

《Spring - 就业技能》

章节	Ch01 - Spring 核心
题目 1)	什么是 spring?
	Spring 是个 java 企业级应用的开源开发框架。Spring 主要用来开发 Java 应用, 但是有些扩展是针对构建 J2EE 平台的 web 应用。Spring 框架目标是简化 Java 企业级应用开发, 并通过 POJO 为基础的编程模型促进良好的编程习惯。
题目 2)	使用 Spring 框架的好处是什么?
	轻量: Spring 是轻量的, 基本的版本大约 2MB。 控制反转: Spring 通过控制反转实现了松散耦合, 对象们给出它们的依赖, 而不是创建或查找依赖的对象们。 面向切面的编程(AOP): Spring 支持面向切面的编程, 并且把应用业务逻辑和系统服务分开。 容器: Spring 包含并管理应用中对象的生命周期和配置。 MVC 框架: Spring 的 WEB 框架是个精心设计的框架, 是 Web 框架的一个很好的替代品。 事务管理: Spring 提供一个持续的事务管理接口, 可以扩展到上至本地事务下至全局事务 (JTA)。 异常处理: Spring 提供方便的 API 把具体技术相关的异常 (比如由 JDBC, Hibernate or JDO 抛出的) 转化为一致的 unchecked 异常。
题目 3)	简述 Spring 配置文件
	Spring 配置文件是个 XML 文件, 这个文件包含了类信息, 描述了如何配置它们, 以及如何相互调用。
题目 4)	解释 AOP 模块
	AOP 模块用于发给我们的 Spring 应用做面向切面的开发, 很多支持由 AOP 联盟提供, 这样就确保了 Spring 和其他 AOP 框架的共通性。这个模块将元数据编程引入 Spring。
题目 5)	什么是 Spring IOC 容器?
	Spring IOC 负责创建对象, 管理对象 (通过依赖注入 (DI), 装配对象, 配置对象, 并且管理这些对象的整个生命周期。

章节	Ch02 - IoC 和 AOP 扩展
题目 1)	IOC 的优点是什么?
	IOC 或 依赖注入把应用的代码量降到最低。它使应用容易测试, 单元测试不再需要单例和 JNDI 查找机制。最小的代价和最小的侵入性使松散耦合得以实现。IOC 容器支持加载服务时的饿汉式初始化和懒加载。
题目 2)	ApplicationContext 通常的实现是什么?
	FileSystemXmlApplicationContext : 此容器从一个 XML 文件中加载 beans 的定义, XML Bean 配置文件的全路径名必须提供给它的构造函数。 ClassPathXmlApplicationContext: 此容器也从一个 XML 文件中加载 beans 的定义, 这里, 你需要正确设置 classpath 因为这个容器将在 classpath 里找 bean 配置。 WebXmlApplicationContext: 此容器加载一个 XML 文件, 此文件定义了一个 WEB 应用的

	所有 bean。
题目 3)	有哪些不同类型的 IOC (依赖注入) 方式?
	构造器依赖注入: 构造器依赖注入通过容器触发一个类的构造器来实现的, 该类有一系列参数, 每个参数代表一个对其他类的依赖。 Setter 方法注入: Setter 方法注入是容器通过调用无参构造器或无参 static 工厂 方法实例化 bean 之后, 调用该 bean 的 setter 方法, 即实现了基于 setter 的依赖注入。
题目 4)	哪种依赖注入方式你建议使用, 构造器注入, 还是 Setter 方法注入?
	你两种依赖方式都可以使用, 构造器注入和 Setter 方法注入。最好的解决方案是用构造器参数实现强制依赖, setter 方法实现可选依赖。
题目 5)	解释 AOP
	面向切面的编程, 或 AOP, 是一种编程技术, 允许程序模块化横向切割关注点, 或横切典型的责任划分, 如日志和事务管理。

