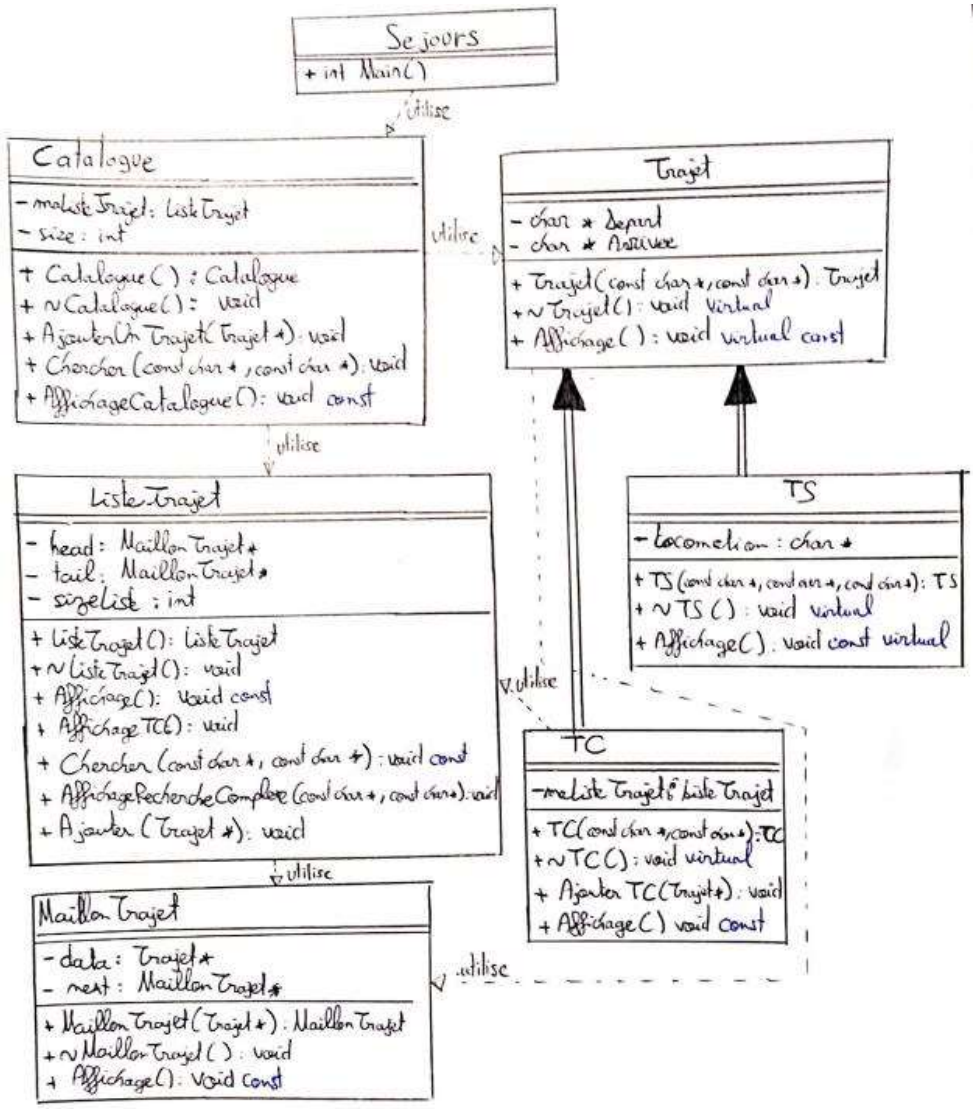
FLANDRE Corentin – THOMAS Colin (B3207)

TPC++ n°2 : Héritage-Polymorphisme

Constitution de voyages à partir d'un catalogue de trajets



Arbre d'héritage de notre code source

I. Conception

Nous cherchons, dans ce projet, à permettre à l'utilisateur de créer des trajets, de les garder dans un catalogue et de les rechercher. Ces trajets peuvent être simples, composés d'un départ, une arrivée, et un moyen de transports, ou compliqués, c'est-à-dire composé de plusieurs trajets simples.

Ainsi, nous avons créé un Catalogue, qui implémente une liste dynamique de trajets.

- Catalogue implémente une ListeTrajet. Il n'implémente que des méthodes qui utilisent simplement les méthodes de ListeTrajet. Il représente la classe métier de ListeTrajet, faisant le lien entre la liste chaînée brute et le catalogue de trajet que souhaite voir l'utilisateur.

Nous avons donc ensuite créé une liste dynamique nommée ListeTrajet, composés de maillons MaillonTrajet qui sont simplement chainés entre eux. Ceux-ci pointe chacun un Trajet.

- ListeTrajet a un pointeur vers le premier maillon de la chaîne, et un pointeur vers le dernier. Elle conserve sa taille en attribut. C'est dans ListeTrajet que l'on a implémenté les fonctions permettant de chercher un Trajet, d'en ajouter un à la liste.

- MaillonTrajet représente les maillons de la liste chaînée. Il implémente un pointeur vers le trajet suivant pour créer une liste simplement chainée, et un pointeur vers le trajet qu'il contient.

Nous avons rassemblé les Trajets Simples, que nous avons nommé TS, et les trajets complexes que nous avons nommé TC, sous le même parent qui est Trajet. En effet, ces deux types de trajets ont en commun d'avoir un Départ, et une Arrivée.

TS implémente en plus un moyen de transport. TC implémente une liste de Trajet, qui réutilise la liste dynamique ListeTrajet que nous avons créé pour le Catalogue.

