## JavaBean

实体（N个属性）+方法

例：

import java.util.Date;

public class User{ //实体User

private int id; //属性 id

public int getId() { //方法 获取id值

return id;

}

public void setId(int id) { //方法 设置id值

this.id = id;

}

内容：

两大实体 用户-记录

用户（ID、UUID、BID）

记录（多个，需要时再创立）

要求：

## 创建工具类

例：

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class JDBCUtils {

// 加载驱动，并建立数据库连接

public static Connection getConnection() throws SQLException,

ClassNotFoundException {

Class.forName("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost:3306/test\_database ";

String username = "root";

String password = "";

Connection conn = DriverManager.getConnection(url, username, password);

return conn;

}

// 关闭数据库连接，释放资源

public static void release(Statement stmt, Connection conn) {

if (stmt != null) {

try {

stmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

stmt = null;

}

if (conn != null) {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

conn = null;

}

}

public static void release(ResultSet rs, Statement stmt, Connection conn) {

if (rs != null) {

try {

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

rs = null;

}

release(stmt, conn);

}

}

要求：

1. 建立数据库连接(startsql)
2. 关闭数据库连接，释放资源(stopsql)

最终实现：

通过配置文件，将他人服务器信息添加到工具类代码中。

## 创建DAO

例：

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

public class UserDao {

// 添加用户的操作

public boolean insert(User user) {

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

try {

// 获得数据的连接

conn = JDBCUtils.getConnection();

// 获得Statement对象

stmt = conn.createStatement();

// 发送SQL语句

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

String registerDate = sdf.format(user.getRegisterDate());

String sql = "INSERT INTO users(id,name,password,email,register\_date) "+

"VALUES("

+ user.getId()

+ ",'"

+ user.getUsername()

+ "','"

+ user.getPassword()

+ "','"

+ user.getEmail()

+ "','"

+ registerDate + "')";

int num = stmt.executeUpdate(sql);

if (num > 0) {

System.out.println("成功添加了"+num+"条数据！");

return true;

}

return false;

} catch (Exception e) {

e.printStackTrace();

} finally {

JDBCUtils.release(rs, stmt, conn);

}

return false;

}

// 查询所有的User对象

public ArrayList<User> findAll() {

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

ArrayList<User> list = new ArrayList<User>();

try {

// 获得数据的连接

conn = JDBCUtils.getConnection();

// 获得Statement对象

stmt = conn.createStatement();

// 发送SQL语句

String sql = "SELECT \* FROM users";

rs = stmt.executeQuery(sql);

// 处理结果集

while (rs.next()) {

User user = new User();

user.setId(rs.getInt("id"));

user.setUsername(rs.getString("name"));

user.setPassword(rs.getString("password"));

user.setEmail(rs.getString("email"));

user.setRegisterDate(rs.getDate("register\_date"));

list.add(user);

}

return list;

} catch (Exception e) {

e.printStackTrace();

} finally {

JDBCUtils.release(rs, stmt, conn);

}

return null;

}

// 根据id查找指定的user

public User find(int id) {

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

try {

// 获得数据的连接

conn = JDBCUtils.getConnection();

// 获得Statement对象

stmt = conn.createStatement();

// 发送SQL语句

String sql = "SELECT \* FROM users WHERE id=" + id;

rs = stmt.executeQuery(sql);

// 处理结果集

while (rs.next()) {

User user = new User();

user.setId(rs.getInt("id"));

user.setUsername(rs.getString("name"));

user.setPassword(rs.getString("password"));

user.setEmail(rs.getString("email"));

user.setRegisterDate(rs.getDate("register\_date"));

return user;

}

return null;

} catch (Exception e) {

e.printStackTrace();

} finally {

JDBCUtils.release(rs, stmt, conn);

}

return null;

}

// 修改用户

public boolean update(User user) {

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

try {

// 获得数据的连接

conn = JDBCUtils.getConnection();

// 获得Statement对象

stmt = conn.createStatement();

// 发送SQL语句

String sql = "UPDATE users set name='" + user.getUsername()

+ "',password='" + user.getPassword() + "',email='"

+ user.getEmail() + "' WHERE id=" + user.getId();

int num = stmt.executeUpdate(sql);

if (num > 0) {

System.out.println("成功修改了"+num+"条数据！");

return true;

}

return false;

} catch (Exception e) {

e.printStackTrace();

} finally {

JDBCUtils.release(rs, stmt, conn);

}

return false;

}

// 删除用户

public boolean delete(int id) {

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

try {

// 获得数据的连接

conn = JDBCUtils.getConnection();

// 获得Statement对象

stmt = conn.createStatement();

// 发送SQL语句

String sql = "DELETE FROM users WHERE id=" + id;

int num = stmt.executeUpdate(sql);

if (num > 0) {

System.out.println("成功删除了"+num+"条数据！");

return true;

}

return false;

} catch (Exception e) {

e.printStackTrace();

} finally {

JDBCUtils.release(rs, stmt, conn);

}

return false;

}

}

要求：

1. 通过SQL语句实现如下功能：

增加表(inserttabale)，修改表()，删除表(deletetabale)；

增加列，修改列，删除列；

增加信息，修改信息，删除信息；

查询信息。

备注：不要求实现详细代码，SQL语句可以暂时省略，作为待实现代码。列出类名。