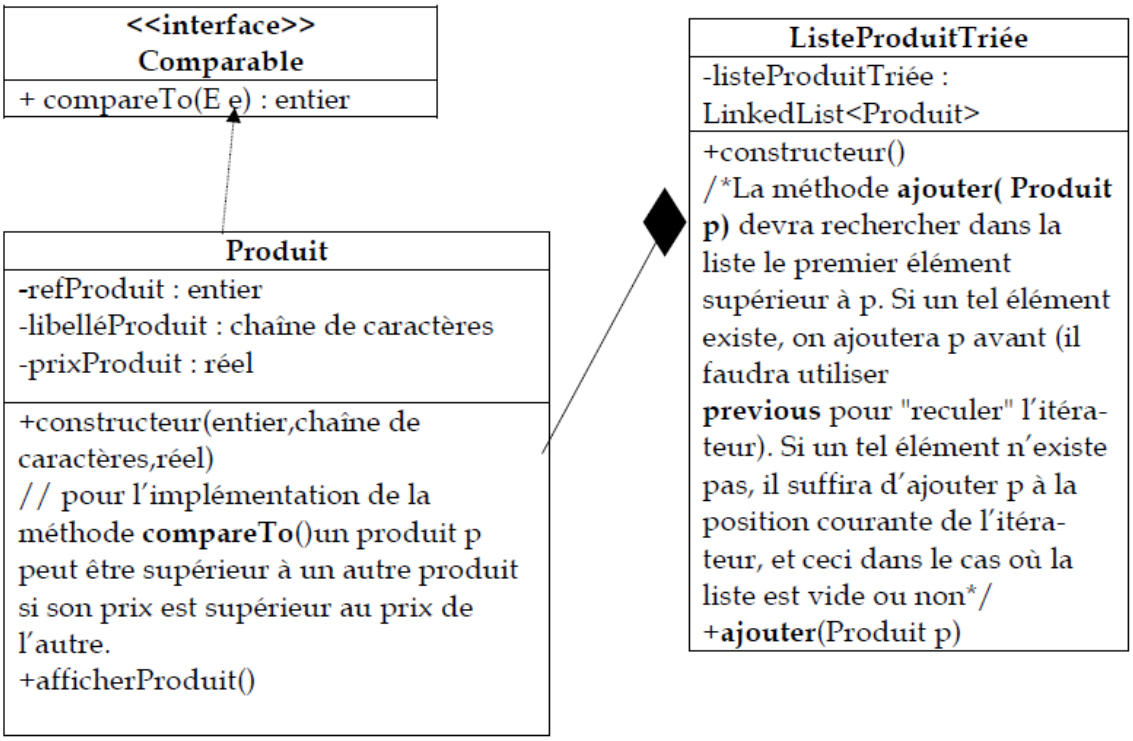
TP1 Les collections

Exercice1 (LinkedList)

Soit le diagramme de classes suivant :



Créer les différentes classes en utilisant les LinkedList

Exercice2 (HashMap)

On souhaite implémenter un annuaire électronique, qui donne pour chaque nom de contact, ses coordonnées (N°tel & Adresse).

1) Définir la classe **Coordonnees**, composée des 2 attributs ***Tel*** et ***Adr*** et implémentant un constructeur par défaut ainsi que des setters pour ses attributs.

2) Définir la classe **Annuaire** qui gère l'annuaire électronique à l'aide d'une collection de type **HashMap**. Cette collection associe à chaque nom, un objet de type **Coordonnees**. La classe **Annuaire** implémente les méthodes suivantes :

- **Ajout (String N, Coordonnees C)** : Ajout d'un nouveau contact
- **AffichCoord (String N):** Affichage des coordonnées d'un contact donné.
- **ModifCoord (String N):** Modification des coordonnées d'un contact donné.
- **Suppression (String N):** Suppression d'un contact donné.
- **ListerNoms ():** Affichage de la liste des contacts (tous les noms).

- **ListerTel ():**Affichage de la liste des N°de Tel disponibles dans l'annuaire.

- **ListerAdr ():**Affichage de la liste des Adresses disponibles dans l'annuaire.

- **AffichAnnuaire ():**Affichage de la totalité de l'annuaire.

3) Ecrire un programme principal, présentant un menu pour la gestion d'un annuaire électronique.

Exercice3 (HashSet)

Soit la classe Patient suivante avec une liste de médicaments sans doublons. Créer cette classe selon le diagramme de classes suivant :

Patient
- **nom** : String - **listeMedicaments** :**set** de String

+ **Patient**(String)
+ **getNom**():String
+ **ordonnanceVide**(): boolean
// ajoute un medicament à la liste des médicaments
+ **ajoutMedicament** (String)
// Teste si la liste des médicaments contient un médicament
+ **contientMedicament** (String) : boolean
//affiche l'ordonnance du patient
+ **afficheOrdonnance** ()
// affiche le nom du patient ainsi que la liste des médicaments de son ordonnance
+**affichePatient**()

Exercice4 (HashMap+TreeSet)

Une pharmacie est caractérisée avec un nom et un ensemble de patientsclients qui est représenté par une **table d'associations** entre noms de patients (String) et objets Patient. Il faut noter que les **noms** des patients doivent être **tous différents**, sans distinction entre minuscules et majuscules. Et pour ce faire avant de faire il faut faire la conversion en minuscules du non du patient (méthode ajoutPatient).

Créer la classe **Pharmacie** selon le diagramme de classes suivant :

Pharmacie
- **nom_Pharmacie** : String - **patients_Clients** :**HashMap**<String,Patient>

+ **Pharmacie** (String)
// ajoute un patient de nom n avec sa liste de médicaments dans
//patients_Clients
+**ajoutPatient**(String n,String[] listeMedicaments)
//Ajoute un nouveau médicament sur un patient déjà existant et renvoie false si le patient n'existe pas et true si l'ajout a été fait avec succès.
+ **ajoutMedicament** (String nomPatient, String med)
//affiche un patient de nom np à partir des patientsClients
+ **affichePatient** (String np)
// affiche tous les patients de la Pharmacie
+**affiche**()
// retourne une collection des noms des patients qui ont pris un médicament donné
// ces noms doivent être triés par ordre croissant
+**PatientAvecMédicament**(String med) : TreeSet<String>
// Méthode qui parcourt la Map et qui enlève tous les patients qui ont une //ordonnance vide
+**enleverOrdonnanceVide**()