- Trajet représente les liens entre TS et TC. Il possède un Départ et une Arrivée, et implémente une méthode d'affichage de ceux-ci.

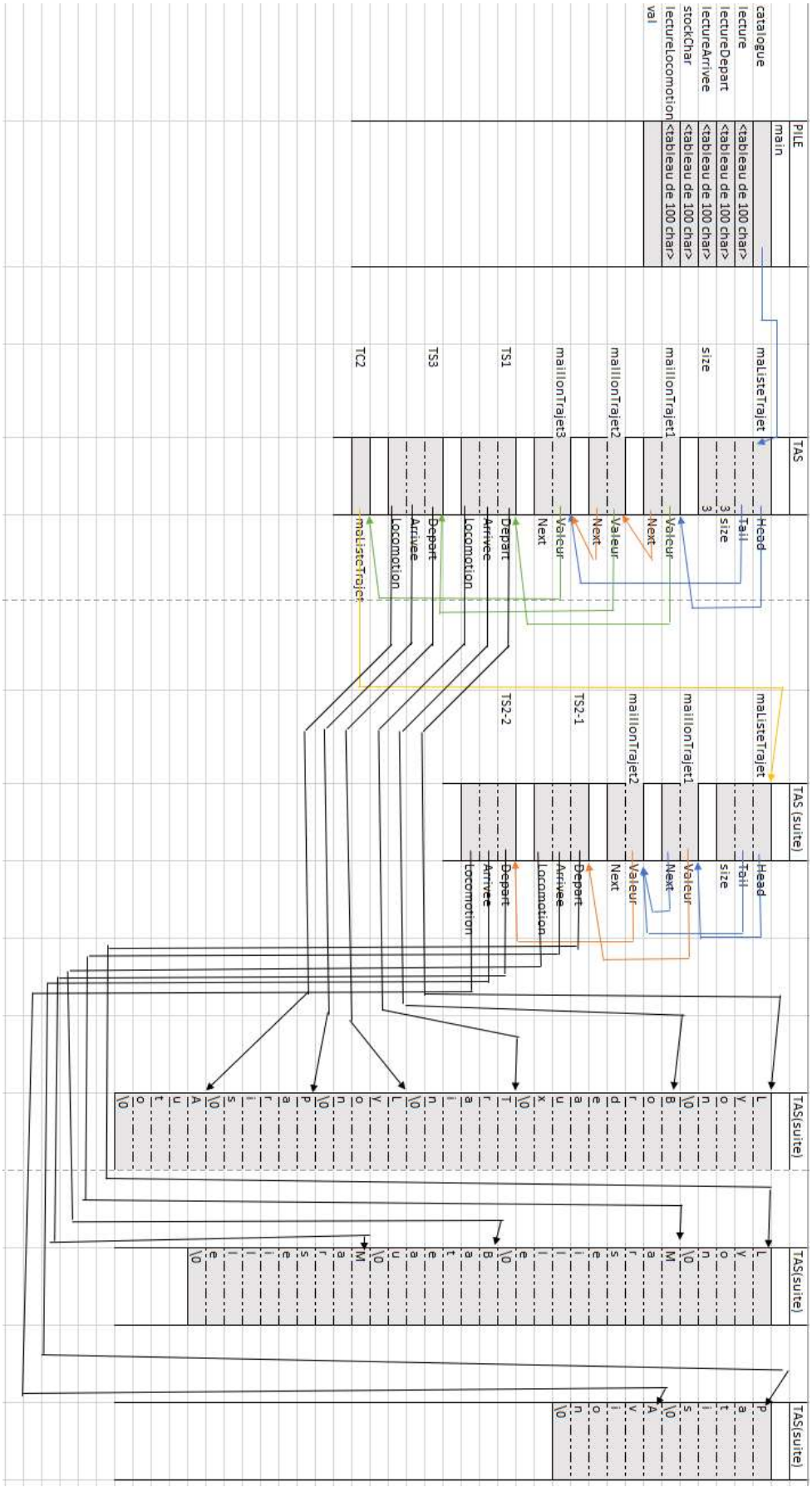
- TS représente le trajet simple, il hérite de Trajet. Il ajoute à ses attributs un moyen de transport Locomotion, et l'ajoute à l'affichage implémenté par Trajet.

- TC représente le trajet complexe, il hérite de Trajet. Il ajoute à ses attributs une ListeTrajet. Il possède des méthodes permettant d'implémenter des trajets dans sa ListeTrajet. Il ajoute l'affichage de sa ListeTrajet à l'affichage de Trajet.

Dans le main, nous avons créé une interface permettant à l'utilisateur d'ajouter un trajet simple ou complexe, de rechercher un trajet à partir de son départ et son arrivée, ou d'afficher simplement tout le catalogue.

- sejours implémente la classe main, qui gère l'interface et les entrées de l'utilisateur, ainsi que des méthodes simplifiant la lecture d'entrées de l'utilisateur.

II. Etat de la mémoire après implémentation de 3 trajets



III. Conclusion

Nous avons appris et appliqué, lors de ce projet, de nombreux principes vus en cours.

Nous avons pu mettre en place un Makefile qui se révéla utile lors des nombreuses compilations.

Nous avons utiliser valgrind pour empêcher toutes les fuites de mémoires possibles, et trouver l'origine des Segmentation Fault.

Nous avons pu mettre en place un héritage, en comprenant les principes d'encapsulation.

Nous avons également pu essayer de créer des classes réutilisables dans différents contextes, comme ici notre liste chaînée.