***TD Algorithmique et Structure de Données***

**Arbre de scores**

Pour le traitement des notes d’un concours des écoles, on utilise un *arbre binaire de recherche* contenant dans chaque nœud le numéro (nombre entier), le nom (chaîne de caractères) et la note (nombre double) d’un élève.

Les élèves y sont rangés par ordre de leur numéro. Ainsi, le numéro d’élève dans un nœud est supérieur à ceux des nœuds de son sous-arbre gauche et inférieur à ceux des nœuds de son sous‑arbre droit. Les numéros sont uniques : il n’y a pas deux élèves avec un même numéro.

Exemple :

number name score

220031 Thomas 12,5

*root*

number name score

220836 Gautier 15

number name score

220029 Nicolas 15,5

number name score

220032 Lemaire 14,5

number name score

219446 Martin 10,5

number name score

220030 Richard 18

number name score

222386 Perrin 15

number name score

219447 Roussel 19

1. Donnez la définition en Java de la classe TNode représentant un nœud de l’arbre.
2. Définissez ensuite la classe ScoreTree représentant l’arbre qui a un seul champ root donnant accès à la racine de l’arbre et qui est initialisé à null par le constructeur et donnez le code des méthodes suivantes :
3. public void listing(double score)qui affiche le numéro et le nom de tous les élèves ayant obtenu le score score
4. public double scoreOf(int num) qui renvoie la note de l’élève numéro num et qui déclenche un NoSuchElementException si aucun élève n’a ce numéro
5. public void add(int num, String name, double score)qui ajoute un nœud contenant ces données à l’arbre en conservant l’ordre si aucun élève n’a le numéro num et qui déclenche un IllegalStateException dans le cas contraire
6. public double bestScore() qui renvoie le score maximale de l’arbre et qui déclenche un NoSuchElementException si l’arbre est vide
7. public void bestStudents()qui affiche le numéro et le nom de tous les élèves ayant obtenu le score maximale
8. private static void successList(TNode r, List<Integer> list, double score) qui ajoute à la liste list le numéro de tous les élèves de l’arbre de racine r ayant un score supérieur ou égale à score
9. public List<Integer> successList(double score) qui renvoie une liste contenant les élèves de l’arbre qui ont un score supérieur ou égal à score
10. public void remove(int num) qui supprime l’élève de numéro num de l’arbre et qui déclenche un NoSuchElementException si aucun élève n’a ce numéro.