

# TP DE JAVA AVANCÉ N°1

---

## Exercice 1

---

Écrire des méthodes paramétrées avec une signature la plus générale possible qui effectue les actions suivantes. Aucune structure de donnée n'est créée par les méthodes demandées ; tout est passé en paramètre.

- Une méthode générique qui supprime toutes les éléments en position paire dans une `List`.
- Une méthode qui prend une `Collection` de `Collection`s et qui lui enlève toutes les `Collection`s vides.
- Une méthode qui insère tous les éléments d'une `Collection` dans une `Map` avec la valeur égale à la clef
- Une méthode qui ajoute dans une `Collection` tous les éléments des `Collection`s présentes dans une `Collection` de `Collection`s.
- Une méthode qui prend une `Collection` d'éléments comparables entre eux (au sens de `Comparable`) et un élément `max` et en supprime de la `Collection` tous les éléments plus grand que `max`.

## Exercice 2

---

1. Écrire une méthode qui prend une `Map` `m` et qui renvoie son inverse : une `Map` où les clefs sont les valeurs de `m` et où un élément `x` sera associé à une `Collection` (`Set` ou `List`, il fait choisir la bonne) des clefs auxquelles la valeur `x` est associée.

Par exemple, si `m` contient les associations :

| 1=>A, 2=>B, 10=>A et 22=>Z

La `Map` retournée contiendra alors :

| A=>[1,10], B=>[2], Z=>[22]

2. Modifier votre code pour que la `Map` retournée soit une vue de `m` : toute modification sur la vue doit être rétroactive vers `m`. Par exemple, avec

l'exemple de `m` ci-dessus, si `v` en est la vue inversée :

- Appeler `v.remove('A')` doit supprimer les associations `1=>A` et `10=>A` de `m`
- Appeler `v.get('A').remove(1)` doit supprimer l'association `1=>A` de `m`