

Compte-rendu du TP n°3

Charles Javerliat, Fabien Narboux

INSA Lyon, 3IF

9 janvier 2020

Mots-clés— C++, Gestion de catalogue, Gestion de fichiers, Sauvegarde, Analyse de formatage.

Résumé

Ce compte-rendu traite du TP n°3 de Programmation Orientée Objet du cours de C++. Il s'agit d'améliorer notre programme afin de permettre l'écriture et la lecture de sauvegardes de collections de trajets hétérogènes depuis des fichiers. Nous ajouterons également une fonctionnalité de filtrage de ces trajets.

Table des matières

1	Nouvelles fonctionnalités	3
1.1	Gestionnaire de fichiers	3
1.1.1	Écriture et lecture de sauvegardes de trajets	3
1.1.2	Gestion du nom des fichiers	4
1.1.3	Convention de formatage des sauvegardes	4
1.1.4	Tokens	4
1.1.5	Attributs d'un TrajetSimple	4
1.1.6	Attributs d'un TrajetCompose	4
1.1.7	Exemple avec le catalogue demo	6
1.1.8	Erreur de formatage	7
1.2	Filtrage des trajets	8
1.3	Exemple avec le catalogue demo	8
1.4	Intégration des deux nouveaux modules	13
2	Exemple d'utilisation	13
2.1	Importation du fichier demo dans le Catalogue	13
3	Conclusion	14
3.1	Difficultés rencontrées	14
3.2	Axes d'évolution et d'amélioration	14
A	Diagramme de classes	15

1 Nouvelles fonctionnalités

Pour respecter la règle selon laquelle chaque classe doit remplir une seule responsabilité, nous avons fait le choix de créer un module à part entière pour la gestion des fichiers nommé `GestionnaireSauvegardeTrajets`. Ainsi `Catalogue` ne contient que les données des trajets et `GestionnaireSauvegardeTrajets` permet de sauvegarder et de charger des trajets dans des fichiers, car il est absurde de penser que le `Catalogue` ait à vérifier le formatage des fichiers lui-même, ce n'est pas son rôle.

De la même façon nous avons fait le choix de créer un module à part entière pour gérer les filtres (type de trajets, ville de départ, ...). Un autre avantage d'avoir séparé le filtre du gestionnaire de fichiers serait de réutiliser le filtre pour améliorer d'autres fonctionnalités. Par exemple on pourrait choisir de filtrer les trajets du `Catalogue` à afficher dans le terminal.

Les méthodes de ces deux classes (`Gestionnaire` et `Filtre`) sont statiques, il est inutile d'avoir une instance de celles-ci, on se sert simplement de leurs fonctions pour réaliser des actions, elles n'ont pas d'attribut particulier.

Le diagramme des classes est disponible en annexe.

Pour tester chaque nouveau module avant de les intégrer ensemble, nous avons réalisé des tests unitaires, présents dans le répertoire `test/`. Il est possible de tester l'ensemble des méthodes en exécutant le binaire de tests `bin/catalogue-test` qui affichera l'état des tests unitaires. L'idée derrière les tests unitaires est ici de s'assurer de la fiabilité, et de la robustesse de notre application en cas de changement. On s'assure ainsi que les fonctionnalités mises en place ne changent pas de comportement quand on en ajoute de nouvelles, qu'on modifie le code source.

1.1 Gestionnaire de fichiers

1.1.1 Écriture et lecture de sauvegardes de trajets

Le module de gestionnaire de fichiers (`GestionnaireSauvegardeTrajets`) a deux fonctions principales :

- la fonction d'écriture : sauvegarde une liste de trajets dans un fichier de nom donné en paramètre en respectant la convention de formatage (Cf. *Convention de formatage des sauvegardes*).
- La fonction de lecture : lit une liste de trajets contenus dans un fichier de nom donné en paramètre si ils respectent la convention, sinon le gestionnaire lèvera une exception pour indiquer où et comment le fichier est mal formaté (Cf. *Erreur de formatage*).

Ces deux méthodes utilisent des méthodes utilitaires protégées pour écrire un trajet, lire un trajet simple et lire un trajet composé. Cela simplifie la lecture du code et évite la redondance (`lireTrajetCompose` appelle `lireTrajetSimple`, cela évite de devoir recopier le code de lecture des attributs des sous-trajets simples). Si on voulait permettre d'avoir des trajets composés de trajets composés cela permettrait aussi de faciliter une forme de récursivité (`lireTrajetCompose` appellerait `lireTrajetCompose`).

