

C++: HERITAGE - POLYMORPHISME

COMPTE RENDU DE TP

Table des matières

1, Description détaillée des classes	2
2, Description détaillée de la structure de données utilisée	3
3, Problèmes rencontrés et axes d'évolution	4

1. Description détaillée des classes

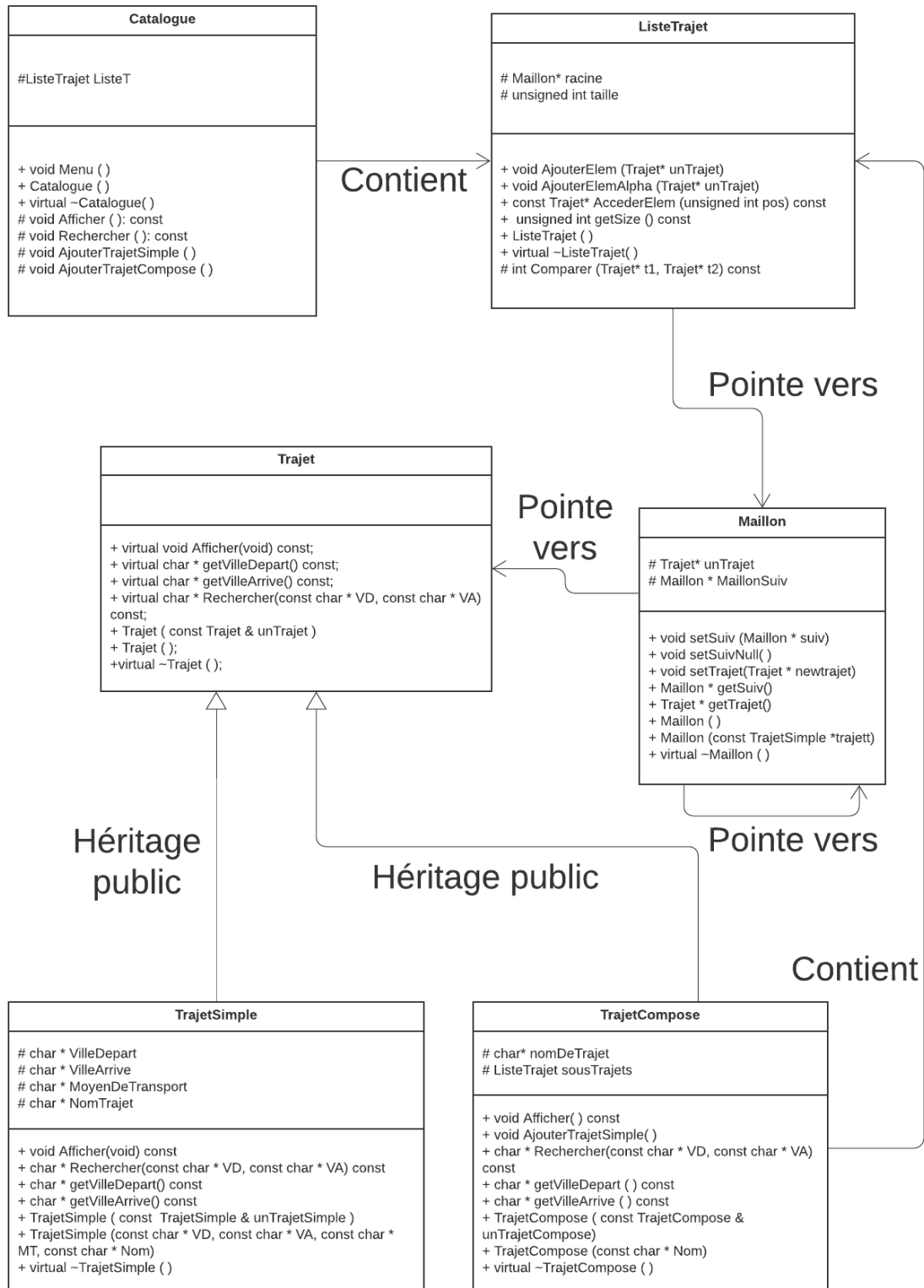
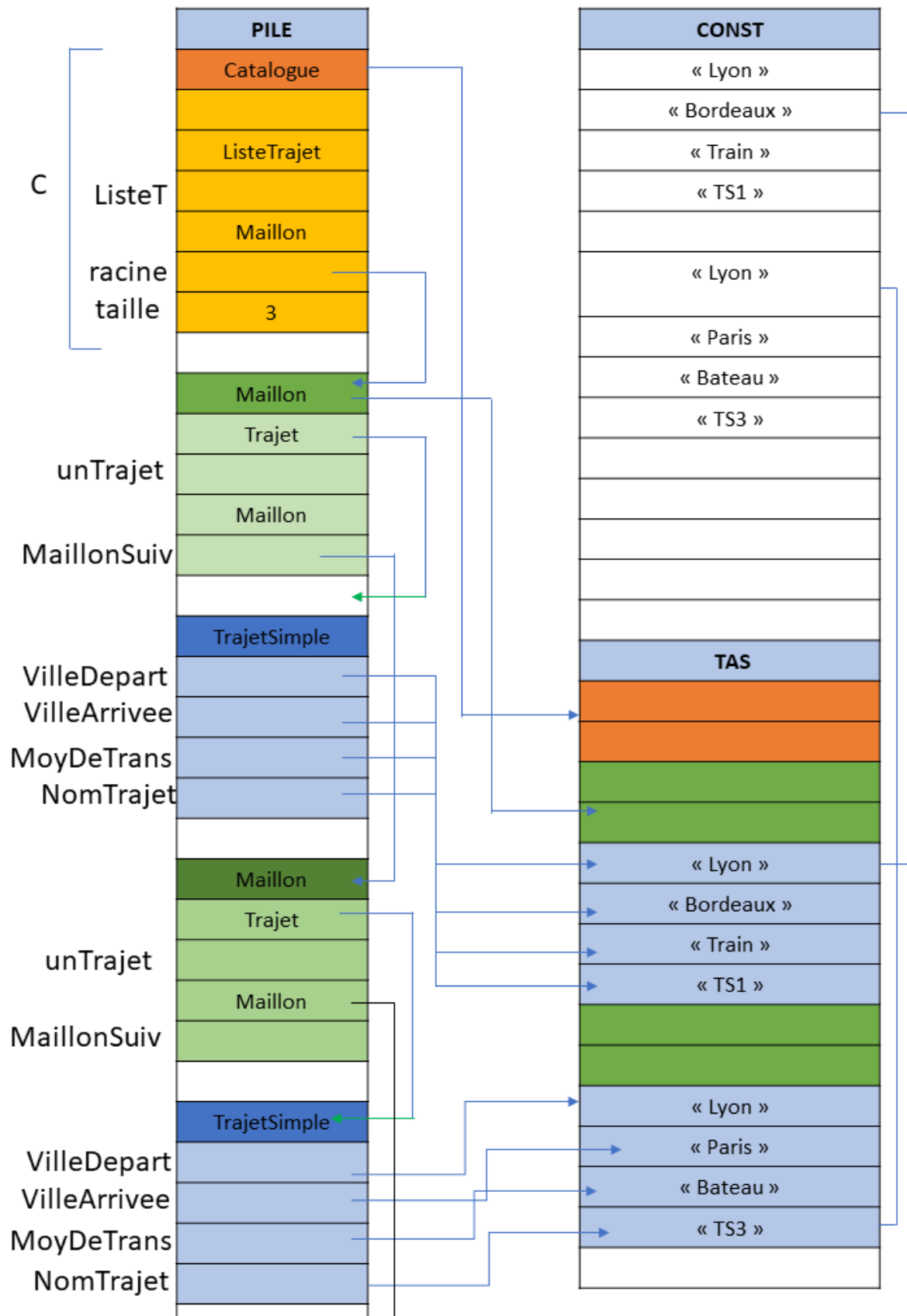
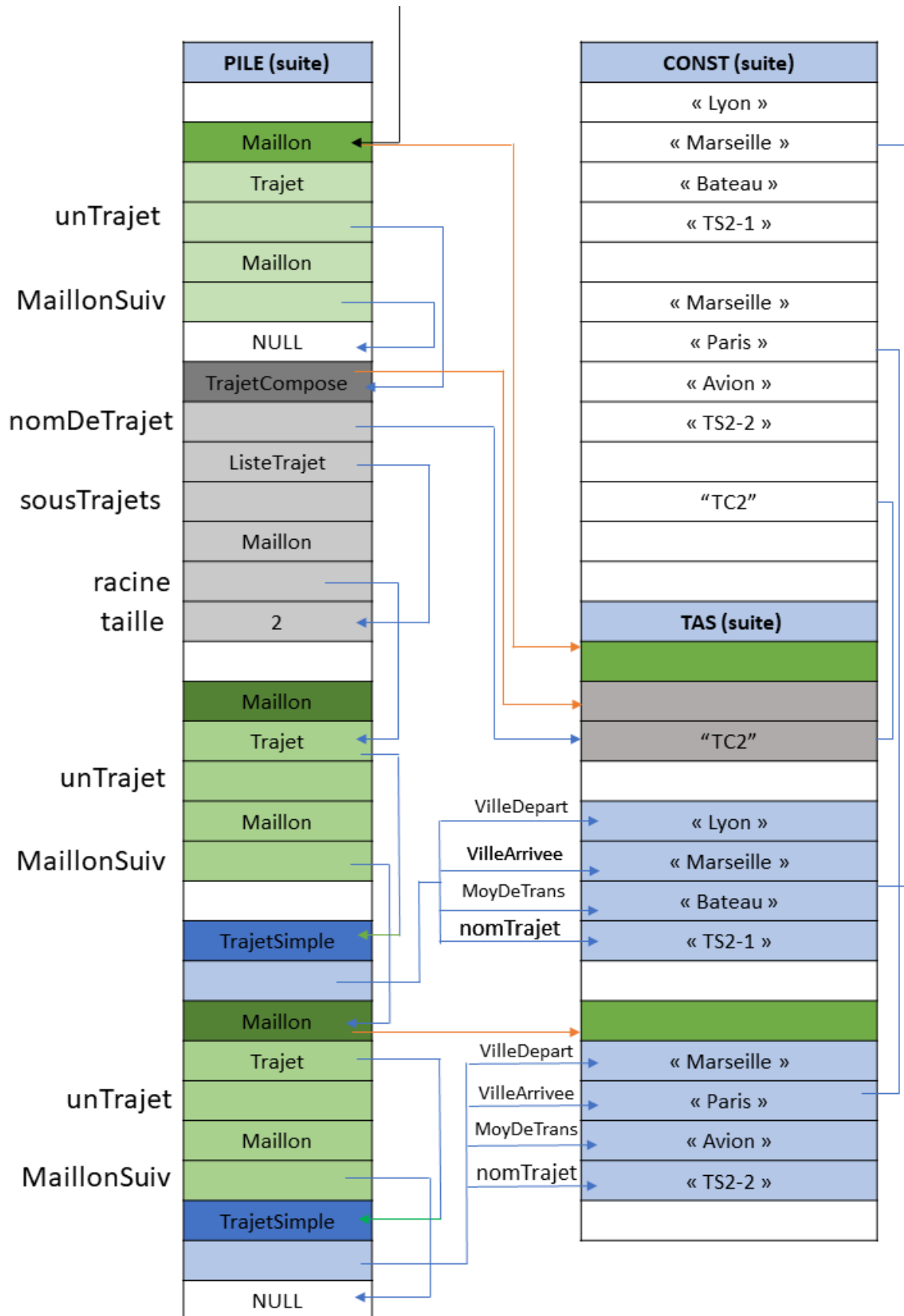


Diagramme des classes UML

2. Description détaillée de la structure de données utilisée





3. Problèmes rencontrés et axes d'évolution

Pour réaliser cette application, nous avons tout d'abord dû réfléchir à une structure de données afin de créer une collection ordonnée de trajets. Nous nous sommes donc rapidement dirigé vers la création d'une liste chaînée qui nous permet de facilement ordonner les trajets de manière optimal, par exemple, avec une liste chaînée chaque trajet contenu dans la liste va permettre de connaître le suivant, ce qui peut être intéressant pour les trajets composés puisque les trajets simples les composants doivent être dans l'ordre chronologique.

En conclusion de ce TP, nous estimons avoir réalisé l'entièreté de l'application demandée, en omettant la recherche avancée, mais nous aurions pu aller plus loin. En effet, l'objectif de ce TP était fixé sur l'héritage et la gestion dynamique de la mémoire, mais dans la réalité, notre application est trop simple. L'interface est énormément simplifiée par rapport à une application réelle et ne satisferait pas les potentiels utilisateurs, ou encore certaines fonctions vitales pour ce genre d'application ne sont pas présentes, comme la suppression d'un trajet une fois que celui-ci s'est déroulé.