

Travaux Pratique (20%)

Vous devez définir une classe **Matrice** qui contient les méthodes suivantes :

Transposer ()

Transpose les valeurs de la matrice. Par exemple si la matrice contient les données suivantes :

1	2	3
4	5	6
7	8	9

On obtiendra ceci :

1	4	7
2	5	8
3	6	9

normaliser()

Rapporte toutes les valeurs de la matrice dans une intervalle entre 0 et 1. La méthode consiste à soustraire la valeur minimale à toutes les autres puis à diviser par la valeur maximale parmi les nouvelles valeurs. Par exemple :

1	2	3
4	5	6
7	8	9

On obtiendra ceci :

0	0.125	0.25
0.375	0.5	0.635
0.75	0.875	1

polariser()

Remplace toutes les valeurs négatives de la matrice par -1 et toutes les valeurs positives par 1. 0 reste à 0.

min()

Retourne la valeur minimale trouvée dans la matrice.

max()

Retourne la valeur maximale trouvée dans la matrice.

additionner(matrice2)

Additionne chacune des valeurs de la matrice passée en paramètre (matrice2) à chacune des valeurs de la matrice courante. Si les deux matrices sont de taille différente, aucune valeur n'est modifiée dans la matrice courante.

multiplier(n)

Multiplie chaque valeur de la matrice par le nombre passé.

getVal(i, j)

Retourne la valeur à un indice spécifique (désigné par les indices i et j) dans la matrice.

setVal(i, j, val)

Écrit dans la matrice la valeur passée à l'indice spécifié par i et j.

afficher()

Affiche à l'écran les valeurs contenues dans la matrice dans un format lisible. Afficher les valeurs avec 3 chiffres de précision (pas besoin d'arrondir). Par exemple la matrice suivante :

1.23333	0.454545454	0.1		1.233	0.454	0.100
1.999999	3.21	2.8		1.999	3.210	2.800
5.7	1.6666666666	2.87	s'affichera comme suit :	5.700	1.666	2.870

Constructeur

Le constructeur de la classe *Matrice* prend un seul paramètre (la taille de la matrice) et doit initialiser toutes les valeurs à 0. Cette taille représente à la fois le nombre de rangées et le nombre de colonnes :

donc si *taille* vaut 4, la matrice contiendra 16 valeurs; si *taille* vaut 5, elle en contient 25; etc.

A noter :

- La classe *Matrice* ne contient que des matrices carrées, i.e. qui contiennent autant de rangées que de colonnes.
- Les valeurs qu'elle contient doivent être de type *float*.
- Votre classe doit fonctionner avec le code fourni (*TestMatrice.py*) sans que vous ayez à modifier celui-ci.
- Toutes les méthodes et toutes les propriétés doivent être publiques.
- Vous pouvez définir des méthodes supplémentaires à celles qui sont spécifiées dans ce travail si cela peut vous aider.

À remettre

Vous devez remettre un fichier nommé *Matrice.py* par Omnivox.

Échéance :

Avant le 10 juillet 23 :55