

Licence Informatique 3^{ème} année

POO & Java

TP n°2 – *Abstraction*

1- Arbres

On veut écrire un programme pour gérer des arbres où chaque noeud contient un objet qui sera soit un entier, soit une chaîne de caractères (String). On veut pouvoir représenter des arbres de n'importe quel type mais tout arbre doit cependant offrir des méthodes pour :

- ajouter un objet comme noeud de l'arbre (l'endroit où on l'ajoute dépend du type d'arbre)
- enlever un objet de l'arbre
- obtenir la hauteur de l'arbre
- déterminer si un objet donné est présent ou pas dans l'arbre

NB : La façon dont ces méthodes vont être implémentées dépend du type d'arbre. Par exemple, pour trouver un objet dans un arbre, on devra faire un parcours exhaustif dans le cas d'un arbre général, alors qu'un parcours dichotomique suffira pour un arbre binaire de recherche.

1.1 Écrire du code objet pour représenter les arbres d'objets en général.

1.2 Écrire du code objet pour représenter les arbres binaires, où chaque noeud a au plus 2 fils. Ce code doit implémenter le calcul de la hauteur d'un arbre binaire.

1.3 Écrire du code objet pour représenter les arbres binaires de recherche, où sur chaque noeud, le fils gauche a une valeur inférieure ou égale à celle du noeud, et le fils droit a une valeur strictement supérieure à celle du noeud. Ce code doit implémenter l'ajout d'un objet comme noeud de l'arbre et la recherche d'un objet donné dans l'arbre. On peut considérer que la méthode de suppression d'un entier est également implémentée à ce niveau mais il n'est pas demandé d'écrire son corps.

Optionnels :

- Écrire du code objet pour représenter les arbres n-aires, où chaque noeud a un nombre quelconque de fils. Ce code doit implémenter le calcul de la hauteur d'un arbre n-aire.
- Adaptez votre programme pour pouvoir charger le dictionnaire « dictionnaire.fich » dans votre arbre binaire de recherche et testez la vitesse d'exécution de la recherche d'un mot dans l'arbre.
- Tout ce qui vous passe par la tête et vous apprend à programmer en POO ...

Le travail peut s'effectuer en binôme. Chaque binôme **remettra au plus tard le vendredi 13 janvier à 23h59** un dossier comprenant : un manuel utilisateur, un diagramme des classes, un bilan du projet, les listings et la documentation (javadoc) des classes. De plus chaque binôme enverra un fichier zip contenant les sources (.java) ainsi que tous les fichiers objets (.class) permettant de tester l'implémentation (sans devoir compiler).