Justification de la conception de notre application

Nous avons choisi de faire hériter d’une classe mère Trajet les classes TrajetSimple et TrajetCompose car elles disposent de nombreuses similitudes : une ville de départ, une ville d’arrivée et la possibilité de l’afficher. De plus, un TrajetCompose n’est qu’une juxtaposition de plusieurs TrajetsSimple. Il sera donc possible de construire un TrajetCompose à partir d’une liste de TrajetSimple.

Nous avons décidé de garder Trajet comme classe abstraite car nous ne voulons pas qu’il soit possible d’instancier un objet de type Trajet : il doit s’agir obligatoirement d’un TrajetSimple ou d’un TrajetCompose.

Classe Trajet

La classe Trajet est une classe abstraite qui est la classe mère de trajetSimple et trajetCompose. Elle possède 2 attributs en protected qui sont des chaines de caractères représentant pour chaque trajet la ville de départ et la ville d’arrivée.

Au niveau des méthodes, getVilleDepart() et getVilleArrivee() seront utiles pour avoir un accès direct aux villes de départ et d’arrivées quand nous voudrons faire des comparaisons dans la classe catalogue. La méthode AfficherTrajet() devra également être redéfinie pour chaque type de trajet tout comme le constructeur.

Classe TrajetSimple : Trajet