Université de Lille 1 SCIENCES ET TECHNOLOGIES

TP1 – Données structurées dans un fichier

Algorithmes des bases de données
Master Informatique – M1S2
2015–2016

Dans ce TP vous allez implémenter quelques fonctionnalités basiques pour stocker et interroger des données structurées :

- structuration des données dans des relations (ou tables) ;
- stockage des données sur disque ;
- les opérations de sélection et projection.

1 Préparatifs

Il est très important de suivre scrupuleusement les instructions de cette section, y compris pour les noms de classes, packages et répertoires. Cela permettra d'avoir la même structure pour les projets de tous les étudiants, et ainsi mécaniser les tests. De plus, tous les TPs qui suivent se baseront sur cette structure commune.

Mise en place du projet

Q 1. Créer un projet pour la matière ABD dans votre environnement de développement favori. Nous travaillerons sur un seul projet pendant tous les TPs.

Q 2. Récupérer l'archive `tp1-projet-initial.zip` sur Moodle, et copiez la dans le répertoire du projet précédemment créé.

Q 3. Décompresser l'archive en utilisant la ligne de commande. Vous voyez apparaître les répertoires `src, src-fourni, src-exercices, test, test-fourni, test-resources-fourni`.

Q 4. Dans l'IDE, rafraîchir le projet pour voir apparaître les répertoires issus de l'archive.

Q 5. Dans l'IDE, indiquer que les répertoires `src, src-fourni, test, test-fourni` sont des répertoires de sources.

Q 6. Si nécessaire, ajouter la librairie *junit4* à votre projet.

À ce stade, vous ne devriez pas avoir d'erreurs de compilation.

Q 7. Exécuter les deux classes de test `abd.TestFactory_FileTable` et `abd.tp1.TestFileTable`. Les tests ne passent pas, car il manque les implémentations.

Lors de ce TP, vous devrez faire une implémentation de l'interface `abd.tp1.FileTable`, et des deux méthodes se trouvant dans la classe `abd.Factory`.

Organisation du projet Les répertoires qui ont été crées par l'archive seront utilisés comme suit :

src-fourni pour le code source que nous allons vous fournir, en particulier les interfaces à implémenter et des bibliothèques ;

src pour le code source des classes que vous allez écrire. Au final, votre SGBD sera composé du code source contenu dans les deux répertoires `src-fourni` et `src` ;

src-exercices pour le code source d'exercices que nous feront pendant les TPs, mais qui ne feront pas partie du SGBD ;

test-fourni pour les classe de test que nous allons fournir, servant à tester que votre implémentation respecte bien la spécification ;

test pour les classes de test que vous allez écrire ;

test-resources-fourni pour les fichiers de données pour tester.

Vous ne devriez jamais placer des classes et/ou fichiers dans les répertoires `*-fourni`.

2 Implémentation

Lors de ce premier TP, vous ne vous occuperez pas des exceptions dues à des erreurs d'entrée sortie sur les fichiers (`java.io.IOException`) ; ces exceptions ne seront jamais captées et provoqueront l'arrêt du programme. Ceci ne dispense pas de libérer les ressources système utilisées par vos programmes, en particulier les fichiers ouverts.

Il est recommandé de suivre l'ordre indiqué ci-dessous.

Q 8. Dans le répertoire `src`, créer une classe qui implémente `abd.tp1.FileTable`. Dans la suite, nous supposerons que cette classe s'appelle `abd.tp1.FileTableImpl`.

Q 9. Étudier la spécification de l'interface `abd.tp1.FileTable`, en utilisant la documentation (javadoc) se trouvant dans le fichier source.

Q 10. Commencer par écrire le constructeur de `abd.tp1.FileTableImpl` et les deux accesseurs.

Q 11. Implémenter les deux méthodes dans `abd.Factory`, et s'assurer que les tests de `abd.TestFactory_FileTable` passent.

Q 12. Continuer en implémentant la méthode d'ajout et l'itérateur dans `abd.tp1.FileTableImpl`. Tester en utilisant les tests liés à ces méthodes dans `abd.tp1.TestFileTable.`

Q 13. Terminer avec la sélection et la projection de `abd.tp1.FileTableImpl`. Tester.

Voici quelques ressources utiles si vous ne connaissez pas bien l'API java de manipulation de fichiers :

- la classe dans `src-exercices/tp1/LectureEcritureFichiers.java` contient un petit exemple, qui illustre quelques primitives qui devraient vous suffire pour ce TP ;
- ce tutoriel en ligne ;
- bien évidemment, la javadoc.

3 Complexité

Supposons qu'une table d'arité 2 est déjà créée et contient les données du tableau ci-dessous. On calcule une sélection des tuples qui contiennent le caractère 'y' dans la première colonne.

Q 14. Combien de tuples le résultat de la sélection contient-il ?

Q 15. Avec votre implémentation, combien de tuples seront lus sur disque pour calculer le résultat de la sélection ?

ax	1
bx	2
cx	3
dx	4
ey	5
fx	6
gx	7
hy	8
ix	9
jx	10
kx	11

4 Effectuer une requête

Q 16. Compléter le fichier `src-exercices/tp1/ExampleQuery.java` pour effectuer la requête demandée.