AOP 有三种植入切面的方法：其一是编译期织入，这要求使用特殊的Java编译器，AspectJ是其中的代表者；其二是类装载期织入，而这要求使用特殊的类装载器，AspectJ和AspectWerkz是其中的代表者；其三为动态代理织入，在运行期为目标类添加增强生成子类的方式，Spring AOP采用动态代理织入切面。

Spring AOP使用了两种代理机制，一种是基于JDK的动态代理，另一种是基于CGLib的动态代理，之所以需要两种代理机制，很大程度上是因为JDK本身只提供基于接口的代理，不支持类的代理。

SPRING是通过动态代理来实现AOP的,SPRING内部提供了2种实现机制  
1.如果是有接口声明的类进行AOP,spring调用的是java.lang.reflection.Proxy类来做处理

2.如果是没有接口声明的类呢?SPRING通过CGLIB包和内部类来实现

1 Action是不是线程安全的？如果不是 有什么方式可以保证Action的线程安全？如果是，说明原因  
不是  
声明局部变量，或者扩展RequestProcessor，让每次都创建一个Action，或者在spring中用scope=”prototype”来管理

3.struts中的几个关键对象的作用(说说几个关键对象的作用)  
Action：控制器类，ActionForm：表单对象，DynaValidatorForm：动态form，ActonMapping：配置文件中action节点的信息……

4.说说AOP和IOC的概念以及在spring中是如何应用的  
AOP:面向方面编程，ioc：依赖注入；声明式事务和编程式事务积极一些通用部分

6.load()和get()的区别  
hibernate对于load方法认为该数据在数据库中一定存在，可以放心的使用代理来延迟加载，load默认支持延迟加载，在用到对象中的其他属性数据时才查询数据库，但是万一数据库中不存在该记录，只能抛异常ObjectNotFoundEcception；所说的load方法抛异常是指在使用该对象的数据时，数据库中不存在该数据时抛异常，而不是在创建这个对象时。由于session中的缓存对于hibernate来说是个相当廉价的资源，所以在load时会先查一下session缓存看看该id对应的对象是否存在，不存在则创建代理（load时候之查询一级缓存，不存在则创建代理）。get()现在一级缓存找，没有就去二级缓存找，没有就去数据库找，没有就返回null ；而对于get方法，hibernate一定要获取到真实的数据，否则返回null。

7.谈谈hibernate的延迟加载和openSessionInView  
延迟加载要在session范围内，用到的时候再加载；opensessioninview是在web层写了一个  
filter来打开和关闭session，这样就表示在一次request过程中session一直开着，保证了延迟  
加载在session中的这个前提。

8.spring的事务有几种方式？谈谈spring事务的隔离级别和传播行为。  
spring事务分两种形式，声明式事务和编程式事务，spring提供了一个事务的接口  
PaltformTractionManager接口，针对不同的事务，spring进行了不同的实现,对hibernate事务  
的实现HIbernateTractionManager,对JDBC的JdbcTractionManager，  
DataSourceTractionManager以及JdoTractionManager。接口platformTractionManager提供了三  
个方法，获取事务，提交和回滚的方法。

一、Spring框架的优点都有什么？

Spring是分层的架构，你可以选择使用你需要的层而不用管不需要的部分

Spring是POJO编程，POJO编程使得可持续构建和可测试能力提高

依赖注入和IoC使得JDBC操作简单化

Spring是开源的免费的

Spring使得对象管理集中化合简单化

二、[描述一下Spring中实现DI（Dependency Injection）的几种方式](http://www.mianwww.com/html/2009/11/5613.html)

方式一：接口注入，在实际中得到了普遍应用，即使在IOC的概念尚未确立时，这样的方法也已经频繁出现在我们的代码中。  
方式二：Type2 IoC: Setter injection对象创建之后，将被依赖对象通过set方法设置进去  
方式三：Type3 IoC: Constructor injection对象创建时，被依赖对象以构造方法参数的方式注入  
Spring的方式

三、[简述你对IoC（Inversion of Control）的理解](http://www.mianwww.com/html/2009/11/5611.html)

一个类需要用到某个接口的方法，我们需要将类A和接口B的实现关联起来，最简单的方法是类A中创建一个对于接口B的实现C的实例，但这种方法显然两者的依赖（Dependency）太大了。而IoC的方法是只在类A中定义好用于关联接口B的实现的方法，将类A，接口B和接口B的实现C放入IoC的 容器（Container）中，通过一定的配置由容器（Container）来实现类A与接口B的实现C的关联。

四、[Spring对多种ORM框架提供了很好的支持，简单描述在Spring中使用Hibernate的方法。](http://www.mianwww.com/html/2009/11/5609.html)

在context中定义DataSource，创建SessionFactoy，设置参数；DAO类继承HibernateDaoSupport，实现具体接口，从中获得HibernateTemplate进行具体操作。在使用中如果遇到OpenSessionInView的问题，可以添加OpenSessionInViewFilter或OpenSessionInViewInterceptor

五、[请介绍一下Spring的事务管理](http://www.mianwww.com/html/2009/09/4215.html)

spring提供了几个关于事务处理的类：

TransactionDefinition //事务属性定义

TranscationStatus //代表了当前的事务，可以提交，回滚。

PlatformTransactionManager这个是spring提供的用于管理事务的基础接口，其下有一个实现的抽象类AbstractPlatformTransactionManager,我们使用的事务管理类例如DataSourceTransactionManager等都是这个类的子类。

十二、[解释一下Dependency injection(DI,依赖注入)和IOC(Inversion of control,控制反转)?](http://www.mianwww.com/html/2009/09/4007.html)

参考答案：依赖注入DI是一个程序设计模式和架构模型， 一些时候也称作控制反转，尽管在技术上来讲，依赖注入是一个IOC的特殊实现，依赖注入是指一个对象应用另外一个对象来提供一个特殊的能力，例如：把一个数据库连接已参数的形式传到一个对象的结构方法里面而不是在那个对象内部自行创建一个连接。控制反转和依赖注入的基本思想就是把类的依赖从类内部转化到外部以减少依赖  
应用控制反转，对象在被创建的时候，由一个调控系统内所有对象的外界实体，将其所依赖的对象的引用，传递给它。也可以说，依赖被注入到对象中。所以，控制反转是，关于一个对象如何获取他所依赖的对象的引用，这个责任的反转。

