École Polytechnique de Montréal

Département de génie informatique et génie logiciel

LOG2810 : Structures discrètes

Tp1 : Graphes – Hiver 2018

Soumis par :

Syphax Ait Yahia 1776868

Georges Louis 1880098

Bassam Ajam 1777689

Groupe 01

03 mars 2018

Table des matières

I. Introduction

Développer un algorithme permettant de trouver le plus petit chemin sur une carte sous certaines conditions et cela tout en gérant les cas exceptionnels. Dans notre cas, on a utilisé la programmation orientée objet avec Java, et on s'est familiarisé avec l'algorithme de Djikstra et on l'a utilisé pour réaliser ce tp.

II. Présentation des résultats avec explications

Dans notre cas, pour réaliser ce genre de travail nous avons développé six classes.

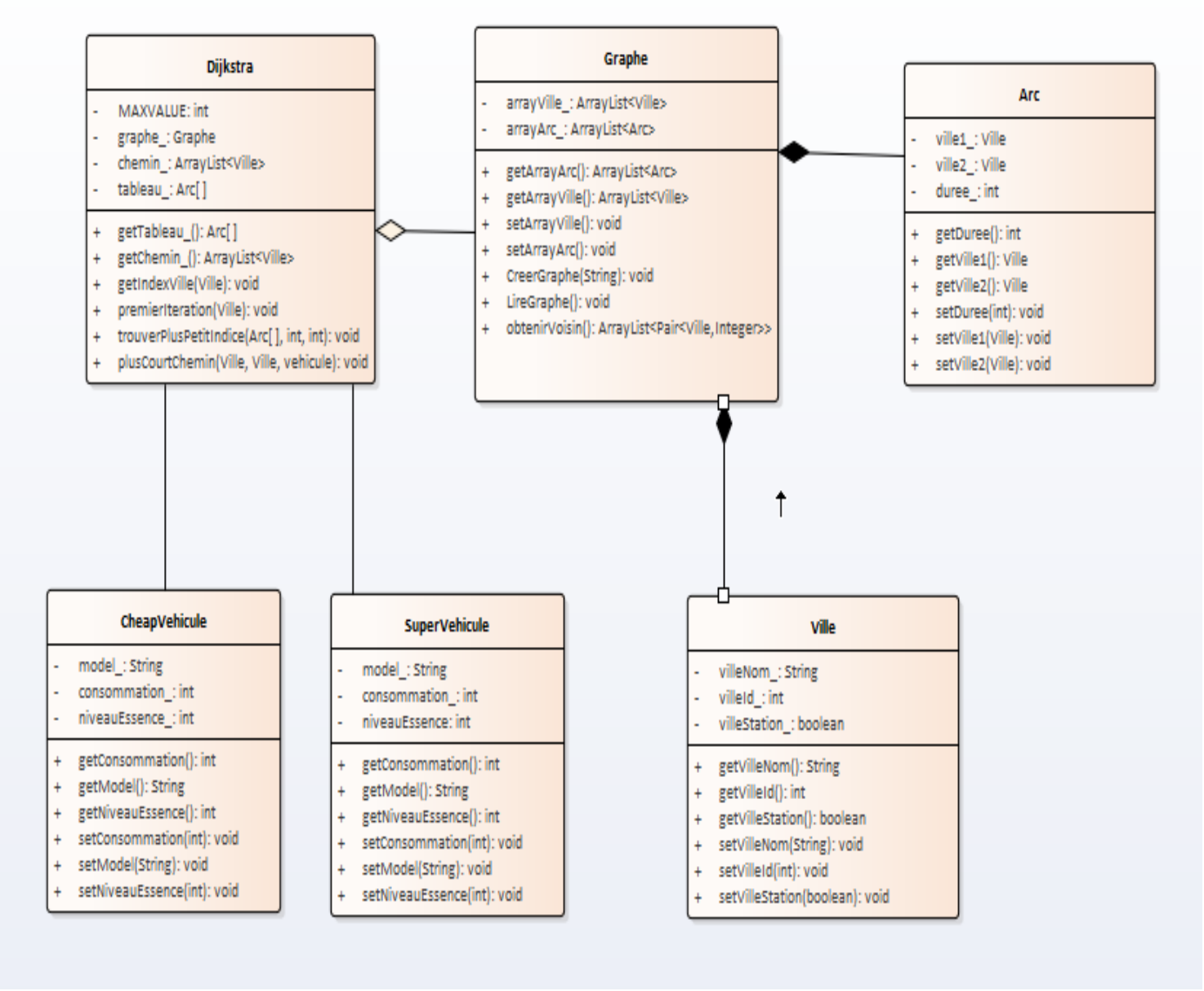
- Classe Ville : Elle permet de créer des instances villes contenant l'id de la ville, le nom et la présence d'une station d'essence.
- Classe Arc : Elle permet de représenter un arc qui est constitué de deux villes et une arête.
- Classe Graphe : Elle est constitué d'un tableau de ville et d'un tableau d'arc. Elle permet de représenter un graphe, elle contient des méthodes permettant de créer un graphe à partir d'un fichier et de lire un graphe c'est-à-dire afficher toute les villes et tous les arcs contenus dans ce graphe.
- Classe CheapVehicule et SuperVehicule : Elles permettent de représenter les différents véhicules présents dans l'énoncé avec leur différentes spécifications (consommation, niveau d'essence… etc.).
- Classe Djikstra : Elle composé d'un graphe qui représente les données de la carte, d'un tableau d'arc qui nous permet de trier les villes selon leur éloignement à une ville donnée par l'utilisateur à l'aide de notre plus importante fonction qui est « trouverPlusCourtChemin () ». Elle contient un tableau de villes qui nous permet de déterminer le chemin à emprunter entre une ville et une autre pour avoir le chemin le plus court.

