Université Blida 1 Saad Dahlab
Faculté des Sciences
Département d'Informatique

Master 2 IL
1er semestre 2024/2025
Architecture Logicielle

- TP 4 -

JDBC et patron DAO

Ce TP a pour objectif de vous familiariser avec l'API JDBC (Java DataBase Connectivity) ainsi qu'au design pattern DAO (Data Access Object).

Exercice 1 :

Ce TP s'inscrit dans la suite du TP3 dans lequel une application de gestion des utilisateurs a été mise en œuvre en utilisant une simple ArrayList pour illustrer les différentes opérations. Au cours de ce TP, une base de données MySQL sera utilisée pour la persistance des données sur les utilisateurs. Pour cela on utilisera l'API JDBC qui est une API Java permettant de se connecter et d'interroger une base de données.

Avant de développer la partie accès à la base de données, on commencera d'abord par créer la base de données ainsi que la table *users* correspondant au schéma suivant :

users (id, name, email)

Pour l'identifiant, on utilisera un champ auto incrémenté.

Une fois la base de données créée, on écrira le code nécessaire pour la persistance des données.

Exercice 2 :

L'inconvénient de la solution proposée lors de l'exercice 1 est que les classes métier (User) contiennent du code lié à la persistance. Cela réduit les possibilités de réutilisation des classes métier puisqu'elles sont fortement liées au SGBD utilisé. De plus, avec une telle conception, le principe Single Responsability Principle (SRP) n'est pas respecté.

Le design pattern DAO (Data Access Object) permet justement d'y remédier aux inconvénients précédents en encapsulant la logique de persistance dans des objets à part. Ces objets sont accessibles à travers des interfaces standards.

Modifier la solution de l'exercice 1 en écrivant une interface UserDAO ainsi qu'une classe UserDAOImpl qui implémente cette interface.

Modifier le contrôleur afin qu'il utilise désormais l'interface UserDAO avec l'injection de la dépendance.