

LE DEVELOPPEMENT EN COUCHES AVEC JAVA SE

Module 4 – Le développement de la couche Data Access Layer (DAL)

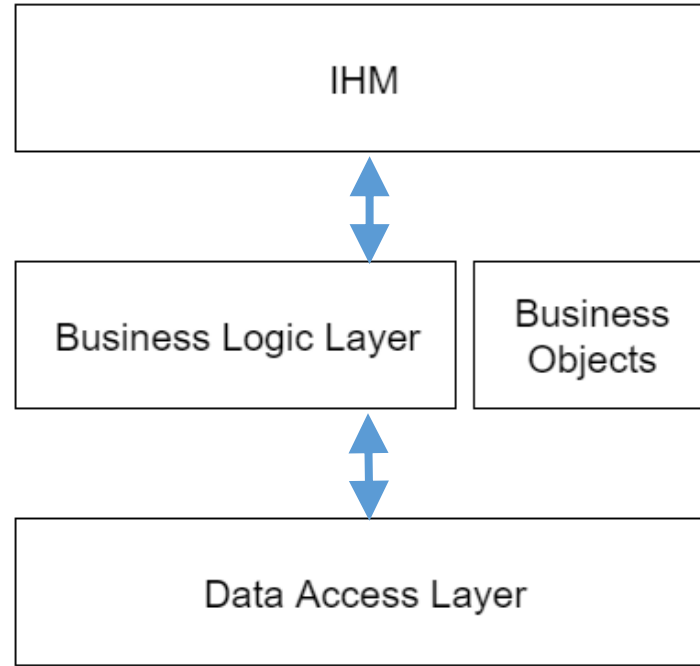


Objectifs

- Situer la couche Data Access Layer (DAL)
- Responsabilités de la couche DAL
- Comprendre l'organisation de la couche DAL
- Savoir implémenter la couche DAL
- Connaître et utiliser l'API JDBC
- Comprendre le design pattern DAO

