Formation : INFO2

INFO
INGÉNIEUR
ESIPE

Rapport du TD n°07
Java Avancé

Guillaume CAU
04/11/2019

UNIVERSITÉ PARIS-EST
MARNE-LA-VALLÉE

Esipe
INGÉNIEURS PARIS EST

INGENIEURS 2000
L'EXCELLENCE AU SERVICE DE L'ALTERNANCE

Exercice 01 :

Question 01 :

La méthode count prend un *Object* en paramètre. En effet, si l'on souhaite stocker des objets qui implémentent A, il y a des situation ou le *extends* ne marche pas. En effet, Si la méthode count récupère un objet qui implémente A et B en même temps, on ne saura pas si il faut le prendre ou pas. Le ? extends A ne prend pas plusieurs « paramètres ».

Question 02 :

La méthode *createSimpleBag* est une méthode statique qui se trouve dans l'interface *Bag*. Elle ne fait que retourner une nouvelle instance de *BagImpl*.

Question 03 :

Pour éviter les effets de bords, la programmation objet utilise plusieurs bonne pratique.
L'une d'elle consiste à mettre toutes le méthodes qui ne sont pas censée être appelé depuis l'extérieur en **private**. De cette manière, seule les méthodes prévu pour être appelée depuis l'extérieur sont en **plublic**.

Question 04 :

Rappel : Un Consumer est une interface fonctionnelle qui prend un argument et qui ne renvoie rien.

Pour implémenter la méthode forEach(), il suffit d'utiliser la méthode forEach de la Map stockée puis d'exécuter n fois le Consumer en fonction du nombre d'élément.

Question 05 :

La méthode Iterator a déjà été vu dans les derniers TP. Voir le dernier TP.

Question 06 :

La Collection renvoyée par la méthode *nCopies* est **immutable**.
Après plusieurs tests, on peut remarquer que les itérateurs créer avec les Stream sont plus efficaces.

Question 07 :

Pour pouvoir créer des forEach ↔ itérateurs avec des objets Bag, il faut ajouter la méthodes iterator() dans l'interface qui sera ensuite implémentée.

Conclusion :

Ce TD m’a permis de rattraper mon retard sur les types paramétrés. En effet, j’avais encore du mal à comprendre les <? super t> et les < ? extends T>. J’ai désormais réussi à comprendre dans quelle situation les utiliser.