

TP N°5 en Java

Cycle ingénieur : 2^{ème} Année GI

Exercice 1 : List, ArrayList, LinkedList et Vector

1. Crée une liste de tableau (`ArrayList`) rempli avec des entiers
 - Insère de nouveaux éléments dans des endroits spécifiés dans la liste.
 - Itérer pour afficher les éléments de la liste, supprimer un élément et réafficher la liste
2. Crée une liste chaînée (`LinkedList`) à partir de la liste de tableau précédente
 - Insère des nouveaux éléments.
 - Supprime des éléments de la liste.
 - Parcourt la liste vers l'avant et vers l'arrière.
3. Faire les mêmes opérations avec la classe `Vector`.

Exercice 2 : Iterable et Iterator

Créer une classe `Paquet<T>` qui implémente l'interface `Iterable<T>` et permette le stockage d'une paquet d'objet de type `T` (variable).

On suppose que les paquets sont de taille inférieure à 50.

La classe dispose de deux attribues, un tableau de type variable `T` et un entier `size` indiquant la taille du paquet. Une méthode `add(T e)` pour ajouter un objet dans le paquet et une méthode `toString()` qui utilise la méthode `iterator()` pour retourner le paquet d'objets sous forme de chaine de caractère. Tester votre programme.

Exercice 3 : Comparable et Comparator, TreeSet, TreeMap

1. Créer la classe `Produit` ayant deux attribues désignation et prix et qui implémente l'interface `Comparable<Produit>`. La méthode `compareTo` compare les prix des produits. Tester votre programme en créant un ensemble ordonné de produit set de type `SortedSet`.
2. Si l'on souhaite ordonner les produits par leurs désignations que faut-il faire. Tester votre programme en créant un deuxième ensemble ordonné de produit set de type `SortedSet` (utiliser le constructeur `TreeSet(Compareur< ? super E> comp)`)

Exercice 4 - HashTable, TreeMap

Ecrire un programme qui compte les occurrences de mots dans un texte et affiche les mots et leurs occurrences dans l'ordre croissant des mots.

- Utilise une table de hachage (`Hashtable`) map pour stocker `<mot, nombre>`
- Pour chaque mot, vérifiez si elle est déjà une clé dans la carte.
 - Sinon, ajoutez la clé et la valeur 1 au map.
 - Sinon, augmentez la valeur du mot (clé) par 1 dans le map.
- utiliser `TreeMap` pour trier le map
- utiliser `TreeSet` pour trier le l'ensemble de clé, valeur.