1.1.2 Gestion du nom des fichiers

Le nom des fichiers de sauvegarde est laissé libre à l'utilisateur, il peut rentrer un chemin relatif à l'emplacement d'exécution (ex : sauvegarde.txt) ou bien absolu (ex : /home/utilisateur/saves/sauvegarde.txt). Il est également libre quant à l'extension du fichier. Dans les faits, cette extension est uniquement utile pour savoir quelle application par défaut doit ouvrir ce type de fichier (.txt associé à gedit, .cpp associé à vim, etc...) mais cela ne change pas le contenu ni le format du fichier. Ainsi, que l'on sauvegarde un fichier en l'appelant sauvegarde.txt, sauvegarde.texte, ou encore sans extension, on pourra toujours le lire en tant que fichier texte car son formatage n'est pas lié à son extension.

Le gestionnaire s'occupe également de renvoyer une erreur si on tente d'écrire un fichier dans un dossier qui n'existe pas (il ne le crée pas par défaut), si le nom de fichier n'est pas correct (nom vide), ou encore si on tente de lire un fichier qui n'existe pas.

1.1.3 Convention de formatage des sauvegardes

Pour analyser un fichier et générer une liste de trajet valide à partir de ce dernier, il est essentiel de mettre en place une convention de formatage. C'est ce que nous avons fait dès le début du TP. Dans un premier temps on pose comme convention que chaque ligne contient une seule information utile.

1.1.4 Tokens

Pour déclarer un nouveau trajet dans un fichier de sauvegarde, on doit d'abord commencer par renseigner sa nature, il s'agit soit d'un `TrajetSimple`, soit d'un `TrajetCompose`. Pour les trajets simples, on place avant les attributs un token `#TS` qui permettra au gestionnaire de savoir qu'il a affaire à un `TrajetSimple`, et qu'il devra donc lire les attributs d'un `TrajetSimple`. Dans le cas d'un `TrajetCompose`, le token s'écrit `#TC`.

1.1.5 Attributs d'un `TrajetSimple`

A la suite d'un token `#TS`, le gestionnaire s'attend à trouver les attributs d'un trajet simple, c'est à dire un fichier de la forme suivante :

Fichier de sauvegarde d'un trajet simple quelconque

```
#TS  
villeDépart  
villeArrivée  
typeTransport
```

1.1.6 Attributs d'un `TrajetCompose`

A la suite d'un token `#TC`, le gestionnaire s'attend à trouver les attributs d'un trajet composé, c'est à dire un entier naturel `n` de sous-trajets simples, suivi de `n` déclarations de trajets simples :

Fichier de sauvegarde d'un trajet composé quelconque

```
#TC
N
#TS
villeA
villeB
typeTransport
#TS
villeB
villeC
typeTransport
[...]
```

1.1.7 Exemple avec le catalogue demo

En suivant nos conventions, nous obtenons la sauvegarde suivante décrivant le contenu de demo :

```
Fichier de sauvegarde demo
#TS
Lyon
Bordeaux
Train
#TS
Lyon
Paris
Auto
#TC
2
#TS
Lyon
Marseille
Bateau
#TS
Marseille
Paris
Avion
```

1.1.8 Erreur de formatage

Les conventions de formatage citées précédemment sont très strictes, si un fichier ne les respecte pas, alors le gestionnaire de fichiers affichera une erreur et ne chargera pas les trajets. Plusieurs types d'erreurs peuvent survenir :

Type d'erreur	Sauvegarde associée	Message d'erreur
Attribut manquant	#TS Lyon Marseille	Erreur de formatage à la ligne 4 : Attribut typeTransport manquant pour le trajet simple.
Token inconnu, on s'attend à avoir #TS ou #TC à la place de Paris	#TS Lyon Marseille TGV Paris	Erreur de formatage à la ligne 5 : Token inconnu Paris
Nombre de sous-trajets incorrect	#TC 2 #TS Lyon Marseille TGV	Erreur de formatage à la ligne 7 : Sous-trajet attendu mais manquant pour le trajet composé.
Impossible d'ajouter un trajet composé à un trajet composé (Cf. cahier des charges du TP2)	#TC 1 #TC 1 #TS Lyon Marseille TGV	Erreur de formatage à la ligne 3 : Token invalide #TC, le trajet composé ne peut contenir que des trajets simples.
Le nombre de sous-trajets doit être >0	#TC -1 #TS Lyon Marseille TGV	Erreur de formatage à la ligne 2 : Le nombre de trajets composants le trajet composé doit être un entier strictement supérieur à 0.
Attribut invalide (Cf. cahier des charges du TP2)	#TS Lyon Lyon TGV	Le nom de la ville d'arrivée ne peut pas être identique au nom de la ville de départ. Erreur de formatage à la ligne 4 : Attributs du TrajetSimple invalides.