三十、[Spring和Struts的区别？](http://www.mianwww.com/html/2009/02/3152.html)

strusts：是一种基于MVC模式的一个web层的处理。  
Spring:提供了通用的服务，ioc/di aop,关心的不仅仅web层，应当j2ee整体的一个服务，可以很容易融合不同的技术struts hibernate ibatis ejb remote springJDBC springMVC

说说AOP和IOC的概念以及在spring中是如何应用的

spring的核心就是IOC,通过指定对象的创建办法,描述对象与服务之间的关系,而不生成对象

6.load()和get()的区别

load()方法认为该数据一定存在,可以放心的使用代理来延时加载 ,如果使用过程中发现了问题,就抛出异常;  
get()方法一定要获取到真实的数据,否则返回null

三十二、[Struts，Spring，Hibernate面试题总结](http://www.mianwww.com/html/2009/02/2310.html)

Hibernate工作原理及为什么要用？

原理：

1. 读取并解析配置文件  
2. 读取并解析映射信息，创建SessionFactory  
3. 打开Sesssion  
4. 创建事务Transation  
5. 持久化操作  
6. 提交事务  
7. 关闭Session  
8. 关闭SesstionFactory

为什么要用：

\* 对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。  
\* Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作  
\* hibernate使用Java反射机制，而不是字节码增强程序来实现透明性。  
\* 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。

Hibernate是如何延迟加载?

当Hibernate在查询数据的时候，数据并没有存在与内存中，当程序真正对数据的操作时，对象才存在与内存中，就实现了延迟加载，他节省了服务器的内存开销，从而提高了服务器的性能。  
Hibernate中怎样实现类之间的关系?(如：一对多、多对多的关系)  
类与类之间的关系主要体现在表与表之间的关系进行操作，它们都市对对象进行操作，我们程序中把所有的表与类都映射在一起，它们通过配置文件中的many-to-one、one-to-many、many-to-many、

如何优化Hibernate？

使用双向一对多关联，不使用单向一对多  
\* 灵活使用单向一对多关联  
\* 不用一对一，用多对一取代  
\* 配置对象缓存，不使用集合缓存  
\* 一对多集合使用Bag,多对多集合使用Set  
\* 继承类使用显式多态  
\* 表字段要少，表关联不要怕多，有二级缓存撑腰

Struts工作机制？为什么要使用Struts？

工作机制：  
Struts的工作流程:  
在web应用启动时就会加载初始化ActionServlet,ActionServlet从struts-config.xml文件中读取配置信息,把它们存放到各种配置对象当ActionServlet接收到一个客户请求时,将执行如下流程.  
(1)检索和用户请求匹配的ActionMapping实例,如果不存在,就返回请求路径无效信息;  
(2)如果ActionForm实例不存在,就创建一个ActionForm对象,把客户提交的表单数据保存到ActionForm对象中;  
(3)根据配置信息决定是否需要表单验证.如果需要验证,就调用ActionForm的validate()方法;  
(4)如果ActionForm的validate()方法返回null或返回一个不包含ActionMessage的ActuibErrors对象, 就表示表单验证成功;  
(5)ActionServlet根据ActionMapping所包含的映射信息决定将请求转发给哪个Action,如果相应的Action实例不存在,就先创建这个实例,然后调用Action的execute()方法;  
(6)Action的execute()方法返回一个ActionForward对象,ActionServlet在把客户请求转发给ActionForward对象指向的JSP组件;  
(7)ActionForward对象指向JSP组件生成动态网页,返回给客户;

为什么要用：  
JSP、Servlet、JavaBean技术的出现给我们构建强大的企业应用系统提供了可能。但用这些技术构建的系统非常的繁乱，所以在此之上，我们需要一个规则、一个把这些技术组织起来的规则，这就是框架，Struts便应运而生。

基于Struts开发的应用由3类组件构成：控制器组件、模型组件、视图组件  
Struts的validate框架是如何验证的？  
在struts配置文件中配置具体的错误提示，再在FormBean中的validate()方法具体调用。

说下Struts的设计模式?

MVC模式:  web应用程序启动时就会加载并初始化ActionServler。用户提交表单时，一个配置好的ActionForm对象被创建，并被填入表单相应的数据，ActionServler根据Struts-config.xml 文件配置好的设置决定是否需要表单验证，如果需要就调用ActionForm的Validate（）验证后选择将请求发送到哪个Action，如果 Action不存在，ActionServlet会先创建这个对象，然后调用Action的execute（）方法。Execute（）从 ActionForm对象中获取数据，完成业务逻辑，返回一个ActionForward对象，ActionServlet再把客户请求转发给 ActionForward对象指定的jsp组件，ActionForward对象指定的jsp生成动态的网页，返回给客户。

spring工作机制及为什么要用?

1. spring mvc请所有的请求都提交给DispatcherServlet,它会委托应用系统的其他模块负责负责对请求进行真正的处理工作。  
   2.DispatcherServlet查询一个或多个HandlerMapping,找到处理请求的Controller.  
   3.DispatcherServlet请请求提交到目标Controller  
   4.Controller进行业务逻辑处理后，会返回一个ModelAndView  
   5.Dispathcher查询一个或多个ViewResolver视图解析器,找到ModelAndView对象指定的视图对象  
   6.视图对象负责渲染返回给客户端。

为什么用：  
AOP 让开发人员可以创建非行为性的关注点，称为横切关注点，并将它们插入到应用程序代码中。使用 AOP 后，公共服务   （比 如日志、持久性、事务等）就可以分解成方面并应用到域对象上，同时不会增加域对象的对象模型的复杂性。

IOC 允许创建一个可以构造对象的应用环境，然后向这些对象传递它们的协作对象。正如单词 倒置 所表明的，IOC 就像反      过来的 JNDI。没有使用一堆抽象工厂、服务定位器、单元素（singleton）和直接构造（straight construction），每一个对象都是用     其协作对象构造的。因此是由容器管理协作对象（collaborator）。

Spring即使一个AOP框架，也是一IOC容器。 Spring 最好的地方是它有助于您替换对象。有了 Spring，只要用 JavaBean 属性和配置文件加入依赖性（协作对象）。然后可以很容易地在需要时替换具有类似接口的协作对象。

3、 简单描述Spring framework与Struts的不同之处，整合Spring与Struts有哪些方法，哪种最好，为什么？

Spring是完整的一站式框架，而Struts仅是MVC框架，且着重于MVC中的C。Spring有三种方式整合Struts：使用 Spring 的 ActionSupport 类整合 Struts；使用 Spring 的 DelegatingRequestProcessor 覆盖 Struts 的 RequestProcessor；将 Struts Action 管理委托给 Spring 框架，动作委托最好。（详见使用Spring 更好地处理Struts 动作）

