



第四章

DAO程序设计

课程目的



- **实体bean的简介:**
 - 数据持久化概念
 - 什么是实体bean
 - 数据库访问对象程序设计:
- **DAO设计模式的好处**
 - 设计模式的实现
 - 设计模式的使用

体验项目——<顾客注册功能>



使用下面已经设计好的DAO实现顾客注册功能，运营时弹出注册窗体，在文本框中输入相应的信息后，点击“拟定”按钮，会把输入的个人信总添加到数据库中。

注册窗体

用户ID: xg

姓名: 小刚

密码: 123

年龄: 23

E-mail: xg@126.com

确定 取消

当点击“拟定”按钮时，顾客信息将被提交到数据库，假如数据添加成功，窗口会弹出“注册成功”对话框，如图所示：



数据持久化概念

数据持久化就是将数据（如内存中的对象）保存到可永久保存的存储设备中（如磁盘）。持久化的主要应用是将内存中的数据存储在关系型的数据库中，当然也能够存储在磁盘文件中或者XML等文件中。



什么是实体bean



实体bean就是数据bean，在成熟的多层面对对象配置当中，能够划分出两种不同的组件配置：

- 应用逻辑组件：这些组件是执行一般任务的提供者。
- 持久的数据组件：这种对象能够提交给永久性存储器，能长时间存在，不会随系统的意外中断而消失，而且能够让多种客户同步访问。

数据库访问对象程序设计

DAO设计模式的好处

未使用DAO的早期开发可能会发觉下列问题：

全部的JDBC代码写在JSP页面之中，维护困难，假如数据库代码某些地方有问题，那么代码就会有大量的改动。

JSP中不应该使用任何SQL包，即：不应该在JSP中直接使用java.sql.*，原因是：JSP只关注于数据的显示，而不关心数据是从哪里来，或向哪里存储，在现实生活中也是一样，例如：企业的业务经理，他只关心他底下的员工的业绩怎样，而不关心员工是怎么做的。在JSP中也一样，它只懂得去调某些措施以到达它想要的成果，详细实现过程由DAO处理。





J2EE的组件层次：

客户端→表达层→业务层→数据层→数据库

DAO属于J2EE数据层的操作，即：用DAO封装了表在一种项目中所应该具有的全部操作。

举例：例如说一种图书馆，图书馆能够增长书籍。

要增长之前，应该把一本书给图书馆才能够。这本书就代表实体bean对象，当插入时要把对象中的数据插入到数据库中，实体bean对象也叫（VO、TO、POJO）（值对象、传播对象、最根本的JAVA对象）即：只包括属性和set()、get()措施的类，客户经过操作VO来操作DAO。

设计模式的实现

数据库脚本的创建代码如下（为了以便，我们直接在pub库里创建此表）：

```
CREATE TABLE person  
(  
    id varchar(32) not null primary key,  
    name varchar(20) not null,  
    password varchar(20) not null,  
    age varchar(20) not null,  
    email varchar(20) not null  
);
```





首先需要设计出整个模块之中对person表执行的全部操作，主要有：

- 增长
- 修改
- 删除
- 按ID查询
- 查询全部
- 模糊查询

按以上要求，要求出操作person表的原则，之后只需针对于不同的数据库实现这些原则即可。在JAVA中只有经过接口才能够定义出原则DAO要求的就是一种接口。

第一步：定义接口

接口是用来定义操作的，在这里应该定义：添、删、改、查（ID查、查询全部、模糊查询）等操作。

```
import java.util.* ;  
// 要求出了操作person表在此项目里的全部措施  
public interface PersonDAO  
{  
    // 增长操作  
    public void insert(Person person) throws Exception;  
    // 修改操作  
    public void update(Person person) throws Exception;  
    // 删除操作  
    public void delete(String id) throws Exception;  
    // 按ID查询操作  
    public Person queryById(String id) throws Exception;  
    // 查询全部  
    public List queryAll() throws Exception;  
    // 模糊查询  
    public List queryByLike(String cond) throws Exception;  
}
```



第二步：实体（数据）bean类Person.java

实体bean就是用来存取数据的类，每个实体bean都应该相应数据库中的唯一一张表，但在操作数据库时所收发的某些信息经过实体bean中的get()、set()措施来实现。

```
public class Person
{
    private String id ;
    private String name ;
    private String password ;
    private int age ;
    private String email ;
    // getter()、setter()措施
    .....
}
```

这个类用来实现前端和数据库之间的数据传递



第三步：数据库连接类DataBaseConnection.java

因为添、删、改、查都需要连接数据库，所以把连接数据库的代码单独提出来生成一种类，这个类主要实现两个功能，即：连接数据库、关闭数据库。主要代码如下：

```
public DataBaseConnection()
{
    try
    {
        Class.forName(DBDRIVER) ;
        this.conn = DriverManager.getConnection(DBURL,DBUSER,DBPASSWORD) ;
    }
    catch (Exception e)
    {}
}
```





// 取得数据库连接

```
public Connection getConnection()
```

```
{
```

```
    return this.conn ;
```

```
}
```

// 关闭数据库连接

```
public void close()
```

```
{
```

```
    try
```

```
    {
```

```
        this.conn.close() ;
```

```
    }
```

```
    catch (Exception e)
```

```
    {
```

```
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/775113341310011303>