

INSA

INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
LYON

C++ - Les bases

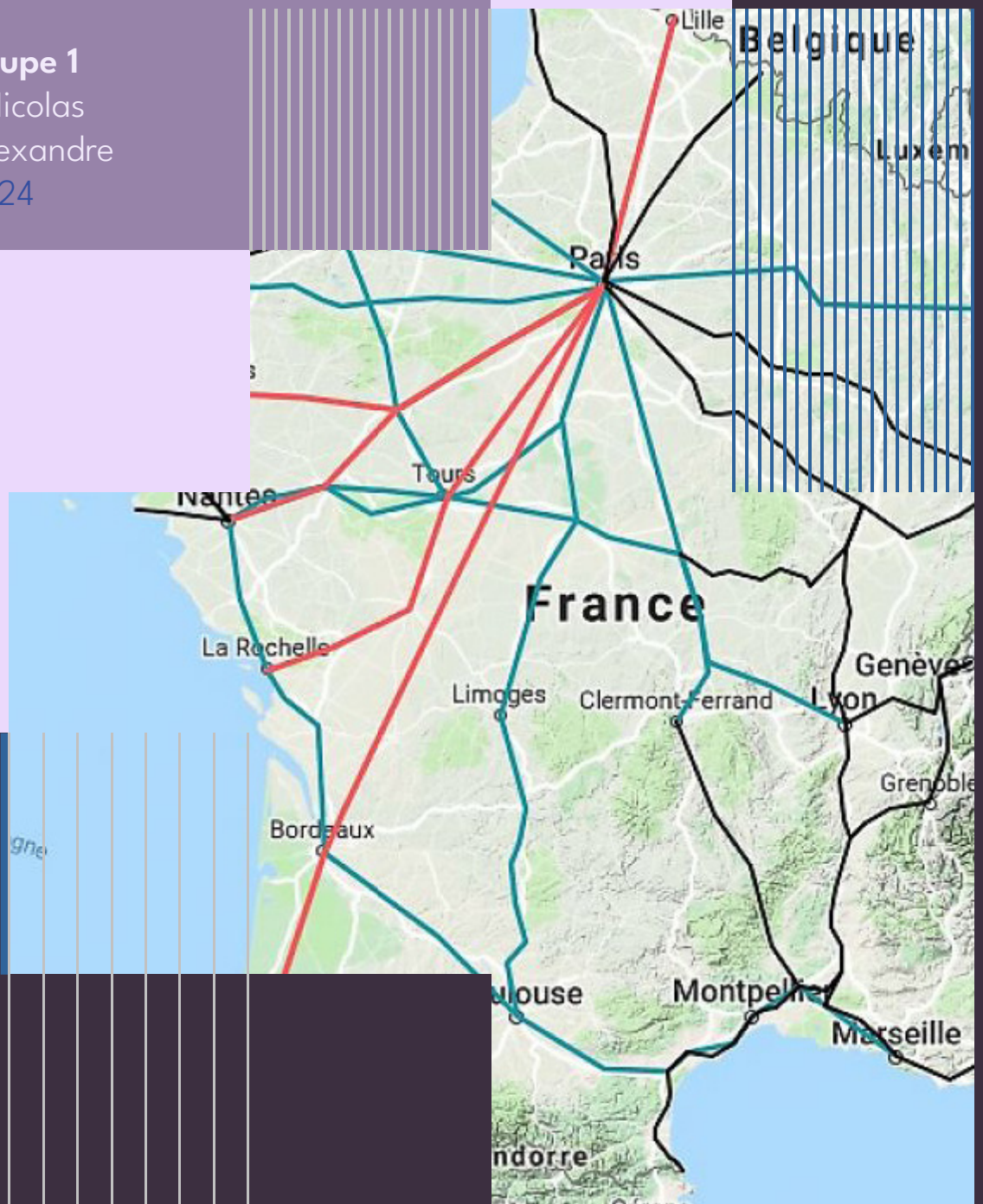
Compte-rendu de TP

3IF - Groupe 1

PUTON Nicolas

DIDIER Alexandre

18/12/2024



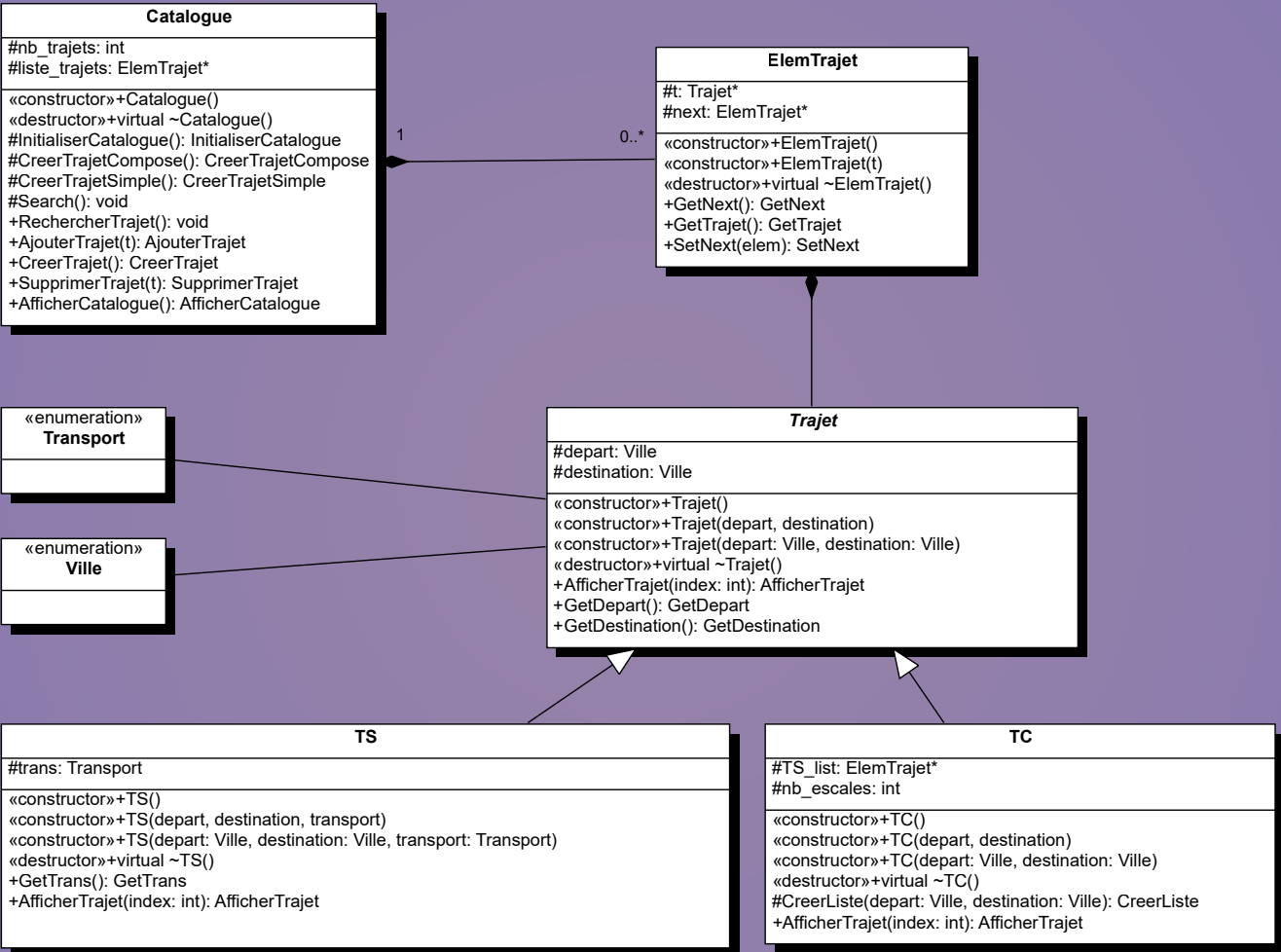
TP°2 - Catalogue de recherche de trajets

Dans ce TP, nous devons créer une application permettant la recherche de trajets dans un catalogue. Ces trajets peuvent être simples ou composées (c'est à dire avec des escales) et se font à bord d'un moyen de transport défini pour chaque trajet.