La liste n'est pas exhaustive et d'autres messages plus ou moins similaires peuvent apparaître si le formatage n'est pas correct.

1.2 Filtrage des trajets

Chaque fonction de cette classe prend en argument une liste chaînée de trajets et les paramètres du filtre (type de trajet, ville départ, ...). Il y a quatre filtres pour notre application :

- Filtrage par type de trajet (simple ou composé)
- Filtrage par ville de départ
- Filtrage par ville d'arrivée
- Filtrage par ville de départ et d'arrivée

Pour vérifier la validité de nos filtres, nous avons réalisé des tests unitaires sur plusieurs jeux de tests. On s'assure ainsi que les trajets sont correctement filtrés.

1.3 Exemple avec le catalogue demo

```
=== PROGRAMME DE GESTION DE CATALOGUE ===
Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 8
Nom du fichier à lire : sauvegarde.txt
Type de filtre à appliquer
  1 - Aucun filtre
  2 - Filtre sur le type de trajet
  3 - Filtre sur la ville de départ
  4 - Filtre sur la ville d'arrivée
  5 - Filtre sur la ville de départ et d'arrivée
  6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
  7 - Quitter
Entrez votre choix: 1
Chargement réussi !
Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 1
===== CATALOGUE =====
Trajet 1 - Trajet simple de Lyon à Bordeaux en Train (indice = 0)
Trajet 2 - Trajet simple de Lyon à Paris en Auto (indice = 1)
Trajet 3 - Trajet composé: (Trajet simple de Lyon à Marseille en Bateau) + (Trajet simple de Marseille à Paris en Avion) (indice = 2)
=== FIN DU CATALOGUE ===
```

FIGURE 1 – Aucun filtre

1.3 Exemple avec le catalogue demo

```

===== PROGRAMME DE GESTION DE CATALOGUE =====
Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 8

Nom du fichier à lire : demo.txt

Type de filtre à appliquer
1 - Aucun filtre
2 - Filtre sur le type de trajet
3 - Filtre sur la ville de départ
4 - Filtre sur la ville d'arrivée
5 - Filtre sur la ville de départ et d'arrivée
6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
7 - Quitter
Entrez votre choix: 2
Type de trajet (1) Simple ou (2) Composé : 1
Chargement réussi !

Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====
Trajet 1 - Trajet simple de Lyon à Bordeaux en Train (indice = 0)
Trajet 2 - Trajet simple de Lyon à Paris en Auto (indice = 1)

===== FIN DU CATALOGUE =====

```

(a) Trajets simples

```

===== PROGRAMME DE GESTION DE CATALOGUE =====
Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 8

Nom du fichier à lire : demo.txt

Type de filtre à appliquer
1 - Aucun filtre
2 - Filtre sur le type de trajet
3 - Filtre sur la ville de départ
4 - Filtre sur la ville d'arrivée
5 - Filtre sur la ville de départ et d'arrivée
6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
7 - Quitter
Entrez votre choix: 2
Type de trajet (1) Simple ou (2) Composé : 2
Chargement réussi !

Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====
Trajet 1 - Trajet composé (Trajet simple de Lyon à Marseille en Bateau) + (Trajet simple de Marseille à Paris en Auto) (indice = 0)
Trajet 2 - Trajet composé (Trajet simple de Lyon à Marseille en Bateau) + (Trajet simple de Marseille à Paris en Auto) (indice = 1)

===== FIN DU CATALOGUE =====

```

(b) Trajets composés

FIGURE 2 – Filtre sur le type

```

===== PROGRAMME DE GESTION DE CATALOGUE =====
Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 8

Nom du fichier à lire : demo.txt

Type de filtre à appliquer
1 - Aucun filtre
2 - Filtre sur le type de trajet
3 - Filtre sur la ville de départ
4 - Filtre sur la ville d'arrivée
5 - Filtre sur la ville de départ et d'arrivée
6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
7 - Quitter
Entrez votre choix: 3
Ville de départ : Bordeaux
Chargement réussi !

Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====
Le catalogue est vide.

===== FIN DU CATALOGUE =====

