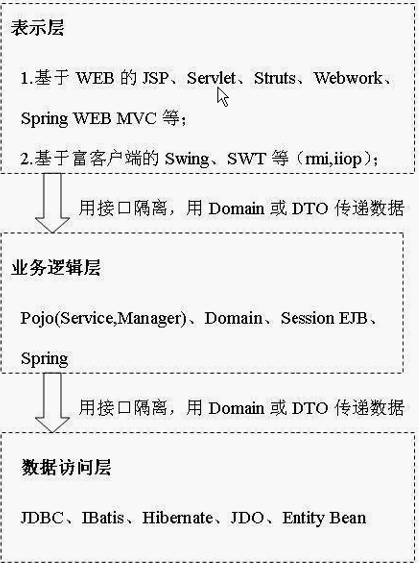
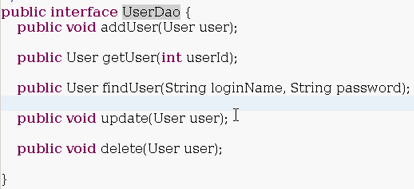
在实际项目中如何应用JDBC

1. **表示层🡪 业务逻辑层🡪数据访问层**

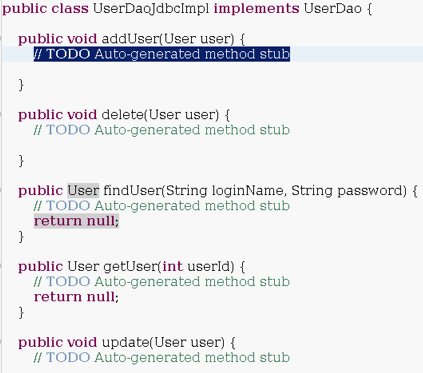


1. 向业务逻辑层提供接口：



1. 然后需要把**接口**进行实现。

利用**接口**的好处是：只需要向业务逻辑层提供接口，具体的实现业务逻辑层不关心，即使更改了，也对业务逻辑层无影响。



1. **数据访问层**：Data Access Layer。

**数据访问对象**：Data Access Object。

**DAO是一种应用程序编程接口（API）。**

1. **下面的最主要的掌握这种异常处理方法，很重要，对异常的处理很关键。这里采用的是为了防止对异常的抛出所带来的麻烦，这里采用通过创建运行时异常类的方法，对此异常进行封装，作为参数传入新的运行时异常类，然后在新的运行时异常中进行处理。**
2. 由于在添加用户的时候会抛出异常，但是如果实现类抛出异常，接口也必须抛出异常，从而需要修改接口，且不同的数据库抛出的异常不一样，就会带来很大的麻烦。如何解决这个问题呢？

方法：把**异常**变成**运行时异常**，而不用**编译时异常**。这样的话，就可以在接口处不用抛出异常了，使代码简洁。

可以把**出现的异常**封装成一个新的**运行时异常**，然后在创建的新的**运行时异常**里面对此异常进行处理，即可。



1. 创建一个新的运行时异常

public class DaoException extends RuntimeException{

public DaoException(String message, Throwable cause) {

super(message, cause);

} }

1. 创建一个接口：

public interface Jiekou {

public void addUser(User user);

public void deleteUser(User user);

public User findUser(int id);

public User findUser(String name,int id);

public void updateUser(User user);

}

1. 创建一个类实现这个接口：

public class Yong implements Jiekou {

public void addUser(User user) {

Connection conn = null;

PreparedStatement psm = null;

conn = DriverUtil.getConnection();

String sql = "insert into users(id,name,birthday) values(?,?,?)";

try {

psm = conn.prepareStatement(sql);

psm.setInt(1, user.getId());

psm.setString(2, user.getName());

psm.setDate(3, new java.sql.Date(user.getBirthday().getTime()));

psm.execute();

} catch (SQLException e) {

**throw new DaoException(e.getMessage(),e);//通过创建一个运行时异常，在出现异常时，可以抛出运行时异常，但是不需要 类抛出也不需要接口抛出此异常，所以在使用此方法时，也不需要抛出异常**

}finally {

if(psm != null) {

try {

psm.close();

} catch (SQLException e) {

}finally {

try {

conn.close();

} catch (SQLException e) {}}}} }

1. 还需要创建一个User类，成员name、id、birthday。
2. 主类：

public class Test1 {

public static void main(String[] args) {

Jiekou jiekou = new Yong();

Date date = new Date();

User user = new User(3333,"SBBB",date);

jiekou.addUser(user);

}

}

1. 删除用户：

public void deleteUser(User user) {

int id = user.getId();

Connection conn = null;

String sql = "delete from users where id = ?";

PreparedStatement psm = null;

try {

conn = DriverUtil.getConnection();

psm = conn.prepareStatement(sql);

psm.setInt(1, id);

psm.execute();

} catch (SQLException e) {

throw new DaoException(e.getMessage(), e);

}

DriverUtil.close(psm,conn);

}

**public** **static** **void** deleteUserMeth(**int** id) {

User user = **new** User(id,"SBBB",**new** Date());

Jiekou jiekou = **new** Yong();//利用接口调用类

jiekou.deleteUser(user);

}

1. **寻找用户findUser**：

public User findUser(int id) {

User user = null;

Connection conn = null;

PreparedStatement psm = null;

ResultSet rs = null;

try {

conn = DriverUtil.getConnection();

psm = conn.prepareStatement("select \* from users where id = ?");

psm.setInt(1, id);

rs = psm.executeQuery();

rs.next();

user = new User(rs.getInt(1), rs.getString(2), new Date(rs.getDate(3).getTime()));

} catch (SQLException e) {

throw new DaoException(e.getMessage(), e);

}

return user;

}

1. UpdateUser：更新用户：

public void updateUser(User user) {

Connection conn = DriverUtil.getConnection();

PreparedStatement psm = null;

String sql = "update users set name= ?,birthday = ? where id = ?";

try {

psm = conn.prepareStatement(sql);

psm.setString(1, user.getName());

psm.setDate(2, new java.sql.Date(user.getBirthday().getTime()+10000000));

psm.setInt(3,user.getId());

int i = psm.executeUpdate();

System.out.println(i);

} catch (SQLException e) {

throw new DaoException(e.getMessage(),e);

}finally {

DriverUtil.close(psm,conn);

} }

1. 综合测试：

public static void main(String[] args) {

Jiekou jiekou = new Yong();

User user = new User(8888888, "发发发发发发阿发", new Date());

jiekou.addUser(user);//添加用户

User user2 = jiekou.findUser(8888888);

System.out.println(user2.getId()+"\t"+user2.getName()+"\t"+user2.getBirthday());

user2.setName("泰勒斯威夫特");

jiekou.updateUser(user2);

jiekou.deleteUser(user2);

}