Université Denis Diderot **Types de données et objets**
L1 S2 **2009–2010**

Examen du 19-Mai 2010. Durée 3 heures. Le barême est indicatif.
Il sera tenu compte de la qualité et de la clarté de votre copie.

A- Listes chaînées (8 points).

On rappelle qu'une modélisation possible des *listes chaînées* consiste en deux classes, une classe `Liste` composée d'un champ *nœud* nommé *début* et d'un champ *nœud* nommé *fin*. La classe Nœud contient elle un champ pour une *valeur* et un champ *suivant*.

1/ Modélisation.

a) Ecrivez la déclaration des classes Liste et Nœud dans le cas des listes d'entiers. En particulier, le champ *suivant* est-il un Noeud ou une Liste ? Justifiez rigoureusement votre choix.

b) Dessinez la représentation d'une liste ℓ_1 à 3 éléments $10, 20, 30$, et d'une autre liste ℓ_2 à 2 éléments $40, 50$, en distinguant clairement dans votre légende les différentes classes.

c) Une liste ℓ_3 est obtenue en partageant les nœuds des valeurs 20 et 30 utilisées dans la construction de la liste ℓ_1. Montrez comment obtenir une telle liste. Vous devrez pour cela écrire des constructeurs de listes et de noeuds, et eventuellement d'autres méthodes, puis dans une méthode *main* le code java qui définit explicitement les listes ℓ_1 et ℓ_3.

2/ Conversion binaire.

d) On s'interesse à l'opération consistant à convertir un entier naturel n en sa représentation binaire :

$$n = b_0.2^0 + b_1.2^1 + \cdots + b_k.2^k\,,$$

Ecrire une méthode **nTObin** qui renvoie la *liste* $b_0, b_1, b_2, \cdots b_k$ de la conversion binaire d'un nombre n donné en entrée. Par exemple, pour $n = 15$ la liste renvoyée doit être

1 1 0 1

car $11 = 1 + 2 + 8$.

e) De même, écrire une méthode **binTOn** qui à partir d'une *liste binaire* $b_0, b_1, b_2, \cdots b_k$ renvoie l'entier correspondant. Par exemple à partir de la liste 0 1 0 1 **binTOn** doit renvoyer 10.

B- Arbres binaires (3 points).

f) Donner une définition récursive d'un arbre binaire d'entiers.

g) Ecrire en Java, les classes Arbre et Nœud correspondant à la définition.

h) Ecrire une méthode de parcours de votre arbre. Décrire ce que fait exactement votre méthode en s'appuyant sur un exemple d'arbre (un schéma est fortement suggéré).

i) Ecrire une méthode de recherche d'un entier x dans un arbre. Quelle est la complexité au pire des cas de cette méthode ? Justifiez votre réponse en exhibant le pire des cas.

C- Piles (3 points).

j) Donner (avec le maximum d'explications commentées) le type **abstrait** des piles (rappel : il doit comporter les 3 fonctions `estVide`, `dépiler` et `empiler` ainsi que des commentaires sur ces fonctionnalités).

k) Ecrire une interface Pile implémentée dans une classe MaPile. En particulier, les mots clefs *interface* et *implements* doivent être présents et expliqués dans votre code.

D- Arbres, récursion et complexité (6 points).

l) Dessiner les **neuf** arbres binaires avec 0, 1, 2, et 3 sommets.

m) Expliquer comment obtenir *tous* les arbres binaires avec n ($n \geq 1$) sommets à partir des arbres binaires avec k sommets et $n-1-k$ sommets. En particulier, donner toutes les valeurs possibles de k.

n) En déduire une fonction récursive **touslesArbres** construisant tous les arbres binaires avec n sommets. Expliciter les *conditions d'arrêt* et le *corps de la récursion* de votre fonction.

1. Soit T_n la complexité de la fonction **touslesArbres** de la question précédente. Sans la résoudre, donner une équation récursive satisfaite par T_n.