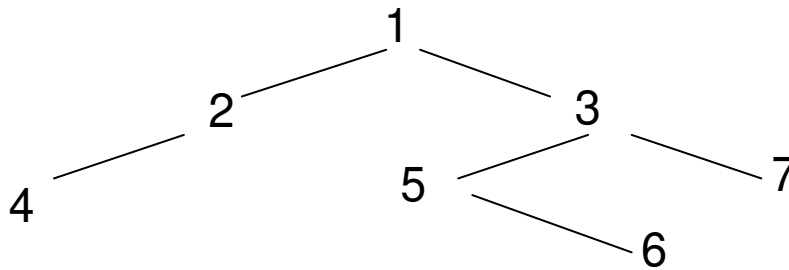


Structures de données avancées TP1 - Arbres binaires

Manipulation de types génériques, structures et méthodes récursives pour les Arbres Binaires.
Tout comme la collection `ArrayList<E>`, nous allons implémenter une collection structurée par un arbre binaire `AB<E>`

1. Un arbre binaire d'objets est une collection d'éléments de type `E`, structurée sous forme d'arbre. Implémenter la classe arbre binaire `AB<E>` en Java qui se décrit par
 - un élément de type `E`,
 - un fils gauche qui est un arbre binaire
 - un fils droit qui est lui aussi un arbre binaire

Test : Construire l'arbre suivant, où les éléments `E` ici sont des entier (Integer en Java) :



2. Implémenter une classe `Disque` qui dispose
 - d'un constructeur
 - des attributs : titre, auteur, nbEcoutés
 - des méthodes d'accès usuelles
 - de la méthode `lire()` qui incrémente le nombre d'écoutes.
 - une surcharge de la méthode `toString()` qui retourne le titre et le nombre d'écoutes

Test : Créer un arbre binaire de quelques disques de type `AB<Disque>`.

3. Implémenter la méthode d'affichage préfixé pour les arbres binaires. La méthode `affichagePrefixe()` d'un `AB<E>` affiche l'élément de l'arbre puis affiche le fils gauche, puis le fils droit. Le résultat sur l'arbre précédent serait 1243567. (pensez aux appels récursifs).
Test : Afficher l'arbre binaire d'entiers et celui de disques créés précédemment.
4. Implémenter la classe `DisquePrefere` étendant la classe `Disque` et implémentant l'interface `Comparable` de tel sorte que la méthode `compareTo(..)` compare deux disques préférés en fonction du nombre d'écoutes. Soient `d1` et `d2` deux instances de `DisquePrefere`, `d1.compareTo(d2)` retourne respectivement un entier négatif, zéro, un entier positif si le nombre d'écoutes de `d1` est respectivement inférieur, égal ou supérieur à `d2`. (cf Java API)
Test : Créer quelques disques préférés, lisez certains et comparer les.
5. Implémenter la méthode de recherche profondeur d'abord DFS d'un élément dans un arbre. La méthode `rechercheDFS(E elmt)` retourne vrai si l'élément `elmt` est égal (par la méthode `compareTo`) à l'élément de l'arbre binaire sinon elle recherche si l'élément est égal à un des éléments contenus dans son fils gauche puis dans son fils droit.(pensez aux appels récursifs)
Test : vérifier la méthode de recherche sur un arbre binaire d'entiers et de disques préférés.