5、 Spring对多种ORM框架提供了很好的支持，简单描述在Spring中使用Hibernate的方法，并结合事务管理。

在context中定义DataSource，创建SessionFactoy，设置参数；DAO类继承HibernateDaoSupport，实现具体接口，从中获得HibernateTemplate进行具体操作。

在使用中如果遇到OpenSessionInView的问题，可以添加OpenSessionInViewFilter或OpenSessionInViewInterceptor。（详见Spring framework 2.0 Reference的12.2节Hibernate）

声明式事务需声明事务管理器，在context中设置指定属性，用确定和。

三十四、[请你谈谈SSH整合](http://www.mianwww.com/html/2009/01/1707.html)

SSH：  
Struts（表示层）+Spring（业务层）+Hibernate（持久层）  
Struts：  
Struts是一个表示层框架，主要作用是界面展示，接收请求，分发请求。  
在MVC框架中，Struts属于VC层次，负责界面表现，负责MVC关系的分发。（View：沿用JSP，HTTP，Form，Tag，Resourse ；Controller：ActionServlet，struts-config.xml，Action）  
Hibernate：  
Hibernate是一个持久层框架，它只负责与关系数据库的操作。  
Spring：  
Spring是一个业务层框架，是一个整合的框架，能够很好地黏合表示层与持久层。

三十八、[请介绍一下Spring框架中Bean的作用域](http://www.mianwww.com/html/2009/01/210.html)

1、singleton作用域

当一个bean的作用域设置为singleton，那么Spring IOC容器中只会存在一个共享的bean实例，并且所有对bean的请求，只要id与该bean定义相匹配，则只会返回bean的同一实例。换言之，当把一个bean定义设置为singleton作用域时，Spring IOC容器只会创建该bean定义的唯一实例。这个单一实例会被存储到单例缓存（singleton cache）中，并且所有针对该bean的后续请求和引用都将返回被缓存的对象实例，这里要注意的是singleton作用域和GOF设计模式中的单例是完全不同的，单例设计模式表示一个ClassLoader中只有一个class存在，而这里的singleton则表示一个容器对应一个bean，也就是说当一个bean被标识为singleton时候，spring的IOC容器中只会存在一个该bean。

2、prototype

prototype作用域部署的bean，每一次请求（将其注入到另一个bean中，或者以程序的方式调用容器的getBean()方法）都会产生一个新的bean实例，相当于一个new的操作，对于prototype作用域的bean，有一点非常重要，那就是Spring不能对一个 prototype bean的整个生命周期负责，容器在初始化、配置、装饰或者是装配完一个prototype实例后，将它交给客户端，随后就对该prototype实例不闻不问了。不管何种作用域，容器都会调用所有对象的初始化生命周期回调方法，而对prototype而言，任何配置好的析构生命周期回调方法都将不会被调用。清除prototype作用域的对象并释放任何prototype bean所持有的昂贵资源，都是客户端代码的职责。（让Spring容器释放被singleton作用域bean占用资源的一种可行方式是，通过使用 bean的后置处理器，该处理器持有要被清除的bean的引用。）

四十、[Spring框架有哪几部分组成？](http://www.mianwww.com/html/2009/01/202.html)

Spring框架有七个模块组成组成，这7个模块(或组件)均可以单独存在，也可以与其它一个或多个模块联合使用，主要功能表现如下：

☞ Spring 核心容器（Core）：提供Spring框架的基本功能。核心容器的主要组件是BeanFactory，她是工厂模式的实现。BeanFactory使用控制反转（Ioc）模式将应用程序的配置和依赖性规范与实际的应用代码程序分开。

☞ Spring AOP：通过配置管理特性，Spring AOP模块直接面向方面的编程功能集成到了Spring框架中，所以可以很容易的使Spring框架管理的任何对象支持 AOP。Spring AOP模块为基于Spring的应用程序中的对象提供了事务管理服务。通过使用Spring AOP，不用依赖于EJB组件，就可以将声明性事务管理集成到应用程序中。

☞ Spring ORM：Spring框架集成了若干ORM框架,从而提供了ORM的对象关系工具,其中包括 JDO、Hibernate、iBatis和TopLink。所有这些都遵从Spring的通用事务和DAO异常层结构。

☞ Spring DAO：JDBC DAO抽象层提供了有意义的异常层次的结构，可用该结构来管理异常处理和不同数据供应商抛出的异常错误信息。异常层次结构简化了错误处理，并且大大的降低 了需要编写的异常代码数量（例如，打开和关系连接）。Spring DAO的面向JDBC的异常遵从通用的DAO异常层结构。

☞ Spring WEB：Web上下文模块建立在上下文模块（Context）的基础之上，为基于Web服务的应用程序提供了上下文的服务。所以Spring框架支持 Jakarta Struts的集成。Web模块还简化了处理多部分请求及将请求参数绑定到域对象的工作。

☞ Spring上下文（Context）：Spring上下文是一个配置文件，向Spring框架提供上下文信息。Spring上下文包括企业服务，例如 JNDI、EJB、电子邮件、国际化校验和调度功能。

☞ Spring MVC：Spring的MVC框架是一个全功能的构建Web应用程序的MVC实现。通过策略接口，MVC框架变成为高度可配置的，MVC容纳的大量视图技术，包括JSP、Velocity、Tiles、iText和Pol

四十一、[使用Spring有什么好处](http://www.mianwww.com/html/2009/01/200.html)

Spring能有效地组织你的中间层对象,。如果你仅仅使用了Struts或其他的包含了J2EE特有APIs的framework，你会发现Spring关注了遗留下的问题，。

◆Spring能消除在许多工程上对Singleton的过多使用。根据我的经验，这是一个主要的问题，它减少了系统的可测试性和面向对象特性。  
◆Spring能消除使用各种各样格式的属性定制文件的需要,在整个应用和工程中，可通过一种 一致的方法来进行配置。曾经感到迷惑，一个特定类要查找迷幻般的属性关键字或系统属性,为此不得不读Javadoc乃至源编码吗？有了Spring，你可 很简单地看到类的JavaBean属性。倒置控制的使用(在下面讨论)帮助完成这种简化。  
◆Spring能通过接口而不是类促进好的编程习惯，减少编程代价到几乎为零。  
◆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。  
◆使用Spring构建的应用程序易于单元测试。  
◆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。  
◆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，它们适于许多web应用。例如,Spring能使用AOP提供声明性事务而不通过使用EJB容器，如果你仅仅需要与单个的数据库打交道，甚至不需要JTA实现。  
■Spring为数据存取提供了一致的框架,不论是使用JDBC或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解决你的问题。这些特性是有很大价值的。  
总结起来，Spring有如下优点：  
◆低侵入式设计，代码污染极低  
◆ 独立于各种应用服务器，可以真正实现Write Once,Run Anywhere的承诺  
◆Spring的DI机制降低了业务对象替换的复杂性  
◆Spring并不完全依赖于Spring，开发者可自由选用Spring框架的部分或全部

