

Les Collections : Les Listes

Les listes (List)

Pour cet exercice, vous pouvez utiliser la documentation en ligne pour vous aider. Par exemple, concernant ArrayList : <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

a) Créer un programme Java qui crée une collection de pays (**Classe Country avec un attribut name**) puis alimenter cette collection avec quelques valeurs et afficher la taille de la collection

Exemple de résultat à obtenir :

```
La collection créée contient 4 pays !
```

b) Compléter le programme pour afficher le contenu de la collection.

Exemple de résultat à obtenir :

```
La collection créée contient 4 pays !  
France  
Allemagne  
Belgique  
Portugal
```

c) Trouver une méthode pour vider la collection et modifier votre programme afin d'afficher un message lorsqu'elle est vide et afficher le contenu lorsqu'elle n'est pas vide.

Exemple de résultat à obtenir :

```
Liste vide
```

d) Après avoir de nouveau alimenté votre liste de pays, modifiez le nom d'un pays et affichez de nouveau la liste des pays.

Remarque : Il y a plusieurs méthodes pour manipuler des éléments dans une collection de type ArrayList (ajouter, modifier ou supprimer). Choisissez celle que vous voulez dans la documentation Oracle.

e) Triez votre collection et réaffichez la liste des pays.

Dans la classe « ArrayList », une méthode « sort » est nécessaire pour trier notre collection.

- **Dans la documentation, trouvez-vous cette méthode ?**

Elle n'existe pas. En revanche, allez voir la documentation de la classe utilitaire « Collections ».

- **Désormais, trouvez-vous cette méthode « sort » ?**

Une « ArrayList » implémente l'interface Collection, elle est donc une collection. Nous pouvons donc trier par ordre alphabétique croissant ou décroissant une ArrayList avec la syntaxe suivante « **Collections.sort(uneArrayList)** » pour trier une ArrayList **de String**.

```
List<String> countries = new ArrayList<>();  
countries.add("FRANCE");  
countries.add("BELGIQUE");  
Collections.sort(countries);
```

Si nous souhaitons trier notre collection dans l'ordre inverse :

```
Collections.sort(countries, Collections.reverseOrder());
```

Tester le code ci-dessous sur une ArrayList de String, puis sur une ArrayList de Country.

- **Que se passe-t-il ?**

Pour utiliser la méthode « sort » sur une collection d'objets autre que String, il nous faut utiliser un comparateur.

Créer une classe **CountryComparator** qui implémente l'interface **Comparator** que vous utiliserez dans votre méthode « **sort** » afin de trier vos **Country** par ordre alphabétique croissant ou décroissant.

```
Collections.sort(countries, new CountryComparator());
```

f) Triez votre collection du plus petit mot au plus grand mot. Réaffichez ensuite la liste des pays.

Exemple de résultat à obtenir :

France
Belgique
Portugal
Allemagne

g) À l'aide de votre boucle sur le résultat exact précédent, supprimez les pays dont le nom est supérieur à 5 caractères. (Faites en sorte d'obtenir une suppression de 2 pays minimum)

Dans un premier temps, vous pouvez utiliser la boucle fori, avec la méthode `liste.remove(i)`.

- **Que se passe-t-il ?**

Désormais, utilisez l'itérateur de votre liste.