

# La classe générique ArrayList<T>

## I La classe générique ArrayList<T>

Pour l'instant nous avons étudié le tableau pour structurer les données:

- Taille statique
- Lent pour la recherche d'éléments particuliers
- Impossibilité d'utiliser un pattern de déplacement dans les éléments

Java propose depuis la version 2 une **classe générique ArrayList<T>** qui est une encapsulation du tableau avec la possibilité de le rendre dynamique en taille.

**La généricité et références :** possibilité de stocker des éléments de type quelconque, pour peu qu'il s'agisse d'objets. Un nouvel élément introduit dans une ArrayList Java est une référence à l'objet et non une copie.

### Exemple:

```
ArrayList<Object> myArrayList = new ArrayList<Object>();
myArrayList.add("Coucou");
myArrayList.add(34);
for (int i = 0; i < myArrayList.size(); i++) {
    Object myObject = myArrayList.get(i);
    if (myObject instanceof String) {
        System.out.println("Chaîne:" + ((String)myObject));
    }
    if (my_object instanceof Integer) {
        System.out.println("Integer:" + ((Integer)myObject));
    }
}
```

## II Iterateurs sur une ArrayList<T>

Les **itérateurs** permettent de parcourir un par un les différents éléments d'une ArrayList.

Il existe deux familles d'itérateurs :

- **monodirectionnels: Iterator**

le parcours de la collection se fait d'un début vers une fin; on ne passe qu'une seule fois sur chacun des éléments.

**hasNext() : boolean** Vérifie s'il y a un prochain

**next() : T** Permet d'obtenir l'objet courant et passe au suivant

**remove()** Supprime le dernier objet renvoyé par **next()**

- **bidirectionnels: ListIterator**

le parcours de la collection peut se faire dans les deux sens ; on peut avancer et reculer à sa guise dans la collection. **Ils seront étudiés plus tard.**

### Exemple:

```
import java.util.*;
public class Exercise {
    public static void main(String[] args) {
        ArrayList<String> students = new ArrayList<String>();
        students.add("Gertrude");
        students.add("James");
        students.add("Patricia");
        students.add("Helen");
        students.add("Hank");
        for(int i = 0; i < students.size(); i++)
            System.out.println("Student: " + students.get(i));
        System.out.println("");
        Iterator iter = students.iterator();
        while(iter.hasNext())
            System.out.println("Student: " + iter.next());
        // ou de manière plus simple en masquant l'itérateur avec une boucle foreach
        for(String student : students)
            System.out.println("Student: " + student);
    }
}
```