Utilisation de classes existantes

1) On voudrait maintenant que le tableau, contenant les données de type InfoContact, ne figure plus dans la classe InfoContact, mais qu'il devienne un Objet de type Agenda.
Créer donc une nouvelle classe Agenda, avec un (ou plusieurs) constructeur(s), dans laquelle on placera désormais les méthodes ajouter, chercher, et AfficheTout. Attention : dans la classe Agenda, ces méthodes deviennent des méthodes d'instances.
Tout les agendas crées auront une taille fixe. Pour indiquer que la taille est une constante, on pourra qualifier la variable taille de final, en plus de static.

2) Modifier la méthode ajouter de la classe Agenda pour que les données (de type InfoContact) d'un agenda soient classées en ordre lexicographique par rapport à leur nom (consulter les méthodes de comparaison de la classe String).
Modifier en consequence la méthode chercher de la classe Agenda.

3) Dans une nouvelle version de la classe Agenda, on voudrait maintenant transformer le tableau, qui contenait jusqu'à présent un nombre fixé de données de type InfoContact, en un "vecteur" de taille variable selon le nombre de données à stocker. Pour cela, on utilisera la classe Vector du package java.util et on utilisera les méthodes proposées dans cette classe.

4) Copier le fichier carrere/MIAGE/IPI/agenda.obj dans votre répertoire.
Consulter dans l'API sur jaguar (à l'adresse : http ://opt/doc/lang/java/jdk1.1.6/api/packages.html), les classes java.io.ObjectInputStream et java.io.FileInputStream. Le fichier agenda.obj contient des objets de la classe InfoContact. En utilisant les méthodes de la classe java.io.ObjectInputStream, il est possible d'extraire un à un, les objets de ce fichier pour les écrire dans le tableau d'un Agenda.
Les objets contenus dans le fichier agenda.obj sont des objets "sérialisables", c'est-à-dire appartenant à une classe, qui imlémente l'interface Serializable. Vous devez donc indiquer dans la classe InfoContact, qu'elle imlémente l'interface Serializable. D'autre part il faudra préciser, pour chaque objet récupéré, qu'il est de type InfoContact, par un "cast" (cela consiste à faire précédé le nom de cet objet par le type InfoContact entre parenthéses).

Ajouter dans la classe Agenda un constructeur Agenda(String NomFichier), qui constuira un agenda contenant toutes les données présentes dans le fichier de nom NomFichier. On utilisera pour cela les méthodes de la classe java.io.ObjectInputStream. Attention : la plupart de ces méthodes peuvent "lever" une exception, c'est-à-dire signaler une erreur, par exemple une erreur du type IOException. Vous devez alors rajouter les mots : throws IOException (ou tout autre type d'exception), juste avant l'accolade ouvrante du code de chacune de vos fonctions, qui utilisent des méthodes levant des exceptions. On testera ensuite ce constructeur avec en argument le nom de fichier agenda.obj.