关于Spring的几道经典试题

1. Spring的Ioc和aop分别是用什么原理实现的?  
  
主要应用了java的反射技术  
  
2.spring 的工作机制以及为什么要用?  
  
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.如何实现国际化?  
  
国际化:不用修改代码，就适用于不同的语言国家  
  
本地化:如果要适应一个国家，要修改源代码  
  
实现过程:  
  
在struts-config.xml中用如下标签说明资源文件名，注意其只是文件名，没有语言\_国家  
  
在资源文件对应的目录struts中加入适应不同语言、国家的properties文件  
  
ApplicationResources\_zh\_CN.properties 中国  
  
ApplicationResources\_en\_US.properties us  
  
如果不是英语要转码  
  
ative2ascii -encoding gb2312 源 ApplicationResources\_zh\_CN.properties  
  
在jsp页面中用  
  
取出信息  
  
4.Spring的BeanFactory与ApplicationContext区别?  
  
ApplicationContext和BeanFacotry相比,提供了更多的扩展功能，但其主要区别在于后者是延迟加载,如果Bean的某一个 属性 没有注入，BeanFacotry加载后，直至第一次使用调用getBean方法才会抛出异常;而ApplicationContext则在初始化自身是 检验，这样有利于检查所依赖属性是否注入;所以通常情况下我们选择使用ApplicationContext。  
  
5.说说你对用ssh框架进行开发的理解?  
  
框架：hibernate、spring、struts;  
  
Hibernate 主要用于数据持久化;  
  
Spring 的控制反转能起到解耦合的作用;  
  
Struts 主要用于流程控制。  
  
6.请解释什么是MVC?(设计模式)  
  
A) M----Model 封装业务数据及业务运算。  
  
B) V-----View 封装用户界面。  
  
C) C-----Controller 封装业务处理流程。  
  
7. MVC的各个部分都有哪些技术来实现?如何实现?  
  
表示层 struts  
  
持久层 hibernate  
  
控制层 spring  
  
也就是最常用的开发框架ssh  
  
8.通过图，说说你对MVC模型的理解。(企业有可能让学生画图)?  
  
1)首先是展示视图给用户，用户在这个视图上进行操作，并填写一些业务数据  
  
2)然后用户会点击提交按钮，来发出请求  
  
3)视图发出的用户请求会到达控制器，在请求中包含了想要完成什么样的业务功能以及相关的数据。  
  
4)控制器会来处理用户请求，会把请求中的数据进行封装，然后选择并调用合适的模型，请求模型进行状态更新，然后选择接下来要展示给用户的视图。  
  
5)模型会去处理用户请求的业务功能，同时进行模型状态的维护和更新  
  
6)当模型状态发生改变的时候，模型会通知相应的视图，告诉视图它的状态发生了改变。  
  
7)视图借到模型的通知后，会向模型进行状态查询，获取需要展示的数据，然后按照视图本身的展示方式，把这些数据展示出来。  
  
8)接下来就是等待用户下一步操作，再次从头轮回了。