

Mini Projet Test et Vérification

Exemple de test de CppUnit

[Tapez le nom de l'auteur]

Exemple de test de CppUnit

Pour tester CppUnit on a choisi de faire une application pour une banque, cette application offre des tests sur des sommes d'argents, l'application doit créer des objets de type Argent qui se caractérisées par deux attributs **m_valeur** et **m_devise**, qui contient respectivement la valeur de la somme d'argent qui est un double et la devise qui est une chaine de caractères.

Cette classe contient aussi les getteurs et les setteurs des deux attributs et une redéfinition des opérateurs += et == pour satisfaire la comparaison en valeur et devise.

On a aussi la Test Suite TestArgent qui contient la suite des tests à effectuer

TestArgent ::testConstructor() qui permet de tester que la valeur initiale de la compte soit égale à 10TND.

TestArgent ::testEqual() qui permet de faire une comparaison entre deux sommes d'argent introduite par l'utilisateur.

TestArgent ::testAdd() qui sert à tester une addition de deux sommes d'argents et bien évidemment le test sera sur la devise puisqu'on ne peut pas additionner deux objets différents.

L'unité de test CppUnit offre une bibliothèque qui contient des méthodes de détection des erreurs tels que CPPUNIT_ASSERT et CPPUNIT_ASSERT_EQUAL.

Si une erreur s'est produit Un message d'erreur sera affiché indiquent la valeur expectée et la valeur introduite sinon un message sera affiché indiquent le nombre de test effectué avec succès.

TestArgent suite contient une déclaration de la suite de test CPP_TEST_SUITE(TestArgent) et la liste des tests à effectuer successivement : CPPUNIT_TEST (testConstructor), CPPUNIT_TEST (testEqual), CPPUNIT_TEST (testAdd), et la clôture de la suite de test CPP_TEST_SUITE_END ().

Pour tester cette suite de test il faut importer « les includes » de CppUnit dans le projet et ajouter la bibliothèque crée à partir de CppUnit code source appelé MyCppUnitStaticLibrary pour pouvoir utiliser les méthodes fournissés par CppUnit, On doit aussi créer un fichier source appelé BanqueApp.cpp qui contient le main pour tester notre test suite, le main contient la registration du test suite déjà crée « TestArgent » dans le registre des tests « CPPUNIT_TEST_SUITE_REGISTRATION »

Puis on doit extraire la dernière suite dans le registre « TestArgent » dans notre cas et on va l'exécuter sur un « Runner ».

Pour notre exemple l'application fournit 3 tests :

1- Test de la vérification de valeur initiale de compte '10TND'

Ce test sert à vérifier que chaque compte crée par l'utilisateur doit être de valeur initiale égale à 10TND pour cela on demande à l'utilisateur de taper la valeur de la somme d'argent initiale et la devise tels que TND, EUR...

Mini Projet Test et Vérification

Exemple de test de CppUnit

[Tapez le nom de l'auteur]

Un test sera lancé pour vérifier si la valeur de la somme introduite est égal à 10 et la devise est TND, ce test est introduit par la méthode `CPPUNIT_ASSERT_EQUAL` fournie par CppUnit.

2- Comparaison de deux sommes d'argent

Pour faire une comparaison entre deux sommes d'argents on doit utiliser les opérateurs `==` et `!=` pour cela on a fait une redéfinition des deux opérateurs pour faire la différence en valeur et en devise d'où on peut tester directement par exemple `CPPUNIT_ASSERT (S1==S2)` où `S1` et `S2` sont deux objets de type `Argent`.

Si les deux objets sont différents en valeur ou en devise CppUnit fournit un message d'erreur.

3- Test Addition de deux sommes d'argent

Pour additionner deux sommes d'argents il faut qu'elles sont sur la même devise pour cela l'utilisateur doit fournir deux sommes d'argents avec leurs devise, un test sera lancé pour vérifier la devise, puisqu'on a modifié l'opérateur `+=` pour additionner deux sommes d'argents « objets de type `Argent` » on a utilisé dans la redéfinition `CPPUNIT_ASSERT_EQUAL ()` pour vérifier que la devise de la deuxième somme d'argent soit identique à celle du première somme.

Exemple d'exécution de l'application :

L'application doit fournir initialement la liste des tests à faire successivement puis commence à faire les tests un par un en attente des valeurs introduite par l'utilisateur.

On va donner deux exemples de test deux test avec des erreurs exprès et un autre test avec des inputs correcte et avec l'affichage de l'application :

1- Test avec des erreurs exprès 1 :

1^{er} test: Test de la vérification de valeur initiale de compte '10TND'

On va introduire une valeur initiale de 20TND pour l'application.

2^{ème} test: Comparaison de deux sommes d'argent

On va introduire deux sommes d'argents 300TND et 20EUR et on doit introduire que les deux sommes sont Egaux « E ».

3^{ème} test: Test Addition de deux sommes d'argent

On va additionner deux sommes d'argents différentes en devise par exemple : 30EUR et 20TND.

Résultat d'exécution :

Mini Projet Test et Vérification

Exemple de test de CppUnit

[Tapez le nom de l'auteur]

Bienvenue au programme de test de la banque !!

Opérations :

- 1) Test de la vérification de valeur initiale de compte '10TND'.
- 2) Comparaison de deux sommes d'argent.
- 3) Test Addition de deux sommes d'argent.