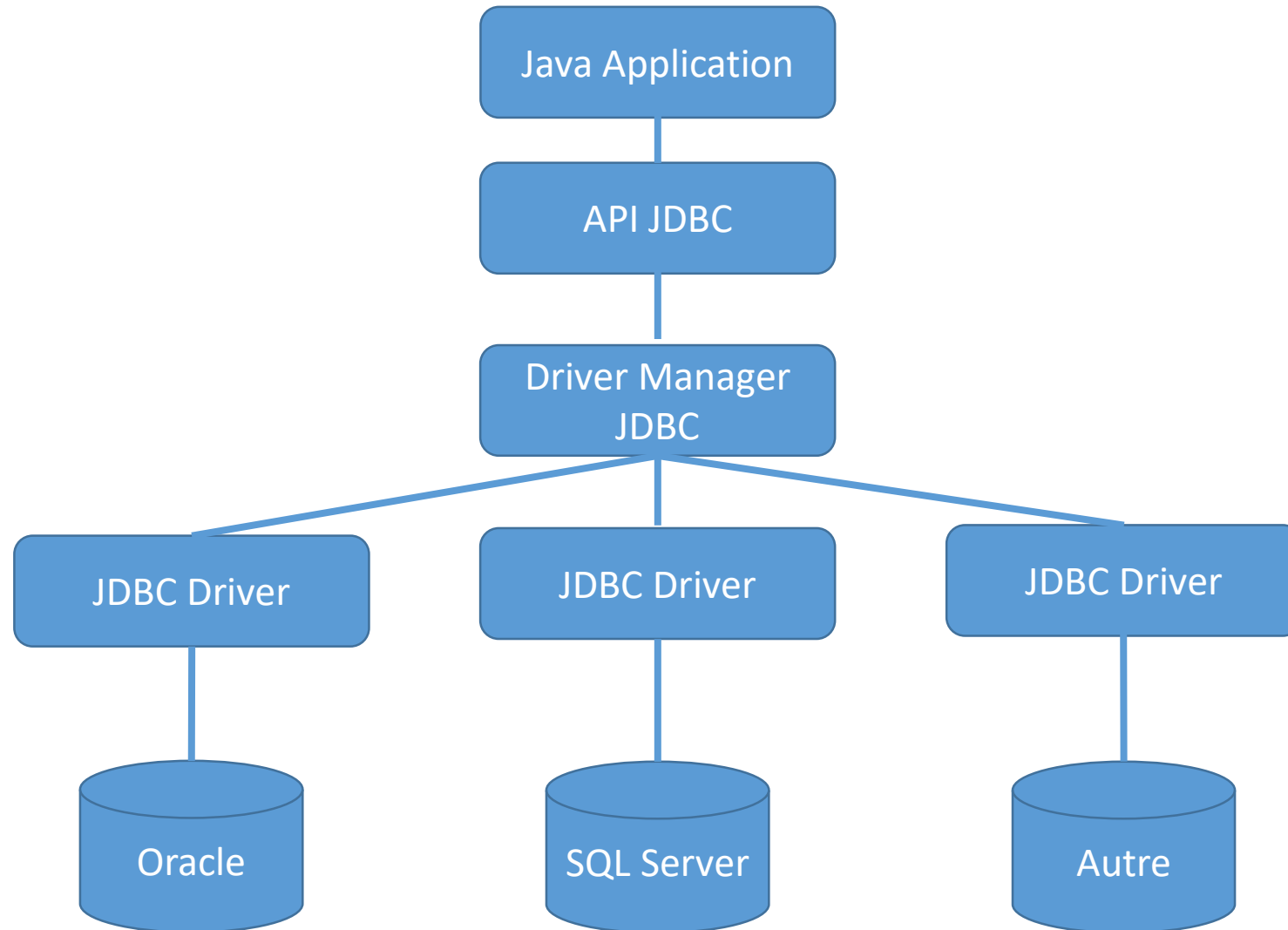
Le développement de la couche DAL

Situer la couche DAL



Le développement de la couche DAL

L'architecture JDBC



Mettre en place l'environnement

- Java est installé
 - JDK pour un poste de développement
 - JRE pour un poste de production
- Une base de données est installée
- Récupérer le pilote correspondant à la base de données installée
- Ajouter le chemin d'accès au pilote dans le « classpath »

Le développement de la couche DAL

Mettre en place l'environnement

Démonstration



Charger le pilote JDBC

- Méthode recommandée par Oracle :

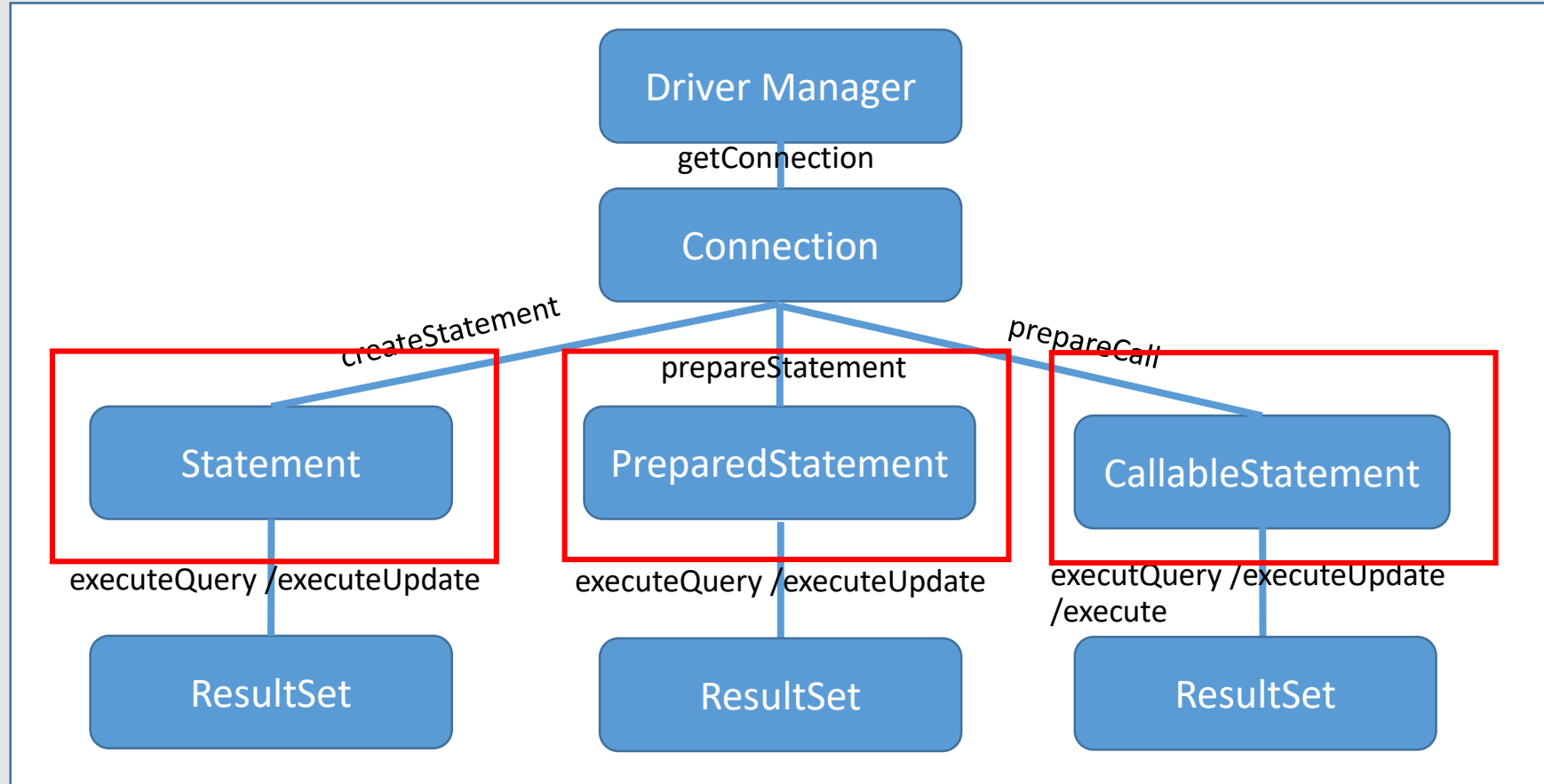
```
DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver());
```

