

# TP DE JAVA AVANCÉ N°3

---

## Exercice 1

---

On considère le champ de type table suivant :

```
HashMap<String,List<V>> map = new HashMap<>();
```

1. Écrire une méthode qui fabrique une liste contenant tous les éléments des listes associées aux clefs de la table à l'aide de la méthode `forEach` de `Map`.
2. Écrire une méthode `toArrayList()` qui remplace les valeurs associées aux clefs par la copie de cette liste dans une nouvelle `ArrayList` si la clef commence par le caractère `'*'`. On utilisera la méthode `replaceAll` de `Map`.
3. Écrire une méthode `append(String k,V v)` qui ajoute `v` à la liste associée à `k` en utilisant la méthode `compute` de l'interface `Map`.

## Exercice 2

---

On se donne le code suivant :

```
<U,T> U fold(Collection<T> c, U u, BiFunction<U,T,U> f) {
    for(T t:c) {
        u = f.apply(u,t);
    }
    return u;
}
```

1. Est-il possible de modifier la méthode pour utiliser la méthode `forEach` de `Collection` ? Pourquoi ?
2. Utiliser la méthode ci-dessus pour calculer la somme des `Integer` d'une liste
3. Même question avec des `String` et leur concaténation. On utilisera la notation `::`
4. Modifier la méthode `fold` pour autoriser un post-traitement une fois la valeur retournée calculée. Essentiellement, cela consiste à ajouter un paramètre `g`

judicieusement typé et remplacer l'instruction `return u;` par `return g.apply(u);`.

5. Utiliser la méthode modifiée pour retourner une représentation sous forme de `String` de la liste des éléments comparable à ce que fait `toString`. On construira la chaîne avec un `StringBuilder`.

## Exercice 3

---

Déclarer une hiérarchie : la classe `O` est superclasse des classes `A`, `B` et `C`.

Implémenter un système de visiteur dans la hiérarchie afin de pouvoir implémenter une méthode `dispatch` tel que, pour un objet `o` de type `O`, l'appel

```
dispatch(o, fA, fB, fC)
```

appelle `fA(o)` si `o` est de type `A`, `fB(o)` si `o` est de type `B` et `fC(o)` si `o` est de type `C`. Votre implémentation doit fonctionner même si les fonctions retournent des valeurs ou lèvent des exception