

Rakib SHEIKH
3A IABD1 et 2 2024-2025
Travaux pratiques : Java orienté objet avancé
rsheikh1@myges.fr

TP Noté Final Semestre 1

Le TP est à rendre sur la plateforme myges, au plus-tard le lendemain à 6h00 du matin.

Règles de pénalités de points : À lire en étant sobre et non stressé.

- Un rendu au-delà des temps impartis de la fin de cours + 30 minutes vous sera facturé -2 pts, jusqu'à 5 h 59 du matin le lendemain. (une fois)
- À partir de 6 h 00 le lendemain (soit la deadline), -5 pts sera facturé à chaque heure de retard. Toute heure entamée est due en son intégralité. (soit déjà - 7 pts en cumulé à 6h00m00s)
- Un rendu ne respectant pas l'architecture des dossiers de paquetage sera facturé -1 pt.
- Les questions ouvertes seront à répondre à l'aide d'un fichier au format markdown. Aucune capture d'écran n'est attendue.

À titre indicatif, la partie 1 et la partie 2 valent pour 10 points chacune.

Partie 1 : Générification d'une Liste Chaînée

Exercice 1 - Les listes chaînées avec un conteneur

Le but de cet exercice est d'écrire une implantation de listes chaînées.

Pour la suite de l'exercice, l'ensemble des classes créées devront être dans le paquetage `fr.esgi`

Nous allons dans un premier temps créer une liste chaînée d'entiers.

1. Créer un record `Link` dans le paquetage `fr.esgi.exo1.data` correspondant à un maillon de la liste chaînée stockant des entiers.
 - En aucun cas, l'utilisateur de la classe ne devra lui-même manipuler des maillons.
 - Pour rappel, un maillon est composé d'une valeur, ainsi qu'un élément suivant.
2. Quelle doit être la visibilité du record `fr.esgi.exo1.Link` ?
3. Écrire un main de test dans cette classe créant deux maillons contenant les valeurs 13 et 144. (C'est à dire à l'intérieur du Record)
4. Donner est la commande pour exécuter le main du record `fr.esgi.exo1.Link` à partir d'un terminal (pas dans IntelliJ) ?
5. Puisque nous n'avons pas vu le fonctionnement d'une lambda, copiez le code de `forEach` dans la classe `LinkedList`.

```
public void forEach(IntConsumer lambda) {  
    Link current = head;  
    while (current != null) {  
        lambda.accept(current.value());  
        current = current.next();  
    }  
}
```

6. Créer une classe `fr.esgi.exo1.LinkedList` qui permettra de manipuler une liste chaînée par son premier maillon, avec :

- une méthode `add(int value)` qui ajoute un élément en tête de la liste.
 - une méthode `get(index)` qui renvoie l'élément à l'index (en commençant à 0).
7. Comment faire en sorte que le code qui vérifie que l'index est valide soit en $O(1)$?
 - Faites les changements qui s'imposent.
 8. Faire une méthode `toString` qui représente la liste chaînée avec des « -> » entre les valeurs. Par exemple, une liste contenant les éléments 3, 2 et 1, on obtient

```
1 --> 2 --> 3
```

Note: il existe une classe `StringJoiner` qui prend à la construction un délimiteur et qui va appliquer le délimiteur entre les chaînes de caractère ajoutées avec `add`.

Pour tester la classe `fr.esgi.data.LinkedList`, créer une classe `Main` dans le package `fr.esgi.exo1.main` dont la méthode `main` contiendra vos tests.

Exercice 2 - Liste chaînée (suite)

1. Dans le but de pouvoir réutiliser la liste dans différents codes, recopiez les classes `fr.esgi.exo1.LinkedList` et `fr.esgi.exo1.Link` dans un nouveau dossier `fr.esgi.exo2.LinkedList` et `fr.esgi.exo2.Link` pour une implantation plus générique à base d'`Object`.
 - Note pour `forEach`, sa signature sera changée vers `Consumer<? Super Object>`
2. Dans la classe `Main`, expliquer pourquoi le code suivant ne fonctionne pas ?

```
var l = new LinkedList();
l.add("hello");
l.add("world");
l.forEach(s -> System.out.println("string " + s + " length " + s.length()));
```

java

Que doit-t-on faire pour que le code fonctionne ?

Exercice 3 - Générification de `LinkedList`

Le but de cet exercice est de « générifier » les classes `fr.esgi.exo2.Link` et `fr.esgi.exo2.LinkedList`

1. Rappeler quel est l'intérêt d'utiliser un type paramétré ici ?
2. Paramétrer la classe `fr.esgi.exo3.LinkedList` pour que celle-ci soit générique.
3. Modifier la classe `fr.esgi.exo3.Main` en conséquence.

Partie 2 : Interface Blockbuster

On souhaite modéliser le catalogue d'un magasin Blockbuster, un magasin qui loue des cassettes vidéos et des laser discs, sachant que l'on veut être capable de lire/écrire un catalogue à partir de fichiers.

Oui, on fait de l'archéologie, avant Netflix, les vrais gens allaient dans un magasin pour louer des films, soit sur des bandes analogiques (des cassettes vidéo) soit sur disques numériques (des laser discs).

Pour simplifier un peu les choses, on va dire que le magasin a deux sortes d'articles, des `VideoTape` et des `LaserDisc`, et que pour un nom de film, il ne peut y avoir qu'un article au maximum dans le catalogue.

