

Implanter une nouvelle collection

Exercice 1 - J'en veux plus

On cherche à implanter la méthode `twice` qui prend une liste d'entiers et qui renvoie une nouvelle liste contenant les valeurs de la liste multipliées par deux.

- 1 Implanter la méthode `twice`.
- 2 Changer l'implantation pour allouer moins d'éléments en utilisant le concept de vue. Pour l'implantation, regarder du côté de `java.util.AbstractList`.
- 3 Quel est le problème de votre implantation dans le cas d'un appel à `twice` avec une `LinkedList` ?
Changer l'implantation de `twice` en fonction du fait que la liste prise en paramètre implante ou non `java.util.RandomAccess`.

Exercice 2 - Pris la main dedans

Il est classique dans plusieurs algorithmes d'utiliser une structure de données appelée Bag. Celle-ci permet de stocker des objets un certain nombre de fois, la structure garde en mémoire le nombre de fois qu'un même objet (au sens de `equals`) est stocké.

- 1 Écrire l'interface `fr.umlv.util.bag.Bag` possédant les méthodes `add`, `remove` et `count` qui respectivement ajoute un objet, retire un objet et renvoie le nombre d'occurrences d'un objet.
- 2 Commenter l'interface écrite.
- 3 Fournir une implantation `BagImpl` de cette interface permettant d'ajouter et de retirer des éléments en temps constant moyen.
- 4 Ajouter une méthode `Iterator<Map.Entry<T,Integer>> iterator()` qui renvoie un itérateur sur les couples (objet, nombre d'occurrences).
- 5 Faire en sorte que l'on puisse choisir l'ordre des objets lors de l'itération par une constante lors de la construction du Bag :

```
Bag<String> b1=new BagImpl<String>(BagOrder.ANY_ORDER);  
// n'importe quel ordre  
Bag<String> b2=new BagImpl<String>(BagOrder.INSERT_ORDER);  
// ordre d'insertion  
Bag<String> b3=new BagImpl<String>(BagOrder.NATURAL_ORDER);  
// ordre naturel de T
```

Quelles sont les contraintes sur `T` en fonction de la collection utilisée ?

- 6 Remplacer les constantes passées lors de la construction par une énumération (`enum`).
- 7 Discuter de l'utilité d'une énumération abstraite ici. Implanter la solution proposée.
- 8 On souhaite que le code suivant marche :

```
Bag<String> bag=new BagImpl<String>(ANY_ORDER);  
for(Map.Entry<String,Integer> entry:bag)  
System.out.println(entry.getKey()+"="+entry.getValue());
```

Que doit-on faire ?

- 9 Pourquoi le code suivant ne marche pas ?

```
Bag<?> bag=new BagImpl<String>(ANY_ORDER);
```

```
Iterator<Map.Entry<?, Integer>> it=bag.iterator();
```

Comment changer le code pour qu'il marche ?

- 10 De quelle interface des collections en Java peut hériter l'interface Bag.
Quelles sont les problèmes que cela pose par rapport au code existant ?
- 11 Faire hériter Bag de cette interface et écrire les méthodes manquantes.