四十二、[Spring面试：什么是Spring, 它有什么特点](http://www.mianwww.com/html/2009/01/198.html)

Spring是一个轻量级的控制反转(IoC)和面向切面(AOP)的容器框架。

◆轻量——从大小与开销两方面而言Spring都是轻量的。完整的Spring框架可以在一个大小只有1MB多的JAR文件里发布。并 且Spring所需的处理开销也是微不足道的。此外，Spring是非侵入式的：典型地，Spring应用中的对象不依赖于Spring的特定类。

◆控制反转——Spring通过一种称作控制反转（IoC）的技术促进了松耦 合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。你可以认为IoC与JNDI相反——不 是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。  
◆面向切面——Spring提供了面向切面编程的丰富支持，允许通过分离应用的 业务逻辑与系统级服务（例如审计（auditing）和事务（）管理）进行内聚性的开发。应用对象只实现它们应该做的——完成业务逻辑——仅此而已。它们 并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。  
◆容器——Spring包含并管理应用对象的配置和生命周期，在这个意义上它是 一种容器，你可以配置你的每个bean如何被创建——基于一个可配置原型（prototype），你的bean可以创建一个单独的实例或者每次需要时都生 成一个新的实例——以及它们是如何相互关联的。然而，Spring不应该被混同于传统的重量级的EJB容器，它们经常是庞大与笨重的，难以使用。  
◆框架——Spring可以将简单的组件配置、组合成为复杂的应用。在Spring中，应用对象被声明式地组合，典型地是在一个XML文件里。Spring也提供了很多基础功能（事务管理、持久化框架集成等等），将应用逻辑的开发留给了你。

四十三、

[主题：spring面试题 对DI , AOP概念的理解](http://winsli.javaeye.com/blog/720285)

spring 的优点？  
1.降低了组件之间的耦合性 ，实现了软件各层之间的解耦  
2.可以使用容易提供的众多服务，如事务管理，消息服务等  
3.容器提供单例模式支持  
4.容器提供了AOP技术，利用它很容易实现如权限拦截，运行期监控等功能  
5.容器提供了众多的辅助类，能加快应用的开发  
6.spring对于主流的应用框架提供了集成支持，如hibernate，JPA，Struts等  
7.spring属于低侵入式设计，代码的污染极低  
8.独立于各种应用服务器  
9.spring的DI机制降低了业务对象替换的复杂性  
10.Spring的高度开放性，并不强制应用完全依赖于Spring，开发者可以自由选择spring的部分或全部

什么是DI机制？

依赖注入（Dependecy Injection）和控制反转（Inversion of Control）是同一个概念，具体的讲：当某个角色  
需要另外一个角色协助的时候，在传统的程序设计过程中，通常由调用者来创建被调用者的实例。但在spring中  
创建被调用者的工作不再由调用者来完成，因此称为控制反转。创建被调用者的工作由spring来完成，然后注入调用者  
因此也称为依赖注入。  
spring以动态灵活的方式来管理对象 ， 注入的两种方式，设置注入和构造注入。  
设置注入的优点：直观，自然  
构造注入的优点：可以在构造器中决定依赖关系的顺序。

什么是AOP？

面向切面编程（AOP）完善spring的依赖注入（DI），面向切面编程在spring中主要表现为两个方面  
1.面向切面编程提供声明式事务管理  
2.spring支持用户自定义的切面

面向切面编程（aop）是对面向对象编程（oop）的补充，  
面向对象编程将程序分解成各个层次的对象，面向切面编程将程序运行过程分解成各个切面。  
AOP从程序运行角度考虑程序的结构，提取业务处理过程的切面，oop是静态的抽象，aop是动态的抽象，  
是对应用执行过程中的步骤进行抽象，，从而获得步骤之间的逻辑划分。

aop框架具有的两个特征：  
1.各个步骤之间的良好隔离性  
2.源代码无关性

1. 为什么用aop 就是为了方便, 就是为了更清晰的逻辑，可以让你的业务逻辑去关注自己本身的业务，而不去想一些其他的事情，这些其他的事情包括：安全，事物，日志等。

.Spring框架的优点都有什么？

Spring是分层的架构，你可以选择使用你需要的层而不用管不需要的部分

Spring是POJO编程，POJO编程使得可持续构建和可测试能力提高

依赖注入和IoC使得JDBC操作简单化

Spring是开源的免费的

Spring使得对象管理集中化合简单化

描述一下Spring中实现DI（Dependency Injection）的几种方式?

方式一：接口注入，在实际中得到了普遍应用，即使在IOC的概念尚未确立时，这样的方法也已经频繁出现在我们的代码中。  
方式二：Type2 IoC: Setter injection对象创建之后，将被依赖对象通过set方法设置进去  
方式三：Type3 IoC: Constructor injection对象创建时，被依赖对象以构造方法参数的方式注入  
Spring的方式

简述你对IoC（Inversion of Control）的理解?

一个类需要用到某个接口的方法，我们需要将类A和接口B的实现关联起来，最简单的方法是类A中创建一个对于接口B的实现C的实例，但这种方法显然两者的依赖（Dependency）太大了。而IoC的方法是只在类A中定义好用于关联接口B的实现的方法，将类A，接口B和接口B的实现C放入IoC的 容器（Container）中，通过一定的配置由容器（Container）来实现类A与接口B的实现C的关联。

谈谈你对Struts的理解。

. struts是一个按MVC模式设计的Web层框架，其实它就是一个大大的servlet，这个Servlet名为ActionServlet，或是ActionServlet的子类。我们可以在web.xml文件中将符合某种特征的所有请求交给这个Servlet处理，这个Servlet再参照一个配置文件（通常为/WEB-INF/struts-config.xml）将各个请求分别分配给不同的action去处理。ActionServlet把请求交给action去处理之前，会将请求参数封装成一个formbean对象, ActionServlet把formbean对象传递给action的execute方法之前，可能会调用formbean的validate方法进行校验，只有校验通过后才将这个formbean对象传递给action的execute方法，否则，它将返回一个错误页面, action执行完后要返回显示的结果视图，这个结果视图是用一个ActionForward对象来表示的，actionforward对象通过struts-config.xml配置文件中的配置关联到某个jsp页面，因为程序中使用的是在struts-config.xml配置文件为jsp页面设置的逻辑名，这样可以实现action程序代码与返回的jsp页面名称的解耦。