- Une `VideoTape` est définie par un nom (`name`) et une durée (`duration`) de type `java.time.Duration` (un type déjà fourni par le JDK).
 - Un `LaserDisc` est uniquement défini par un nom.
 - Un `Catalog` permet d'ajouter (`add`) des articles, de chercher (`lookup`) un article par son nom, de charger (`load`) des articles à partir d'un fichier et de sauvegarder (`save`) les articles du catalogue dans un fichier.
1. Écrire les types `VideoTape` et `LaserDisc` tels que le code suivant fonctionne

```
var laserDisc = new LaserDisc("Jaws");
var videoTape = new VideoTape("The Cotton Club", Duration.ofMinutes(128));
var videoTape2 = new VideoTape("Mission Impossible", Duration.ofMinutes(110));
```

Attention à ne pas oublier les pré-conditions.

2. On souhaite maintenant écrire un type `Catalog` avec une méthode

- `add` qui permet d'ajouter une cassette vidéo ou un laser disc.

Attention, cette méthode ne doit pas permettre d'ajouter deux articles ayant le même nom.

- `lookup` qui permet de rechercher un article par son nom.

a. Quel doit être le type du paramètre de `add` et le type de retour de `lookup` ?

b. Que doit renvoyer `lookup` s'il n'y a ni cassette vidéo ni laser disc ayant le nom demandé dans le catalogue ?

c. Implanter le type `Catalog` sachant que l'on souhaite que le code suivant fonctionne :

```
var catalog = new Catalog();
catalog.add(laserDisc);
catalog.add(videoTape);
catalog.add(videoTape2);
// catalog.add(new LaserDisc("Mission Impossible")); // exception !
System.out.println(catalog.lookup("Jaws"));
System.out.println(catalog.lookup("The Cotton Club"));
System.out.println(catalog.lookup("Indiana Jones"));
```

3. Pourquoi `fromText` est-elle une méthode statique alors que `toText` est une méthode d'instance ?

Le format textuel est composé du type de l'article (`LaserDisc` ou `VideoTape`) suivi du nom de l'article et, dans le cas de la cassette vidéo, de la durée en minutes (il existe une méthode `duration.toMinutes()` et une méthode `Duration.ofMinutes()`). Les différentes parties du texte sont séparées par des :

Voici un exemple de fichier contenant un laser disc et une cassette vidéo.

```
LaserDisc:Jaws
VideoTape:The Cotton Club:128
```

4. Dans un premier temps, écrire la méthode `toText` de telle façon que le code suivant est valide

```
var laserDiscText = laserDisc.toText();
var videoTapeText = videoTape.toText();
System.out.println(laserDiscText); // LaserDisc:Jaws
System.out.println(videoTapeText); // VideoTape:The Cotton Club:128
```

5. Puis écrire le code de la méthode `fromText` sachant qu'il existe une méthode `string.split()` pour séparer un texte suivant un délimiteur et que l'on peut faire un `switch` sur des `Strings`. Le code suivant devra fonctionner :

```
var laserDisc2 = Article.fromText(laserDiscText);
var videoTape3 = Article.fromText(videoTapeText);
System.out.println(laserDisc.equals(laserDisc2)); // true
System.out.println(videoTape.equals(videoTape3)); // true
```

Note faire en sorte que les noms « LaserDisc » et « VideoTape » soit définis sous forme de constantes pour que le code soit plus lisible.

6. On souhaite maintenant ajouter une méthode `save` qui permet de sauvegarder les articles d'un catalogue dans un fichier.
 - a. Quelle méthode doit-on utiliser pour créer un écrivain sur un fichier texte à partir d'un `Path` ?
 - b. Comment doit-on faire pour garantir que la ressource système associée est bien libérée ?
 - c. Comment doit-on gérer l'exception d'entrée/sortie ?
 - d. Écrire la méthode `save` afin que le code suivant fonctionne :

```
var catalog2 = new Catalog();
catalog2.add(laserDisc);
catalog2.add(videoTape);
catalog2.save(Path.of("catalog.txt"));
```

7. Comme `Catalog` est mutable, on va écrire la méthode `load` comme une méthode d'instance et non pas comme une méthode statique.
 - a. Expliquer quel est l'intérêt.
 - b. Écrire la méthode `load` dans `Catalog` afin que le code suivant fonctionne :

```
var catalog3 = new Catalog();
catalog3.load(Path.of("catalog.txt"));
System.out.println(catalog3.lookup("Jaws")); // LaserDisc:Jaws
System.out.println(catalog3.lookup("The Cotton Club")); // VideoTape:The Cotton Club:128
```

Note pour `load`, on ne vous demande pas de vérifier au préalable que le fichier est bien formé (lignes de la bonne taille, formats numériques corrects, ...). Vous pouvez laisser filer les exceptions susceptibles de survenir dans ces cas là.

10. Tout le monde s'est plus ou moins mis d'accord pour que l'UTF-8 soit le format utilisé pour la stockage, malheureusement, il reste encore plein de Windows XP / Windows 7 qui ne sont pas en UTF8 par défaut. On va donc ajouter deux surcharges à `load` et `save` qui prennent en paramètre l'encodage. Le code suivant doit fonctionner :

```
var catalog4 = new Catalog();
catalog4.add(new LaserDisc("A Fistful of €"));
catalog4.add(new VideoTape("For a Few €s More", Duration.ofMinutes(132)));
catalog4.save(Path.of("catalog-windows-1252.txt"), Charset.forName("Windows-1252"));

var catalog5 = new Catalog();
catalog5.load(Path.of("catalog-windows-1252.txt"), Charset.forName("Windows-1252"));
System.out.println(catalog5.lookup("A Fistful of €"));
System.out.println(catalog5.lookup("For a Few €s More"));
```

11. Écrire les deux méthodes et partager le code entre les surcharges pour ne pas dupliquer de code.

Note il existe une classe `StandardCharsets` qui est une énumération des encodages standard et qui contient l'encodage UTF-8.