 - **Observation:** Assure la robustesse du système par la vérification de la conformité des paramètres.
- **Test de montant nul (testZeroAmount) et taux de change nul (testInvalidExchangeRateZero) :**
 - Teste le comportement avec un montant nul et un taux de change nul.
 - **Critère :** Cas limites et gestion des valeurs spéciales.
 - **Observation:** Assurer que les cas de paramètres inattendus sont bien gérés et renvoie une erreur correspondante, sinon.
- **Test de taux de change négatif (testInvalidExchangeRateNegative) :**
 - Évalue la réaction de la méthode face à un taux de change négatif.
 - **Critère :** Validation des entrées et gestion des erreurs.
 - **Observation:** Alerte et vérifie que des cas spéciaux n'aient lieu.

Analyse des résultats (TestCurrency):

Les tests unitaires de la classe TestCurrency dans le logiciel « Currency Converter » garantissent l'exactitude et la fiabilité des calculs de conversions de devises. Ils se concentrent sur deux aspects principaux : la vérification de la précision des conversions de devises en fonction des taux de change préétablis et le respect des contraintes et spécifications, comme les limites de montant et la gestion des taux de change négatifs ou nuls.

Les résultats attendus de ces tests confirment non seulement la justesse des calculs pour des entrées valides, mais ils mettent aussi en évidence la robustesse de la méthode face à des entrées inhabituelles ou erronées. Cette approche garantit que le logiciel reste fiable même dans des scénarios de cas inattendu.

Tache 2: Boite Blanche

Explication du choix des tests:

Lors de la conception des tests des deux méthodes convert, l'approche adoptée visait à couvrir le plus de scénarios possibles. Les cas de test incluent des valeurs positives standards pour simuler des conditions d'utilisation normales, des valeurs nulles pour tester la robustesse de la méthode face à des entrées potentiellement problématiques mais qui peuvent arriver, ainsi que des valeurs négatives pour évaluer le comportement des deux méthodes dans des situations non standards. De plus, certains tests avec des valeurs extrêmes, comme des montants très grands ou très petits, ont été inclus pour repousser les limites des méthodes afin de vérifier leur comportement et leur fiabilité.

Analyse des résultats:

Les résultats des tests ont révélé que la méthode convert fonctionne bien dans la plupart des scénarios standards pour les deux classes, mais présente des lacunes dans la gestion des valeurs négatives et nulles. Le succès des tests avec des valeurs positives et extrêmes indique que la méthode est capable de gérer une gamme étendue de situations réalistes et hypothétiques en termes de montants et de valeurs d'échange. Cependant, l'échec des tests impliquant des valeurs négatives et nulles souligne une vulnérabilité dans les méthodes, suggérant que des mesures de validation ou de gestion d'erreur supplémentaires pourraient être nécessaires pour gérer ces cas. Cette constatation met en lumière l'importance d'inclure des scénarios de test variés, non seulement pour valider la fonctionnalité dans des conditions normales mais aussi pour garantir la robustesse et la sécurité du système dans des situations exceptionnelles ou erronées.

Observations sur les différentes approche et critères de test:

L'analyse des résultats des tests pour les deux méthodes convert l'une dans la classe Currency et l'autre dans la classe MainWindow - offre des perspectives importantes sur l'application et l'efficacité des critères de test suivants :

1.Couverture des Instructions :

Currency : Ce critère a été pleinement vérifié. Chaque ligne de la méthode convert a été exécutée, assurant une évaluation complète de la fonctionnalité.

MainWindow : Semblablement, la couverture des instructions a été assurée, testant chaque partie du code pour garantir qu'aucune instruction n'est laissée non testée.

2.Couverture des Arcs du Graphe de Flot de Contrôle :

Currency : La méthode est relativement simple avec un seul chemin donc pas très pertinent.

MainWindow : Ce critère a plus de pertinence, vu que la méthode comporte plusieurs branches de décision. Les tests étaient conçus pour parcourir différents chemins logiques.

3.Couverture des Chemins Indépendants :

Currency : Moins pertinent en raison de la simplicité de la méthode. Les "chemins" sont linéaires, avec peu de variations.

MainWindow : Assez utile car la méthode contient des boucles et des branches multiples. Les tests visaient à couvrir différents chemins d'exécution indépendants.

4.Couverture des Conditions :

Currency : Pas directement applicable car la méthode ne contient pas de logique conditionnelle complexe.

MainWindow : La méthode a plusieurs conditions donc les tests l'ont vérifié.

5.Couverture des i-Chemins :

Currency : Compte tenu de la structure directe de la méthode, ce critère n'est pas fortement pertinent.

MainWindow : Ce critère est significatif pour s'assurer que tous les chemins individuels possibles à travers le code sont testés, particulièrement dans les méthodes avec des logiques plus ou moins complexes comme celle-ci.