

Ateliers « Mise en pratique des tests avec JUnit »

Pré-requis :

- JDK 8+
- IDE (IntelliJIDEA, Eclipse, STS, VSCode)

TP1 : Concepts JUnit

Objectifs

L'objectif de ce TP est de se familiariser avec les concepts de JUnit. En particulier, les notions de TestCase, de Fixture. Ensuite, l'approche XP est entrevue.

1.1 : Mise en place du projet

Créer un projet Maven et ajouter la dépendance suivante dans le *pom.xml*

```
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter</artifactId>
  <version>5.7.0</version>
  <scope>test</scope>
</dependency>
```

Récupérer la classe *Money*

1.2 : Premières méthodes de test

Nous disposons d'une classe *Money* qui comprend deux attributs : le montant (amount) et la monnaie (currency). Cette classe contient déjà une méthode permettant d'additionner deux objets de type *Money*.

Nous proposons d'écrire les TestCase permettant de tester la méthode *add()* et la méthode *equals()*

1. Créer la classe *MoneyTest*.
2. Implémenter les méthode vous paraissant adéquates pour tester les différents cas de test

1.3 : Approche TDD

Jusqu'à maintenant, nous avons utilisé des sommes d'argent de même type.

Pour additionner des sommes d'argent de monnaie différente, nous créons un nouvelle classe *MoneyBag* qui représente une collection de sommes d'argent de monnaies différentes.

Les classes *Money* et *MoneyBag* implémentent l'interface *IMoney* qui spécifie une seule méthode :

Nous voulons implémenter pour ces 2 classes la méthode *IMoney add(IMoney money)* .

Avant de les implémenter, définir les différents cas de test et les implémenter dans une classe de test. Réfléchir à la notion de *MoneyBag* vide

Procéder comme suit :

1. Définir les fixtures nécessaires et les initialiser

2. Écrire les méthodes de tests et s'assurer qu'elles échouent
3. Implémenter la méthode *add()* de la classe **Money**, réexécuter les tests
4. Implémenter la méthode *add()* de la classe **MoneyBag** et s'assurer que tous les tests passent

1.4 : Exécution Maven

Quelle plugin et quelle version par défaut correspond à la phase maven **test** ?

Modifier le **pom.xml** afin que les tests JUnit5 soit détectés.

TP2 : Matcher Hamcrest

2.1: Ajout de dépendance

Ajouter la dépendance Hamcrest dans le *pom.xml*

```
<dependency>
  <groupId>org.hamcrest</groupId>
  <artifactId>hamcrest-all</artifactId>
  <version>1.3</version>
  <scope>test</scope>
</dependency>
```

2.1: Matcher

Réécrire toutes les assertions avec des Matchers. Vous pouvez utiliser :

- Des contraintes sur le type
- Des contraintes sur des propriétés de JavaBean
- Des contraintes sur les collections
- Des *FeatureMatcher*
- Des opérateurs de coimbinaison

Faire échouer des tests pour visualiser le reporting