

# Rapport du TD n°09 Java avancé

Guillaume CAU  
29/11/2019

## 1) Question 01 :

Pour trouver une case  $i,j$  dans un tableau à une seule dimension de taille  $i_{\max} * j_{\max}$ , on utilise la fonction suivante :

$$f: \mathbb{N}^2 \rightarrow \mathbb{N}$$

Définie par :

$$f(i, j) = a * i + j$$

C'est un peu comme si l'on utilisait un nombre en base  $a$  et que le chiffre des dizaine représente le numéro de la ligne et que le chiffre des unité représente la colonne.

## 2) Question 02 :

C'est à cause de l'erasure que nous ne pouvons pas créer directement un tableau de  $T$ . Les éléments de  $T$  sont vu comme des Object à la compilation. On doit donc créer un tableau d'objets à la place.

On peut supprimer le Warning car un tableau d'Object pourra de tout manière stocker des  $T$ .

## 3) Question 03 :

On peut remarquer que la classe *MatrixGraph* n'apporte pas de nouvelles méthodes par rapport aux méthodes de l'interface *Graph* donc il n'est pas nécessaire que la classe *MatrixGraph* soit publique.

En effet, il n'est pas nécessaire que la classe *Graph* soit visible depuis l'extérieur, les utilisateurs peuvent directement utiliser la classe *MatrixGraph*.

#### 4) Question 04 :

En Java, nous devons régulièrement renvoyer des valeurs à la fin de nos fonctions. Le problème est que, dans certains cas, la fonction peut retourner un résultat ou non. Si la fonction ne renvoie pas de résultat, on peut renvoyer **null**. Le problème est que, dans certains cas, on peut oublier de tester si la valeur est null et ainsi lever un *NullPointerException*.

Le Java fournit un objet Optional qui permet, de façon sécurisé, de contenir un élément vide ou plein. De cette manière, il est impossible de directement accéder au contenu. Le *NullPointerException* n'est donc plus possible.

#### 5) Question 05 :

La méthode `addEdge` insère l'argument passé en paramètre à la position  $f(i, j)$  du tableau.

#### 6) Question 06 :

La méthode `edge` doit exécuter le `EdgeConsumer` passé en paramètre sur chaque arête du sommet passé en paramètre.

#### 7) Question 07 :

Dans la question 7, l'objectif était de développer l'itérateur de l'interface. Un itérateur parcourt une liste ou un tableau.

## Conclusion :

Ce TP m'a permis d'apprendre à utiliser plusieurs notions. Tout d'abord, je me suis amélioré dans l'utilisation et la création des itérateurs. J'ai également compris comment fonctionnait un Spliterator, mais je n'ai pas pu l'implémenter dans cet TP. Je n'ai pas eu le temps de m'attarder dessus.