

POA – Laboratoire Matrice

Objectif

Le but de ce laboratoire est d'implémenter en C++ des matrices et leur calculs (Addition, Soustraction, Multiplication) en Orienté Objet.

Conception

Classe **Matrix**

Classe pour la création des matrices, elle se compose de variables n et m pour la taille, modulo pour indiquer quelle valeur max peuvent avoir nos valeurs dans la matrice et d'un tableau de tableau nommé data.

Il y a 3 constructeurs dans notre classe, un privé et deux publiques. Le constructeur privé prend en paramètres n, m, modulo et un booléen pour indiquer si l'on veut générer les valeurs de la matrice aléatoirement, il permet de factoriser la création des matrices et est utilisé par les constructeurs publiques. Le premier constructeur public a les mêmes paramètres que le constructeur privé sauf le booléen en moins, mais fait appel à ce dernier avec « false » pour le booléen. Le troisième est le constructeur par copie de notre classe. Pour finir, il y a un destructeur qui s'occupe de désallouer le tableau data qui est alloué dynamiquement.

Aux niveaux des calculs, nous avons décidé d'utiliser des méthodes qui font appel à la classe **Operation**. Il y a trois méthodes différentes, une pour effectuer les calculs directement sur la matrice qui appelle celle-ci (*operation(...)*), ensuite une qui renvoie une nouvelle matrice allouée statiquement en mémoire et qui est renvoyée en valeur et non par référence (*operationByValue(...)*). Et une dernière, qui renvoie un pointeur sur une matrice allouée dynamiquement en mémoire (*operationByPointer(...)*). Chacune de ces méthodes prend en paramètres une matrix et un Object de type **Operation**. Pour finir, ces méthodes sont appelées par des fonctions qui effectuent les opérations arithmétiques demandées (addition, soustraction, multiplication).

Classe **Operation**

Pour éviter d'avoir beaucoup de méthodes et fonctions dans notre classe **Matrix**, nous avons créé une classe **Operation** qui exécute le calcul d'un opérande avec un autre. Chaque sous-classe de cette dernière correspond à une nouvelle opération arithmétique (addition, soustraction, multiplication, etc...).

Étant donné que les méthodes operation de matrice sont publiques, un utilisateur de la classe peut créer ses propres opérations en créant une classe qui hérite de Operation. Et les appliquer composante par composante sans modifier la classe **Matrix**.

Classe Random

Comme conseillé dans la documentation, nous avons créé une classe **Random** pour générer nous même les nombres aléatoires qui seront dans la matrice. Nous avons fait de la classe un singleton afin de n'avoir qu'un seul seed lors de l'instanciation de la classe.

Test

Constructeur/Destructeur

Test effectué	Résultat attendu	Résultat obtenu
Génération de matrice avec valeurs aléatoires	Ok	Ok
Génération de matrice avec n et/ou m égal ou inférieur à 0	Erreur	Erreur
Génération de matrice avec modulo égal à 0	Erreur	Erreur
Génération de matrice avec le constructeur par copie	Ok	Ok
Fonctionnement du destructeur	Ok	Ok

Operateur « = » et « << »

Test effectué	Résultat attendu	Résultat obtenu
Affichage des matrices avec l'opérateur de flux	Ok	Ok
Utilisation de l'opérateur d'affectation	Ok	Ok
Utilisation de l'opérateur d'affectation avec matrices de taille différente	Ok	Ok

Add

Test effectué	Résultat attendu	Résultat obtenu
Addition inline de matrices de même taille et modulo	Ok	Ok
Addition par valeur de matrices de même taille et modulo	Ok	Ok
Addition par pointeur de matrices de même taille et modulo	Ok	Ok
Addition inline de matrices de taille différente mais même modulo	Ok	Ok
Addition par valeur de matrices de taille différente mais même modulo	Ok	Ok
Addition par pointeur de matrices de taille différente mais même modulo	Ok	Ok
Addition inline de matrices de différente taille et modulo	Erreur	Erreur
Addition par valeur de matrices de différente taille et modulo	Erreur	Erreur

Addition par pointeur de matrices de différente taille et modulo	Erreur	Erreur
Addition de matrices de taille N x M et M x N, même modulo	Ok	Ok
Addition de matrices de taille N x M et M x N, différent modulo	Erreur	Erreur
Addition avec une matrice rempli de 0 et même modulo	Ok	Ok
Addition avec une matrice rempli de 0 et différent modulo	Erreur	Erreur

Sub

Test effectué	Résultat attendu	Résultat obtenu
Soustraction inline de matrices de même taille et modulo	Ok	Ok
Addition par valeur de matrices de même taille et modulo	Ok	Ok
Addition par pointeur de matrices de même taille et modulo	Ok	Ok
Addition inline de matrices de taille différente mais même modulo	Ok	Ok
Addition par valeur de matrices de taille différente mais même modulo	Ok	Ok
Addition par pointeur de matrices de taille différente mais même modulo	Ok	Ok
Addition inline de matrices de différente taille et modulo	Erreur	Erreur
Addition par valeur de matrices de différente taille et modulo	Erreur	Erreur
Addition par pointeur de matrices de différente taille et modulo	Erreur	Erreur
Addition de matrices de taille N x M et M x N, même modulo	Ok	Ok
Addition de matrices de taille N x M et M x N, différent modulo	Erreur	Erreur
Addition avec une matrice rempli de 0 et même modulo	Ok	Ok
Addition avec une matrice rempli de 0 et différent modulo	Erreur	Erreur

Multiply

Test effectué	Résultat attendu	Résultat obtenu
Addition inline de matrices de même taille et modulo	Ok	Ok
Addition par valeur de matrices de même taille et modulo	Ok	Ok
Addition par pointeur de matrices de même taille et modulo	Ok	Ok
Addition inline de matrices de taille différente mais même modulo	Ok	Ok
Addition par valeur de matrices de taille différente mais même modulo	Ok	Ok
Addition par pointeur de matrices de taille différente mais même modulo	Ok	Ok
Addition inline de matrices de différente taille et modulo	Erreur	Erreur
Addition par valeur de matrices de différente taille et modulo	Erreur	Erreur
Addition par pointeur de matrices de différente taille et modulo	Erreur	Erreur
Addition de matrices de taille $N \times M$ et $M \times N$, même modulo	Ok	Ok
Addition de matrices de taille $N \times M$ et $M \times N$, différent modulo	Erreur	Erreur
Addition avec une matrice rempli de 0 et même modulo	Ok	Ok
Addition avec une matrice rempli de 0 et différent modulo	Erreur	Erreur

Matrix

