Saé 2.01+2.02 : dev d’une application et exploration algorithmique

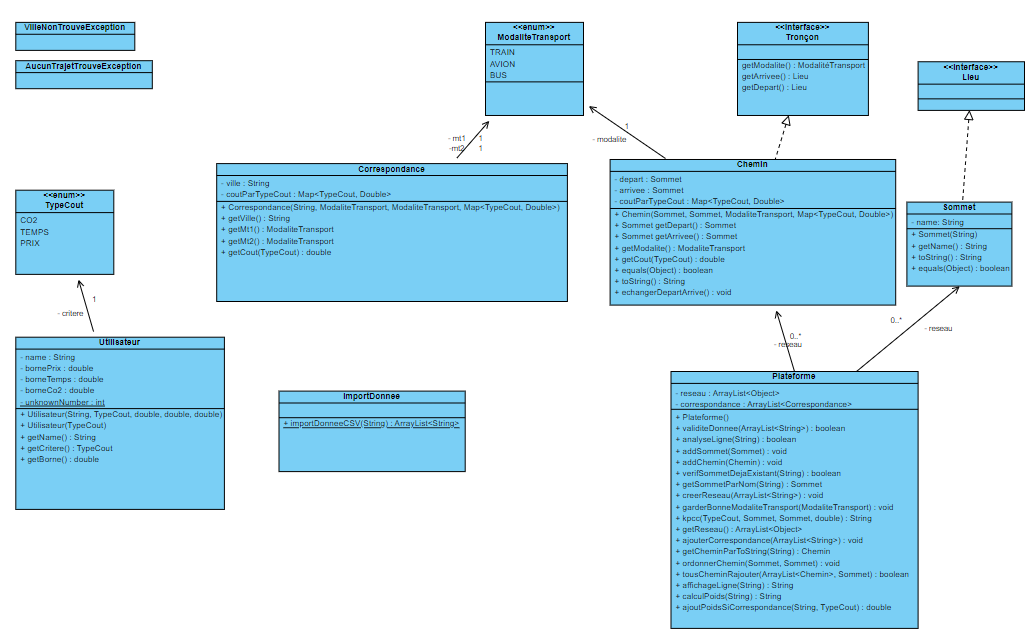
Rapport de la Saé Version 2

### 1. Introduction

Ce projet a pour objectif de développer une application permettant d'optimiser les trajets multimodaux en termes de coût, temps et émissions de CO2. L'application prend en compte différents modes de transport tels que le train, l'avion et le bus.

### 2. Description Générale du Système

L'application est structurée autour de plusieurs classes principales, chacune ayant un rôle spécifique dans le traitement et l'optimisation des trajets. Les classes principales incluent Utilisateur, Correspondance, Chemin, Sommet, Plateforme, et ImportDonnee. Ces classes collaborent pour modéliser le réseau de transport et fournir les itinéraires optimaux selon les critères de l'utilisateur.



### 3. Description des Classes Principales

#### Classe Utilisateur

* **Rôle**: Gérer les préférences de l'utilisateur en matière de coût, temps, et émissions de CO2.
* **Pourquoi l'utiliser**: Cette classe permet de centraliser et de gérer facilement les préférences de l'utilisateur, ce qui est essentiel pour fournir des résultats personnalisés.

#### Classe Correspondance

* **Rôle**: Représenter les correspondances entre différents modes de transport dans une ville.
* **Pourquoi l'utiliser**: Elle facilite la gestion des transitions entre différents moyens de transport, permettant ainsi une meilleure optimisation des trajets.

#### Classe Chemin

* **Rôle**: Modéliser un trajet entre deux villes avec une certaine modalité de transport.
* **Pourquoi l'utiliser**: Cette classe est essentielle pour représenter les segments de trajet et calculer les coûts associés, ce qui est crucial pour l'optimisation.

#### Classe Sommet

* **Rôle**: Représenter une ville dans le réseau de transport.
* **Pourquoi l'utiliser**: Elle permet de modéliser les points de départ et d'arrivée des trajets, simplifiant ainsi la structure du réseau.

#### Classe Plateforme

* **Rôle**: Gérer l'ensemble du réseau de transport, y compris les villes (sommets) et les trajets (chemins).
* **Pourquoi l'utiliser**: Cette classe centralise la gestion du réseau, facilitant l'ajout, la suppression et la modification des trajets et des correspondances.

#### Classe ImportDonnee

* **Rôle**: Importer et valider les données du réseau de transport à partir d'un fichier CSV.
* **Pourquoi l'utiliser**: Elle automatise le processus de chargement des données, assurant leur cohérence et leur intégrité, ce qui est essentiel pour des calculs précis.

### 4. Nouvelles Fonctionnalités et Améliorations

Par rapport à la version précédente, cette nouvelle version inclut plusieurs améliorations et nouvelles fonctionnalités :

#### Ajout des Exceptions

* **VilleNonTrouveException** et **AucunTrajetTrouveException**
  + **Rôle**: Gestion des erreurs spécifiques, augmentant la robustesse du système en fournissant des messages d'erreur clairs et adaptés.

#### Méthodes de Calcul Optimisées

* **getCout dans Chemin et Correspondance**
  + **Rôle**: Calcul précis et optimisé des coûts pour chaque type de coût, permettant une évaluation rapide des options de trajet.

#### Amélioration de la Plateforme

* **Validation et Analyse des Données**
  + **Rôle**: Les méthodes validiteDonnee et analyseLigne assurent la validation rigoureuse des données, garantissant que seules des données correctes et formatées de manière appropriée sont utilisées.
* **Algorithme KPCC**
  + **Rôle**: Implémentation de l'algorithme KPCC pour trouver les chemins optimaux, assurant des solutions efficaces pour les utilisateurs.

#### Importation des Données

* **ImportDonnee**
  + **Rôle**: Facilite l'importation de données CSV, rendant l'application plus flexible et capable de s'adapter à différents ensembles de données.

### 5. Conclusion

Cette version 2 de l'application d'optimisation des trajets multimodaux introduit des améliorations significatives par rapport à la version précédente. Les nouvelles fonctionnalités et les optimisations apportées rendent le système plus robuste, flexible et efficace. Les classes et les méthodes améliorées permettent une gestion plus fine des préférences de l'utilisateur et des données du réseau de transport, assurant des résultats optimaux et personnalisés



Prototype basse fidélité de l’application pour notre Saé.