- Méthode évitant la dépendance forte

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

- Chargement automatique à partir de JDBC v4.

Présentation de l'API JDBC



Etablir une connexion

- Créer une connexion

```
String url = "jdbc:sqlserver://localhost:1433;databaseName=EVAL_DB";  
Connection uneConnection = DriverManager.getConnection(url, "sa", "Pa$$w0rd");
```

- Fermer une connexion

```
uneConnection.close();
```

Créer un Statement

- L'interface `java.sql.Statement` représente une requête SQL
- Créer un Statement

```
Statement unStmt = uneConnection.createStatement();
```

Exécuter un Statement

- Requête sans modification de données

```
ResultSet rs = unStmt.executeQuery("select prenom, nom, email from stagiaires");
```

- Requête avec modification de données

```
int nbRows = unStmt.executeUpdate(requeteSql);
```

Fermer un Statement

- TOUJOURS FERMER LES STATEMENTS APRES UTILISATION

```
unStmt.close();
```

Le développement de la couche DAL

Exploiter le résultat d'un Statement

- ResultSet : données structurées de manière tabulaire
- Se déplacer dans les lignes
 - beforeFirst() (Position par défaut)
 - first()
 - next()
 - previous()
 - last()
 - ...
- Récupérer les valeurs des colonnes de la ligne courante
 - Méthodes *getTypeColonne*(colonne)
 - getString(colonne)
 - getInt(colonne)
 - getDate(colonne)
 - ...
- Identifier une valeur NULL avec wasNull()



Se déplacer dans un ResultSet

- createStatement(type, mode)
- Types de déplacement
 - ResultSet.TYPE_FORWARD_ONLY
 - ResultSet.TYPE_SCROLL_INSENSITIVE
 - ResultSet.TYPE_SCROLL_SENSITIVE
- Mode d'ouverture
 - ResultSet.CONCUR_READ_ONLY
 - ResultSet.CONCUR_UPDATEABLE

```
stmt = connection.createStatement(resultSet.TYPE_FORWARD_ONLY, resultSet.CONCUR_READ_ONLY);
```

La requête paramétrée

```
public void update(String prenom, String nom, int noStagiaire) throws SQLException{  
    String sql = "update STAGIAIRES set prenom= " + prenom + ",nom=" + nom  
        + " where noStagiaire=" + noStagiaire;  
  
    Connection connection = JdbcTools.getInstance().getConnection();  
    Statement stmt = connection.createStatement();  
  
    stmt.executeUpdate(sql);  
}
```

```
String sql = "update STAGIAIRES set prenom=?, nom=?,email=?,motDePasse=? where noStagiaire=?";  
PreparedStatement stmt = connection.prepareStatement(sql);
```

Utiliser une requête paramétrée

```
String sql = "update STAGIAIRES set prenom=?, nom=? where noStagiaire=?";
```

- Créer un objet PreparedStatement

```
PreparedStatement stmt = connection.prepareStatement(sql);
```

- Fixer les valeurs des paramètres

```
stmt.setString(1, "Bob");  
stmt.setString(2, "Leponge");  
stmt.setInt(3, 10);
```

- Remarque : Utiliser la méthode setNull() pour initialiser un paramètre à Null

- Exécuter

```
stmt.executeUpdate();
```


Le développement de la couche DAL

Exécuter une requête SQL depuis Java

Démonstration



Le développement de la couche DAL

Gestion d'une papeterie - partie 2

TP



Appeler une procédure stockée

- Utiliser la classe `java.sql.CallableStatement`
- Création du `callableStatement` depuis une instance de `Connection`
`CallableStatement prepareCall(String sql) throws SQLException`
- Syntaxe de la requete sql :
`{ [?=] call procedure-name([parameter][, [parameter]]...) }`

