

I. classe Item

II. classe
ItemEnStock

III. classe Stock

IV. classe
ChaîneNisagam

Tout magasin de la marque Nisagam a un lieu de stockage dans laquelle 1000 objets maximum peuvent être entreposés. Chaque objet (item) est caractérisé par un code composé de 5 caractères numériques et par un prix de vente. Chaque objet peut être en plusieurs exemplaires dans le stock et la marque Nisagam propose un choix de 100 objets différents.

Il faudra donc penser aux exceptions pour gérer :

- un code incorrect
- un stock plein

sachant que les exceptions lancées devront être attrapées soit dans la méthode lançant l'exception soit par une méthode appelant cette méthode.

classe Item

I. classe Item

II. classe ItemEnStock

III. classe Stock

IV. classe ChaîneNisagam

Ecrire la classe Item avec

- ses constructeurs
- ses attributs privés
- ses accesseurs en lecture et modification
- la réécriture des méthodes equals et toString

classe ItemEnStock

La classe `ItemEnStock` qui hérite de la classe `Item` comme attribut le nombre d'exemplaires présent en stock.

Ecrire la classe `ItemEnStock` avec

- ses attributs privés
- ses constructeurs
- ses accesseurs en lecture et modification
- la réécriture des méthodes `equals` et `toString`

classe Stock

La classe Stock est caractérisée par

- le nom du magasin
- l'ensemble des Items en stock dans le magasin
- le nombre total d'Items en stock dans le magasin

Ecrire la classe Stock avec

- ses constantes (le nombre d'Items différents 100 et la capacité d'un stock 1000)
- ses attributs privés
- ses accesseurs en lecture et modification.
- une méthode d'ajout d'une quantité donnée en paramètre d'un Item donné par son code (exceptions)
- une méthode de retrait d'une quantité donnée en paramètre d'un Item donné par son code (exception) et qui renvoie le nombre d'Items réellement retirés.
- une méthode qui renvoie la valeur totale du stock
- une méthode qui renvoie l'Item qui est le plus présent dans le stock

Pour l'attribut représentant l'ensemble des Items en stock dans le magasin vous pourrez utiliser des tableaux ou des ArrayList.

classe ChaineNisagam

I. classe Item

II. classe
ItemEnStock

III. classe Stock

IV. classe
ChaineNisagam

On considère une chaîne de 10 magasins Nisagam. On s'intéresse donc à gérer globalement les stocks de ces 10 magasins.

Ecrire la classe ChaineNisagam avec

- ses constantes (10 magasins)
- son ou ses attributs privés
- ses accesseurs en lecture et modification.
- une méthode d'ajout Stocker d'un Item donné par son code dans le magasin dont le stock est le moins chargé et qui renvoie le nom de ce magasin (voir les exceptions lancées par la méthode d'ajout de la classe Stock)
- une méthode qui renvoie le nom du magasin qui a le moins de valeur marchande en stock
- une méthode de classe qui renvoie la valeur marchande totale des 10 stocks

I. classe Item

II. classe
ItemEnStock

III. classe Stock

IV. classe
ChaineNisagam

Dans la méthode `main`, vous devrez

- créer un Stock pour un magasin Nisagami pour chaque $i \in \{0..9\}$
- créer une chaîne de 10 magasins Nisagam1, ... Nisagam10
- ajouter 5 exemplaires d'un Item
- afficher le nom de magasin qui a le moins de valeur marchande en stock
- afficher la valeur marchande totale des 10 stocks

Rappels :

- la classe `java.util.Random` contient la méthode `public int nextInt(int k)` qui renvoie un entier dans $\{0, \dots, k - 1\}$.
- la méthode `String concat(String s)` ajoute la chaîne de caractères `s` derrière la chaîne objet.
- la méthode `static String valueOf(int i)` de la classe `String` renvoie la chaîne de caractères représentant l'entier en argument