

Exercices dirigés

séance n°7

Exercice 1 : Matrice creuse

Une matrice creuse est une matrice d'entiers essentiellement composée de 0 . Les valeurs sont comprises entre 0 et 10 ($[0,10[$).

| | | | | |
|---|---|---|---|---|
| 0 | 2 | 0 | 0 | 0 |
| 1 | 0 | 3 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 6 | 0 |

l'élément situé
ligne 4, colonne 4
vaut 6

On souhaite représenter une matrice creuse par une liste chaînée, contenant uniquement les éléments non nuls, avec leurs indices.

`[(2,(1,2)); (1,(2,1)); (3,(2,3)); (6,(4,4)); (0, (4,5))]`

Les éléments sont rangés par indice croissant (ligne puis colonne).

Le dernier élément de la liste permet de connaître l'indice maximal de la matrice même si celui-ci est nul (`(0, (4,5))` dans l'exemple).

Question 1

Définir la classe `Element`, avec ses variables d'instance, un constructeur qui les initialise et au moins la méthode `println()` pour afficher l'état d'une instance.

Question 2

Définir la classe `ListeCreuse`. en s'inspirant de la classe `Liste` vue en cours et en ED. On y ajoutera une méthode d'affichage de la liste.

Question 3

Définir la classe `MatriceCreuse` comme un tableau à 2 dimensions. On initialisera cette matrice de manière aléatoire en utilisant la méthode `Math.random()` du package `java.lang` qui retourne un `double` compris entre 0 et 1 ($[0,1[$).

Question 4

Définir une fonction `ajouter` qui ajoute un élément en tête à la liste.

Question 5

Écrire la fonction `transformer` qui transforme une matrice creuse en une liste chaînée selon le modèle présenté ci-dessus.

Question 6

Écrire un petit programme de test qui crée une matrice creuse, l'affiche, la transforme en une liste creuse et affiche cette dernière.

Exercice 2 : vente par correspondance

Un magasin de vente par correspondance souhaite disposer d'un logiciel de gestion de ses ventes. Chaque article disponible dans le magasin est décrit par sa référence (entier), par sa dénomination (une chaîne de caractères) et par son prix. L'ensemble des articles (le stock) est représenté par un tableau.

Question 1

Définir la classe `Article` et la classe `Stock`.

Question 2

Les commandes des clients sont également représentées par un tableau. Une commande groupe l'ensemble des achats effectués par un client. Elle est caractérisée par un numéro de commande, le nom du client, les achats et le solde. Un achat est caractérisé par un article, le nombre d'exemplaires commandés et le nom du client. Les achats du client sont représentés par une table de hachage (`Hashtable`).

Définir la classe `Achat`, c'est à dire variables d'instance, constructeur avec initialisation et accesseurs (`getter` et `setter`)

Définir la classe `Commande`.

On utilisera la classe `java.util.Hashtable`, pour réunir au sein d'une structure commune tous les couples (référence, article).

Exemple de Hashtable :

```
Hashtable<String,Integer> numbers = new Hashtable<String,Integer> ();
numbers.put("one", 1);
numbers.put("two", 2);
numbers.put("three", 3);
// parcours de la table
Integer i = null;
for (Enumeration<Integer> iter=numbers.elements();iter.hasMoreElements();)
    i = iter.nextElement();
```

Question 3

Ecrire une méthode qui permet d'ajouter un achat à une commande.

Question 4

Ecrire une méthode qui modifie, dans une commande, le nombre d'exemplaires d'un article acheté connaissant sa référence.

Question 5

Ecrire une méthode qui retourne le nombre d'articles achetés lors d'une commande.

Question 6

Ecrire l'algorithme puis la méthode qui retourne le solde d'une commande.