谈谈你对Hibernate的理解。

答:  
1. 面向对象设计的软件内部运行过程可以理解成就是在不断创建各种新对象、建立对象之间的关系，调用对象的方法来改变各个对象的状态和对象消亡的过程，不管程序运行的过程和操作怎么样，本质上都是要得到一个结果，程序上一个时刻和下一个时刻的运行结果的差异就表现在内存中的对象状态发生了变化。  
2.为了在关机和内存空间不够的状况下，保持程序的运行状态，需要将内存中的对象状态保存到持久化设备和从持久化设备中恢复出对象的状态，通常都是保存到关系数据库来保存大量对象信息。从Java程序的运行功能上来讲，保存对象状态的功能相比系统运行的其他功能来说，应该是一个很不起眼的附属功能，java采用jdbc来实现这个功能，这个不起眼的功能却要编写大量的代码，而做的事情仅仅是保存对象和恢复对象，并且那些大量的jdbc代码并没有什么技术含量，基本上是采用一套例行公事的标准代码模板来编写，是一种苦活和重复性的工作。  
3.通过数据库保存java程序运行时产生的对象和恢复对象，其实就是实现了java对象与关系数据库记录的映射关系，称为ORM（即Object Relation Mapping），人们可以通过封装JDBC代码来实现了这种功能，封装出来的产品称之为ORM框架，Hibernate就是其中的一种流行ORM框架。使用Hibernate框架，不用写JDBC代码，仅仅是调用一个save方法，就可以将对象保存到关系数据库中，仅仅是调用一个get方法，就可以从数据库中加载出一个对象。  
4.使用Hibernate的基本流程是：配置Configuration对象、产生SessionFactory、创建session对象，启动事务，完成CRUD操作，提交事务，关闭session。  
5.使用Hibernate时，先要配置hibernate.cfg.xml文件，其中配置数据库连接信息和方言等，还要为每个实体配置相应的hbm.xml文件，hibernate.cfg.xml文件中需要登记每个hbm.xml文件。  
6.在应用Hibernate时，重点要了解Session的缓存原理，级联，延迟加载和hql查询。

你对Spring的理解。

1.Spring实现了工厂模式的工厂类（在这里有必要解释清楚什么是工厂模式），这个类名为BeanFactory（实际上是一个接口），在程序中通常BeanFactory的子类ApplicationContext。Spring相当于一个大的工厂类，在其配置文件中通过元素配置用于创建实例对象的类名和实例对象的属性。  
2. Spring提供了对IOC良好支持，IOC是一种编程思想，是一种架构艺术，利用这种思想可以很好地实现模块之间的解耦。IOC也称为DI（Depency Injection），什么叫依赖注入呢？

3. Spring提供了对AOP技术的良好封装， AOP称为面向切面编程，就是系统中有很多各不相干的类的方法，在这些众多方法中要加入某种系统功能的代码，例如，加入日志，加入权限判断，加入异常处理，这种应用称为AOP。实现AOP功能采用的是代理技术，客户端程序不再调用目标，而调用代理类，代理类与目标类对外具有相同的方法声明，有两种方式可以实现相同的方法声明，一是实现相同的接口，二是作为目标的子类在，JDK中采用Proxy类产生动态代理的方式为某个接口生成实现类，如果要为某个类生成子类，则可以用CGLI B。在生成的代理类的方法中加入系统功能和调用目标类的相应方法，系统功能的代理以Advice对象进行提供，显然要创建出代理对象，至少需要目标类和Advice类。spring提供了这种支持，只需要在spring配置文件中配置这两个元素即可实现代理和aop功能

Struts优缺点?

优点：  
1. 实现MVC模式，结构清晰,使开发者只关注业务逻辑的实现.  
2．有丰富的tag可以用 ,Struts的标记库(Taglib)，如能灵活动用，则能大大提高开发效率  
3. 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。  
4. 提供Exception处理机制 .  
5. 数据库链接池管理  
6. 支持I18N

缺点  
一、 转到展示层时，需要配置forward，如果有十个展示层的jsp，需要配置十次struts，而且还不包括有时候目录、文件变更，需要重新修改forward，注意，每次修改配置之后，要求重新部署整个项目，而tomcate这样的服务器，还必须重新启动服务器  
二、 二、 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。  
三、 测试不方便. Struts的每个Action都同Web层耦合在一起，这样它的测试依赖于Web容器，单元测试也很难实现。不过有一个Junit的扩展工具Struts TestCase可以实现它的单元测试。  
四、 类型的转换. Struts的FormBean把所有的数据都作为String类型，它可以使用工具Commons-Beanutils进行类型转化。但它的转化都是在Class级别，而且转化的类型是不可配置的。类型转化时的错误信息返回给用户也是非常困难的。  
五、 对Servlet的依赖性过强. Struts处理Action时必需要依赖ServletRequest 和ServletResponse，所有它摆脱不了Servlet容器。  
六、 前端表达式语言方面.Struts集成了JSTL，所以它主要使用JSTL的表达式语言来获取数据。可是JSTL的表达式语言在Collection和索引属性方面处理显得很弱。  
七、 对Action执行的控制困难. Struts创建一个Action，如果想控制它的执行顺序将会非常困难。甚至你要重新去写Servlet来实现你的这个功能需求。  
八、 对Action 执行前和后的处理. Struts处理Action的时候是基于class的hierarchies，很难在action处理前和后进行操作。  
九、 对事件支持不够. 在struts中，实际是一个表单Form对应一个Action类(或DispatchAction)，换一句话说：在Struts中实际是一个表单只能 对应一个事件，struts这种事件方式称为application event，application event和component event相比是一种粗粒度的事件

STRUTS的应用(如STRUTS架构)?

