

- TD 10 -

Test Conteneur

Le but de ce TP est d'écrire les tests pour une classe `Conteneur` censée implémenter un ensemble. Les sources nécessaires sont disponibles sur la page du cours.

1.Interface

L'interface `Conteneur` définit la spécification voulue. Il s'agit d'une collection d'objets similaire à `java.util.Set`, mais moins souple (pas de redimensionnement automatique, impossible d'ajouter un élément 2x, ou d'enlever un élément non présent, ...).

Lisez-là bien, c'est elle qui vous guidera sur les tests à écrire (ou pas).

2.Implémentation

`ConteneurImpl` est une implémentation buguée (et inefficace). Il n'est pas utile de la lire. Mais sachez que les bugs ne sont pas évidents à trouver.

3.Travail

Il s'agit d'écrire une série de tests pour identifier les bugs de l'implémentation (sans les corriger).

Les méthodes considérées comme "too small to fail" sont

- le constructeur
- `estVide()`
- `taille()`
- `capacite()`
- `contient(Object)` : pour cette fonction, il est moins évident à sa lecture qu'elle soit "too small to fail", mais dans le cas contraire les tests seraient beaucoup moins aisés à écrire puisque vous devriez alors tester en vous basant sur la structure interne de la collection.

Les autres méthodes doivent être testées de la manière la plus complète possible.

Conseil : Pour débuter l'écriture de vos tests sous NetBeans utilisez le menu "Tools > Create JUnit Tests".

4.Méthode ajouter

Les tests pour la méthode `ajouter()` sont décrits ci-dessous :

- Right
 - Ajouter 1 élément
 - Ajouter 2 éléments différents
- Boundary
 - Ajouter null
 - Ajouter 2 fois le même élément
 - Ajouter 2 éléments égaux (dans le sens de `Object.equals()`)
 - Ajouter 2 éléments dans un Conteneur de capacité 2
 - Ajouter 2 éléments dans un Conteneur de capacité 1
- Crosscheck
 - Ajouter 4 éléments dans un Conteneur et dans un `java.util.HashSet`, vérifier que tous les éléments du `HashSet` sont présents dans le Conteneur

Trois de ces tests devraient échouer.

5.Méthode `retirer`

A vous de définir les test de la méthode `retirer`. Attention, ici l'ordre des appels (`ajouter/retirer`) a son importance.

Indication : il y a trois erreurs dans la méthode `retirer`.

6.Méthode `redimensionner`

Ecrire 5 tests différents pour la méthode `redimensionner()` dont 3 Right.

Indication : il y a deux erreurs dans la méthode `redimensionner`.