Exemple de procédure stockée

```
CREATE PROCEDURE dbo.insertPersonne
    @prenom nvarchar(30),
    @nom nvarchar(30),
    @dateNaissance date,
    @idPersonne integer output
AS
    insert into dbo.PERSONNE (prenom, nom, dateNaissance, archive)
        values (@prenom, @nom, @dateNaissance, 0 );
    set @idPersonne = @@Identity;
GO
```

Appeler une procédure stockée

- Créer un objet CallableStatement

```
String sql = "{ call dbo.insertPersonne( ?, ?, ?, ?) }";  
CallableStatement callStmt = con.prepareStatement(sql);
```

- Renseigner les valeurs des paramètres in

```
callStmt.setString(1, "Nordine");  
callStmt.setString(2, "NATEUR");  
callStmt.setDate(3, new Date(1aDate.getTime()));
```

- Définir les paramètres out

```
callStmt.registerOutParameter(4, Types.INTEGER);
```

- Exécuter

```
callStmt.execute();
```

- Récupérer la valeur du paramètre

```
callStmt.getInt(4)
```

Gérer les transactions

- Activer/Désactiver le mode auto-commit

```
connection.setAutoCommit(false);
```

- Gestion d'une transaction

- Début implicite
- Valider une transaction

```
connection.commit();
```

- Annuler les modifications

```
connection.rollback();
```

Externaliser la chaîne de connexion

- Par fichier Properties

- Format texte (extension .properties)

```
# Paramètres de connexion à SQLServer
driverdb=com.microsoft.sqlserver.jdbc.SQLServerDriver
urldb=jdbc:sqlserver://127.0.0.1;databasename=EVAL_DB;
userdb=sa
passworddb=Pa$$w0rd
```

- Format XML (DTD : <http://java.sun.com/dtd/properties.dtd>)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>Paramètres de connexion à SQLServer</comment>
    <entry key="driverdb">com.microsoft.sqlserver.jdbc.SQLServerDriver</entry>
    <entry key="urldb">jdbc:sqlserver://127.0.0.1;databasename=EVAL_DB</entry>
    <entry key="userdb">sa</entry>
    <entry key="passworddb">Pa$$w0rd</entry>
</properties>
```

Le développement de la couche DAL

Externaliser la chaîne de connexion

Démonstration



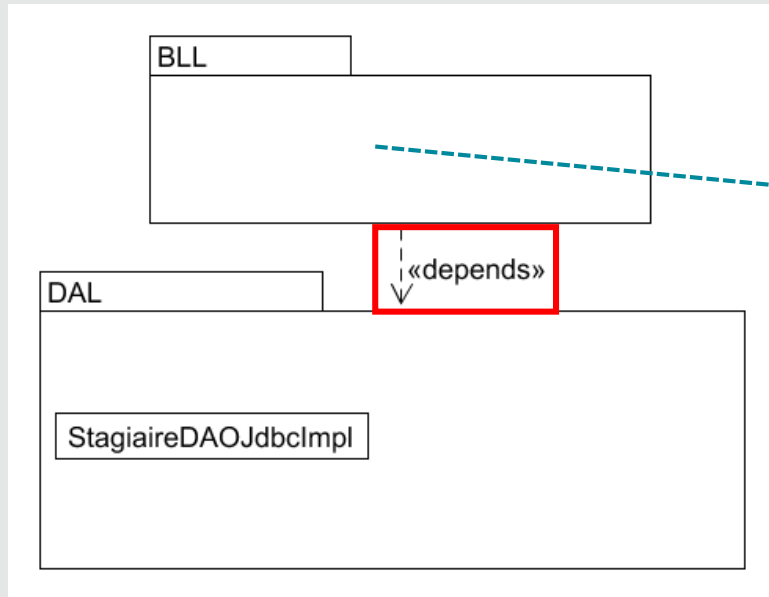
Le développement de la couche DAL

Définir le Data Access Object (DAO)



Le développement de la couche DAL

Utiliser la DAL depuis la BLL

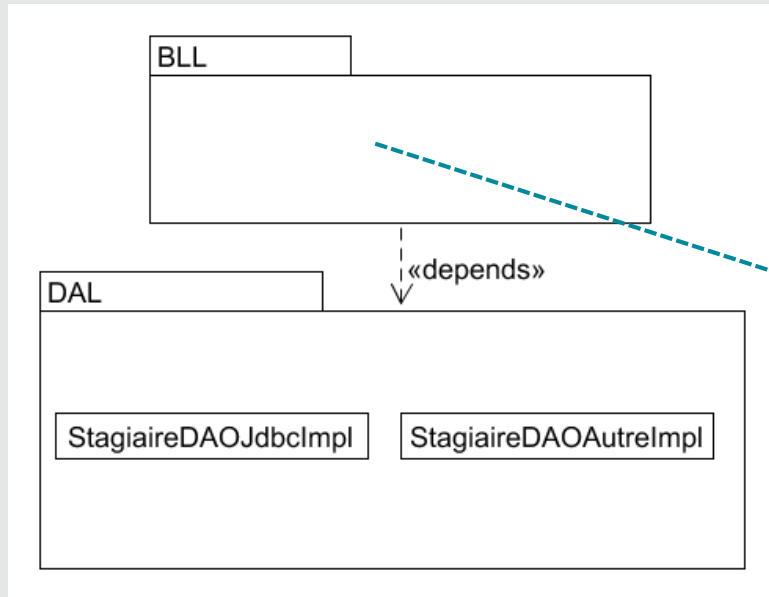


