

Programmation orientée objet Java

TP4 : Collections

Exercice 1

1. Écrire une classe `Entier` qui a un attribut `val` (de type entier) en ajoutant un constructeur, des getters et des setters.
2. Écrire un programme Java qui crée une `LinkedList` `L` d'objets `Entier` avec des valeurs aléatoires de l'intervalle `[0,100]`.
3. Écrire un programme qui affiche le maximum et le minimum des éléments de la liste.
4. Écrire un programme qui à l'aide d'un itérateur modifie la liste en multipliant par deux les entiers supérieurs à 5.
5. Comparer les collections `ArrayList`, `LinkedList` et `TreeSet` par rapport à la complexité en temps d'exécution de l'ajout, de la suppression et de la recherche dans une collection de `n` éléments.

Exercice 2

1. Créer une classe `Personne` avec deux attributs prénom et nom de type `String`. La classe contient un constructeur avec deux arguments, une méthode `affiche()` qui affiche le prénom et nom séparées par espace, des getters et setters.
2. Créer une classe `ListePersonnes` qui contient un attribut `personnes` de type `HashMap` avec clés de type `String` et valeurs de type `Personne`. La clé étant la concaténation du prénom et nom. La classe contient les méthodes suivantes :
 - a. `void ajouterPersonne(Personne p)` qui permet d'ajouter une personne au Map. Cette méthode lève une exception `ErrAjoutPersonne` (à créer) si la personne à ajouter existe déjà.
 - b. getters et setters
 - c. `void afficher()` qui affiche les éléments du Map.
3. Créer une classe `Test` avec une méthode `main` qui teste la création, l'ajout et l'affichage d'une liste de personnes. La méthode `main` doit gérer l'exception `ErrAjoutPersonne`.