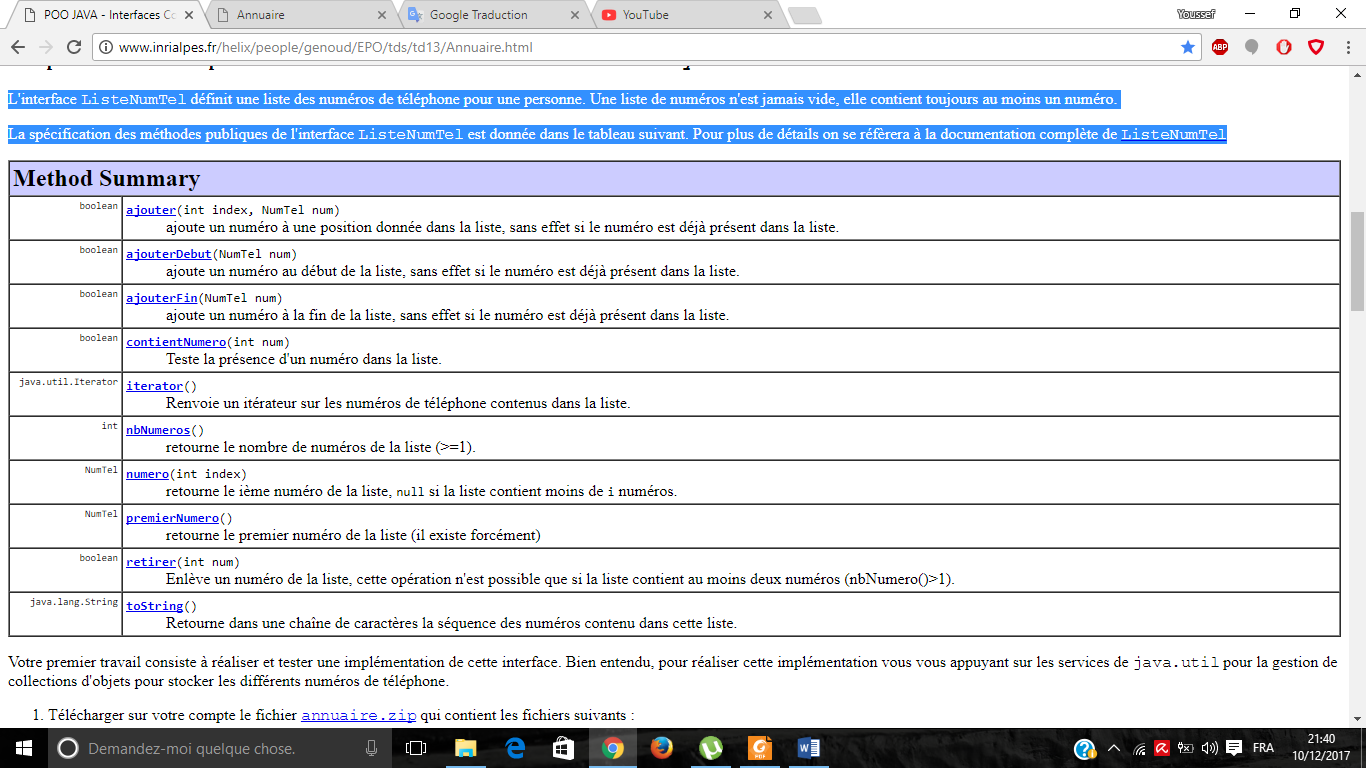
**TP : Les collections en Java**

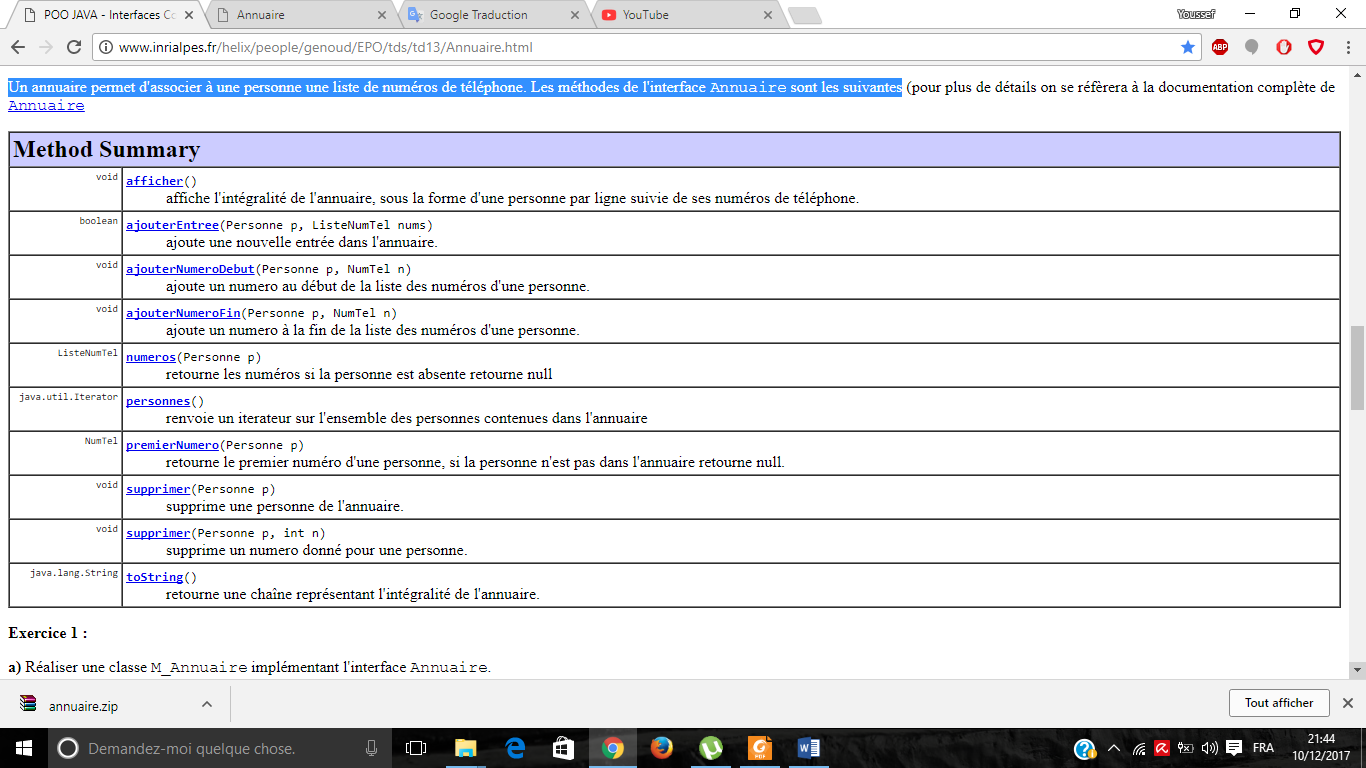
On souhaite réaliser une application de gestion d'un annuaire téléphonique qui permet d'associer à une personne un ou plusieurs numéros de téléphone. Cette application s'articulera autour des classes et interfaces suivantes :

