ISN
TP2 - Arbres

Nous allons dans cet exercice nous familiariser avec la structure d'Arbre Binaire de Recherche.

1 Opérations sur les arbres binaires de recherche

Prenez sur le wiki le fichier *arbres.jvs* et ouvrez le avec JavaScool. Ce fichier est composé de trois classes. La première, `Arbre`, contient une implémentation d'un arbre binaire de recherche pouvant contenir des valeurs entieres. Cette implémentation utilise une classe `Noeud`. La troisième classe, `Afficheur` sert a afficher dans une fenêtre graphique un arbre construit.
1) Compilez et exécutez le projet. Admirez l'affichage!
2) Commencez par rajouter à l'objet Noeud un champs parent qui sera mis à jour lorsqu'il est ajouté dans un arbre. Ce champs va nous permettre de retrouver facilement le parent d'un noeud.
3) Écrivez une méthode *Noeud successeur(Noeud n)* qui retourne le noeud successeur du noeud n. L'algorithme permettant de trouver le successeur est :

```
successeur(noeud x) {
  si x.droit ≠ NILL
  alors retourner minimum(x.droit)
  y ← x.parent
 tant que y ≠ NILL et x = y.droit
      x ← y
      y ← y.parent

retourner y
```

4) Implémentez la méthode supprimer dans la classe Arbre et vérifiez son bon fonctionnement en exécutant le main d'Arbre.

2 Arbres équilibrés

Écrire sa propre implémentation d'arbres équilibrés peut se révéler difficile, nous allons donc utiliser celle fournit en standard dans Java : les `TreeMap`. Attention, vous ne pourrez pas utiliser

l'afficheur de l'exercice précédent pour visualiser votre arbre, il faudra prévoir une méthode "à la main".
5) La documentation indique qu'il s'agit de *Red-Black tree*, expliquez brièvement le principe.
6) Vérifiez expérimentalement qu'il s'agit d'arbres équilibrés.