TP - Tests Unitaires

Dans ce TP, nous allons apprendre à écrire une classe de test. L'objectif est de créer une classe Color composée sur le modèle RGB.

Voici les spécifications :

Votre classe Color doit avoir **deux constructeurs** :

1. Un constructeur à trois arguments :
    a. Argument 1 : un entier pour le rouge.
    b. Argument 2 : un entier pour le vert.
    c. Argument 3 : un entier pour le bleu.

Une exception de type IllegalArgumentException est levée si un argument est incorrect (pas compris entre 0 et 255 inclus).

2. Un constructeur à un argument :
    a. Argument 1 : une chaîne de caractères au format hexadécimal.

Une exception de type IllegalArgumentException est levée si l'argument est incorrect.
Format de la chaîne de caractères voulu :

- Commence obligatoirement par #
- Puis d'un triplet hexadécimal qui forme un nombre hexadécimal à 6 chiffres. Avec les deux premiers pour le rouge, les deux suivants pour le vert et les deux derniers pour le bleu. La valeur possible d'un chiffre hexadécimal est compris entre 0 et 9 ou A, B, C, D, E, F.

Votre classe Color doit avoir **les méthodes get et set** suivantes :

- getRed / setRed
- getGreen / setGreen
- getBlue / setBlue
- getHexValue / setHexValue

Une exception est levée pour les mêmes raisons que pour les constructeurs.

Votre classe Color doit avoir **une méthode toString()** qui retourne la chaîne de caractères suivante :

```
[value=#D58D35, r=213, g=141, b=53]
```

Réalisation :

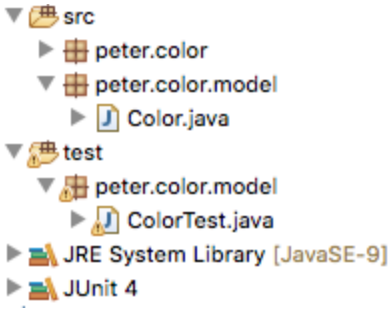
Créez un projet ColorJava.
Ajoutez la classe Color dans un package bien nommé (ex : peter.color.model).

Ajoutez à la même hauteur que de dossier src/ un nouveau dossier de source (clic sur le projet, new source folder), nommez-le test.

Dans ce dossier, créez un nouveau package avec le même nom que celui créé dans le dossier src. Ajoutez-y la classe de test : ColorTest.

Il n'y a pas de méthode main dans votre projet.

Vous devez avoir :



Rappel d'**un cycle de TDD** (Test Driven Development) :

1. Écrire un premier test.
2. Vérifier qu'il échoue (car le code qu'il teste n'existe pas), afin de vérifier que le test est valide.
3. Écrire juste le code suffisant pour passer le test.
4. Vérifier que le test passe.
5. Réviser le code (refactoring), i.e. l'améliorer tout en gardant les mêmes fonctionnalités.

Maintenant que vous avez votre structure, à vous de développer vos 2 classes, dans le but d'obtenir une classe Color fonctionnelle et dépourvue de bugs.