

M1 IFPRU - Ingénierie Logiciel

TD-TP No 3

Frameworks :

Tests et réutilisation

Typage statique réutilisation

1 Tests et Réutilisation

Rappels

- test fonctionnels (ou test de boîte noire) :
on élabore les critères de test sur la base de l'interface de la classe testée. Ces tests doivent être réalisés en situation de client de la classe.
- Test structurels (ou tests de boîte blanche) :
on ajoute de nouveaux critères sur la base de la connaissance de l'implantation, ces tests peuvent nécessiter (non obligatoire) l'accès aux ressources privées. Si c'est le cas ils sont nécessairement définis dans des méthodes de la classe testée.

Dans cet exercice, vous allez faire un seul paquet de jeux de test, mélangeant boîte noire et boîte blanche. Il vous est demandé de tester la classe `FastDictionary`

Questions

1. Proposer un ensemble raisonnable de critères portant a) sur les spécifications (par exemple : "la clé recherchée par la méthode "get" est ou n'est pas dans le dictionnaire") et b) sur les éléments d'implantation ,par exemple ("les tableaux sont de taille paire ou impaire").
Croisez les critères pour obtenir un ensemble de situations couvrant le maximum de situations possibles. Par exemple un exemple de situation sera : le dictionnaire contient deux tableaux de taille impaire n'ayant jamais grossi, l'insertion de l'élément provoquera un grossissement du tableau.
2. Construire une (ou plusieurs) méthodes automatisant les tests pour la classe "FastDictionary". Si plusieurs méthodes sont définies, une lancera toutes les autres. Chaque méthode de test doit afficher "ok" ou "échec" au terminal.
3. Réflexion : est-il possible de définir une partie des méthodes de test au niveau de la classe abstraite (classe `AbstractDictionary`).

2 Réutilisation

1. Réalisez une extension de la classe `OrderedDictionary`, implémentant des dictionnaires à accès séquentiel ayant une capacité, i.e. créés avec une taille donnée et incrémentés d'une taille donnée par l'utilisateur à chaque grossissement.
2. Intéressons nous maintenant à la classe "SortedDictionary". Les dictionnaires ordonnés que vous avez réalisés (classe `OrderedDictionary`) sont ordonnés, en interne, selon l'ordre d'insertion des éléments. Les dictionnaires triés, à définir ici, via la classe `SortedDictionary` sont triés selon un ordre défini par le type des clés. Par exemple si les clés sont des *String*, l'ordre est lexicographique (alphabétique).
3. On souhaite implanter une recherche séquentielle des éléments. ¹ Proposez le schéma UML de la solution la plus économique possible, du point de la quantité de code écrit, pour implanter `SortedDictionary`.

¹Notez qu'il n'est pas stupide d'avoir un dictionnaire trié avec une recherche séquentielle, cela permet par exemple d'imprimer le dictionnaire directement, les couples clés-valeurs apparaîtrons dans l'ordre alphabétique.

4. Même question que la précédente si l'on souhaitait, dans une solution différente, implanter une recherche dichotomique.
5. Pour la mise en oeuvre on implémente la solution 1, il est nécessaire de comparer les clés entre elles. On utilisera pour cela le type `Comparable`. Un objet de type "comparable" est un objet auquel on peut envoyer le message `compareTo(Object another)`.

```
public int compareTo(Object o)
```

Compares this object with the specified object for order. Returns a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.

6. Sur cette base, et dans le contexte de la solution 1, que pensez vous d'une solution dans laquelle :

- (a) le tableau des clés déjà déclaré comme tableau d'objets² dans `AbstractDictionary` serait spécialisé en tableau de "Comparable" dans `SortedDictionary`,
- (b) la méthode `newIndexOf(Object key)` de `AbstractDictionary` serait spécialisée en `newIndexOf(C key)` sur `SortedDictionary`?

Cette solution fonctionnerait-elle en Java?

7. Donnez puis implantez et testez en TP une version utilisable en Java de la classe `SortedDictionary` dans la version No 1 de la solution.

²Tous ceux qui ont en fait déclaré un tableau de `String` dans `AbstractDictionary` seront confrontés à une variante du problème.