Algorithmique Programmation (Ing39) : Tp Noté sur les collections

8 octobre 2023

Ce que vous devez faire

Dans ce tp noté, on vous demande d'implémenter un certain nombre de classes pour gérer un catalogue et des factures. Dans chacune de ces classes, nous vous demandons d'implémenter certaines méthodes décrites par leurs contrats. Les tests fournis (ET D'AUTRES !) serviront pour la notation. N'oubliez pas de commiter et pusher votre projet au fur et à mesure du TP. Toute perte accidentelle de code est de votre responsabilité.
Très important :

1. le profil des méthodes NE DOIT PAS CHANGER. Si vous voulez malgré tout ajouter une exception au comportement d'une méthode, il FAUT utiliser `RuntimeException`.
2. Il y a certaines contraintes d'implantation explicitées plus bas. Lisez bien l'énoncé et les contrats dans le code.

Projet gitlab Le projet maven du TP est dans votre projet git (faites un `git pull`). Vous ne devez pas renommer les classes existantes, y compris les classes de tests. Vérifiez au début du TP que vos commits et push fonctionnent.

Rappel : si vous n'avez pas de clone déjà prêt, pour cloner votre projet le plus simple est, dans un terminal, de vous mettre dans un répertoire (**qui n'est pas déjà dans un git ou un autre projet java**) et de faire la commande :

```
git clone <adresse de votre projet git>
```

Ensuite `import/Existing maven project`.

La classe `Produit` Un produit contient un identifiant, un prix et une désignation. L'identifiant est unique, cependant comme dans un catalogue un produit n'apparaît qu'une seule fois, l'égalité par défaut (comparaison des adresses) est suffisante pour nos besoins. **Il n'est donc PAS nécessaire de redéfinir les méthodes `equals` et `hashCode`**.

La classe `Catalogue` **Contraintes d'implantation** : cette classe doit **impérativement** avoir un constructeur par défaut, sinon *tous* les tests échoueront et *vous aurez la note de 0/20*. Le catalogue de produits **doit être implanté** par un Map.

Cette classe comporte beaucoup de méthodes. Écrivez-les et testez-les une par une, dans l'ordre dans lequel elles apparaissent dans l'interface. SURTOUT ne testez pas à la fin du TP mais au fur et à mesure.

La première méthode (`creerProduit`) sera beaucoup utilisée par les tests, car elle servira à créer les produits. C'est ce qu'on appelle une *factory method* (une méthode servant à créer un nouvel objet).

La classe `LigneFacture` La classe `LigneFacture` est très simple et représente une paire (produit,quantité). *Les objets de cette classes sont immuables.* C'est-à-dire qu'on ne peut pas les modifier, seulement en créer de nouveaux.

La classe `Facture` Les objets de la classe `Facture` devraient contenir une collection de `LigneFacture`. Lisez bien l'invariant de la classe Facture et comprenez-le avant de commencer.