#### Министерство образования и науки Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение

#### высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №3

по дисциплине   
«Распределённые программные системы»

Тема: «Работа с реляционной СУБД средствами JDBC API»

Выполнил:

Ст.гр.ПРИ-116

Чернова Д.В.

Принял:

Проскурина Г.В.

Владимир, 2018 г.

ЦЕЛЬ РАБОТЫ.

Получить навыки практического использования средства JDBC API для реализации объектов доступа к данным (DAO).

ХОД РАБОТЫ.

Интерфейс DAO.java:

**package** ru.shvartz.lab2;  
  
**import** java.sql.Connection;  
**import** java.sql.SQLException;  
**import** java.util.List;  
  
**public interface** DAO {  
 **void** insertTable(Connection connection, UserDAO user) **throws** SQLException;  
 **void** updateTable (Connection connection, UserDAO user) **throws** SQLException;  
 List<UserDAO> selectTable(Connection connection) **throws** SQLException;  
 **void** deleteTable(Connection connection, **int** id) **throws** SQLException;  
}

Модель UserDAO.java:

**package** ru.shvartz.lab2;  
  
**import** java.sql.Connection;  
  
**public class** UserDAO {  
 **private int id**;  
 **private** String **name**;  
 **private** String **email**;  
 **private int courseId**;  
  
  
 **public int** getId() {  
 **return id**;  
 }  
  
 **public** UserDAO(**int** id, String name, String email) {  
 **this**.**id** = id;  
 **this**.**name** = name;  
 **this**.**email** = email;  
 *//this.courseId = courseId;* }  
  
 **public** UserDAO(Connection connection) {  
 **this**.**id** = 1;  
 **this**.**name** = **"name1"**;  
 **this**.**email** = **"mail"**;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public** String getEmail() {  
 **return email**;  
 }  
  
 **public int** getCourseId() {  
 **return courseId**;  
 }  
}

Модель CourseDAO.java:

**package** ru.shvartz.lab2;  
  
**public class** CourceDAO {  
 **private int id**;  
 **private** String **name**;  
 **private** String **description**;  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public** String getDescription() {  
 **return description**;  
 }  
}

Класс ConnectionModel.java:

**package** ru.shvartz.lab2;  
  
**import** java.sql.Connection;  
**import** java.sql.DriverManager;  
**import** java.sql.SQLException;  
  
**public class** ConnectionModel {  
 **private** String **host**;  
 **private int port**;  
 **private** String **database**;  
 **private** String **user**;  
 **private** String **password**;  
 **private** String **url**;  
 **private** String **driverName**;  
  
 **public** ConnectionModel(String host, **int** port, String database, String user, String password) {  
 **this**.**host** = host;  
 **this**.**port** = port;  
 **this**.**database** = database;  
 **this**.**user** = user;  
 **this**.**password** = password;  
 **this**.**url** = **"jdbc:mysql://"** + host + **":"** + port + **"/"** + database + **"?serverTimezone=UTC"**;  
 **this**.**driverName** = **"com.mysql.jdbc.Driver"**;  
 }  
  
 **public** String getUser() { **return user**; }  
 **public** String getPassword() { **return password**; }  
 **public** String getUrl() { **return url**; }  
 **public** String getDriverName() { **return driverName**; }  
  
 **public static** Connection getDBConnection() {  
 ConnectionModel connectionModel = **new** ConnectionModel(**"localhost"**, 3306, **"lab2"**, **"root"**, **"AkwcEcsE"**);  
 Connection connection = **null**;  
 **try** {  
 Class.*forName*(connectionModel.getDriverName());  
 } **catch** (ClassNotFoundException e) {  
 System.***out***.println(e.getMessage());  
 }  
 **try** {  
 connection = DriverManager.*getConnection*(connectionModel.getUrl(), connectionModel.getUser(), connectionModel.getPassword());  
 **return** connection;  
 } **catch** (SQLException e) {  
 System.***out***.println(e.getMessage());  
 }  
 **return** connection;  
 }  
  
}

Класс ConnectionClass.java:

**package** ru.shvartz.lab2;  
  
**import** java.sql.\*;  
**import** java.util.Scanner;  
  
**public class** ConnectionClass {  
  
  
 **public static void** main(String[] args) **throws** SQLException {  
 Connection connection = ConnectionModel.*getDBConnection*();  
 Scanner scan = **new** Scanner(System.***in***);  
  
 System.***out***.println(**"Введите команду"**);  
  
 **int** command = 0;  
  
 CRUDOperations crudOperations = **new** CRUDOperations();  
 **try** {  
 **do** {  
 System.***out***.println(**"1 - insert, 2 - update, 3 - select, 4 - delete"**);  
 command = scan.nextInt();  
 **switch** (command) {  
 **case** 1: {  
 crudOperations.insertTable(connection, **new** UserDAO(4, **"namelksjd"**, **"mail@mail.ru"**));  
 **break**;  
 }  
 **case** 2: {  
 crudOperations.updateTable(connection, **new** UserDAO(2,**"nameupdated"**, **"mailupdated"**));  
 **break**;  
 }  
 **case** 3: {  
 crudOperations.selectTable(connection);  
 **break**;  
 }  
 **case** 4: {  
 System.***out***.println(**"Введите id объекта"**);  
 crudOperations.deleteTable(connection, scan.nextInt());  
 **break**;  
 }  
 }  
 } **while** (command != 0);  
 } **catch** (SQLException e) {  
 System.***out***.println(e.getMessage());  
 } **finally** {  
 connection.close();  
 }  
 }  
}

Класс CRUDOperations.java:

