



Base de données Postgres

Rappel Connection via jdbc.

Les classes de JDBC permettent des connections entre une base de données relationnelles et Java. La base de données que nous utilisons est postgres (qui s'installe très facilement sous Ubuntu) et dont le manuel d'utilisation est disponible ici <http://www.lif.univ-mrs/~lugiez/Enseignement/Master2/Doc/>. A chaque type de base de données correspond un pilote particulier qu'on peut récupérer sous forme d'archive .jar à rajouter dans le repertoire des librairies externes de Java ou à mettre dans un repertoire mentionné dans votre *CLASSPATH*. Ensuite le pilote java permet de créer une connection par un objet connection qui permet de faire des requêtes sur la base.

Remplir une base de données postgres pour les soutenances de stages des étudiants sachant que :

1. Un étudiant a un nom, prenom, statut (fsi ou bda) et son identifiant.
2. Une plage de soutenance a un jour (1 à 5), une plage horaire (1 à 10) et un numéro de salle et son identifiant.
3. Une soutenance a un titre, un étudiant et une plage de soutenance.

Utilisation de base de données via JDBC

Les classes java peuvent dialoguer avec une base de données grâce aux classes de JDBC (Java Data Base Connectivity) et exécuter des commandes SQL (importer java.sql.*). Pour chaque type de la base de donnée MySQL, postgres,.. il faut charger le pilote correspondant qui devra être sur les chemins de recherche (via CLASSPATH ou lib/ext de l'installation de la machine virtuelle java). L'instruction *Class.forName("nom_du_driver");* initialise le Driver manager.

Un objet de type *Connection* est créé avec l'appel *getConnection(url,user,passwd)* sur l'objet *DriverManager* avec *url* la chaine donnant l'adresse de la base décrite par par *jdbc :type_de_la_base ://adresse_IP/nom_de_la_base*

avec *adresse_IP* est localhost sur la machine locale, *user* est le nom d'utilisateur, *passwd* son mot de passe.

Les méthodes utiles sur l'objet *connection* sont *close()*, *createStatement()*. Les exceptions déclenchées sont usuellement de type *SQLException*.

Ecrire une classe *InteractBD* qui établit une connection avec une base de donnée et contient les méthodes *connect()*, *disconnect()*, *isConnected()*, *insertion*, *deletion*, *update*,

Test JUnit à la main

JUnit permet de réaliser des tests sur la classe *InteractBD* comme pour tout autre classe.

1. Définir un jeu de test pour tester les aspects connection à la base et écrire la classe JUnit *InteractBDtestConnection* correspondante. Vérifier que les tests passent.
2. Définir un jeu de test pour tester les opérations d'interrogations de la base de donnée et écrire la classe JUnit *InteractBDtest* . Vérifier que les tests passent.

Test avec DbUnit

DbUnit (<http://www.dbunit.org/>) est un logiciel permettant d'automatiser les tests sur les bases de données. Il permet de réaliser les tests sur des données structurées sous forme de tables.

- L'interface *IDatabaseConnection* représente une connection avec une base de données.
- L'interface *IDataSet* représente une suite de tables.
- La classe abstraite *DataBaseOperation* représente une opération effectuée avant et après chaque test.

Les données sont usuellement données via un fichier sous format *flat XML file* (mais on peut les obtenir aussi par une requête sql sur une base de données, un fichier excell,... Reprendre le travail précédent en utilisant *dbunit*.

Travail à rendre

Le travail demandé est d'une part les classes des tests effectués directement avec une connexion à la base de donnée et d'autre part les test effectués en utilisant *dbunit* pour simuler celle-ci.

- Les programmes devront respecter la norme SUN (disponible sur la page www du cours)
- La programmation se fera par groupe de deux, l'un programmant l'autre conseiller.
- Le résultat sera mis dans un fichier dont le nom est
 - **NOM DU CHEF DU PROJET-identifiant du projet.suffixe**
Les suffixes sont les suffixes classiques .txt, .pdf, .ps
 - Ce fichier contiendra les sources des classes *InteractBD.java*, *InteractBDTest.java* et *InteractBDConnectionTest.java*, *ListPointExcepTest.java* imprimées à l'aide d'un formateur comme *a2ps*.
 - L'en-tête du fichier contiendra les noms des membres du projets en identifiant clairement le responsable ainsi qu'une description succincte du travail demandé.
 - L'identifiant de ce projet est *tpDB*.