

TP autour des Collections

On souhaite réaliser une application de gestion d'un annuaire téléphonique permettant d'associer à une personne un ou plusieurs numéros de téléphone.

On va donc travailler avec deux classes `Personne` et `NumTel` qui seront fournies en TP et dont la documentation est donnée en annexe.

- **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).

1. Créer une classe **ListeNumTel** permettant de gérer une liste de numéros de téléphone.

Proposer un constructeur sachant qu'une liste doit toujours contenir au moins un numéro de téléphone ainsi que les méthodes suivantes:

- `boolean ajouter(int index, NumTel num)`

Ajoute un numéro à une position donnée dans la liste, sans effet si le numéro est déjà présent dans la liste.

- `boolean ajouterFin(NumTel num)`

Ajoute un numéro à la fin de la liste, sans effet si le numéro est déjà présent dans la liste.

- `boolean contientNumero(int num)`

Teste la présence d'un numéro dans la liste.

- `java.util.Iterator<NumTel> iterator()`

Renvoie un itérateur sur les numéros de téléphone contenus dans la liste.

- `int nbNumeros()`

Retourne le nombre de numéros de la liste (≥ 1).

- `NumTel numero(int index)`

Retourne le ième numéro de la liste.

- `NumTel premierNumero()`

Retourne le premier numéro de la liste (il existe forcément)

- `boolean retirer(int num)`

Enlève un numéro de la liste. Si le numéro existe, retourne `true` sinon `false`.

Cette opération n'est possible que si la liste contient au moins deux numéros (`nbNumeros() > 1`).

- `java.lang.String toString()`

Retourne dans une chaîne de caractères la séquence des numéros contenu dans cette liste.

2. Créer une classe **Annuaire** permettant d'associer à une personne une liste de numéros de téléphone. Chaque personne ne doit se trouver qu'une seule fois dans l'annuaire.

Cette classe proposera un constructeur sans paramètre et les méthodes suivantes :

- `boolean ajouterEntree(Personne p, ListeNumTel nums)` : ajoute une nouvelle entrée dans l'annuaire.

Si `p` n'existe pas, on crée une nouvelle association (`p,nums`) et le booléen `true` est retourné; sinon le booléen `false` est retourné et la méthode est sans effet.

- `ListeNumTel numeros (Personne p) : retourne la liste des numéros d'une personne. Si la personne est absente retourne null.`

Quelle méthode doit-on ajouter à la classe `Personne` pour effectuer cette recherche ?
Proposer une implantation.

- `java.util.Iterator <Personne> personnes () : retourne un itérateur sur l'ensemble des personnes contenues dans l'annuaire.`
- `void ajouterNumeroFin (Personne p, NumTel n) : ajoute un numéro à la fin de la liste des numéros d'une personne.`

Si la personne n'existe pas, on crée une nouvelle entrée pour cette personne avec comme liste de numéros associée la liste constituée du numéro passé en paramètre.

- `public NumTel premierNumero (Personne p) : retourne le premier numéro d'une personne si cette personne est présente dans l'annuaire, sinon retourne null.`
- `public void supprimer (Personne p) : supprime une personne dans l'annuaire si celle-ci est présente.`
- `public void supprimerNumero (int n, Personne p) : supprime le numéro pour une personne donnée. Si le numéro est le seul numéro de la personne, retire la personne.`
- `public Set<Personne> lesEntreesPour (String nom) : retourne l'ensemble des personnes ayant le nom passé en paramètre.`
- `public String toString () : retourne une chaîne de caractères décrivant l'ensemble des personnes de l'annuaire. Chaque ligne contient les informations d'une seule personne.`

3. On souhaiterait consulter les personnes de l'annuaire dans l'ordre alphabétique.

Quelles modifications devrait-on apporter et dans quelles classes?

4. Proposer une classe `TestAnnuaire` avec une méthode `main` permettant d'insérer trois personnes dans l'annuaire, chacune avec une liste de numéros de téléphone associée. Parmi les trois personnes deux auront le même nom et même prénom.

Afficher le contenu de l'annuaire. Quel problème constate-t-on à l'affichage ? Compléter la classe `Personne` pour régler le problème.