Struts是采用Java Servlet/JavaServer Pages技术，开发Web应用程序的开放源码的framework。 采用Struts能开发出基于MVC(Model-View-Controller)设计模式的应用构架。 Struts有如下的主要功能： 一.包含一个controller servlet，能将用户的请求发送到相应的Action对象。 二.JSP自由tag库，并且在controller servlet中提供关联支持，帮助开发员创建交互式表单应用。 三.提供了一系列实用对象：XML处理、通过Java reflection APIs自动处理JavaBeans属性、国际化的提示和消息。

说说struts1与struts2的区别。

1.都是MVC的WEB框架,  
2 struts1的老牌框架，应用很广泛，有很好的群众基础，使用它开发风险很小，成本更低！struts2虽然基于这个框架，但是应用群众并多，相对不成熟，未知的风险和变化很多，开发人员相对不好招，使用它开发项目的风险系数更大，用人成本更高！  
3.struts2毕竟是站在前辈的基础设计出来，它会改善和完善struts1中的一些缺陷，struts1中一些悬而未决问题在struts2得到了解决。  
4.struts1的前端控制器是一个Servlet，名称为ActionServlet，struts2的前端控制器是一个filter，在struts2.0中叫FilterDispatcher，在struts2.1中叫StrutsPrepareAndExecuteFilter。  
5.struts1的action需要继承Action类，struts2的action可以不继承任何类；struts1对同一个路径的所有请求共享一个Action实例，struts2对同一个路径的每个请求分别使用一个独立Action实例对象，所有对于struts2的Action不用考虑线程安全问题。  
6.在struts1中使用formbean封装请求参数，在struts2中直接使用action的属性来封装请求参数。  
7.struts1中的多个业务方法放在一个Action中时（即继承DispatchAction时），要么都校验，要么都不校验；对于struts2，可以指定只对某个方法进行校验，当一个Action继承了ActionSupport且在这个类中只编写了validateXxx()方法，那么则只对Xxx()方法进行校验。

iBatis与Hibernate有什么不同?

相同点：屏蔽jdbc api的底层访问细节，使用我们不用与jdbc api打交道，就可以访问数据。  
jdbc api编程流程固定，还将sql语句与java代码混杂在了一起，经常需要拼凑sql语句，细节很繁琐。  
ibatis的好处：屏蔽jdbc api的底层访问细节；将sql语句与java代码进行分离;提供了将结果集自动封装称为实体对象和对象的集合的功能，queryForList返回对象集合，用queryForObject返回单个对象；提供了自动将实体对象的属性传递给sql语句的参数。

Hibernate是一个全自动的orm映射工具，它可以自动生成sql语句,ibatis需要我们自己在xml配置文件中写sql语句，hibernate要比ibatis功能负责和强大很多。因为hibernate自动生成sql语句，我们无法控制该语句，我们就无法去写特定的高效率的sql。对于一些不太复杂的sql查询，hibernate可以很好帮我们完成，但是，对于特别复杂的查询，hibernate就很难适应了，这时候用ibatis就是不错的选择，因为ibatis还是由我们自己写sql语句。

spring+Hibernate中委托方案怎么配置?

解决方案一，按照Object[]数据取出数据，然后自己组bean  
解决方案二，对每个表的bean写构造函数，比如表一要查出field1,field2两个字段，那么有一个构造函数就是Bean(type1 filed1,type2  
field2) ，然后在hql里面就可以直接生成这个bean了。

介绍一下Hibernate的二级缓存

：（1）首先说清楚什么是缓存，（2）再说有了hibernate的Session就是一级缓存，即有了一级缓存，为什么还要有二级缓存，（3）最后再说如何配置Hibernate的二级缓存。  
（1）缓存就是把以前从数据库中查询出来和使用过的对象保存在内存中（一个数据结构中），这个数据结构通常是或类似Hashmap，当以后要使用某个对象时，先查询缓存中是否有这个对象，如果有则使用缓存中的对象，如果没有则去查询数据库，并将查询出来的对象保存在缓存中，以便下次使用

2）Hibernate的Session就是一种缓存，我们通常将之称为Hibernate的一级缓存，当想使用session从数据库中查询出一个对象时，Session也是先从自己内部查看是否存在这个对象，存在则直接返回，不存在才去访问数据库，并将查询的结果保存在自己内部。由于Session代表一次会话过程，一个Session与一个数据库连接相关连，所以Session最好不要长时间保持打开，通常仅用于一个事务当中，在事务结束时就应关闭。并且Session是线程不安全的，被多个线程共享时容易出现问题。通常只有那种全局意义上的缓存才是真正的缓存应用，才有较大的缓存价值，因此，Hibernate的Session这一级缓存的缓存作用并不明显，应用价值不大。Hibernate的二级缓存就是要为Hibernate配置一种全局缓存，让多个线程和多个事务都可以共享这个缓存。我们希望的是一个人使用过，其他人也可以使用，session没有这种效果。

（3）二级缓存是独立于Hibernate的软件部件，属于第三方的产品，多个厂商和组织都提供有缓存产品，例如，EHCache和OSCache等等。在Hibernate中使用二级缓存，首先就要在hibernate.cfg.xml配置文件中配置使用哪个厂家的缓存产品，接着需要配置该缓存产品自己的配置文件，最后要配置Hibernate中的哪些实体对象要纳入到二级缓存的管理中。明白了二级缓存原理和有了这个思路后，很容易配置起Hibernate的二级缓存。扩展知识：一个SessionFactory可以关联一个二级缓存，也即一个二级缓存只能负责缓存一个数据库中的数据，当使用Hibernate 的二级缓存后，注意不要有其他的应用或SessionFactory来更改当前数据库中的数据，这样缓存的数据就会与数据库中的实际数据不一致。

Jdo是什么?  
JDO是Java对象持久化的新的规范

[ibatis与hibernate有什么区别](http://www.mianwww.com/html/2013/08/18552.html)

iBatis的sql语句需要自己手动写，而Hibernate能够在程序运行时自动生成。但是不要就这样以为Hibernate比iBatis方便，其 实二者熟悉之后的效率差不多。而Hibernate还能够自动建表等等，这样的好处就在于你带着这个程序，无论到什么机器上，你都不需要数据库，应为它能 自动为你完成，而iBatis就必须要有相应的数据库表才能进行移植  
Hibernate的特点：  
Hibernate功能强大，数据库无关 性好，O/R（对象/关系）映射能力强，如果你对Hibernate相当精通，而且对Hibernate进行了适当的封装，那么你的项目整个持久层代码会 相当简单，需要写的代码很少，开发速度很快，非常爽。Hibernate对数据库结构提供了较为完整的封装，Hibernate的O/R Mapping实现了POJO（实体类） 和数据库表之间的映射，以及SQL 的自动生成和执行。程序员往往只需定义好了POJO 到数据库表的映射关系，即可通过Hibernate 提供的方法完成持久层操作。程序员甚至不需要对SQL 的熟练掌握， Hibernate/OJB 会根据制定的存储逻辑，自动生成对应的SQL 并调用JDBC 接口加以执行。Hibernate的缺点就是学习门槛不低，要精通门槛更高，而且怎么设计O/R映射，在性能和对象模型之间如何权衡取得平衡，以及怎样用 好Hibernate方面需要你的经验和能力都很强才行，但是Hibernate现在已经是主流O/R Mapping框架，从文档的丰富性，产品的完善性，版本的开发速度都要强于iBATIS。

