

**НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА**
Институт радиоэлектроники и информационных технологий
Кафедра “Прикладная математика”

**Лабораторная работа №2 по курсу «Базы данных»
«SQL + JDBC + DAO »**

Выполнила:
Горенкова А. В.
Группа 16-ПМ
Проверил:
Моисеев А.Е.

НИЖНИЙ НОВГОРОД
2018 г.

Оглавление

Введение.....	3
Задание.....	4
Решение.....	5
Результаты работы.....	7
Листинг.....	8

Введение

Во время выполнения предыдущей лабораторной работы мы познакомились с языком SQL (*structured query language* — «язык структурированных запросов») и JDBC, стандартом взаимодействия приложения на языке Java с различными СУБД с помощью языка SQL. Однако, часто при разработке приложений встаёт проблема написания интерфейса взаимодействия между пользователем и базой данных. Появляется необходимость в прослойке, которая осуществляла бы работу с базой данных. В этом может помочь один из шаблонов проектирования — DAO (Data Access Object) - это объект, который предоставляет абстрактный интерфейс к какому-либо типу базы данных, та самая прослойка между БД и системой.

Вершиной иерархии DAO является абстрактный класс или интерфейс с описанием общих методов, которые будут использоваться при взаимодействии с базой данных. Как правило, это методы поиска, удаление по ключу, обновление и т. д. Набор методов не является завершённым, он зависит от конкретной системы.

Рассмотрим DAO-шаблон в работе с Java-приложением и JDBC-интерфейсом.

Задание

- Добавить в Л.Р.№1 интерфейс DAO, а так же добавить вторую таблицу, связанную с таблицей из Л.Р.№1.
- Сделать несколько запросов к таблицам.

Решение

1. Создадим классы Furniture и Shop, которые будут описывать элементы в таблицах мебели и мебельных магазинах

2. Создадим интерфейс для работы с таблицей Furniture:

Создаём класс FurnitureFace , в котором реализуем методы соединения с базой данных, получения всех данных, удаления, добавления и обновления элемента в таблице Furniture.

```
public class FurnitureFace {  
    public Connection getConn();  
    public Set<Furniture> getAll();  
    public boolean delete(int id);  
    public boolean add(Furniture obj);  
    public boolean update(int id, Furniture obj);  
}
```

3. Создадим интерфейс для работы с таблицей Shop:

Создаём класс ShopFace , в котором реализуем методы соединения с базой данных и получения данных определённого магазина из таблицы Shop.

```
public class ShopFace {  
    public Connection getConn();  
    public boolean getShop(int id);  
}
```

4. Импортируем необходимые для работы библиотеки:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.Statement;  
import java.util.Scanner;
```

5. Подключаемся к базе данных и создаем Statement:

```
Class.forName(JDBC_DRIVER).newInstance();  
Connection con = DriverManager.getConnection(DB_URL);  
Statement stm = con.createStatement();
```

6. Создаём и заполняем таблицы

```

stm.executeUpdate("CREATE TABLE Furniture" +
    "(id INTEGER NOT NULL AUTO_INCREMENT, " +
    " name VARCHAR(255), " +
    " color VARCHAR(255), " +
    " price INTEGER, " +
    " shopId INTEGER NOT NULL)");
stm.executeUpdate("CREATE TABLE Shop" +
    "(id INTEGER NOT NULL AUTO_INCREMENT, " +
    " name VARCHAR(255), " +
    " adress VARCHAR(255))");

stm.executeUpdate("INSERT INTO Furniture (name, color, price, shopId) VALUES
('Table', 'White', 3200, 1)");
stm.executeUpdate("INSERT INTO Furniture (name, color, price, shopId) VALUES
('Table', 'Black', 1000, 1)");
...

stm.executeUpdate("INSERT INTO Shop (name, adress) VALUES ('Akson',
'Komsomolskoe shosse, 14')");
stm.executeUpdate("INSERT INTO Shop (name, adress) VALUES ('IKEA', 'TRC MEGA,
pos Fediakovo')");
...

```

7. Напишем небольшое меню для работы с базой данных

