

### Objectifs :

- ✓ Manipulation des interfaces en java
- ✓ Gestion des collections ( List)

### Exercice 1 :

1. Définir une classe **Personne** avec les attributs suivants : *code, Nom et prénom*.
2. Définir les accesseurs aux différents attributs de la classe (Getters/ Setters).
3. Définir un **constructeur** permettant d'initialiser les attributs d'un objet **Personne** par des valeurs saisies par l'utilisateur.
4. Définir la méthode **toString ( )** permettant d'afficher les informations de la classe en cours.
5. Créer l'interface **IPerson** avec les méthodes suivantes :
  - void save ( Personne p ) : Méthode permettant d'ajouter un objet p de type **Personne**.
  - void delete (int code) : Méthode permettant de supprimer un objet p dont le code est passé en paramètre..
  - boolean update (Personne p) : Méthode permettant de modifier un objet p de type **Personne**.
  - Personne findById (int code) : Méthode permettant de renvoyer un objet dont le code est passé en paramètre.
  - List<Personne> findAll() : Méthode permettant de renvoyer la liste des personnes
6. Créer la classe « **PersonneService** » qui implémente l'interface **IPerson**. Dans cette classe les données seront stockées dans une collection de type **ArrayList**.
7. Dans une classe de test :
  - Créer cinq personnes.
  - Afficher la liste des personnes.
  - Supprimer une personne.
  - Modifier les informations d'une personne.
  - Rechercher une personne par son code.
  - Afficher à nouveau la liste des personnes

### Exercice 2 : LinkedList

1. Dans un code java, créer une liste « liste1 » de type « **LinkedList** » contenant des entiers
2. Insérer dans la liste un élément à une position spécifiée
3. Insérer des éléments dans la liste à la première et à la dernière position
4. A l'aide d'un itérateur de l'interface « **Iterator** » afficher tous les éléments de la liste
5. Afficher à nouveau les éléments de la liste mais cette fois en utilisant une boucle **for**
6. Afficher la première et la dernière position de la liste
7. Supprimer un élément donné de la liste.
8. Créer une autre liste « liste2 » de type **LinkedList**, puis copier à l'envers tous les éléments de la liste « liste1 » dans la liste « liste2 »