* **Personne**: classe représentant une personne définie par un nom, un prénom et une civilité (M., Mlle, Mme).
* **NumTel** : classe représentant un numéro de téléphone, défini par le numéro proprement dit et un code qui indique la nature du numéro (numéro de portable, numéro de poste fixe professionnel, numéro de poste fixe à domicile, numéro de Fax).
* **ListeNumTel**: interface définissant les fonctionnalités d'une liste de numéros de téléphone.
* **Annuaire**: interface définissant les fonctionnalité d'un annuaire téléphonique, c'est à dire d'un ensemble d'associations <p,ln> avec p une Personne et ln une ListeNumTel.

L'interface ListeNumTel définit une liste des numéros de téléphone pour une personne. Une liste de numéros n'est jamais vide, elle contient toujours au moins un numéro. La spécification des méthodes publiques de l'interface ListeNumTel est donnée dans le tableau suivant.



Un annuaire permet d'associer à une personne une liste de numéros de téléphone. Les méthodes de l'interface Annuaire sont les suivantes :



**Exercice 1:**

**a)** Réaliser une classe M\_Annuaire implémentant l'interface Annuaire.

On utilisera comme structure de données pour l'annuaire une table associative (interface  **Map**) : la personne jouant le rôle de clé et la liste des numéros de téléphone étant la valeur associée à cette clé (on prendra une HashMap comme implémentation). Dans un premier temps on se contentera d'écrire une implémentation complète uniquement pour les méthodes ajouterEntree(Personne, ListeNumTel),personnes(), afficher() et numeros() de la classe Annuaire.

**b)** Tester ces méthodes avec un petit programme principal qui :

* crée un annuaire vide,
* ajoute une entrée pour une personne de nom DURAND, prénom Sophie, Civilité Mlle. avec comme numéro 151171
* ajoute une entrée pour une personne de nom DUPONT, prénom Jean, Civilité M. avec comme numéro 151171
* ajoute une entrée pour une personne de nom DUSCHMOL, prénom Louis, Civilité M. avec comme numéro 146761
* ajoute une entrée pour une personne de nom AARGH, prénom Robert, Civilité M. avec comme numéro 140361
* affiche le contenu de l'annuaire.
* effectue une recherche des numéros de la personne Sophie DURAND et une recherche des numéros de Jean DUPONT

Constatez que la recherche du numéro pour Jean DUPONT a échoué alors qu'elle a réussi pour Sophie DURAND. Quelle explication donnez vous à cela ?

**c)** Modifiez la classe Personne de manière à corriger les erreurs détectées précédemment.

**Exercice 2 :**

On voudrait que l'itérateur fournissant les différentes personnes enregistrées dans l'annuaire permette d'obtenir celles-ci selon l'ordre alphabétique, ce qui n'est pas garanti par une HashMap. Pour cela il vaudrait mieux utiliser comme implémentation de l'annuaire un objet instance de la classe TreeMap.

1. Dans votre implémentation de l'annuaire remplacez HashMap par TreeMap
2. Recompilez votre application et exécutez. Quel problème cela pose-t'il ? Quelle doit être la propriété de la classe **Personne**?
3. Faites les modifications nécessaires, refaites le test précédent et constatez les modifications lors de l'affichage de l'annuaire.

**Exercice 3 :**

Ecrivez l'implémentation complète de la classe Annuaire et un programme de test de cette classe.

**Exercice 4 :**

On veut ajouter la méthode suivante à l’annuaire :

/\*\*

\* donne l'ensemble de toutes les personnes de l'annuaire dont le nom   
 \* débute par une chaîne donnée.  
 \* @param s1 la chaine pour la recherche  
 \* @return l'ensemble des personnes de l'annuaire dont le nom débute  
 \* par s1  
 \*/  
 public Set entreesPourChaine(String s1)

Réaliser cette évolution et tester cette nouvelle méthode. (Indication : avant d'écrire ce code examinez attentivement toutes les méthodes de TreeMap).