```

(a) Départ de Bordeaux

```

===== PROGRAMME DE GESTION DE CATALOGUE =====
Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 8

Nom du fichier à lire : demo.txt

Type de filtre à appliquer
1 - Aucun filtre
2 - Filtre sur le type de trajet
3 - Filtre sur la ville de départ
4 - Filtre sur la ville d'arrivée
5 - Filtre sur la ville de départ et d'arrivée
6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
7 - Quitter
Entrez votre choix: 3
Ville de départ : Lyon
Chargement réussi !

Que désirez-vous faire ?
1 - Afficher le catalogue
2 - Ajouter un trajet simple
3 - Ajouter un trajet composé
4 - Supprimer un trajet
5 - Recherche de trajet simple
6 - Recherche de trajet avancée
7 - Sauvegarde dans un fichier
8 - Lecture et ajout de trajets depuis un fichier
9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====
Trajet 1 - Trajet simple de Lyon à Bordeaux en Train (indice = 0)
Trajet 2 - Trajet simple de Lyon à Paris en Auto (indice = 1)
Trajet 3 - Trajet composé (Trajet simple de Lyon à Marseille en Bateau) + (Trajet simple de Marseille à Paris en Auto) (indice = 2)

===== FIN DU CATALOGUE =====

```

(b) Départ de Lyon

FIGURE 3 – Filtre sur la ville de départ

1.3 Exemple avec le catalogue demo

```
=== PROGRAMME DE GESTION DE CATALOGUE ===
Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 8

Nom du fichier à lire : demo.txt

Type de filtre à appliquer
  1 - Aucun filtre
  2 - Filtre sur le type de trajet
  3 - Filtre sur la ville de départ
  4 - Filtre sur la ville d'arrivée
  5 - Filtre sur la ville de départ et d'arrivée
  6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
  7 - Quitter
Entrez votre choix: 4
Ville d'arrivée : Paris
Chargement réussi !

Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====
Trajet 1 - Trajet simple de Lyon à Paris en Auto (indice = 0)
Trajet 2 - Trajet composé: (Trajet simple de Lyon à Marseille en Bateau) + (Trajet simple de Marseille à Paris en Avion) (indice = 1)
=== FIN DU CATALOGUE ===
```

FIGURE 4 – Filtre sur la ville d’arrivée

```
=== PROGRAMME DE GESTION DE CATALOGUE ===
Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 8

Nom du fichier à lire : demo.txt

Type de filtre à appliquer
  1 - Aucun filtre
  2 - Filtre sur le type de trajet
  3 - Filtre sur la ville de départ
  4 - Filtre sur la ville d'arrivée
  5 - Filtre sur la ville de départ et d'arrivée
  6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
  7 - Quitter
Entrez votre choix: 5
Ville de départ : Lyon
Ville d'arrivée : Bordeaux

Chargement réussi !

Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====

Trajet 1 - Trajet simple de Lyon à Bordeaux en Train (indice = 0)

=== FIN DU CATALOGUE ===
```

FIGURE 5 – Filtre sur la ville de départ et d'arrivée

```
=== PROGRAMME DE GESTION DE CATALOGUE ===
Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 8

Nom du fichier à lire : demo.txt

Type de filtre à appliquer
  1 - Aucun filtre
  2 - Filtre sur le type de trajet
  3 - Filtre sur la ville de départ
  4 - Filtre sur la ville d'arrivée
  5 - Filtre sur la ville de départ et d'arrivée
  6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
  7 - Quitter
Entrez votre choix: 6
Début de l'intervalle (entre 1 et 3) : 1
Fin de l'intervalle (entre 1 et 3): 2

Chargement réussi !

Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====

Trajet 1 - Trajet simple de Lyon à Bordeaux en Train (indice = 0)
Trajet 2 - Trajet simple de Lyon à Paris en Auto (indice = 1)

=== FIN DU CATALOGUE ===
```

FIGURE 6 – Filtre sur un intervalle

1.4 Intégration des deux nouveaux modules

Pour la sauvegarde, on filtre tout d'abord le catalogue avec le filtre choisi par l'utilisateur. La fonction de filtre retourne une liste de trajets que l'on sauvegarde avec le gestionnaire dans un fichier dont le nom à été choisit par l'utilisateur.

