



Développement WEB : JEE

TP 1 : Les Collections

Exercice 1 :

Interface Collection:

```
public interface Collection<E> extends Iterable<E> {
    // Basic operations
    int size();
    boolean isEmpty();
    boolean contains(Object element);
    boolean add(E element);
    boolean remove(Object element);
    Iterator<E> iterator();

    // Bulk operations
    boolean containsAll(Collection<?> c);
    boolean addAll(Collection<? extends E> c);
    boolean removeAll(Collection<?> c);
    boolean retainAll(Collection<?> c);
    void clear();

    // Array operations
    Object[] toArray();
    <T> T[] toArray(T[] a);
}

import java.util.ArrayList;
import java.util.Random;

public class Ex01 {
    public static void main(String args[]){
        Random gen = new Random();
        ArrayList<Integer> l = new ArrayList<Integer>();
        for (int i = 0; i < 20; i++){
            l.add(gen.nextInt(5));
        }
        for (int e : l)
            System.out.println(e);
    }
}
```

Explication :

ArrayList implémente Collection, ce qui fait que d'une part la méthode l.add, qui permet d'ajouter des éléments à la collection l, existe et d'autre part, on peut itérer sur le contenu de l avec la construction for (int e : l)

- Réécrivez le programme précédent en remplaçant ArrayList par les différentes implémentations de Collection (LinkedList, Set), en particulier par HashSet. Que constatez-vous?

Exercice 2 :

1. Définissez une classe Rationnel contenant entre autre les champs numérateur et dénominateur, et dont la méthode public String toString() retourne une chaîne de type “numérateur/dénominateur”.
2. Écrivez un programme qui demande à l'utilisateur d'entrer des rationnels, jusqu'à ce qu'il entre un dénominateur qui vaut 0. Le programme met les rationnels dans un ArrayList au fur et à mesure, et à la fin, affiche le contenu de cet ensemble.
3. Écrivez une classe CompareRationnels implements Comparator<Rationnel> permettant d'ordonner des nombres rationnels selon leur ordre habituel en mathématiques. (vous devez implémenter la méthode int compare(Rationnel r1, Rationnel r2) qui retourne -1 si $r1 < r2$, 0, si $r1 = r2$ et 1 si $r1 > r2$).
4. Utilisez la méthode Collections.sort ainsi qu'un comparateur instancié de CompareRationnels pour créer une liste ordonnée de rationnels de l'ensemble créé précédemment.

Exercice 3 (Tableau associatif) :

On vous demande d'écrire une classe **Annuaire** pour mémoriser des numéros de téléphone et d'adresses. Chaque entrée est représentée par une fiche à plusieurs champs: un nom, un numéro et une adresse. La structure des fiches est décrite par une classe **Fiche** que vous devez écrire.

Écrivez également une classe **Annuaire** comportant une table associative (**Map<String,Fiche>**) qui sera faite d'associations (*un_nom, une_fiche*).

Parcourez l'annuaire pour afficher l'ensemble des fiches en utilisant deux méthodes :

- Boucle for
- Boucle while + iterator