Le code est donné avec ce document au format .zip, mais il est également disponible sur GitHub :

<https://github.com/alex-ddr/TrajetsCPP.git>

1 Diagramme de classe



Nous avons donc implémenté plusieurs classes, dont une utilisant de l'héritage, ainsi que des **enums** pour la liste des villes et moyens de transports disponibles. Au niveau du stockage, nous avons décidé d'utiliser une **liste chaînée** contenant les différents Trajets. Cette structure de données sera utilisé autant dans le catalogue que pour le stockage des trajets d'un trajet composé.

2 Description des classes

Catalogue

La classe est le point d'entrée de notre programme. Elle est instanciée dans le main d'où les entrées de l'utilisateur vont directement appeller des méthodes sur le catalogue. Il s'agit donc de l'intermédiaire entre les entrées et la base de données.

Cette classe va donc inclure un **pointeur sur notre liste chaînée** (dont nous détaillerons le fonctionnement plus tard), un **compteur du nombre de trajets** et plusieurs méthodes permettant (suivant les choix de l'utilisateur) de :

- Afficher les trajets du catalogue
- Ajouter un trajet dans le catalogue
- Rechercher un trajet dans le catalogue
- Supprimer un trajet présent dans le catalogue

Trajet

Cette classe représente un trajet entre une ville de départ et une ville de destination. Cependant, cette classe n'est jamais instanciée, il s'agit d'une **classe abstraite** qui nous permet de traiter de la même manière les deux classes TS et TC représentant respectivement les trajets simples et les trajets composés.

Nous avons fait ce choix car peut importe qu'il s'agisse d'un trajet simple ou composé, il aura **toujours une ville de départ et une ville de destination** et faire les **mêmes actions**, à savoir afficher les étapes du trajet et le(s) moyen(s) de transport. La méthode AfficherTrajet est donc une **méthode virtuelle** que l'on va redéfinir dans les classes TS et TC, pour adapter l'affichage aux besoins de ces classes.

TS

La classe TS hérite de la classe Trajet, et il s'agit clairement du **cas le plus simple**, car un trajet simple est seulement composé de ses villes de départ et des destination et de son moyen de transport.

Nous avons donc juste ajouté un **attribut pour stocké le moyen de transport** et une méthode pour le récupérer.

La méthode AfficherTrajet a aussi été adaptée pour afficher l'affichage standard de la classe parent et afficher à la suite le moyen de transport.

TC

La classe TC hérite elle aussi de la classe Trajet, mais les choses sont plus corsées car un trajet composé est composé de plusieurs trajets simples.

Nous avons donc utiliser notre structure de **liste chaînée** pour stocker les différents trajets simples constituant le trajet composé. Les différents trajets entrées par l'utilisateur lors de la création d'un trajet composé sont donc **successivement transformées en TS et stockés dans la liste**.

Nous avons aussi adapté la méthode AfficherTrajet, qui itère à travers la liste et appelle la méthode AfficherTrajet de TS.

ElemTrajet

La classe ElemTrajet joue un rôle primordiale puisqu'elle est utilisée aussi bien par notre classe Catalogue que notre classe TC. Il s'agit d'une structure de liste chaînée classique, mais adaptée en classe.

Chaque cellule ElemTrajet contient un pointeur sur un trajet alloué dynamiquement sur le tas, ainsi qu'un pointeur vers la cellule suivante. Nous avons aussi créer deux getters pour accéder à ces attributs, et un setter pour définir le pointeur sur la cellule suivante, car nous en avons besoin pour l'ajout de trajet depuis la classe Catalogue.

Exemple

Essayons de visualiser le fonctionnement de notre structure de données avec ces données :