Les opérations seront exécutées dans l'ordre si dessus :

```
.
1) Test de la vérification de valeur initiale de compte '10TND'
   Donner la valeur du somme d'argent : 20
   Donner la devise du somme d'argent : TND, EUR... TND
F.
2) Comparaison de deux sommes d'argent
   Donner la valeur du 1ere somme d'argent : 300
   Donner la devise du 1ere somme d'argent : TND, EUR... TND
   Valeur introduite : 300TND
   Donner la valeur du 2eme somme d'argent : 20
   Donner la devise du 2eme somme d'argent : TND, EUR... EUR
   Valeur introduite : 20EUR
300TND ET 20EUR
   Pour Tester les deux sommes vous taper 'E' pour tester s'ils sont égaux et 'D'
   pour différents
E
F.
3) Addition de deux somme d'argent
   Donner la valeur du 1ere somme d'argent : 30
   Donner la devise du 1ere somme d'argent : TND, EUR... EUR
   Combien voulez-vous ajouter à cette somme : 30EUR ?!
   Donner la valeur du 2eme somme d'argent : 20
   Donner la devise du 2eme somme d'argent : TND, EUR... TND
F
```

TestArgent.cpp:48:Assertion

Test name: TestArgent::testConstructor

equality assertion failed

- Expected: 10

- Actual : 20

TestArgent.cpp:87:Assertion

Test name: TestArgent::testEqual

assertion failed

- Expression: SommePrimaire == SommeDeuxieme

Argent.h:64:Assertion

Test name: TestArgent::testAdd

equality assertion failed

- Expected: EUR

- Actual : TND

Failures !!!

Run: 3 Failure total: 3 Failures: 3 Errors: 0

Mini Projet Test et Vérification

Exemple de test de CppUnit

[Tapez le nom de l'auteur]

2- Test avec des erreurs exprès 2 :

1^{er} test: Test de la vérification de valeur initiale de compte '10TND'

On va introduire une valeur initiale de 10TND pour l'application.

2^{ème} test: Comparaison de deux sommes d'argent

On va introduire deux sommes d'argents 30TND et 30EUR et on doit introduire que les deux sommes sont Egaux « E ».

3^{ème} test: Test Addition de deux sommes d'argent

On va additionner deux sommes d'argents différentes en devise par exemple : 30EUR et 200EUR.

Résultat d'exécution :

Bienvenue au programme de test de la banque !!

Opérations :

- 1) Test de la vérification de valeur initiale de compte '10TND'.
- 2) Comparaison de deux sommes d'argent.
- 3) Test Addition de deux sommes d'argent.

Les opérations seront exécutées dans l'ordre si dessus :

```
.
1) Test de la vérification de valeur initiale de compte '10TND'
   Donner la valeur du somme d'argent : 10
   Donner la devise du somme d'argent : TND, EUR... TND
.
2) Comparaison de deux sommes d'argent
   Donner la valeur du 1ere somme d'argent : 30
   Donner la devise du 1ere somme d'argent : TND, EUR... TND
   Valeur introduite : 30TND
   Donner la valeur du 2eme somme d'argent : 30
   Donner la devise du 2eme somme d'argent : TND, EUR... EUR
   Valeur introduite : 30EUR
30TND ET 30EUR
   Pour Tester les deux sommes vous taper 'E' pour tester s'ils sont égaux et 'D'
   pour différents
E
F.
3) Addition de deux somme d'argent
   Donner la valeur du 1ere somme d'argent : 30
   Donner la devise du 1ere somme d'argent : TND, EUR... EUR
   Combien voulez-vous ajouter à cette somme : 30EUR ?!
   Donner la valeur du 2eme somme d'argent : 200
```

Mini Projet Test et Vérification

Exemple de test de CppUnit

[Tapez le nom de l'auteur]

Donner la devise du 2eme somme d'argent : TND, EUR... EUR

```
TestArgent.cpp:87:Assertion
Test name: TestArgent::testEqual
assertion failed
- Expression: SommePrimaire == SommeDeuxieme
```

Failures !!!

Run: 3 Failure total: 1 Failures: 1 Errors: 0

1- Test avec des entrées valides :

1^{er} test: Test de la vérification de valeur initiale de compte '10TND'

On va introduire une valeur initiale de 10TND pour l'application.

2^{ème} test: Comparaison de deux sommes d'argent

On va introduire deux sommes d'argents 30TND et 30EUR et on doit introduire que les deux sommes sont Différents « D ».

3^{ème} test: Test Addition de deux sommes d'argent

On va additionner deux sommes d'argents différentes en devise par exemple : 30EUR et 200EUR.

Résultat d'exécution :

Bienvenue au programme de test de la banque !!

Opérations :

- 1) Test de la vérification de valeur initiale de compte '10TND'.
- 2) Comparaison de deux sommes d'argent.
- 3) Test Addition de deux sommes d'argent.

Les opérations seront exécutées dans l'ordre si dessus :

- .
- 1) Test de la vérification de valeur initiale de compte '10TND'
Donner la valeur du somme d'argent : 10
Donner la devise du somme d'argent : TND, EUR... TND
 - .
 - 2) Comparaison de deux sommes d'argent
Donner la valeur du 1ere somme d'argent : 30
Donner la devise du 1ere somme d'argent : TND, EUR... TND
Valeur introduite : 30TND
Donner la valeur du 2eme somme d'argent : 30
Donner la devise du 2eme somme d'argent : TND, EUR... EUR
Valeur introduite : 30EUR
30TND ET 30EUR

Mini Projet Test et Vérification

Exemple de test de CppUnit

[Tapez le nom de l'auteur]

Pour Tester les deux sommes vous taper 'E' pour tester s'ils sont égaux et 'D' pour différents

D

.

3) Addition de deux somme d'argent

Donner la valeur du 1ere somme d'argent : 30

Donner la devise du 1ere somme d'argent : TND, EUR... EUR

Combien voulez-vous ajouter à cette somme : 30EUR ?!

Donner la valeur du 2eme somme d'argent : 200

Donner la devise du 2eme somme d'argent : TND, EUR... EUR

OK (3)