```
import fr.eni.evaluations.dal.jdbc.StagiaireDAOJdbcImpl;
```

```
StagiaireDAOJdbcImpl stagiaireDAO = new StagiaireDAOJdbcImpl();
```

Le développement de la couche DAL

Utiliser la DAL depuis la BLL



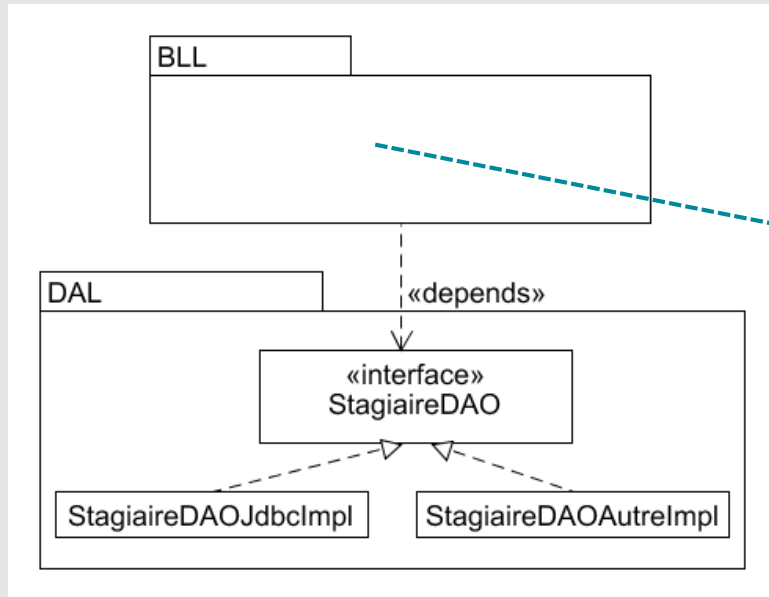
```
import fr.eni.evaluations.dal.jdbc.StagiaireDAOJdbcImpl;
```

```
StagiaireDAOJdbcImpl stagiaireDAO = new StagiaireDAOJdbcImpl();
```

```
StagiaireDAOAutreImpl stagiaireDAO = new StagiaireDAOAutreImpl();
```

Le développement de la couche DAL

Utiliser la DAL depuis la BLL



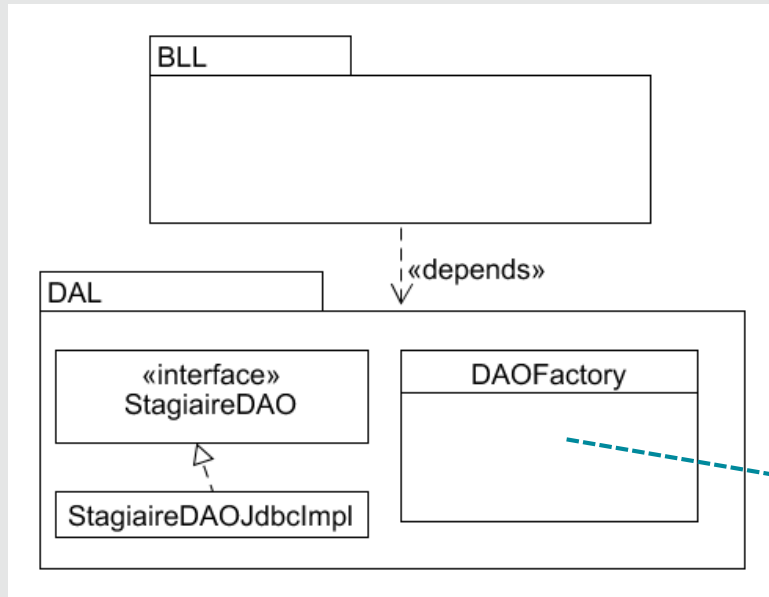
```
import fr.eni.evaluations.dal.StagiaireDAO;  
import fr.eni.evaluations.dal.jdbc.StagiaireDAOJdbcImpl;
```