Clarification de la fonction plusCourtChemin () :

- Elle prend deux villes en entrées et à l'aide de l'algorithme de Djikstra elle permet de trouver le plus court chemin entre elles.
- Elle permet d'afficher ce chemin.
- Elle effectue le test pour savoir si c'est possible d'utiliser un véhicule de type CheapCar pour arriver au but voulu, sinon elle effectuera le test avec un véhicule de type SuperCar. Si le résultat est négatif dans les deux cas, elle nous avertira que le but ne pourrait être atteint.

III. Diagramme de classes

Figure 1 : Diagramme de classes



IV. Difficultés rencontrées

Le travail pratique est trop long, et les exigences sont trop complexes surtout ce qui suit :

« À cet effet, l'algorithme doit éviter de donner à Joe un chemin qui fait passer le niveau d'essence de son véhicule en dessous de **12%**. Si c'est le cas, on préférera une autre chemin plus coûteux en termes de temps mais plus sécuritaire pour la réussite de fuite. Si une telle trajectoire n'existe pas, c'est seulement à ce moment-là qu'on utilisera un véhicule de chez "Super Car". Si les mêmes conditions ne sont pas respectées non plus pour le véhicule de chez "Super Car", on ne devra pas effectuer le braquage. »

Car un plus petit chemin il en existe qu'un seul, par contre un chemin plus couteux en terme de temps il pourrait y en avoir une infinité, c'est pour cela que la tache de trouver un chemin devient très difficile. Nous trouvons que nous manquons de temps pour réaliser un algorithme d'une tel complexité.
C'est-à-dire :

1- Trouver le plus court chemin
2- Effectuer les tests avec le véhicule cheap Car
3- Vérifier s'il y a une panne sèche
4- Chercher le deuxième plus court chemin
5- Effectuer le test à nouveau…ainsi de suite
6- Si à la fin de toute les recherche il y a toujours une panne sèche refaire les étapes de 1 à 5 avec un véhicule de type SuperCar
7- S'il y a toujours une panne sèche alors le braquage est annulé

V. Conclusion

Ce travail pratique est le premier travail où nous avons la liberté de structurer notre code comme nous le voulons, sans se baser sur un ancien code et prendre la responsabilité de notre code source. C'était vraiment long et nous avons manqué un peu de temps pour mener à bien ce travail. Nous espérons qu'au prochain tp nous aurons plus de temps pour le compléter.