

# Exercice pour Test unitaire et test d'intégration

## TABLE DES MATIERES

<b><i>Section 1 Généralités.....</i></b>	<b><i>3</i></b>
1.1 Déroulement de l'exercice.....	3
<b><i>Section 2 Les classes à développer.....</i></b>	<b><i>4</i></b>
2.1 Diagramme de classe.....	4
2.2 Classe Money.....	4
2.3 Classe MoneyBag.....	4

## Section 1 Généralités

---

### 1.1 Déroulement de l'exercice

A partir d'un diagramme de classe vous allez implémenter le code en Java.

Vous allez choisir la classe la plus 'interne' et appliquer la méthode de développement TDD, jusqu'à l'obtention d'un code correct et testé.

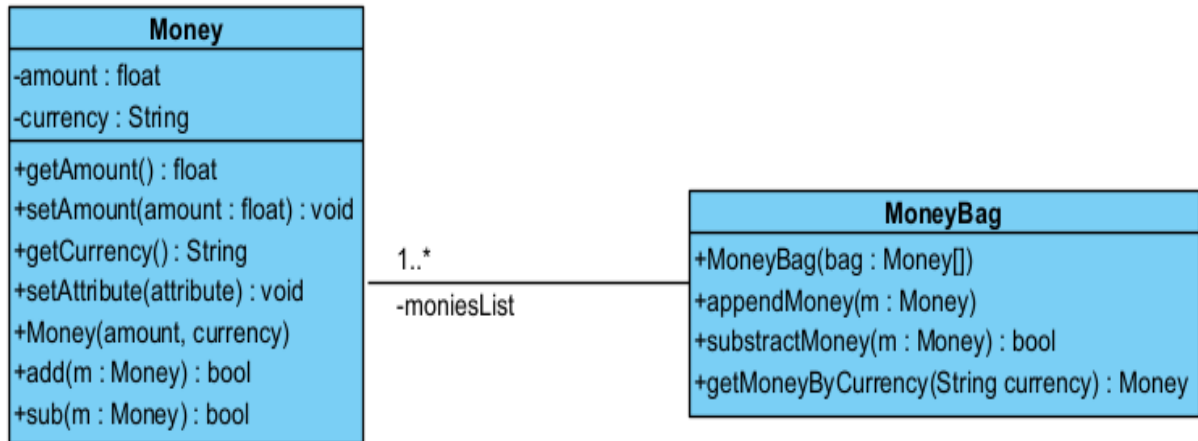
Vous développez ensuite la seconde classe en TDD, en imaginant un bouchon de la première ; jusqu'à l'obtention d'un code correct et testé.

Vous réalisez ensuite une intégration des 2 classes, donc sans bouchon

## Section 2 Les classes à développer

---

### 2.1 Diagramme de classe



### 2.2 Classe Money

Tout objet de la classe Money permet d'enregistrer une somme d'argent et une devise.

La méthode add permet d'ajouter les sommes dans l'objet pourvu que les 2 devises soient identiques. Rend false sinon.

La méthode sub permet de retrancher les sommes si les devises sont identiques. Rend false sinon. Le résultat peut être une somme négative.

SetAmount() génère une exception si la somme est négative.

### 2.3 Classe MoneyBag

Ce porte monnaie contient des objets Money.

Il ne peut pas y avoir plusieurs objets Money avec la même devise.

appendMoney() permet d'ajouter un objet Money ou tout au moins la somme de l'objet Money avec celui existant avec la même devise. Toute somme résultante de 0 fait supprimer l'objet de MoneyBag.

subtractMoney() permet d'enlever une somme à un objet existant de même devise, ou d'ajouter cet objet au porte monnaie (avec la bonne somme). La somme résultante peut être négative. Toute somme résultante de 0 fait supprimer l'objet.

GetMoneyByCurrency() rend l'objet Money correspondant à la devise. Rend null sinon.