姓 名：陆玉近

学 号：201502442

专业班级：信管11501班

**Java的三大架构与MVC设计模式的介绍**

**一、J2EE的介绍**

J2EE是一种利用Java 2平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。

J2EE技术的基础就是核心Java平台或Java 2平台的标准版，J2EE不仅巩固了标准版中的许多优点，例如"编写一次、随处运行"的特性、方便存取数据库的JDBC API、CORBA技术以及能够在Internet应用中保护数据的安全模式等等，同时还提供了对 EJB（Enterprise JavaBeans）、Java Servlets API、JSP（Java Server Pages）以及XML技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。

J2EE体系结构提供中间层集成框架用来满足无需太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台，J2EE降低了开发多层应用的费用和复杂性，同时提供对现有应用程序集成强有力支持，完全支持Enterprise JavaBeans，有良好的向导支持打包和部署应用，添加目录支持，增强了安全机制，提高了性能。

**二、MVC框架**

1、MVC 编程模式

mvc是一种设计模式 model、view、controller。而三大框架就像是工具：

struts 处理前台（servlet的处理）；

hibernate 处理后台（数据库交互）；

spring 通过bean来进行统一处理。

MVC 模式同时提供了对 HTML、CSS 和 JavaScript 的完全控制；通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

2、MVC 是一种使用 MVC（Model View Controller 模型-视图-控制器）设计创建 Web 应用程序的模式：

（1）Model（模型）表示应用程序核心（比如数据库记录列表）。

Model（模型）是应用程序中用于处理应用程序数据逻辑的部分；通常模型对象负责在数据库中存取数据。

（2）View（视图）显示数据（数据库记录）。

View（视图）是应用程序中处理数据显示的部分；通常视图是依据模型数据创建的。

（3）Controller（控制器）处理输入（写入数据库记录）。

Controller（控制器）是应用程序中处理用户交互的部分。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。

MVC开始是存在于桌面程序中的，M是指业务模型，V是指用户界面，C则是控制器，使用MVC的目的是将M和V的实现代码分离，从而使同一个程序可以使用不同的表现形式。比如一批统计数据可以分别用柱状图、饼图来表示。C存在的目的则是确保M和V的同步，一旦M改变，V应该同步更新。

模型－视图－控制器（MVC）是Xerox PARC在二十世纪八十年代为编程语言Smalltalk－80发明的一种软件设计模式，已被广泛使用。后来被推荐为Oracle旗下Sun公司Java EE平台的设计模式，并且受到越来越多的使用ColdFusion和PHP的开发者的欢迎。模型－视图－控制器模式是一个有用的工具箱。

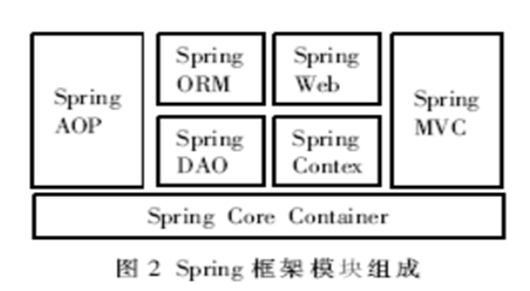
**三、java 三层架构SSH（Spring、Struts、Hibernate)**

一个spring+hibernate+struts2组合框架，使用spring的 IoC来管理应用的 所有bean，包括struts2的 action，充分发挥了spring轻量级框架的优势。

**1、业务逻辑层框架Spring**

spring 是为了解决企业应用开发的复杂性而创建的，使用基本的JavaBean来完成以前只可能由EJB完成的事情。Spring是一个开源框架，Spring是一个解决了许多J2EE开发中常见问题并能够替代EJB技术的强大的轻量级框架。这里所说的轻量级指的是 Spring框架本身，而不是指Spring只能用于轻量级的应用开发。Spring的轻盈体现在其框架本身的基础结构以及对其他应用工具的支持和装配能力。与EJB这种庞然大物相比，Spring可使程序研发人员把各个技术层次之间的风险降低。