```
StagiaireDAOAutreImpl stagiaireDAO = new StagiaireDAOAutreImpl();
```

```
StagiaireDAO stagiaireDAO = new StagiaireDAOAutreImpl();
```

Le développement de la couche DAL

Utiliser une Factory



```
StagiaireDAO stagiaireDAO = new StagiaireDAOJdbcImpl();
```

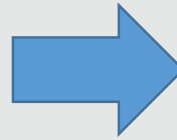
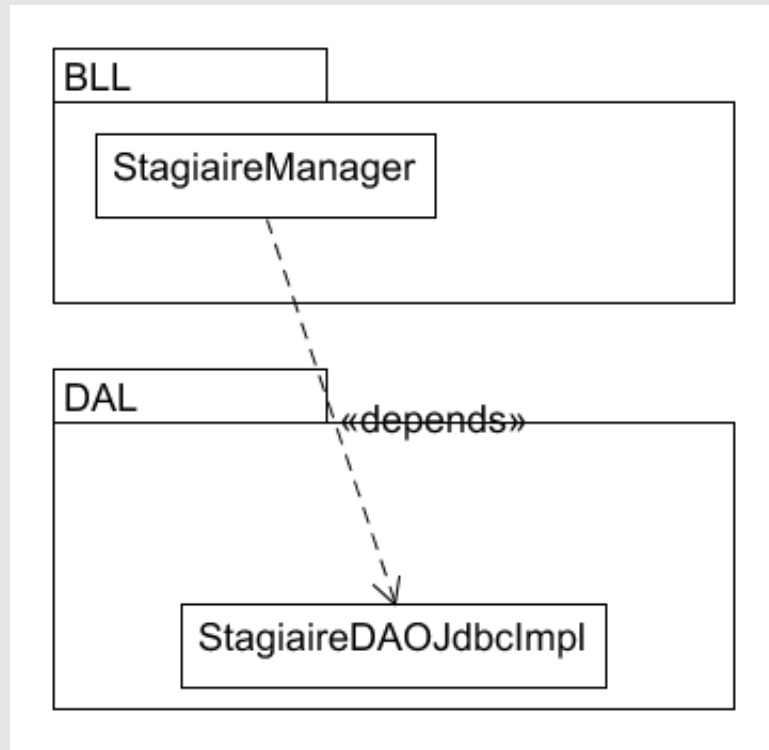
```
StagiaireDAO stagiaireDAO = DAOFactory.getStagiaireDAO();
```

