TP5 et TP6 — Héritage

Objectif : Dans ce TP, on reprendra la classe **`Matrice`** du TP4 représentant des matrices de $\mathcal{M}_n(\mathbb{R})$, et on ajoutera des classes filles représentant d'autres façons de stocker les coefficients.

Rappels du TP4

Dans ce TP, on repart des classes **`Matrice`** et **`Vecteur`** du TP4. On pourra par exemple s'inspirer du corrigé disponible sur `moodle`.

Quelques notations

Soit $A \in \mathcal{M}_n(\mathbb{R})$, que l'on écrit de la façon suivante.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix}$$

On s'intéressera à des matrices particulières : triangulaire inférieure et supérieure, tridiagonale, et diagonale.

Les matrices triangulaires inférieures L et supérieures U auront les formes suivantes :

$$L = \begin{pmatrix} l_{1,1} & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} \end{pmatrix} \qquad \text{et} \qquad U = \begin{pmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & u_{2,2} & \dots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{n,n} \end{pmatrix},$$

et les matrices tridiagonales T et diagonales D vérifieront :

$$T = \begin{pmatrix} t_{1,1} & t_{1,2} & 0 & 0 & \dots & 0 \\ t_{2,1} & t_{2,2} & t_{2,3} & 0 & \dots & 0 \\ 0 & t_{3,2} & t_{3,3} & t_{3,4} & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & t_{n-1,n-2} & t_{n-1,n-1} & t_{n-1,n} \\ 0 & 0 & \dots & 0 & t_{n,n-1} & t_{n,n} \end{pmatrix} \quad \text{et} \quad D = \begin{pmatrix} d_{1,1} & 0 & 0 & 0 & \dots & 0 \\ 0 & d_{2,2} & 0 & 0 & \dots & 0 \\ 0 & 0 & d_{3,3} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & d_{n-1,n-1} & 0 \\ 0 & 0 & \dots & 0 & 0 & d_{n,n} \end{pmatrix}.$$

1. Hiérarchie de l'héritage

On veut créer quatre classes héritant (potentiellement indirectement) de la classe **`Matrice`** :
les classes **`MatriceTriangulaireInf`**, **`MatriceTriangulaireSup`**, **`MatriceTridiagonale`** et **`MatriceDiagonale`**.

Comment organiser l'héritage entre ces différentes classes ?

2. Écriture des classes

Pour chacune des quatre classes ci-dessus, écrire :

— un constructeur (qui pourra soit utiliser celui de la classe mère, soit le remplacer),

— une implémentation du produit entre une matrice et un vecteur colonne.

Conseil : Commencer par le cas des matrices tridiagonales et diagonales, avant de s'intéresser aux matrices triangulaires.

3. Héritage multiple

On remarque qu'une matrice diagonale est un cas particulier de matrice tridiagonale, de matrice triangulaire inférieure, et de matrice triangulaire supérieure.

Utiliser les principes de l'*héritage multiple* pour rendre compte de cette hiérarchie. On pourra par exemple implémenter les méthodes suivantes, en réfléchissant bien à la classe où elles doivent se trouver :

— surcharge de l'opérateur d'addition,

— calcul du déterminant de la matrice.

4. Pour aller plus loin

Dans certains cas, il est facile de résoudre un système linéaire, c'est-à-dire, étant donnés un vecteur colonne $b \in \mathbb{R}^n$ et une matrice carrée $A \in \mathcal{M}_n(\mathbb{R})$, trouver un vecteur colonne $x \in \mathbb{R}^n$ tel que $Ax = b$.

— Implémenter une méthode `solve` dans la classe **`MatriceDiagonale`** permettant de résoudre ce problème.

— Faire de même dans les classes **`MatriceTriangulaireInf`** et **`MatriceTriangulaireSup`**.