

Java avancée

Exercice 1 – Location de voiture

Question 5

Les Stream permettent de réaliser des traitements sur les collections sans avoir à intégralement les calculer en amont.

La classe List dispose d'une méthode `List::stream` qui retourne un Stream de la List.

`Stream::filter` prend en argument un prédicat, lambda retournant un booléen, pour retourner un Stream ne contenant que les éléments ayant répondu positivement au prédicat.

`Stream::map` prend en argument une fonction qui réalisera un traitement sur chaque élément du Stream et retournera un nouveau Stream contenant les résultats du traitement.

`Stream::collect` prend en argument un Collector pour récupérer le résultat des différents traitements réalisés dans un format défini par le Collector

`Collectors::joining` permet de concaténer les éléments d'un stream en un String séparé par un délimiteur passé en argument.

Donc pour simplifier on peut récupérer le Stream de la liste de `CarRental` avec `List::stream` et utiliser `Stream::collect` avec en argument `Collectors.joining('\n')`.

Question 6

Si un utilisateur essaie de retirer un véhicule non présent dans `CarRental`, il est plus sûr de jeter une exception par souci de sécurité. En effet, cette état n'est pas supposé être autorisé, le manque de réponse pourrait être interprété comme un faux positif.

Le test 6 fonctionne car les records ont par défaut les méthodes `equals`, `hashCode` et `toString` définies.

Question 7

La méthode `CarRental::findAllByYear` laisse l'utilisateur comprendre qu'il y a un résultat, le fait qu'il n'y ai aucun véhicule répondant au critère n'est pas anormal, donc il n'y a pas lieu de jeter une exception. Dans ce cas nous pouvons retourner null, en spécifiant dans la documentation les circonstances possible.

Question 10

Les interfaces sont indispensables car `Car` et `Camel`, et possiblement les suivants, sont des records. Ce format impose l'utilisation des interfaces pour généraliser les contraintes.

Conclusion

Ce TP m'a permis de reprendre les bonnes habitudes du développement en POO. Durant la période en entreprise ce n'était pas une contrainte qui m'était imposée et je n'ai pas codé en Java donc c'est l'occasion pour moi de tout revoir rapidement.

De plus, j'ai appris à mieux utiliser les Stream pour répondre aux diverses questions de ce TP sans trop perdre de temps.