

Mathématiques 2 — DM n°2

À rendre le mardi 6 avril 2019

Le but du devoir est de coder en Java plusieurs opérations que nous avons vues sur les matrices. Le code avait été écrit dans des fichiers `Rational.java` (où sont représentés des nombres rationnels), `Matrice.java` (où sont écrites les fonctions de manipulation de matrices) et `Main.java` (un fichier de test).

Ici, on a enlevé certains morceaux de code du fichier `Matrice.java`, et c'est à vous de les réécrire.

Vous pouvez effectuer de CM en binôme. Dans ce cas, le nom des deux élèves ayant travaillé ensemble devra apparaître clairement.

Pour ce DM, il faudra rendre sur e-learning un compte-rendu, nommé `cr.txt`, ainsi que le code complété du fichier `Matrice.java` :

- les exercices à une étoile consistent à compléter différentes fonctions du fichier `Matrice.java` ;
- les exercices à deux étoiles sont des questions auxquelles il faudra répondre dans le compte-rendu.

Exercice 1**.

Trouvez un problème qui peut être résolu grâce à un système d'équations linéaires.

Plus le problème sera intéressant, plus vous aurez de points bonus.

Exercice 2*.

Remplir le code de la fonction `plus` pour qu'elle renvoie la somme de `this` et de `M`.

Exercice 3*.

Remplir le code de la fonction `times` pour qu'elle renvoie le produit de `this` et de `M`.

Exercice 4*.

Remplir le code de la fonction `transpose` pour qu'elle renvoie la transposée de `this`.

On rappelle que la transposée d'une matrice M est la matrice M^* dont les coefficients sont donnés par la relation $M_{i,j}^* = M_{j,i}$.

Exercice 5*.

Remplir le code de la fonction `swapRows` pour qu'elle échange les lignes `i` et `j` de `this`.

Exercice 6*.

Remplir le code de la fonction `transvection` pour qu'elle ajoute `a` fois la ligne `i` de `this` à la ligne `j`.

Exercice 7*.

Remplir le code de la fonction `multiplyRow` pour qu'elle multiplie par `a` la ligne `i` de `this`.

Exercice 8*.

Remplir le code de la fonction `identity` pour qu'elle renvoie la matrice identité de taille $n \times n$ (si `this` est déjà une matrice de taille $n \times n$).

Exercice 9*.

Remplir le code de la fonction `inverse` pour qu'elle renvoie l'inverse de la matrice `this`. Cette fonction devra renvoyer une `ArithmeticException("Division par zéro")` si `this` est une matrice carré (de taille $n \times n$) qui n'a pas d'inverse.

On pourra procéder en utilisant le pivot de Gauss. Ce faisant, si, à un moment, une des lignes de la copie de `this` ne contient que des 0, alors c'est que `this` n'a pas d'inverse.

Exercice 10**.

En calculant l'inverse d'une matrice, on peut résoudre les systèmes d'équations ayant autant de variables que d'inconnues et ayant exactement une solution. Écrire un pseudo-code pour traiter les cas où l'on a plus de variables que d'équations.

Exercice 11**.

Utiliser le programme écrit en question 9 pour résoudre le problème proposé en question 1. Vous devrez indiquer comment vous avez obtenu les données numériques de votre problème, puis discuter les solutions obtenues.