```
public class DAOFactory {  
  
    public static StagiaireDAO getStagiaireDAO() {  
        StagiaireDAO stagiaireDAO= new StagiaireDAOJdbcImpl();  
        return stagiaireDAO;  
    }  
}
```

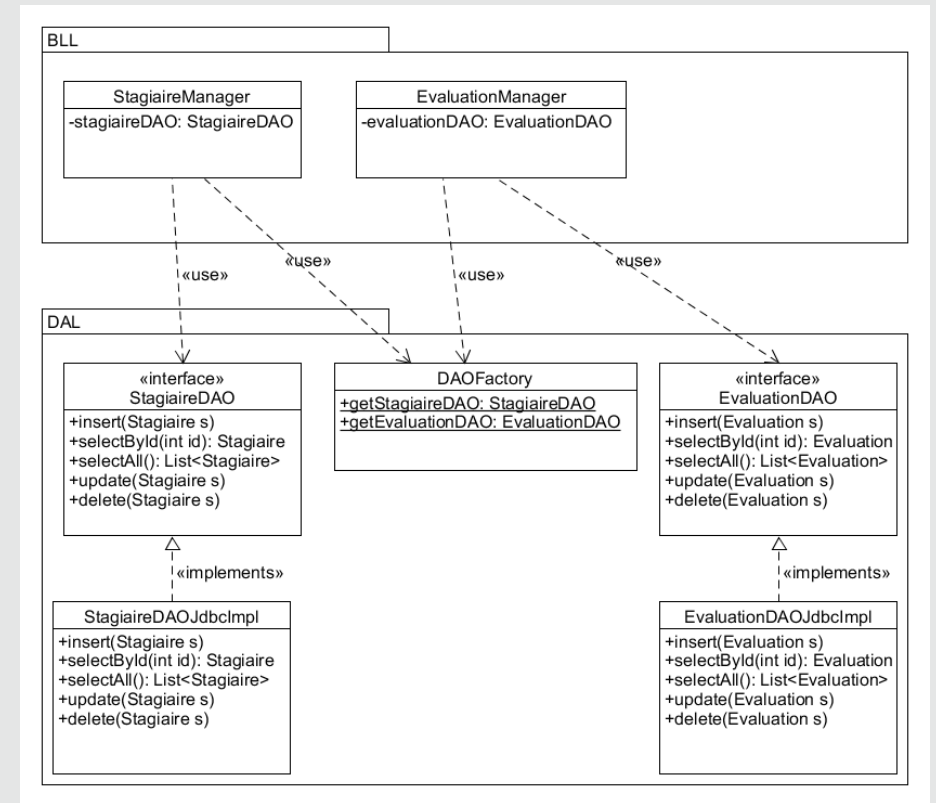
Le développement de la couche DAL

Utiliser la DAL depuis la BLL

AVANT

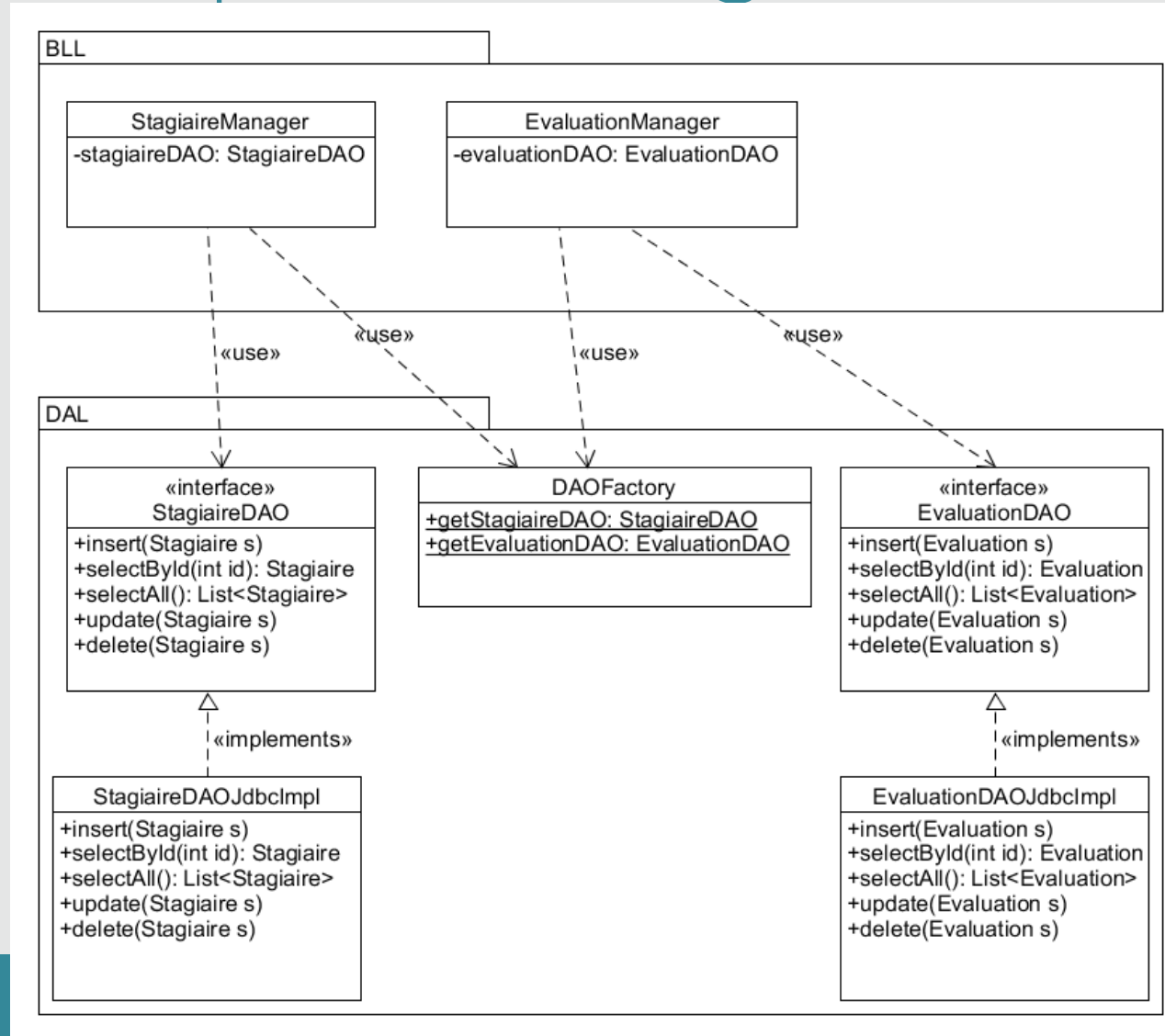


