Dr. Nastaran FATEMI
Florian POULIN

Moteur d'indexation

Laboratoire Nº 2

A. Objectifs

Ce laboratoire a pour but d'implémenter un moteur d'indexation pour une collection de documents, plus précisément sur la collection CACM.

Les points étudiés dans ce laboratoire seront :

1. Construction d'un fichier d'index
2. Construction d'un fichier d'index inversé
3. Implémentation de fonctions de recherche élémentaires

B. Références

Cours «Recherche d'Information Multimédia » de Nastaran Fatemi.

C. Rapport

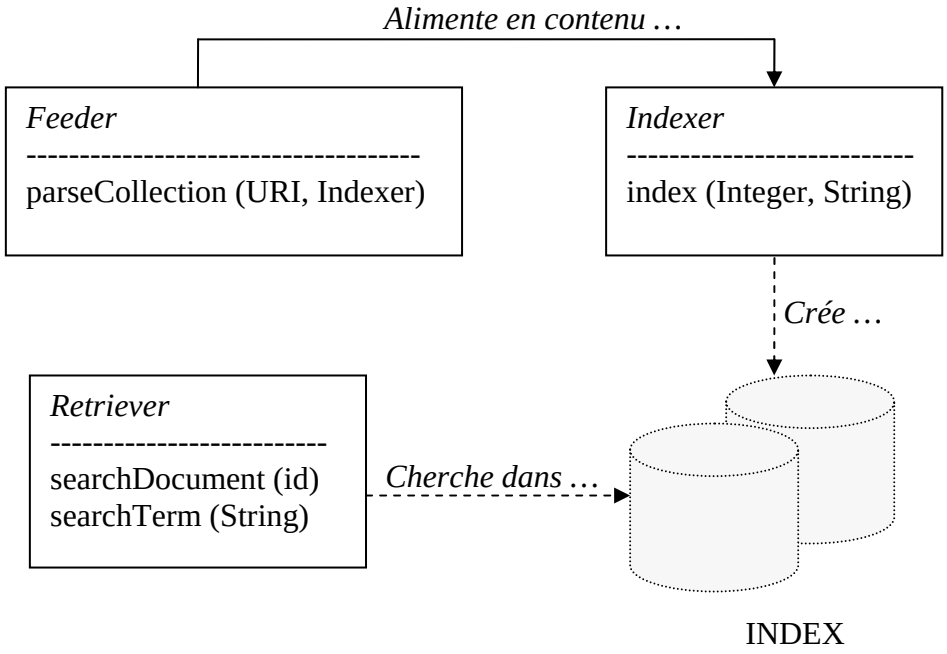
A remettre : Vos sources et un rapport comportant les objectifs, la description des démarches adoptées, l'analyse des résultats obtenus, l'analyse de la complexité des opérations d'indexation développées et une conclusion personnelle.

A remettre : Au plus tard au début de la séance du 31 octobre 2007.

D. Donnée

1. RIM Framework

Afin de vous aidez dans votre travail, un framework Java composé de quelques classes et interfaces vous est fourni. Votre travail consistera dans un premier temps à comprendre son fonctionnement, puis à en implémenter certaines classes. Le schéma suivant présente les trois interfaces du framework et montre leurs principales interactions :



Dans ce premier laboratoire vous allez réaliser un moteur de recherche élémentaire pour la collection CACM. Votre travail consistera à implémenter l'*Indexer* et le *Retriever*, ainsi que le programme principal de l'application.

Une implémentation du *Feeder* pour la collection CACM vous est fournie. Il s'agit de la classe *CACMFeeder* présente dans le package *rim.cacm*. Celle-ci parcourt la collection CACM (*cacm.all*), et pour chaque document, invoque la méthode *index(Integer, String)* de l'*Indexer* connu en lui transmettant l'identifiant du document et son contenu à indexer. Le contenu en question est une simple concaténation du titre et du résumé.

2. Indexer

Il vous est demandé de compléter le contenu de la classe *CACMIndexer* présente dans le package *rim.cacm*. L'indexation d'un document peut être décomposée comme suit :

- Il faut commencer par extraire les *tokens* du contenu reçu pour l'indexation. Il faut donc séparer le String reçu en une séquence de mots. Les espaces et la ponctuation constituent les séparateurs ;
- Il faut ensuite procéder à l'élimination des « stop words », à savoir le filtrage des mots qui n'ont pas de signification. Ces mots sont fournis dans la collection CACM. Pour ce filtrage, il ne faut pas tenir compte de la casse ;
- La lemmatisation ou la troncature des mots significatifs pourrait être réalisée ici mais elle ne vous est pas demandée ;
- On stocke ensuite pour chaque document traité, les mots qu'il contient ainsi que leur fréquence ;
- On génère également un index inversé contenant, pour chaque mot, les id des documents les contenant ainsi que leur fréquence.

Afin de faciliter le débogage (et la correction). **Les index devront être imprimés dans des fichiers**. Pour l'index, vous devez générer un fichier contenant une ligne par document avec le format ci-dessous (strictement) respecté, soit :

```
[id1] {<terme4,freq><terme7,freq> … }
[id2] {<terme1,freq><terme3,freq> … }
…
```

Les documents doivent être **triés par identifiant**, et les termes contenus dans chaque document devront être **triés alphabétiquement**. Pour l'index inversé, même consignes en ce qui concerne le tri, le fichier ressemblera donc à ceci :

```
[terme1] {<id3,freq><id8,freq> … }
[terme2] {<id3,freq><id4,freq> … }
…
```

3. Retriever

Vous devrez également compléter le contenu de la classe *CACMRetriever* présente dans le package *rim.cacm*. Il s'agit d'implémenter deux fonctions de recherche simples : une première qui permet de connaître quels mots apparaissent dans un document et avec quelles fréquences, et une seconde qui permet, en fonction d'un mot donné, de connaître les documents qui le contiennent et leur fréquence relative.

4. Programme principal

Finalement, il s'agit de compléter la classe *Labo2* présente dans le package *rim*. Cette classe contient le point d'entrée de l'application. Elle est chargée d'orchestrer les différentes opérations à réaliser. Elle doit donc effectuer l'instanciation des objets (*feeder*, *indexer*, *retriever*), l'indexation de la collection CACM mise à disposition dans le dossier *ressources*, et enfin la démonstration du bon fonctionnement des méthodes de recherche (pour ce faire, vous n'êtes pas obligés d'implémenter un dialogue avec l'utilisateur, mais cela serait apprécié).

E. Indications

- Ne modifiez pas les interfaces mises à disposition ;
- Lisez la *javadoc* fournie avec le Framework, elle contient des informations utiles ;
- Faites attention aux structures de données utilisées (notamment pour les index) ! Aidez-vous de l'API de Java, et en particulier de *java.util* ;
- Ecrivez du code efficace, mais tout de même lisible (documenté et indenté) ;
- Pour le rapport, faites apparaître clairement les cinq points demandés. Construisez vos idées, soyez précis et soignez l'orthographe ;
- En cas de doute, n'hésitez pas à poser des questions à l'assistant ou au professeur.