iBATIS的特点：  
iBATIS入门简单， 即学即用，提供了数据库查询的自动对象绑定功能，而且延续了很好的SQL使用经验，对于没有那么高的对象模型要求的项目来说，相当完美。iBATIS的缺 点就是框架还是比较简陋，功能尚有缺失，虽然简化了数据绑定代码，但是整个底层数据库查询实际还是要自己写的，工作量也比较大，而且不太容易适应快速数据 库修改。当系统属于二次开发,无法对数据库结构做到控制和修改,那iBATIS的灵活性将比Hibernate更适合。系统数据处理量巨大，性能要求极为 苛刻，这往往意味着我们必须通过经过高度优化的SQL语句（或存储过程）才能达到系统性能设计指标。在这种情况下iBATIS会有更好的可控性和表现。  
现在就市场来说，Hibernate的需求要大于iBatis的需求，而Hibernate对个人能力的要求更高，即使公司使用iBatis，你在Hibernate的基础上也能快速掌握。

[hibernate缓存机制详解](http://www.mianwww.com/html/2013/03/18096.html)?

Hibernate缓存的作用：  
Hibernate是一个持久层框架，经常访问物理数据库，为了降低应用程序对物理数据源访问的频次，从而提高应用程序的运行性能。缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据Hibernate缓存分类：

Hibernate缓存包括两大类：Hibernate一级缓存和Hibernate二级缓存Hibernate一级缓存又称为“Session的缓存”，它是内置的，不能被卸载（不能被卸载的意思就是这种缓存不具有可选性，必须有的功能，不可以取消session缓存）。由于Session对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。第一级缓存是必需的，不允许而且事实上也无法卸除。在第一级缓存中，持久化类的每个实例都具有唯一的OID。 Hibernate二级缓存又称为“SessionFactory的缓存”，由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。第二级缓存是可选的，是一个可配置的插件，在默认情况下，SessionFactory不会启用这个插件。

什么样的数据适合存放到第二级缓存中？  
1 很少被修改的数据  
2 不是很重要的数据，允许出现偶尔并发的数据  
3 不会被并发访问的数据  
4 常量数据  
不适合存放到第二级缓存的数据？  
1 经常被修改的数据  
2 .绝对不允许出现并发访问的数据，如财务数据，绝对不允许出现并发  
3 与其他应用共享的数据。

Hibernate查找对象如何应用缓存？  
当Hibernate根据ID访问数据对象的时候，首先从Session一级缓存中查；查不到，如果配置了二级缓存，那么从二级缓存中查；如果都查不到，再查询数据库，把结果按照ID放入到缓存,删除、更新、增加数据的时候，同时更新缓存。Hibernate管理缓存实例无论何时，当你给save()、update()或saveOrUpdate()方法传递一个对象时，或使用load()、 get()、list()、iterate() 或scroll()方法获得一个对象时, 该对象都将被加入到Session的内部缓存中。 当随后flush()方法被调用时，对象的状态会和数据库取得同步。 如果你不希望此同步操作发生，或者你正处理大量对象、需要对有效管理内存时，你可以调用evict() 方法，从一级缓存中去掉这些对象及其集合。

**1．Hibernate的检索方式**

答：①导航对象图检索  ②OID检索  ③HQL检索  ④QBC检索  ⑤本地SQL检索

1.导航对象图检索方式。（根据已经加载的对象，导航到其他对象。）

2.OID检索方式。（按照对象的OID来检索对象。）

3.HQL检索方式。（使用面向对象的HQL查询语言。）

4.QBC检索方式。（使用QBC(Qurey By Criteria) API来检索对象。）

5.本地SQL检索方式。（使用本地数据库的SQL查询语句。）

**3．持久化对象的三种状态，代表含义。**

答： 临时状态——刚用new语句创建，未被持久化，不处于session的缓存中。

持久化状态——已经被持久化，处于session的缓存中。

游离态——已经被持久化，但不处于session的缓存中。

**4、三种检索策略是什么，分别适用于哪种场合？**

答：立即检索——

优点：对应用程序完全透明，缺点：select语句数目多。适用：类级别。

延迟检索——

优点： 由应用程序决定加载哪些对象，可以避免执行多余的select语句以及避免加载不需要访问的对象，节省内存空间，提高检索效率。

缺点： 应用程序如果要访问游离态的代理类实例，必须保证它在持久化时已经被初始化。

适用： 一对多或多对多关联。应用程序不需要立即访问或者根本不会访问的对象。

迫切左外连接检索：

优点：对应用程序完全透明，不管对象处于持久化状态还是游离状态，应用程序都可以方便的从一个对象导航到另一个与它相关联的对象。使用了外连接，select语句数目少。

缺点：可能会加载程序不许要访问的对象。复杂的数据库表连接形象检索性能。

适用：一对一或多对一关联。应用程序需要立即访问的对象。数据库系统具有良好的表连接性能。

**4、ORM解决的不匹配问题**

答： 域模型是面向对象的，关系模型是面向关系的。

域模型中有继承关系，关系模型中不能直接表示继承关系。

域模型中有多对多关联关系，关系模型中通过连接表来表示多对多关联关系。

域模型中有双向关联关系，关系模型中只有单向参照关系，而且总是many参照one方。

域模型提倡精粒度模型，关系模型提倡粗粒度模型。

**5、映射继承关系的三种方式？**P 424

答：（1）继承关系树的每个具体类对应一张表：在具体类对应的表中，不仅包含和具体类属性对应的字段，还包括与具体类的父类属性对应的字段。

（2）继承关系树的根类对应一张表：在根类对应的表中，不仅包括根类属性对应的字段，

还包括根类的所有子类属性对应的字段。

（3）继承关系树中的每个类对应一张表，每个表中只包括和这个类本身属性对应的字段，子类的表参照父类对应的表。

**14.请简述Session的特点有哪些？**

答案：

(1)不是线程安全的，因此在设计软件架构时，应该避免多个线程共享同一个Session实例。

(2)Session实例是轻量级的，所谓轻量级是指它的创建和销毁不需要消耗太多的资源。这意味着在程序中可以经常创建或销毁Session对象，例如为每个客户请求分配单独的Session实例，或者为每个工作单元分配单独的Session实例。

