TP C++ n°3 : Gestion des entrées / sorties

Sauvegarde et restitution d’un catalogue de trajets

# Objectifs

L’objectif de ce TP est de reprendre le catalogue de trajets créé au TP précédent et de lui ajouter deux fonctionnalités : la sauvegarde de trajets dans un fichier et l’importation de trajets depuis un fichier.

# Description du fichier de sauvegarde

|  |  |
| --- | --- |
| 1  2  3  4 | s,Lyon,Bordeaux,Train;  c,Lyon,Paris,{s,Lyon,Marseille,Bateau;s,Marseille,Paris,Avion;}  s,Lyon,Paris,Auto; |

Figure : Contenu du fichier demo

Dans les fichiers de sauvegarde, il y a un trajet par ligne et chaque trajet est décomposé en 4 sections séparées par des virgule suivi d’un caractère pour indiquer la fin du trajet. Nous avons fait ce choix pour la praticité et la lisibilité du fichier. En effet un caractère de séparation entre les informations des trajets est indispensable car on ne connait pas à l’avance la taille des noms de ville. De plus, avoir un caractère séparateur entre les trajets permets de lire un trajet entièrement en une commande. Enfin, le retour à la ligne sert uniquement à lisibilité, créée une séparation claire et visuelle entre les différents trajets et permet dans une certaine mesure de garder un maximum d’information sur l’écran.

La première section est composée d’un caractère indiquant si le trajet est un trajet simple ou un trajet composé, ‘s’ pour un trajet simple et ‘c’ pour un trajet composé. Avoir cette information en début de ligne, un endroit qui est fixe, permet de déterminer très rapidement le type de trajet sans avoir à faire d’analyse plus poussée

Les deuxième et troisième sections sont respectivement le nom de la ville de départ et le nom de la ville d’arrivée.

La quatrième section varie en fonction du type de trajet. Pour un trajet simple, elle est le mode de transport suivi d’un point-virgule pour indiquer la fin du trajet simple. Pour un trajet composée, elle est la liste des trajets composant le trajet composé. Les trajets de cette liste sont sous le même format que les autres trajet du fichier de sauvegarde mais il n’y a pas de retour à la ligne entre chaque trajet. Changer de caractère de séparation entre les trajets pour la liste permet de toujours pouvoir lire l’entièreté du trajet composé en une commande. De plus la liste de ces trajets est mise entre accolade. L’accolade fermante permet de déterminer la fin de la liste de trajet et donc du trajet composé. L’accolade ouvrante quant à elle n’est pas nécessaire mais accroit grandement la lisibilité.

Pour les trajets composés, la deuxième et troisième section créé une redondance d’information. En effet, on pourrait déterminer la ville de départ et d’arrivée en analysant la liste de trajets qui compose le trajet principal. Mais cela serait couteux et peu efficace lors d’un chargement de trajets ayant un départ et/ou une arrivée spécifiés. Ainsi nous avons décidé de les mettre dans des sections a part, cela permet également d’avoir un format très similaire au trajet simple et ainsi de traiter les trajet indépendamment de leur type pour le chargement avec départ et/ou arrivée spécifiés.

# Description de la répartition des fonctionnalités

La classe catalogue est responsable de l’ouverture des fichiers et la création des flux d’entrée et de sortie.

Les classes descendante de trajet implémente les méthodes d’exportation vers un flux de sortie et d’importation depuis un flux d’entrée.

# Conclusion

## Problèmes marquants

Pour le fichier de sauvegarde, nous avions initialement prévu de mettre des points-virgules uniquement dans la liste de trajets des trajets composés puis nous avons remarquer que mettre des points virgule a la fin de tout les trajets simples permettait de réutiliser facilement la fonction d'importation des trajets simples. Nous modifié le format de notre fichier de sauvegarde

## Axe d’évolution et d’amélioration