**package** ru.shvartz.lab2;  
  
**import** java.sql.Connection;  
**import** java.sql.PreparedStatement;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** CRUDOperations **implements** DAO {  
  
 **private static final** String ***insertIntoTable*** = **"INSERT INTO users (id, name, email) VALUES(?,?,?)"**;  
 **private static final** String ***selectFromTable*** = **"SELECT** *\** **FROM users LIMIT 10"**;  
 **private static final** String ***deleteFromTable*** = **"DELETE FROM users where id = ?"**;  
 **private static final** String ***updateTableName*** = **"UPDATE users SET name = ?, email = ? where id = ?"**;  
  
  
 **public void** getById(Connection connection, **int** id) **throws** SQLException {  
 PreparedStatement preparedStatement= **null**;  
 **try** {  
 preparedStatement = connection.prepareStatement(**"Select from user where id = ?"**);  
 preparedStatement.setInt(1, id);  
 preparedStatement.execute();  
 System.***out***.println(**"record with id = "** + id);  
 } **catch** (SQLException e) {  
 System.***out***.println(e.getMessage());  
 }  
 }  
  
 @Override  
 **public void** insertTable(Connection connection, UserDAO user) **throws** SQLException {  
 PreparedStatement preparedStatement = **null**;  
 **try** {  
 preparedStatement = connection.prepareStatement(***insertIntoTable***, preparedStatement.***RETURN\_GENERATED\_KEYS***);  
 preparedStatement.setInt(1, user.getId());  
 preparedStatement.setString(2, user.getName());  
 preparedStatement.setString(3, user.getEmail());  
 preparedStatement.execute();  
 System.***out***.println(**"insert the record. id = "** + user.getId());  
 } **catch** (SQLException e) {  
 System.***out***.println(**"CreateTable method failed"**);  
 System.***out***.println(e.getMessage());  
 }  
 }  
  
 @Override  
 **public void** updateTable (Connection connection, UserDAO user) **throws** SQLException{  
 PreparedStatement preparedStatement = connection.prepareStatement(***updateTableName***);  
  
 **try** {  
 preparedStatement.setInt(3, user.getId());  
 preparedStatement.setString(1, user.getName());  
 preparedStatement.setString(2, user.getEmail());  
 preparedStatement.executeUpdate();  
 } **catch** (SQLException e) {  
 System.***out***.println(**"updateTable failed"**);  
 System.***out***.println(e.getMessage());  
 }  
 }  
  
 @Override  
 **public** List<UserDAO> selectTable(Connection connection) **throws** SQLException {  
  
 List<UserDAO> users = **new** ArrayList<>();  
 PreparedStatement preparedStatement = connection.prepareStatement(***selectFromTable***);  
 ResultSet resultSet = preparedStatement.executeQuery();  
  
 **try** {  
 **while** (resultSet.next()) {  
 users.add(**new** UserDAO(resultSet.getInt(**"id"**), resultSet.getString(**"name"**), resultSet.getString(**"email"**)));  
 System.***out***.print(resultSet.getInt(1) + **"|"**);  
 System.***out***.print(resultSet.getString(2) + **"|"**);  
 System.***out***.println(resultSet.getString(3) + **"|"**);  
 }  
 } **catch** (SQLException e) {  
 System.***out***.println(**"selectTable failed"**);  
 System.***out***.println(e.getMessage());  
 }  
 **return** users;  
 }  
  
 @Override  
 **public void** deleteTable(Connection connection, **int** id) **throws** SQLException {  
 PreparedStatement preparedStatement = connection.prepareStatement(***deleteFromTable***);  
 **try** {  
 preparedStatement.setInt(1, id);  
 preparedStatement.execute();  
 } **catch**(SQLException e) {  
 System.***out***.println(**"deleteTable was failed"**);  
 System.***out***.println(e.getMessage());  
 }  
 }  
  
}

Результат работы программы представлен на рис.1.

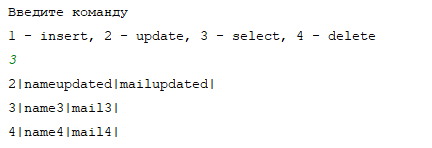


Рисунок . Метод select

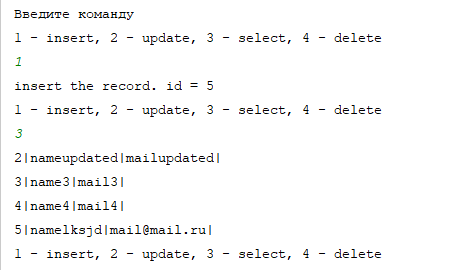


Рисунок .Метод insert

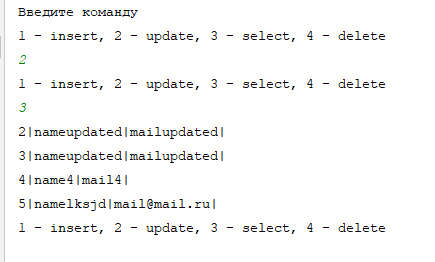


Рисунок . Метод update

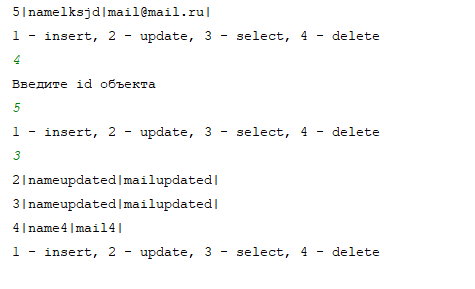


Рисунок . Метод delete

ВЫВОДЫ.

В ходе выполнения лабораторной работы я ознакомилась с паттерном доступа к данным DAO и выполнила практический пример.