APRES



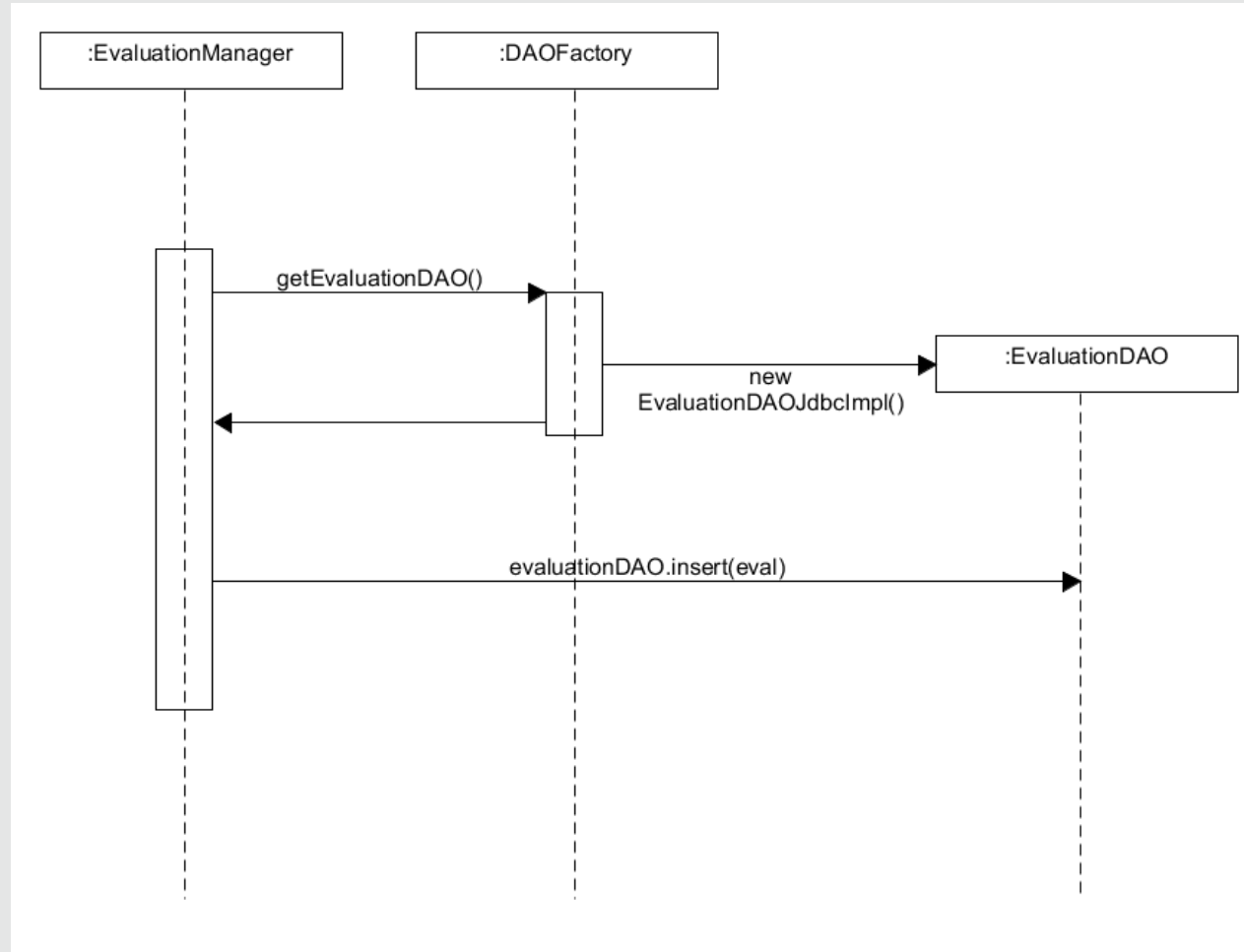
Le développement de la couche DAL

Le modèle statique du Design Pattern DAO



Le développement de la couche DAL

Le modèle dynamique du Design Pattern DAO



Le développement de la couche DAL

Le Design Pattern DAO

Démonstration



Le développement de la couche DAL

Gestion d'une papeterie - partie 3

TP

