



TP Convertisseur euro/dollar

avec la classe pré-existante BigDecimal

Sujet : Les méthodes, classes wrapper java.math.BigDecimal, l'énumération java.math.RoundingMode, test unitaire (non-JUnit)

1 Item : méthode euroToDollar

Un outil de conversion euro vers dollar

Avec votre IDE, et si nécessaire, vous créez le nouveau projet FrameWorkAfpa.

Dedans, et si nécessaire, vous créez une classe fr.afpa.math.Math. Dans cette classe, vous créez une méthode static **euroToDollar** auquel on passe une valeur en euro et qui retourne une valeur en dollar (voir la signature plus bas).

Pour conserver une précision maximum, vous utilisez les objets BigDecimal et leurs méthodes divide, multiply, setScale, ...

Conseils : Vous codez une méthode de classe avec la signature suivante:

```
public static BigDecimal euroToDollar( BigDecimal euro );
```

Conseil : Pour le taux de conversion entre les deux monnaies, vous pouvez utiliser une constante. Par exemple :

```
final BigDecimal TAUX_EURO_DOLLAR = new BigDecimal("1.18674");
```

Faites le premier test dans le main de Math

2 Item : développement de l'applcatif

L'applcatif est codé dans le main de la classe fr.afpa.util.Convertisseur (à créer).

L'applcatif utilise la méthode euroToDollar précédente.

Votre applcatif (la classe fr.afpa.util.Convertisseur) demande à l'opérateur une valeur en euro puis affiche en dollars la valeur convertie. Pour la saisie, vous pouvez utiliser la méthode showInputDialog qui ouvre une boîte de dialogue modale. Pour l'affichage du résultat, vous pouvez utiliser la méthode showMessageDialog.

3 item: conversion dollar vers euro

Codez un convertisseur dollars vers euro. Le préalable est de coder la méthode fr.afpa.math.Math.dollarToEuro .

Complétez l'applcatif fr.afpa.util.Convertisseur avec cette nouvelle fonctionnalité.

4 Item : tests unitaires des deux méthodes

Développez un test unitaire (sans le framework JUnit pour l'instant) qui vérifie, par exemple, que 10 euros valent 11.87 dollars.

Vous créez une classe (le cas de test) fr.afpa.math.TestMath (Le sourceFolder est toujours src). Dans TestMath, vous codez les méthodes de test avec la convention de nom qui est la suivante : euroToDollarTest1, euroToDollarTestDeux, ...dollarToEuroTestUn, dollarToEuroTest2, ...

Généralement, il faut un seul test par méthode (une seule assertion par méthode). Pour les tests, on pourrait chaîner les deux méthodes et retrouver la valeur initiale.

Exemple de pseudo-code pour un test unitaire euroToDollarTestDix:

```
Si 10 euros valent 11.87 dollars alors
    Afficher "test réussi"
Sinon
    Afficher (en rouge) "test en échec: 10 euros valent 11.87 dollars"
```