

TP6

Interfaces

Vous avez déjà rencontré des *interfaces*, par exemple dans la méthode:

```
static void swap(List<?> list, int i, int j)
```

List n'est pas une classe, mais une *interface*. L'interface List représente le concept de liste, c'est-à-dire tout objet qui aura une méthode add, une méthode get, une méthode size, etc. En java on parle d'interface, mais de façon générale on parle de *type abstrait de données*.

1. Que connaissez vous comme classes qui implémentent List, c'est-à-dire qui ont au moins ces méthodes ?

L'interface List contient beaucoup de méthodes (25 dont quelques unes compliquées), et même si vous pourriez écrire un code qui *implémente* la List on va commencer par quelque chose de plus simple (et en fait de plus utile).

Comparator

L'interface Comparator<T> représente le concept de quelque chose qui est capable de comparer des objets de type T entre eux. Elle contient une seule méthode :

- `public int compare(T o1, T o2)` Compares its two arguments for order. Returns a negative integer, zero, or a positive integer as the first argument is less than, equal to, or greater than the second.

On peut écrire une classe qui *implémente* la classe Comparator de la manière suivante (dans cet exemple, un comparateur de chaînes de caractères) :

```
public class CompChaine implements Comparator<String>{
    public int compare(String s1, String s2) // Attention : il faut que cette méthode
    soit public
    {
        if(s1.length() == s2.length())
            return 0;
        if(s1.length() > s2.length())
            return 1;
        return -1;
    }
}
```

1. Que fera le code suivant :

```
Comparator<String> c = new CompChaine();  
System.out.println(c.compare("toto", "tut"));
```

2. Que fait la méthode suivante sur le comparateur c de la question précédente ?

```
public static void methode(Comparator<String> c){  
    if(c.compare("Bonjour", "toto") > 0)  
        System.out.println("Oh Oh");  
    else  
        System.out.println("Ah Ah");  
}
```

3. Écrivez un comparateur tel que si on appelle la méthode précédente sur une instance de ce comparateur, l'affichage soit "Ah Ah".
4. Écrivez une méthode statique prenant en entrée un comparateur de String et un tableau de String, et renvoyant vrai si le premier élément du tableau de String est plus grand (strictement) que tous les autres selon le comparateur et faux sinon.
5. Écrivez un comparateur permettant, à l'aide la méthode précédente, de savoir si le premier mot d'un tableau de String est le plus court.

algorithmes

1. La classe Collections contient une méthode statique sort prenant en argument une List<T> et un Comparator<T> et triant la liste par ordre croissant selon le comparateur. Étant donnée une ArrayList<Integer> que faut-il faire pour trier cette liste par ordre croissant ?
2. Écrivez une implémentation de Comparator permettant de trier une liste d'entiers par ordre décroissant.