```

Scanner sc = new Scanner(System.in);
System.out.println("1 - print all \n2 - add \n3 - delete \n4 - print all in
shop");
int key = sc.nextInt();

```

Результаты работы

1. Получаем из базы данных всю мебель:

```
FurnitureFace tf = new FurnitureFace();  
for(Furniture f : tf.getAll()) {  
    System.out.println("id: " + f.getId() + " name: " + f.getName() + " color:  
" + f.getColor() + " price: " + f.getPrice() + " shopId: " + f.getShopId());  
}
```

```
1 - print all  
2 - add  
3 - delete  
4 - print all in shop  
1  
id: 2 name: Table color: Black price: 1000 shopId: 1  
id: 16 name: Cupboard color: Green price: 8000 shopId: 4  
id: 18 name: Shelf color: Red price: 800 shopId: 4  
id: 9 name: Shelf color: White price: 8000 shopId: 3  
id: 10 name: Cupboard color: Brown price: 20000 shopId: 3  
id: 20 name: Chair color: White price: 500 shopId: 4  
id: 3 name: Cupboard color: Blue price: 9000 shopId: 1  
id: 1 name: Table color: White price: 3200 shopId: 1  
id: 15 name: Shelf color: Green price: 1000 shopId: 3  
id: 17 name: Chair color: Brown price: 800 shopId: 4  
id: 5 name: Char color: Brown price: 2000 shopId: 2  
id: 12 name: Shelf color: Brown price: 200 shopId: 3  
id: 6 name: Char color: Green price: 1800 shopId: 2  
id: 8 name: Table color: Black price: 10000 shopId: 3  
id: 4 name: Chair color: Blue price: 500 shopId: 2  
id: 11 name: Table color: Red price: 9000 shopId: 3  
id: 14 name: Cupboard color: Brown price: 5000 shopId: 3  
id: 7 name: Chair color: Green price: 400 shopId: 2  
id: 19 name: Table color: Green price: 9000 shopId: 4  
id: 13 name: Chair color: White price: 500 shopId: 3
```

2. Получаем из базы данных данные магазина с id = 1 и всю мебель из этого магазина:

```
System.out.println("Введите id магазина");  
int idSh = sc.nextInt();  
ShopFace sh = new ShopFace();  
sh.getShop(idSh);
```

```
1 - print all  
2 - add  
3 - delete  
4 - print all in shop  
4  
Введите id магазина  
1  
name: Akson adress: Komsomolskoe shosse, 14  
name: Table color: White price: 3200  
name: Table color: Black price: 1000  
name: Cupboard color: Blue price: 9000
```

Листинг

Main.java:

```
package lab2;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.util.Scanner;

public class Main {
    static final String JDBC_DRIVER = "org.h2.Driver";
    static final String DB_URL = "jdbc:h2:file:~/base2";
    public static void main(String[] args) {
        try {
            Class.forName(JDBC_DRIVER).newInstance();
            Connection con = DriverManager.getConnection(DB_URL);
            Statement stm = con.createStatement();
            stm.executeUpdate("CREATE TABLE Furniture" +
                "(id INTEGER NOT NULL AUTO_INCREMENT, " +
                " name VARCHAR(255), " +
                " color VARCHAR(255), " +
                " price INTEGER, " +
                " shopId INTEGER NOT NULL)");
            stm.executeUpdate("CREATE TABLE Shop" +
                "(id INTEGER NOT NULL AUTO_INCREMENT, " +
                " name VARCHAR(255), " +
                " adress VARCHAR(255))");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Table', 'White', 3200, 1)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Table', 'Black', 1000, 1)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Cupboard', 'Blue', 9000, 1)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Chair', 'Blue', 500, 2)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Char', 'Brown', 2000, 2)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Char', 'Green', 1800, 2)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Chair', 'Green', 400, 2)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Table', 'Black', 10000, 3)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Shelf', 'White', 8000, 3)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Cupboard', 'Brown', 20000, 3)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Table', 'Red', 9000, 3)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Shelf', 'Brown', 200, 3)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Chair', 'White', 500, 3)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Cupboard', 'Brown', 5000, 3)");
            stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Shelf', 'Green', 1000, 3)");
```



