

TP3 : Généricité, Collection et sttream.

Exercice 1 :

On souhaite créer une classe générique appelée StorageGenerique qui peut stocker des éléments de n'importe quel type. La classe devrait avoir un attribut nommé elements de type

List pour stocker les éléments. Ajouter à la classe StorageGenerique les méthodes suivante :

- `Public void addElement(T o)` : permettant d'ajouter un élément à la collection.
- `public T getElement(int index)` : permettant de récupérer un élément à une position donnée dans la collection.
- `public void removeElement(int index)` permettant de supprimer un élément spécifique de la collection.
- `public int getSize()` permettant d'obtenir la taille actuelle de la liste.

Créez une classe de test appelée App pour tester votre implémentation. Utilisez la classe StorageGenerique avec différents types (par exemple, Integer, String, Double). Effectuez des opérations d'ajout, de récupération, de suppression et de vérification sur la liste générique.

Exercice 2 :

1. Créer un map de type Hashmap qui stocke les notes des étudiants. La clé est de type String représentant le nom de l'étudiant, et la valeur est de type double représentant la note.
2. Insérer des notes des étudiants.
3. Augmenter la note d'un étudiant.
4. Supprimer la note d'un étudiant.
5. Afficher la taille du map.
6. Afficher la note moyenne, max et min.
7. Vérifier s'il y a une note égale à 20.

Remarque : Après chaque opération afficher la liste en utilisant la boucle forEach avec l'expression lambda.

Exercice 3: Les Sets

1. Créer deux Sets de type HashSet nommées groupeA et groupeB, contenant les nom des étudiants des deux groupes A et B.
2. Ajouter des noms des étudiants à chaque HashSet.
3. Afficher l'intersection des deux HashSets.
4. Afficher l'union des deux HashSets.

Exercice 4 : les streams

On suppose que nous avons une liste de mots et que nous souhaitons effectuer les opérations suivantes :

1. Filtrer les mots qui contiennent la lettre "a".
2. Filtrer les mots qui ont une longueur supérieure à 3 et transformer chaque mot en son inverse.
3. Filtrer les chaînes qui contiennent la lettre "e" et aplatir chaque chaîne en une liste de ses caractères.
4. Transformer chaque chaîne en sa version en majuscules.
5. Convertir chaque chaîne en sa longueur.
6. Transformer chaque chaîne en une liste de ses caractères, puis aplatir toutes les listes en une seule liste de caractères.
7. À partir d'une liste de mots, transformer chaque nom en une chaîne de la forme "**Nom-Index**" où l'index représente la position du nom dans la liste.

Exercice 5: les streams

Créer une liste de type **ArrayList** d'employés. Chaque **employé** ait un **nom** (String), un **département**(String) et un **salair**e (double). Vous devez effectuer les opérations suivantes :

1. Utiliser **mapToDouble** et **sum** pour calculer la somme totale des salaires de tous les employés.
2. Utiliser **sorted** pour trier la liste des employés par ordre alphabétique du nom.
3. Utiliser **min** pour trouver l'employé avec le salaire le plus bas.

4. Utiliser **filter** pour obtenir la liste des employés ayant un salaire supérieur à une valeur donnée.
5. Utilisez **reduce** pour trouver l'employé avec le salaire le plus élevé dans l'ensemble de l'entreprise.
6. Utilisez **reduce** pour concaténer les noms de tous les employés