# Design patterns, Java

# Exercices généralités, lambdas

### Préambule

Copier le programme principal et la méthode auxiliaire fournis :

public static void main(String[] args) {

List<String> laListe = Arrays.asList("Un", "Deux", "Trois", "Quatre", "Cinq", "Six", "Sept", "Huit", "Neuf");

affiche(laListe);

Collections.sort(laListe, new Comparator() {

public int compare(Object s1, Object s2) {

return ((Comparable)s1).compareTo(s2);

}

});

affiche(laListe);

}

public static void affiche(List<String> laListe) {

for (String suivante : laListe) {

System.out.println(" " + suivante);

}

System.out.println(" ---------");

}

### Lambda

Dans cet exercice, on modifie le code existant et on en ajoute. Il ne faut utiliser que des lambdas expressions

1. Remplacer les paramètres de Collections.sort par une lambda expression
2. Afficher ensuite le contenu de la liste trié par ordre alphabétique inverse (en utilisant une lambda expression bien sûr)
3. Afficher ensuite le contenu de la liste trié par longueur de chaîne croissante (en utilisant une lambda)
4. Afficher ensuite le contenu de la liste en mettant d’abord les chaînes possédant un ‘e’, les autres chaînes ensuite (en utilisant une lambda)

### Référence à une méthode

1. Dans le tri par ordre alphabétique, remplacer la lambda par une référence à une méthode existante de la classe String
2. Dans le tri par ordre alphabétique inverse, remplacer la lambda par une référence à une méthode qu’il faudra écrire

### Interface fonctionnelle

1. Créer l’interface fonctionnelle « MeilleureChaine » qui déclare la méthode « meilleure ». Cette méthode prend deux chaînes en paramètres et renvoie un booléen.
2. Créer une fonction statique « String getMeilleure(String inS1, String inS2, MeilleureChaine inSelecteur) ». Cette fonction renvoie la première chaîne si inSelecteur.meilleure(inS1, inS2) vaut true, la seconde sinon.

Appeler cette fonction statique avec les chaines « Un », « Deux » et la lambda adéquate pour obtenir la première chaîne par ordre alphabétique

Appeler cette fonction statique avec les mêmes chaines et la lambda adéquate pour obtenir la chaîne de plus grande longueur.

Appeler cette fonction statique avec les mêmes chaines et une lambda adéquate pour obtenir une chaîne contenant un « e » s’il y en a une, sinon le premier argument de l’appel.