
Exercice 1 – La classe `MatriceEntiere`

Le but de cet exercice est d'écrire une classe `MatriceEntiere` offrant les fonctionnalités suivantes :

- la possibilité d'initialiser les éléments de la matrice à partir de données contenues dans un fichier ASCII ;
- la possibilité d'initialiser une matrice à zéro ;
- les opérations de base : somme de matrices, produit de matrices, produit d'une matrice par un scalaire ;
- le calcul de la matrice transposée ;
- une méthode `toString()` et une méthode d'affichage ;

L'ensemble des exceptions potentiellement levées par ces différentes opérations devra être géré soit en interne par la méthode qui lève l'exception, soit par le programme appelant, en choisissant la solution la mieux adaptée.

Pour pouvoir tester notre classe, nous aurons besoin de saisir les valeurs contenues dans une matrice. Cette opération étant fastidieuse, nous allons lire ces valeurs dans un fichier ASCII.

Question 1

Recherchez dans l'API Java la méthode permettant de transformer une chaîne représentant un entier en la valeur correspondante.

Question 2

Les attributs d'une matrice entière sont le nombre de lignes, le nombre de colonnes et un tableau à deux dimensions permettant de stocker ses éléments.

Construisez une classe `MatriceEntiere` avec :

- un constructeur `MatriceEntiere(int lignes, int colonnes)`, qui crée une matrice sans l'initialiser ;
- un deuxième constructeur `MatriceEntiere(File fichier)` : les dimensions sont lues dans le fichier, et la matrice est initialisée à partir des valeurs lues dans le fichier ;
- deux accesseurs :
 - `int getElem(int i, int j)` qui renvoie la valeur de l'élément en ligne `i` et colonne `j`.
 - `void setElem(int i, int j, int val)` qui place la valeur `val` à l'intersection de la ligne `i` et de la colonne `j`.

Question 3

Ajoutez à la classe `MatriceEntiere` une méthode `toString()` et une méthode d'affichage, et écrivez un programme permettant de construire une matrice à partir des données contenues dans un fichier et de l'afficher. Vous pourrez tester votre classe avec les fichiers de données fournis (`donnees_*`), accessibles dans le répertoire :

`/Infos/lmd/2019/licence/ue/LU3IN001-2019oct/TME`

On rappelle que le produit des matrices $A(nbl_A, nbc_A)$ et $B(nbl_B, nbc_B)$ peut être calculé si $nbc_A = nbl_B$. Le résultat est une matrice $C(nbl_A, nbc_B)$ dont l'élément en position (i, j) a pour valeur :

$$c_{i,j} = \sum_{k=1}^{nbc_A} a_{i,k} \cdot b_{k,j}$$

Question 4

Complétez la classe `MatriceEntiere` en lui ajoutant les méthodes permettant d'effectuer les opérations suivantes : initialisation à zéro, calcul de la transposée, somme de matrices, produit d'une matrice par un scalaire et produit de deux matrices.

Vous devrez vous assurer, lorsqu'une opération implique plusieurs matrices, que les dimensions de ces matrices sont compatibles avec l'opération demandée. Vous devrez pour cela créer votre propre classe `TaillesNonConcordantesException`.

Vous pourrez tester les méthodes en utilisant les fichiers de données fournis. Le fichier `resultats_tests` contient les valeurs attendues lorsqu'on somme les matrices initialisées à partir des valeurs contenues dans `donnees_somme1`

et `donnees_somme2`, ainsi que le produit de la matrice initialisée à partir de `donnees_produit1` par la matrice initialisée à partir de `donnees_produit2`.