(3)在Session 中，每个数据库操作都是在一个事务(transaction)中进行的，这样就可以隔离开不同的操作（甚至包括只读操作）。

**12、Hibernate中采用XML文件来配置对象-关系映射的优点有那些？**

答案：Hibernate既不会渗透到上层域模型中，也不会渗透到下层数据模型中。软件开发人员可以独立设计域模型，不必强迫遵守任何规范。数据库设计人员可以独立设计数据模型，不必强迫遵守任何规范。对象-关系映射不依赖于任何程序代码，如果需要修改对象-关系映射，只需修改XML文件，不需要修改任何程序，提高了软件的灵活性，并且使维护更加方便。

**14、 叙述Session的缓存的作用**

答案：

（1）减少访问数据库的频率。应用程序从内存中读取持久化对象的速度显然比到数据库中查询数据的速度快多了，因此Session的缓存可以提高数据访问的性能。

（2）保证缓存中的对象与数据库中的相关记录保持同步。当缓存中持久化对象的状态发生了变换，Session并不会立即执行相关的SQL语句，这使得Session能够把几条相关的SQL语句合并为一条SQL语句，以便减少访问数据库的次数，从而提高应用程序的性能。

2乐观锁和悲观锁。  
一对多中，一的一方有多的一方集合，多的一方有以的一方的实体类。

Dbutils:

Common Dbutils是操作数据库的组件，对传统操作数据库的类进行二次封装，可以把结果集转化成List。

补充一下，传统操作数据库的类指的是JDBC（java database connectivity：java数据库连接，java的数据库操作的基础API。）

DBUtils是java编程中的数据库操作实用工具，小巧简单实用，

特色：

1.对于数据表的读操作，他可以把结果转换成List，Array，Set等java集合，便于程序员操作；

2.对于数据表的写操作，也变得很简单（只需写sql语句）

3.可以使用数据源，使用JNDI，数据库连接池等技术来优化性能--重用已经构建好的数据库连接对象，

而不像php，asp那样，费时费力的不断重复的构建和析构这样的对象。

session:

存储需要在整个用户会话过程中保持其状态的信息，例如登录信息或用户浏览 Web应用程序时需要的其它信息。

存储只需要在页重新加载过程中或按功能分组的一组页之间保持其状态的对象。

Session 的作用就是它在 Web服务器上保持用户的状态信息供在任何时间从任何页访问。

因为浏览器不需要存储任何这种信息，所以可以使用任何浏览器，即使是像 PDA 或手机这样的浏览器设备。

servlet:

Servlet是一种服务器端的Java应用程序，具有独立于平台和协议的特性，可以生成动态的Web页面。

它担当客户请求（Web浏览器或其他HTTP客户程序）与服务器响应（HTTP服务器上的数据库或应用程序）的中间层。

Servlet是位于Web 服务器内部的服务器端的Java应用程序，与传统的从命令行启动的Java应用程序不同，

Servlet由Web服务器进行加载，该Web服务器必须包含支持Servlet的Java虚拟机。

Filter: 是过滤器，可以阻止url请求，过滤url请求，当请求符合要求时才会批准请求，进行响应

Listenner:

listener是servlet的监听器，他可以监听客户端的请求，服务端的操作等，通过监听器，可以自动激发一些

操作，比如监听在线用户数量，当增加一个httpsession时，给在线人数加1，编写监听器需要实现相应

的接口，编写完后在web.xml文件中配置一下，就要可以起作用了

拦截器

一、理解Struts2拦截器

1. Struts2拦截器是在访问某个Action或Action的某个方法，字段之前或之后实施拦截，并且Struts2拦截器是可插拔的，拦截器是ＡＯＰ的一种实现．

2. 拦截器栈（Interceptor Stack）。Struts2拦截器栈就是将拦截器按一定的顺序联结成一条链。

在访问被拦截的方法或字段时，Struts2拦截器链中的拦截器就会按其之前定义的顺序被调用。

二、实现Struts2拦截器原理

Struts2拦截器的实现原理相对简单，当请求struts2的action时，Struts 2会查找配置文件，并根据其配置实例化相对的

拦截器对象，然后串成一个列表，最后一个一个地调用列表中的拦截器

三、定义Struts2拦截器。

Struts2规定用户自定义拦截器必须实现com.opensymphony.xwork2.interceptor.Interceptor接口。该接口声明了3个方法，

其中，init和destroy方法会在程序开始和结束时各执行一遍，不管使用了该拦截器与否，只要在struts.xml中声明了该Struts2拦截器就会被执行。

intercept方法就是拦截的主体了，每次拦截器生效时都会执行其中的逻辑。

值栈

值栈：简单的说，就是存放action的堆栈，当我们提交一个请求道服务器端 action时，就有个堆栈，如果action在服务器端进行跳转，

所有action共用一个堆栈，当需要保存在action中的数据时，首先从栈顶开始 搜索，

若找到相同的属性名（与要获得的数据的属性名相同）时，即将值取出

持久化对象：

已经被持久化,加入到session的缓存中，对象和session有了关联，对象处于hibernate框架管理中.处于持久化的java对象被称为持久化对象

JDBC：

是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。

JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序

数据源：

数据源（Data Source）是提供某种所需要数据的器件或原始媒体。顾名思义，数据的来源。

在数据源中存储了所有建立数据库连接的信息。就象通过指定文件名称可以在文件系统中找到文件一样，

通过提供正确的数据源名称，你可以找到相应的数据库连接。

JBPM简要过程：

1、定义流程（利用JPDL）

2、部署流程（部署到数据库）

3、创建公文并与流程实例绑定

4、可通过JBPM的接口，触发流程向下流动

5、可通过JBPM的接口，获得流动到某个用户那里的文档（即待处理任务列表）

6、可通过JBPM的接口，结束某个用户的任务（这将触发流程继续向下流动）

7、如此，直到结束

IOC：控制反转，就是应用本身不负责依赖对象的创建和维护，依赖对象的创建及维护是由外部容器负责，这样控制权由应用转移到了外部容器

控制权的转移就是所谓的反转，目的为了获得更好的扩展性和良好的可维护性。