

## TD/TP 2 - Entreprise

Programmation Java : Utilisation des collections et exceptions.

---

On poursuit le travail sur les employés en définissant une nouvelle classe **Entreprise**.

**En TP, les méthodes seront testées dans un main au fur et à mesure.**

Une entreprise est un rassemblement d'employés, elle doit donc contenir la liste de ses employés. Elle doit de plus permettre de rechercher un employé en fonction de son nom. De plus, elle a obligatoirement un patron qui est un cadre qui ne peut pas être renvoyé. Il lui faut donc l'interface suivante :

- public **void** ajouteEmploye(**Employe** e) qui permet d'ajouter un employé à la liste.
- public **void** affiche() qui affiche l'ensemble des employés en commençant par le patron.
- public **Employe** rechercheParNom(**String** nom) throws Exception pour retrouver un employé par son nom. Cette fonction doit générer une exception si elle ne trouve pas l'employé.

On supposera que chaque personne a un nom différent.

- Q.1)** - Décrire la classe entreprise, quelle structure de données faut-il pour stocker les employés ?
- Q.2)** - Écrire le constructeur de la classe.
- Q.3)** - Écrire la méthode public **void** ajouteEmploye(**Employe** e).
- Q.4)** - Écrire la méthode public **void** affiche().
- Q.5)** - Écrire la méthode public **Employe** rechercheParNom(**String** nom) throws Exception.

### Questions facultatives.

- Q.6)** - Compléter la méthode ajouteEmploye afin qu'elle génère une exception si l'employé est déjà dans cette entreprise.
- Q.7)** - Écrire une méthode public **void** retireEmploye(**String** nom) throws Exception qui permet de retirer un employé de l'entreprise. Cette fonction doit générer une exception s'il n'y a pas d'employé de ce nom dans l'entreprise. De plus, si l'on retire un cadre, l'ensemble de ses subordonnés directs et indirects doivent être également retirés de l'entreprise.
- Q.8)** - Dans la classe **Cadre**, pour stocker la liste des subordonnées changer la structure de données tableau en un tableau dynamique de type **Vector**. Modifier en conséquence les méthodes de la classe **Cadre**.
- Q.9)** - Compléter la méthode ajouteSubordonne de **Cadre** afin qu'elle génère une exception dans le cas où l'employé est déjà au service de ce cadre.