

# Rapport de POO

## Rapport POO Version 1

Manière de lancement : **Run** du debugger java

Diagramme UML avec *StarUML* :

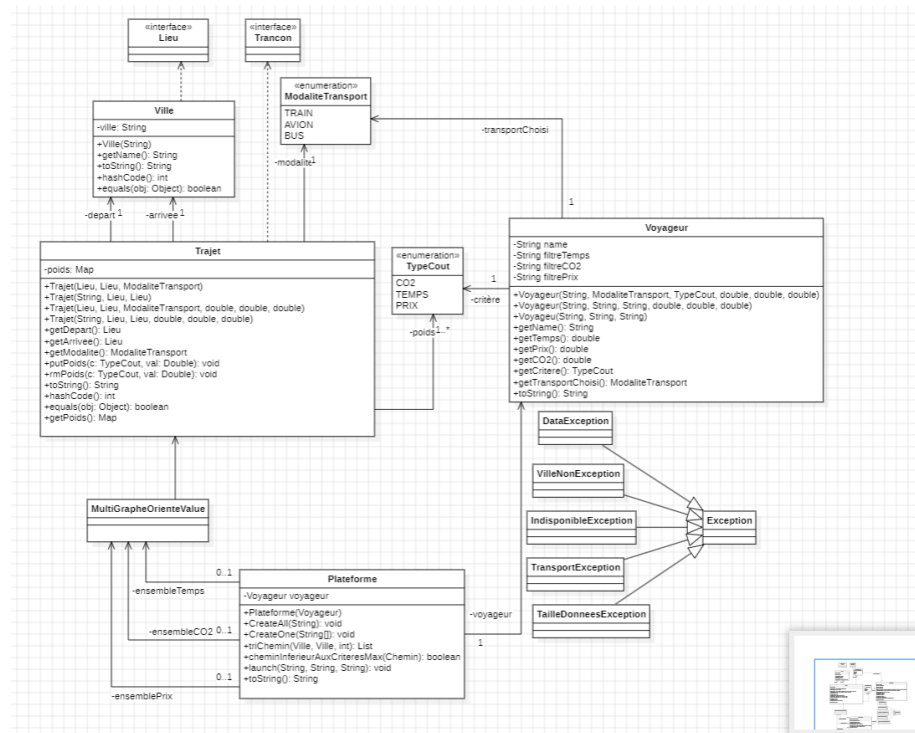


Figure 1: umlv

### Analyse technique

Dans cette version nous avons pris en compte des différentes class et interface mise à disposition afin de les utilisés dans notre programme java. Nous avons utiliser la librairie de graphe pour calculer les chemins les plus courts en prenant un type **CO2**, **Prix** ou **Temps**.

La class Plateforme.java nous permet de calculer les chemins les plus courts en prenant compte du critère choisi.

On a implémenté des exceptions dans notre code java afin de mieux les traiter en cas d'erreur.

Les class de test **UsePlateforme.java** et **UsePlateformeAlice.java** , nous permet de tester les fonctionnalité de la version 1 de la SAE, c'est à dire prendre en compte un ville de départ, arrivé et un critère. Grace à ces testes on peut vérifier le bon fonctionnement du code quand l'utilisateur choisi un critère et que les deux autres critère ne seront pas prisent en compte.

Manière de lancement : **Run** du debugger java

[illegible]

Dans cette personne nous avons mis en place une methode scanner dans Plateforme.java afin de pouvoir lire les données CSV tout en passant par une vérification des donnés et si jamais la vérification échoue, le code renvoie une exception. L'ajout de la multi modalite permet de mettre en place les correspondances entre

La class de test UsePlateformeAlice.java a été modifié pour couvrir les nouvelles implementation du scanner et de pouvoir vérifier la multi modalite étant donné qu'on a ajouté des critère de correspondance dans les paramètres du constructeur de Voyageur.java.

# Rapport POO Version 3

Manière de lancement : **Run** du debugger java

Diagramme UML avec *StarUML* :

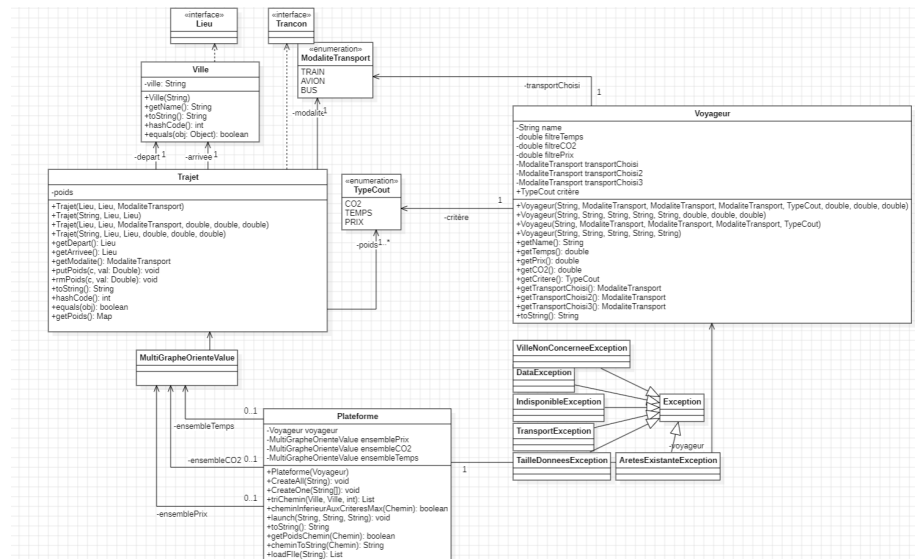


Figure 3: umlv3

## Analyse technique

On a mis en place une historique via la s rialisation des objets dans Plateforme.java, la methode n'a pas  t  assez developp  pour pouvoir l'utiliser pleine-

ment dans le code notamment dans l'application graphique.

### **Analyse quantitative/qualitative des tests**

La même class de test `UsePlateformeAlice.java` sera utilisé pour tester la serialisation.