TS1

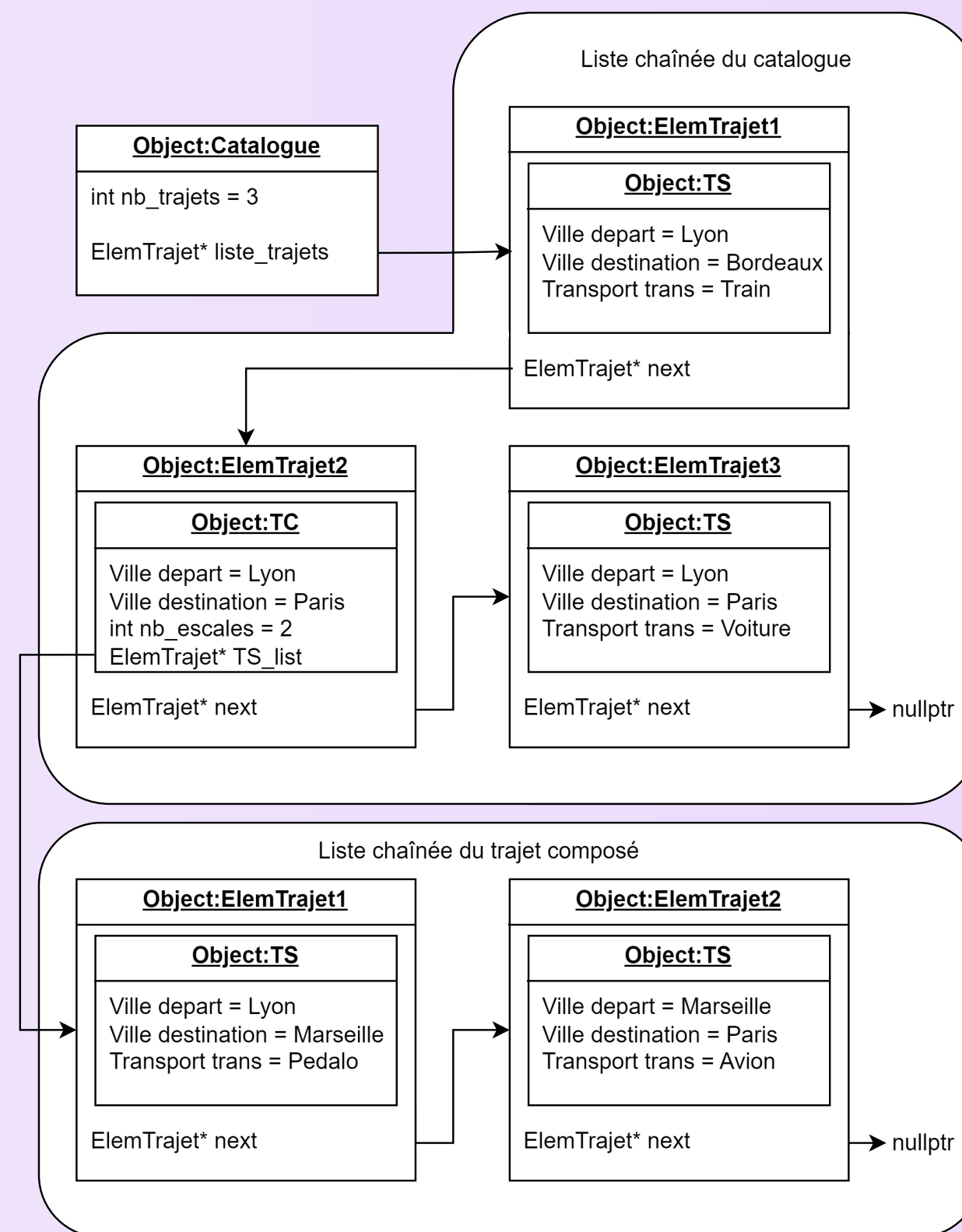
Lyon - Bordeaux
Train

TS3

Lyon - Paris
Auto

TC2

TS2-1 Lyon - Marseille | Bateau
TS2-2 Marseille - Paris | Avion



Conclusion

Ce TP a été une bonne initiation à la programmation orientée objet et surtout au concept d'héritage. Malheureusement, nous avons passé davantage de temps sur la gestion de la structure de données et surtout sur l'algorithme de recherche de trajet qui est assez compliqué puisqu'il s'agit d'un algorithme de Depth-First Search/Backtracking. Le seul point que nous voudrions d'ailleurs bien améliorer concerne cet algorithme puisque dans sa version courante il ne regarde pas les villes intermédiaires dans les trajets composés, mais seulement les villes de départ et de destination finale, et donc certains trajets ne peuvent pas être affichés.



INSA LYON

Campus de la Doua
69100, Villeurbanne

Tél : +33 (0)4 72 43 83 83

www.insa-lyon.fr



Visuel inspiré de la “Charte Graphique Écoles” V2.3
Par Keyne PEKAR & Nicolas PUTON

Sous l'accord de Céline Authemayou et Daphné Loquet-Caffiaux