

TD9/TP9 Arbre Générique

Pour le TP, recopier le répertoire : <http://www.lirmm.fr/~chaumont/download/cours/structuresdedonnees/tp9/>

I. Implémentation de la classe Arbre (en respectant l'interface donnée)

L'interface Arbre (ci-dessous) vous est donnée.

```
public interface Arbre<Element> {  
  
    public Element getElement ();           // Retourne la racine  
    public int getDegree ();                // Retourne le degré du « nœud » courant  
    public Arbre<Element> getNode(int i);   // Retourne le sous-arbre i  
    public boolean isEmpty ();              // Retourne vrai si l'arbre est vide  
    public boolean isLeaf ();               // Retourne vrai si le nœud est une feuille  
  
    public void setElement(Element e);       // Positionne la valeur courante du nœud racine  
    public void addNode(Arbre<Element> a);  // Ajoute un sous-arbre fils (utilisation du constructeur par clonage)  
    public void deleteNode(int i);           // Supprime le ième nœud fils  
  
    public String toString();               // Affichage des Elements  
    public boolean equals(Arbre<Element> a); // Test d'égalité  
    public int getHeight();                 // Retourne la hauteur de l'arbre  
}
```

Vous devez « implémenter » l'interface **Arbre**. Donnez-lui par exemple le nom **ArbreGene**.

```
import java.util.*; // Pour pouvoir utiliser les collections...

public class ArbreGene<Element> implements Arbre<Element> {

    private Element element;
    private LinkedList<ArbreGene<Element>> fils;

    public ArbreGene() {
        // à compléter
    }

    public ArbreGene(Element e) {
        // à compléter
    }

    public ArbreGene(Arbre<Element> a) { // A partir d'un Arbre (implémentation quelconque) on crée un ArbreGene
        // à compléter
    }

    // à compléter

} //Fin de la classe ArbreGene
```

II. En tp...

- Finir d'implémenter (et de tester) l'interface **Arbre**,
- La méthode `toString()` vue en td implémente un parcours en profondeur de l'arbre avec utilisation de la pile de récursivité. Essayer d'implémenter la méthode `toString()` de sorte que le parcours soit effectué en profondeur mais cette fois-ci à travers l'utilisation d'une pile !
- Essayer d'implémenter la méthode `toString()` de sorte que le parcours soit effectué en largeur. Le parcours en largeur consiste à traverser l'arbre par strates de profondeurs égales. Pour cela, on utilise une file...