```

        stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Cupboard', 'Green', 8000, 4)");
        stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Chair', 'Brown', 800, 4)");
        stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Shelf', 'Red', 800, 4)");
        stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Table', 'Green', 9000, 4)");
        stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('Chair', 'White', 500, 4)");
        stm.executeUpdate("INSERT INTO Shop (name, adress) VALUES
('Akson', 'Komsomolskoe shosse, 14')");
        stm.executeUpdate("INSERT INTO Shop (name, adress) VALUES
('IKEA', 'TRC MEGA, pos FediaKovo')");
        stm.executeUpdate("INSERT INTO Shop (name, adress) VALUES
('Mebelnui bazar', 'Moskovskii vokzal')");
        stm.executeUpdate("INSERT INTO Shop (name, adress) VALUES
('OBI', 'TRC Fantastika')");

Scanner sc = new Scanner(System.in);
System.out.println("1 - print all \n2 - add \n3 - delete \n4 -
print all in shop");
int key = sc.nextInt();
FurnitureFace tf = new FurnitureFace();

switch(key) {
case 1:
    for(Furniture f : tf.getAll()) {
        System.out.println("id: " + f.getId() + " name: "
+ f.getName() + " color: " + f.getColor() + " price: " + f.getPrice() + "
shopId: " + f.getShopId());
    }
    break;
case 2:
    Furniture fnew = new Furniture();
    sc.nextLine();
    System.out.println("Имя:");
    fnew.setName(sc.nextLine());
    System.out.println("Цвет:");
    fnew.setColor(sc.nextLine());
    System.out.println("Цена:");
    fnew.setPrice(sc.nextInt());
    System.out.println("id магазина:");
    fnew.setShopId(sc.nextInt());
    tf.add(fnew);
    break;
case 3:
    System.out.println("Введите id");
    int idDel = sc.nextInt();
    tf.delete(idDel);
    break;
case 4:
    System.out.println("Введите id магазина");
    int idSh = sc.nextInt();
    ShopFace sh = new ShopFace();
    sh.getShop(idSh);

    break;
default:
    return;
}
} catch (Exception e) {

```

```

        e.printStackTrace();
    }
}

```

Furniture.java:

```

package lab2;

public class Furniture {
    private int id;
    private String name;
    private String color;
    private int price;
    private int shopId;

    public Furniture() {

    }

    public Furniture(int _id, String _name, String _color, int _price, int
    _shopId) {
        id = _id;
        color = _color;
        name = _name;
        price = _price;
        shopId = _shopId;
    }

    public int getId() {
        return id;
    }

    public int getShopId() {
        return shopId;
    }

    public void setShopId(int _shopId) {
        shopId = _shopId;
    }

    public String getName() {
        return name;
    }

    public void setName(String _name) {
        name = _name;
    }

    public String getColor() {
        return color;
    }

    public void setColor(String _color) {
        color = _color;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int _price) {
        price = _price;
    }

}

```

FurnitureFace.java:

```
package lab2;

import java.sql.*;
import java.util.HashSet;
import java.util.Set;

public class FurnitureFace {
    static final String JDBC_DRIVER = "org.h2.Driver";
    static final String DB_URL = "jdbc:h2:file:~/base2";

    public Connection getConn() {
        try {
            Class.forName(JDBC_DRIVER).newInstance();
            Connection con = DriverManager.getConnection(DB_URL);
            return con;
        } catch (SQLException se) {
            se.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public Furniture extractFurnituresFromResult(ResultSet rs) throws
    SQLException {
        Furniture tmp = new Furniture(rs.getInt("id"),
            rs.getString("name"), rs.getString("color"),
            rs.getInt("price"), rs.getInt("shopId"));
        return tmp;
    }

    public Furniture getObj(int id) {
        try {
            Connection con = getConn();
            Statement stm = con.createStatement();
            ResultSet rs = stm.executeQuery("SELECT * FROM Furniture WHERE
id=" + id);

            if(rs.next()) {
                stm.close();
                con.close();
                return extractFurnituresFromResult(rs);
            }
            stm.close();
            con.close();

