Programmation orientée objet

Classe C

Rapport du laboratoire n°5 : Matrices

# Notes

## Gestion des exceptions

Nous avons décidé d’englober notre programme dans un bloc try-catch, afin d’afficher la stack trace des exceptions qui auraient pu survenir dans ce dernier. Dans cette optique, nous n’avons volontairement pas vérifié si des objets donnés en paramètres de nos méthodes étaient null. En effet, une NullPointerException, ou NumberFormatException étant des sous-classes de RuntimeException, nous traitons toujours ces cas.

Le seul cas différent est dans la méthode StringArrayToIntArray de la classe Util, car nous estimons que le message de l’exception pouvait être plus explicite.

## Classe Util

Nous avons décidé de créer une classe Util contenant une collection de méthodes statiques utilitaires, car elles pourraient être utilisées dans n’importe quelle classe de manière générale.

## Matrices vides et constructeur par défaut

Nous n’avons pas reçu de consignes particulières concernant la construction ou l’utilisation de matrices vides. Nous avons donc décidé d’interdire la construction de tels objets. C’est pour cette raison que nous n’avons pas implémenté de constructeur vide. De plus, nous estimons que dans notre cas d’utilisation, des matrices vides n’ont pas de sens d’exister.

## Tests unitaires

Nous avons conçu nos tests unitaires avec JUnit et Maven. Pour les exécuter, il faut posséder le ficher pom.xml et à son emplacement exécuter la commande « mvn clean test ». Maven peut être installé suivant ce [guide](https://maven.apache.org/install.html).

Nous avons nommé nos tests de manière à ce que l’on sache exactement l’objectif du test, comme appris au cours DAI.

Le but de la classe « ProgramTest » est de visualiser sur la console le résultat d’appel de notre programme principal avec différent arguments.

## Modélisation UML

Nous n’avons pas modélisé les classes « Program », « Util », « ProgramTest », et « MatrixTest », car cela apporte peu à la compréhension du fonctionnement de notre programme.

# Diagramme de classe

