




TP MELOG n°II: application des structures linéaires aux fichiers

Octobre 2009

L'objectif de ce TP est d'appliquer les principes de la structuration des données pour mettre en œuvre un programme de traitement de fichiers. La taille importante des fichiers considérés impose que la conception des algorithmes des programmes demandés conduise au choix des structures les plus adaptées pour éviter d'en arriver à des temps d'exécution excessifs, voire rédhibitoires.

0 Préliminaires

Utiliser les fichiers joints ( `Tp.java` et  `Contenu.java` et  `texte.txt`) pour se familiariser avec la lecture d'un fichier de texte, le découpage et un traitement sur ses mots, et l'écriture dans un nouveau fichier.

Question :

- ▶ Exécuter le programme `Tp`.

1 Mise au point d'un premier lexique

Nous cherchons dans un premier temps à créer un lexique de tous les mots contenus dans un texte donné.

1.1 Premier texte

Le texte en question est dans un fichier qui s'appelle  `deLaTerreALaLune.txt` et il contient le roman « De la terre à la lune » de Jules Verne. Il est fourni (cliquer sur le nom du fichier dans ce paragraphe).

Question :

- ▶ Définissez et implantez en java les classes nécessaires pour effectuer au moins les opérations suivantes :
 - constituer, à partir d'un texte contenu dans un fichier d'entrée, un *lexique* des mots utilisés avec le nombre d'occurrences pour chaque mot ;
 - sauvegarder le contenu du lexique ainsi produit dans un fichier de sortie ;
 - afficher le nombre de mots différents présents dans le lexique ;
 - tester la présence d'un mot dans le lexique et retourner son nombre d'occurrences ;
 - supprimer un mot du lexique.

Les algorithmes non triviaux (à l'instar de l'algorithme permettant de construire le lexique) devront figurer dans votre rapport.

Le lexique sera constitué dans un ordre que vous choisirez et préciserez dans votre compte-rendu.

- ▶ Illustrez le bon fonctionnement de votre programme sur des jeux d'essai basés sur la manipulation du lexique associé au texte `deLaTerreALaLune.txt`.

1.2 Choix d'une meilleure structure de données

Le programme que vous venez de finir utilise une structure de données de liste chaînée. Son exécution prend un certain temps en raison du temps de recherche des éléments dans la liste. Les tables de hachage ont été introduites en cours, l'objectif de cette seconde question est de les utiliser afin d'améliorer les temps d'exécution du programme, avant de passer à l'étape suivante.


Question :

- Créez la ou les nouvelles classes pour effectuer la même tâche, mais en mettant en œuvre une table de hachage et plus une liste chaînée. Conservez la version précédente de votre programme ;
- Comparez les temps d'exécution des deux programmes. Expliquez.

2 Enrichissement du lexique

Nous cherchons maintenant à enrichir le premier lexique avec les mots contenus dans un deuxième texte.

2.1 Premier texte

Ce deuxième texte se trouve dans un nouveau fichier qui s'appelle  `autourDeLaLune.txt` et il contient le roman « Autour de la lune » de Jules Verne. Il est également fourni (cliquer sur le nom du fichier dans ce paragraphe).

Questions :

- Écrivez l'algorithme qui permette d'examiner un texte contenu dans un second fichier, d'enrichir le lexique avec les mots nouveaux et de mettre à jour le nombre d'occurrences de chacun des mots. Complétez les classes java précédemment définies pour que pareille opération puisse être assurée.
- Énumérez les nouveaux mots repérés dans cette seconde phase.
- Illustrez le bon fonctionnement de votre programme.

3 Construction d'un lexique complet

Pour chaque mot, nous voulons maintenant enregistrer la liste de toutes les lignes où ce mot apparaît dans le texte.

(remarque : les instructions de lecture en Java (voir ci-dessous) permettent de lire ligne par ligne ; c'est ceci qui permettra d'attribuer un numéro à chaque ligne)

Question :

- Revoyez la conception du programme de constitution du lexique en prenant en compte cette nouvelle information à fournir.

4 Analyse des mots employés par Jules Verne

Nous voulons maintenant connaître les mots les plus employés par Jules Verne (dans les romans que nous analysons).

Question :

- Constituez la liste des quinze mots de plus de trois lettres le plus fréquemment trouvés.
-

Quelques indications et rappels sur la lecture d'un fichier texte, la gestion par ligne, et par mot :

- Lecture d'un fichier texte (création d'un flot de caractères) :

```
Reader reader = new FileReader("xxx.txt");
```

- Prise en compte d'une ligne (création d'un flot de lignes) :

```

BufferedReader in = new BufferedReader(reader);
String ligne = in.readLine();
while (ligne != null) {
    // ... analyse de la ligne courante ...
    ligne = in.readLine(); // lecture de la ligne suivante
}

```

- L'analyse d'une ligne se fait à l'aide d'un `StringTokenizer` qui permet de la découper en mots séparés les uns des autres par un ou plusieurs caractères délimiteurs. Ces caractères sont spécifiés dans une chaîne de caractères. Les mots sont obtenus successivement en invoquant la méthode `nextToken`, tandis que `hasMoreTokens` teste l'existence d'un mot suivant :

```

String delimiters = " '.,; :?!(){}[] «»<>\"\\t";
StringTokenizer st = new StringTokenizer(ligne, delimiters);
while (st.hasMoreTokens()) {
    String mot = st.nextToken();
    // ... traitement du mot courant ...
}

```