        } catch (SQLException se) {
            se.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public Set<Furniture> getAll() {
        try {
            Connection con = getConn();
            Statement stm = con.createStatement();
            ResultSet rs = stm.executeQuery("SELECT * from Furniture");
```

```

        Set<Furniture> furnitureSet = new HashSet<Furniture>();
        while(rs.next()) {
            Furniture furniture = extractFurnituresFromResult(rs);
            furnitureSet.add(furniture);
        }
        return furnitureSet;
    } catch(SQLException se) {
        se.printStackTrace();
    } catch(Exception e) {
        e.printStackTrace();
    }
    return null;
}

public boolean delete(int id) {
    try {
        Connection con = getConn();
        Statement stm = con.createStatement();

        if(stm.executeUpdate("DELETE FROM Furniture WHERE id=" + id)
== 1) {

            stm.close();
            con.close();
            return true;
        }
        stm.close();
        con.close();

    } catch(SQLException se) {
        se.printStackTrace();
    } catch(Exception e) {
        e.printStackTrace();
    }
    return false;
}

public boolean add(Furniture obj) {
    try {
        Connection con = getConn();
        Statement stm = con.createStatement();

        if(stm.executeUpdate("INSERT INTO Furniture (name, color, price,
shopId) VALUES ('" + obj.getName() + "', '" + obj.getColor() + "', " +
obj.getPrice() + ", " + obj.getShopId() + ")") == 1){
            con.close();
            return true;
        }
        con.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } catch(Exception e) {
        e.printStackTrace();
    }
    return false;
}

public boolean update(int id, Furniture obj) {
    try {
        Connection con = getConn();

```

```

        PreparedStatement ps = con.prepareStatement("UPDATE Furniture SET
name=?, color=?, price=?, shopId=? WHERE id=?");
        ps.setString(1, obj.getName());
        ps.setString(2, obj.getColor());
        ps.setInt(3, obj.getPrice());
        ps.setInt(4, obj.getShopId());
        ps.setInt(5, id);
        if(ps.executeUpdate() == 1) {
            con.close();
            return true;
        }
        con.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } catch(Exception e) {
        e.printStackTrace();
    }
    return false;
}
}

```

Shop.java:

```

package lab2;

public class Shop {
    private String name;
    private String address;
    private int id;

    public Shop() {
    }

    public Shop(String _name, String _address) {
        name = _name;
        address = _address;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }
}

```

ShopFace.java:

```

package lab2;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

```

```

public class ShopFace {
    static final String JDBC_DRIVER = "org.h2.Driver";
    static final String DB_URL = "jdbc:h2:file:~/base2";

    public Connection getConn() {
        try {
            Class.forName(JDBC_DRIVER).newInstance();
            Connection con = DriverManager.getConnection(DB_URL);
            return con;
        } catch(SQLException se) {
            se.printStackTrace();
        } catch(Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public boolean getShop(int id) {
        try {
            Connection con = getConn();
            PreparedStatement ps = con.prepareStatement("SELECT * from Shop WHERE
id=?");
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();
            while(rs.next()) {
                System.out.println("name: " + rs.getString("name") + " address:
" + rs.getString("adress"));
            }

            ps = con.prepareStatement("SELECT * from Furniture WHERE shopId=?");
            ps.setInt(1, id);
            rs = ps.executeQuery();
            while(rs.next()) {
                System.out.println("name: " + rs.getString("name") + " color:
" + rs.getString("color")+ " price: " + rs.getInt("price"));
            }

            con.close();
            return true;
        } catch(SQLException se) {
            se.printStackTrace();
        } catch(Exception e) {
            e.printStackTrace();
        }
        return false;
    }
}

```