**WEB程序设计课后思考**

**J2EE是什么**

J2EE是一套全然不同于传统应用开发的技术架构，包含许多组件，主要可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。

J2EE核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循J2EE架构的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容，企业内部或外部难以互通的窘境。

**J2EE的三大架构**

**1·Struts**

MVC即Model-View-Controller的缩写，是一种常用的设计模式。MVC 减弱了业务逻辑接口和数据接口之间的耦合，以及让视图层更富于变化。MVC的工作原理：Struts 是MVC的一种实现，它将Servlet和JSP 标记（属于 J2EE 规范）用作实现的一部分。Struts继承了MVC的各项特性，并根J2EE的特点，做了相应的变化与扩展。

优点：

a.开源软件，能更深入的了解其内部实现机制。

b.Taglib标记库，灵活动用，能大大提高开发效率。

c.页面导航使系统的脉络更加清晰。通过一个配置文件，即可把握整个系统各部分之间的联系，这对于后期的维护有着莫大的好处。尤其是当另一批开发者接手这个项目时，这种优势体现得更加明显。

d. 提供Exception处理机制

e. 数据库链接池管理

f. Struts 的Action必需是thread－safe方式，它仅仅允许一个实例去处理所有的请求。所以action用到的所有的资源都必需统一同步，这个就引起了线程安全的问题。

**2·Spring**

Spring 它是一个开源的项目，而且目前非常活跃；它基于IoC（Inversion of Control，反向控制）和AOP的构架多层j2ee系统的框架，但它不强迫你必须在每一层中必须使用Spring，因为它模块化的很好，允许你根据自己的需要选择使用它的某一个模块；它实现了很优雅的MVC，对不同的数据访问技术提供了统一的接口，采用IoC使得可以很容易的实现bean的装配，提供了简洁的AOP并据此实现Transcation Managment，等等

优点：

a. Spring能有效地组织你的中间层对象，不管你是否选择使用了EJB。如果你仅仅使用了Struts或其他为J2EE的API特制的framework，Spring致力于解决剩下的问题。

b. Spring能消除在许多工程中常见的对Singleton的过多使用。根据我的经验，这是一个很大的问题，它降低了系统的可测试性和面向对象的程度。

c. 通过一种在不同应用程序和项目间一致的方法来处理配置文件，Spring能消除各种各样自定义格式的属性文件的需要。曾经对某个类要寻找的是哪个魔法般的属性项或系统属性感到不解，为此不得不去读Javadoc甚至源编码？有了Spring，你仅仅需要看看类的JavaBean属性。Inversion of Control的使用（在下面讨论）帮助完成了这种简化。

d.通过把对接口编程而不是对类编程的代价几乎减少到没有，Spring能够促进养成好的编程习惯。

e. Spring被设计为让使用它创建的应用尽可能少的依赖于他的APIs。在Spring应用中的大多数业务对象没有依赖于Spring。

f. 使用Spring构建的应用程序易于单元测试。

g. Spring能使EJB的使用成为一个实现选择,而不是应用架构的必然选择。你能选择用POJOs或local EJBs来实现业务接口，却不会影响调用代码。

h. Spring帮助你解决许多问题而无需使用EJB。Spring能提供一种EJB的替换物，它们适用于许多web应用。例如，Spring能使用AOP提供声明性事务管理而不通过EJB容器，如果你仅仅需要与单个数据库打交道，甚至不需要一个JTA实现。

i. Spring为数据存取提供了一个一致的框架,不论是使用的是JDBC还O/R mapping产品（如Hibernate）。Spring确实使你能通过最简单可行的解决办法来解决你的问题。而这是有有很大价值的。

**3·Hibernate**

Hibernate使用了J2EE架构中的如下技术：JDBC、JTA、JNDI。其中JDBC是一个支持关系数据库操作的一个基础层；它与JNDI和JTA一起结合，使得Hibernate可以方便地集成到J2EE应用服务器中去。在这里，我们不会详细地去讨论Hibernate API接口中的所有方法，我们只简要讲一下每个主要接口的功能，如果你想了解得更多的话，你可以在Hibernate的源码包中的net.sf.hibernate子包中去查看这些接口的源代码。

优点：

a.Hibernate是JDBC的轻量级的对象封装，它是一个独立的对象持久层框架。Hibernate可以用在任何JDBC可以使用的场合，例如Java应用程序的数据库访问代码，DAO接口的实现类，甚至可以是BMP里面的访问数据库的代码。

b.Hibernate是一个和JDBC密切关联的框架，所以Hibernate的兼容性和JDBC驱动，和数据库都有一定的关系，但是和使用它的Java程序，和App Server没有任何关系，也不存在兼容性问题。

c.Hibernate不能用来直接和Entity Bean做对比，只有放在整个J2EE项目的框架中才能比较。并且即使是放在软件整体框架中来看，Hibernate也是做为JDBC的替代者出现的，而不是Entity Bean的替代者出现的。

d.Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以随心所欲的使用对象编程思维来操控数据库。

e.Hibernate可以应用在任何使用JDBC的场合。

f.Hibernate 使用Java 反射机制而不是字节码增强程序来实现透明性。

g.Hibernate 的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系。

**MVC与J2EE的三大架构之间对应的关系**

MVC与J2EE架构的对应关系是:View处于Web Tier或者说是[Client Tier](http://www.itisedu.com/phrase/200604241148385.html)，通常是JSP/[Servlet](http://www.itisedu.com/phrase/200603091005185.html)，即页面显示部分。Controller也处于Web Tier，通常用Servlet来实现，即页面显示的逻辑部分实现。Model处于[Middle Tier](http://www.itisedu.com/phrase/200604241147165.html)，通常用服务端的[JavaBean](http://www.itisedu.com/phrase/200604251741535.html)或者[EJB](http://www.itisedu.com/phrase/200604241156485.html)实现，即业务逻辑部分的实现。