

# Rapport du TD n°06 Java avancé

Guillaume CAU  
01/11/2019

## Exercice 01 :

## Question 02 :

Pour effectuer l'affichage demandé, j'ai utilisé la méthode *Collectors.joining*. Celle-ci permet de transformer en String tous les éléments d'une Stream. On peut choisir les séparateur, un début et une fin.

## Question 03 :

Ma méthode of prend un (Object ...). Elle utilise ensuite la méthode from préalablement créée. Elle crée donc un Seq en envoyant la liste à from.

## Question 04 :

Pour réaliser la méthode foreach, j'ai converti l'ArrayList en une stream et utilisé la méthode forEach de la Stream.

## Question 05 :

Pour réaliser la méthode map, j'ai ajouté un champ Function à la classe Seq. Celui-ci garde en mémoire la/les fonctions à appliquer une fois une opération demandée. L'interface Function dispose d'une méthode andThen qui permet de créer une liste de fonctions à appliquer successivement. Je crée donc un nouvel objet Seq à chaque appel de la méthode map et affecte la liste des fonctions +1 au nouvel objet.

## Question 06 :

La méthode FindFirst doit renvoyer un Optional pour éviter les NullPointerException. Cet Optional est obtenu par la méthode findFirst de stream().

## Question 07 :

Pour pouvoir effectuer un for-each-in sur notre objet Seq, il faut qu'il implémente l'interface Iterable.

J'ai donc implémenté les méthodes de cette interface pour rendre la classe Seq itérable,

## Exercice 02 :

## Question 01 :

Il s'avère que j'ai utilisé les ArrayList dès le début pour que le get soit en  $O(1)$  comme demandé.

## Conclusion

Je n'ai malheureusement pas compris grand-chose de ce TP. J'ai essayé de faire au mieux mais certains tests ne passent pas, je ne comprends pas pourquoi. J'essayerai de comprendre plus tard, je ne peux pas maintenant, le temps disponible est écoulé.