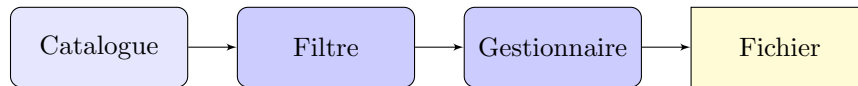


FIGURE 7 – Actions réalisées pour sauvegarder le catalogue dans un fichier.

Pour la lecture, on charge tous les trajets contenus dans le fichier choisi par l'utilisateur dans une liste de trajets. On applique ensuite le filtre désiré à cette liste puis on l'ajoute dans le catalogue.

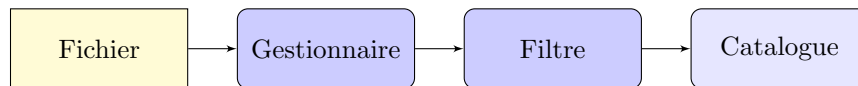


FIGURE 8 – Actions réalisées pour charger une liste de trajets dans le catalogue.

Pour le choix du nom de fichier par l'utilisateur, le chemin peut être relatif ou absolu par rapport à l'endroit où l'exécutable est lancé.

2 Exemple d'utilisation

2.1 Importation du fichier demo dans le Catalogue

On importe dans le catalogue le fichier demo tel que décrit précédemment. L'exécution du programme sera la suivante :

```

=== PROGRAMME DE GESTION DE CATALOGUE ===
Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 8
Nom du fichier à lire : sauvegarde.txt
Type de filtre à appliquer
  1 - Aucun filtre
  2 - Filtre sur le type de trajet
  3 - Filtre sur la ville de départ
  4 - Filtre sur la ville d'arrivée
  5 - Filtre sur la ville de départ et d'arrivée
  6 - Filtre sur un intervalle (trajet numéro n à trajet numéro m inclus)
  7 - Quitter
Entrez votre choix: 1
Chargement réussi !
Que désirez-vous faire ?
  1 - Afficher le catalogue
  2 - Ajouter un trajet simple
  3 - Ajouter un trajet composé
  4 - Supprimer un trajet
  5 - Recherche de trajet simple
  6 - Recherche de trajet avancée
  7 - Sauvegarde dans un fichier
  8 - Lecture et ajout de trajets depuis un fichier
  9 - Quitter
Entrez votre choix: 1

===== CATALOGUE =====
Trajet 1 - Trajet simple de Lyon à Bordeaux en Train (indice = 0)
Trajet 2 - Trajet simple de Lyon à Paris en Auto (indice = 1)
Trajet 3 - Trajet composé: (Trajet simple de Lyon à Marseille en Bateau) + (Trajet simple de Marseille à Paris en Avion) (indice = 2)

=== FIN DU CATALOGUE ===
  
```

FIGURE 9 – Déroulement du programme de chargement de demo

3 Conclusion

3.1 Difficultés rencontrées

3.2 Axes d'évolution et d'amélioration

Nous avons rencontré quelques difficultés lors de ce TP :

- Nous avons pris notre temps pour s'organiser au niveau du formatage du fichier et de la répartition des tâches. Il s'agissait surtout pour l'un de nous deux de découvrir le code de l'autre. Nous avons donc passé un peu de temps à choisir lequel utiliser avant de commencer le TP.
- Il ne faut pas oublier de désallouer la mémoire des trajets chargés depuis un fichier mais non rajoutés au catalogue (éliminé lors du filtrage).
- On s'est posé la question du cas des doublons, si deux trajets ont exactement les mêmes attributs, doit-on permettre à l'utilisateur de rajouter les deux ? En relisant le cahier des charges rien n'était spécifié à ce propos nous avons donc fait le choix de permettre les doublons dans le catalogue.

On aurait pu utiliser namespaces plutôt que class dans les fichiers d'entête pour GestionnaireSauvegardeTrajets et FiltreTrajets étant donné qu'elles ne contiennent que des fonctions statiques.

Il serait possible de rendre le fichier contenant des trajets plus lisible en ajoutant des tabulations. On pourrait utiliser des fichiers binaires lors de la sauvegarde pour optimiser la taille des fichiers de sauvegarde.

A Diagramme de classes

