

## Interface List et classe ArrayList

La documentation de la classe générique `ArrayList<E>` de la JDK se trouve ici :  
<https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/ArrayList.html>

### 1 Utilisation

1. Ecrire une fonction (une méthode statique) qui prend en argument un entier `n`, crée une liste d'entiers (donc de type `ArrayList<Integer>`) allant de 1 à `n`, et la renvoie.
2. Ecrire une procédure (une méthode statique qui renvoie `void`) qui prend en argument une liste d'entiers et les affiche un par un.
3. Ecrire une fonction qui prend en argument une liste d'entiers, calcule la somme de ces entiers et renvoie le résultat.

### 2 Implantation

L'objectif de cet exercice est d'implanter l'interface `List<E>` de la JDK en utilisant un tableau pour stocker les éléments (de manière similaire à la classe `ArrayList` de la JDK, mais avec un tableau de taille fixe dans un premier temps). On partira de l'implantation partielle suivante :

```
public class BoundedArrayList<E> {  
  
    E[] elements = null;  
    int size = 0;  
  
    BoundedArrayList(E[] elements) {  
        this.elements = elements;  
        this.size = elements.length;  
    }  
}
```

1. Ajouter un constructeur qui prend en paramètre la capacité initiale du tableau.
2. Vérifier le comportement par défaut des méthodes `equals` et `toString`.
3. Implanter les méthodes `get(int)` et `size()` de l'interface `List<E>`.
4. Compléter l'implantation en utilisant la classe abstraite `AbstractList<E>` de la JDK.
5. Pourquoi l'implantation des méthodes `equals` et `toString` est-elle devenue inutile ? Que dit la documentation de l'interface `List<E>` de la JDK à ce sujet ?
6. Pour rendre la liste mutable, implanter la méthode `set(int, E)`.
7. Pour permettre d'ajouter des éléments à la liste (dans la limite de la capacité du tableau), implanter aussi les méthodes `add(E)`.
8. Pour respecter le contrat de `AbstractList`, implanter aussi les méthodes `add(int, E)` et `remove(int)`.
9. Compléter la méthode `add(int, E)` pour augmenter la capacité du tableau quand c'est nécessaire (on pourra doubler la capacité par exemple). On pourra alors aussi renommer `BoundedArrayList` en `ArrayList`.