Rapport labo 05 POO

# Modélisation UML

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

# Choix de conception

Nous avons décidé de créer 2 constructeurs différents pour la classe Matrix afin de couvrir les différents moyens d’instanciation de l’objet.

La fonction checkIfModsEquals renvoie void car elle jettera directement une exception si les modulos ne sont pas égaux. Elle est également privée car elle ne sera jamais utilisée en dehors de la classe Matrix.

L’interface Operator et ses sous-classes ont été crées afin d’être utilisées dans la fonction performOperation de Matrix afin de ne pas avoir besoin de créer une fonction différente dans Matrix pour chaque type de calcul.

Les fonctions doOperation des classes Addition, Substraction et Mulitplication vont toutes override la fonction de l’interface Operator.

# Tests et résultats

|  |  |  |  |
| --- | --- | --- | --- |
| **Numéro** | **Description du test** | **Résultat** | **Commentaires** |
| 1 | Instanciation d’une Matrice avec le constructeur utilisant la taille et le modulo | OK | La Matrice est bien générée avec des valeurs aléatoires correctes. |
| 2 | Instanciation d’une Matrice avec le constructeur utilisant la taille, le modulo et les valeurs. | OK |  |
| 3 | Instanciation d’une Matrice avec des valeurs plus grandes que le modulo | OK | Les valeurs sont adaptées. |
| 4 | Instanciation d’une matrice avec un tableau de valeurs ne correspondant pas à la taille spécifiée | OK | La Matrice remplace les trous par des 0 et ignore les valeurs en dehors de la taille spécifiée |
| 5 | Exécution d’une opération avec des Matrices de même modulo. | OK | Les opérations sont correctes. |
| 6 | Exécution d’une opération avec des Matrices de modulos différents. | OK | Le système indique que les modulos ne sont pas égaux et renvoie « null » à chaque opération. |
| 7 | Exécution d’une opération avec des Matrices de même taille. | OK | Les opérations sont correctes. |
| 8 | Exécution d’une opération avec des Matrices de tailles différentes. | OK | La taille de la Matrice de résultat est adaptée et les opérations sont correctes. |

Output des tests :

Test numéro 1 :

The modulus is 5

one :

0 3 2

3 4 4

4 1 3

Test numéro 2 : (int[][] valuesTest = {{1, 2, 3, 4}, {1, 2, 3, 4}})

The modulus is 5

one :

1 2 3 4

1 2 3 4

Test numéro 3 : (int[][] valuesTest = {{1, 2, 3, 8}, {1, 7, 3, 4}}):

The modulus is 5

one :

1 2 3 3

1 2 3 4

Test numéro 4 : (int[][] valuesTest = {{1, 2, 3}, {1, 4}})

M = 5 N = 5

The modulus is 5

one :

1 2 3 0 0

1 4 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

Test numéro 5 :

The modulus is 5

one :

1 2 3 4

1 2 3 4

two :

1 2 3 4

1 2 3 4

one + two:

2 4 1 3

2 4 1 3

one - two:

0 0 0 0

0 0 0 0

one x two:

1 4 4 1

1 4 4 1

Test numéro 6:

one + two:

The modulus are not equal

Test numéro 7: Prouvé dans le test 5

Test numéro 8:

The modulus is 5

one :

1 2 3

1 2 3

two :

1 2 3 4

1 2 3 4

one + two:

2 4 1 4

2 4 1 4

one - two:

0 0 0 1

0 0 0 1

one x two:

1 4 4 0

1 4 4 0

# Listing du code

Le listing du code est présent à la fin de ce document

# Annexes

Le code source de ce projet est disponible dans le répertoire « Code source ».