

TP13 : Pour ceux qui ont fini le TP3 sur les streams

Marianne Simonot

Donner notre propre implémentation de SortedSet

Dans ce TP, vous allez prendre le rôle des concepteurs de Java : vous allez donner votre propre implémentation de l'interface `SortedSet`. L'implémentation connue est `TreeSet` qui utilise des arbres pour stocker efficacement des éléments ordonnés. Comme vous avez programmé en algo les ABR, vous devrez utiliser un ABR pour stocker les éléments du `sortedSet`.

On simplifie un peu le problème en implémentant `SortedSet<Integer>`. Autrement dit, votre classe aura l'entête suivant :

```
public class MonTreeSetInt implements SortedSet<Integer> {
```

- Vous ne coderez pas toutes les méthodes demandées. On veut au moins : `size()`, `isEmpty()`, `iterator()`, `contains()`, `add()`, `remove()`, `toString()`, `findFirst()` et `findLast()`.
- Vous prendrez garde à respecter la spécification d'un `SortedSet` : c'est une structure où il n'y a pas de répétition et où les éléments sont ordonnés. Vous écrirez les tests Junit permettant de vérifier la conformité par rapport à cette spécification.
- Comme on s'intéresse aux `SortedSet<Integer>`, l'ordre sera l'ordre naturel sur les entiers. On ne fournira pas de constructeur permettant de choisir le comparateur.
- Dans un second temps, vous pouvez chercher à rendre votre classe générique.