Spring框架的核心是控制翻转IoC(Inversion of Control)/依赖注入DI(Dependence Injection)机制。IoC是指由容器中控制组件之间的关系（这里，容器是指为组件提供特定服务和技术支持的一个标准化的运行时的环境）而非传统实现中由程序代码直接操控，这种将控制权由程序代码到外部容器的转移，称为“翻转”。DI是对IoC更形象的解释，即由容器在运行期间动态地将依赖关系(如构造参数、构造对象或接口)注入到组件之中。 Spring采用设值注入(使用Setter方法实现依赖)和构造子注入(在构造方法中实现依赖)的机制，通过配置文件管理组建的协作对象，创建可以构造组件的IoC容器。这样，不需要编写工厂模式、单例模式或者其他构造的方法，就可以通过容器直接获取所需的业务组件。Spring框架的结构如图2所示。

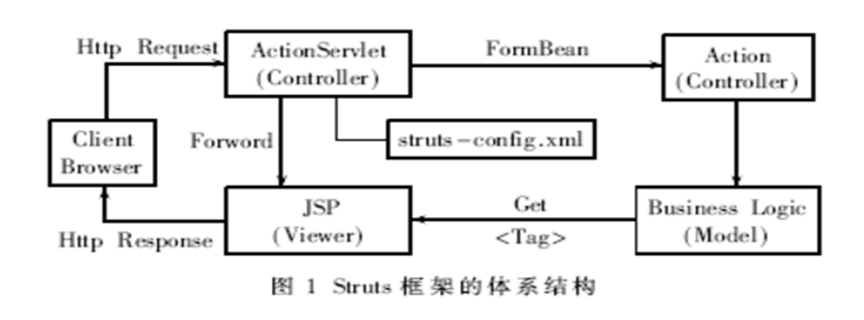


**2、表示层框架Struts**

struts 其实就是采用JavaServlet/JSP技术，实现了基于JavaEEWeb应用的MVC设计模式的应用框架。Struts有一组相互协作的类（组件）、Servlet以及jsp tag lib组成。Struts是一个在JSP Model2基础上实现的MVC框架，主要分为模型(Model)、视图(Viewer)和控制器(Controller)三部分，其主要的设计理念是通过控制器将表现逻辑和业务逻辑解耦，以提高系统的可维护性、可扩展性和可重用性。

Struts有其自己的控制器（Controller），同时整合了其他的一些技术去实现模型层（Model）和视图层（View）。在模型层，Struts可以很容易的与数据访问技术相结合，包括EJB，JDBC和Object Relation Bridge。在视图层，Struts能够与JSP, Velocity Templates,XSL等等这些表示层组件相结合。

Struts框架的体系结构如图1所示：



下面就图1所示的体系结构图分析Struts框架中的MVC组件：

（1）视图：视图部分主要由JSP页面组成，其中没有流程逻辑、业务逻辑和模型信息，只有标记。Struts自身包含了一组标记库(TagLib)，这也是Struts的精华之一，灵活运用它们可以简化JSP页面的代码，提高开发效率。

（2）控制器：Struts中的Controller主要是其自身提供的ActionServlet。ActionServlet接收所有来自客户端的请求并根据配置文件(struts-config.xml)中的定义将控制转移到适当的Action对象。

（3）模型：Struts没有定义具体Model层的实现，Model层通常是和业务逻辑紧密相关的，有持续化的要求。目前在商业领域和开源世界，都有一些优秀的工具可以为Model层的开发提供便利。

**3、数据持久层框架Hibernate**

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纵数据库。（说白了hibernate就是实现了Java的对象模型和数据库的关系模型之间的交互）。

O/R mapping技术是为了解决关系型数据库和面向对象的程序设计之间不匹配的矛盾而产生的。Hibernate是目前最为流行的O/R mapping框架，它在关系型数据库和Java对象之间做了一个自动映射，使得程序员可以以非常简单的方式实现对数据库的操作。

Hibernate通过对JDBC的封装，向程序员屏蔽了底层的数据库操作，使程序员专注于OO程序的开发，有助于提高开发效率。程序员访问数据库所需要做的就是为持久化对象编制xml映射文件。

Hibernate有自己的面向对象的查询语言HQL，HQL功能强大，支持目前大部分主流的数据库，如Oracle、DB2、MySQL、 Microsoft SQL Server等，是目前应用最广泛的O/R映射工具。Hibernate为快速开发应用程序提供了底层的支持。

Hibernate工作原理如图3所示：