章节	Ch03 - Spring MVC 体系结构和处理请求控制器
题目 1)	springMVC 工作流程
	1、用户发送请求至前端控制器 DispatcherServlet 2、DispatcherServlet 收到请求调用 HandlerMapping 处理器映射器。 3、处理器映射器找到具体的处理器, 生成处理器对象及处理器拦截器(如果有则生成)一并返回给 DispatcherServlet。 4、DispatcherServlet 调用 HandlerAdapter 处理器适配器 5、HandlerAdapter 经过适配调用具体的处理器(Controller, 也叫后端控制器)。 6、Controller 执行完成返回 ModelAndView 7、HandlerAdapter 将 controller 执行结果 ModelAndView 返回给 DispatcherServlet 8、DispatcherServlet 将 ModelAndView 传给 ViewResolver 视图解析器 9、ViewResolver 解析后返回具体 View 10、DispatcherServlet 根据 View 进行渲染视图 (即将模型数据填充至视图中)。 11、DispatcherServlet 响应用户
题目 2)	如何解决 POST 请求中文乱码问题, GET 的又如何处理呢?
	在 web.xml 中加入: CharacterEncodingFilter 解决 post 请求乱码问题。 对于 get 请求中文参数出现乱码解决方法有两个 a.修改 tomcat 配置文件添加编码 URIEncoding="utf-8" b.对参数进行重新编码: String userName = new String(request.getParamter("userName").getBytes("ISO8859-1"),"utf-8")
题目 3)	SpringMVC 与 Struts2 的主要区别?
	①springmvc 的入口是一个 servlet 即前端控制器, 而 struts2 入口是一个 filter 过滤器。 ②springmvc 是基于方法开发, 传递参数是通过方法形参, 可以设计为单例或多例(建议单例), struts2 是基于类开发, 传递参数是通过类的属性, 只能设计为多例。 ③Struts 采用值栈存储请求和响应的数据, 通过 OGNL 存取数据, springmvc 通过参数解析器是将 request 对象内容进行解析成方法形参, 将响应数据和页面封装成 ModelAndView 对象, 最后又将模型数据通过 request 对象传输到页面。 Jsp 视图解析器默认使用 jstl。
题目 4)	SpringMvc 的控制器是不是单例模式,如果是,有什么问题,怎么解决

	是单例模式,所以在多线程访问的时候有线程安全问题,不要用同步,会影响性能的,解决方案是在控制器里面不能写字段
题目 5)	如果在拦截请求中,我想拦截 get 方式提交的方法,怎么配置
	可以在@RequestMapping 注解里面加上 method=RequestMethod.GET

章节	Ch04 - Spring MVC 核心应用-1
题目 1)	@RequestMapping 注解用在类上面有什么作用
	用于类上, 表示类中的所有响应请求的方法都是以该地址作为父路径
题目 2)	怎么样在方法里面得到 Request,或者 Session
	直接在方法的形参中声明 request 或者 session, SpringMvc 就自动把 request 或者 session 对象传入
题目 3)	@RequestParam 作用
	通过@RequestParam 对简单类型(8 种基本类型都可以)的参数进行绑定, 如果不使用此注解, 要求 request 传入的参数名和 controller 方法的形参要一致才可以。
题目 4)	@RequestBody 作用
	@RequestBody 注解用于读取 http 请求的内容(字符串), 通过 springmvc 提供的 <code>HttpMessageConverter</code> 接口将读到的内容(json 数据)转换为 java 对象并绑定到 Controller 方法的参数上。 @RequestBody 注解实现接收 http 请求的 json 数据, 将 json 数据转换为 java 对象进行绑定
题目 5)	@ResponseBody 作用
	@ResponseBody 注解用于将 Controller 的方法返回的对象, 通过 springmvc 提供的 <code>HttpMessageConverter</code> 接口转换为指定格式的数据如: json,xml 等, 通过 Response 响应给客户端 @ResponseBody 注解实现将 Controller 方法返回 java 对象转换为 json 响应给客户端。

章节	Ch05 - Spring MVC 核心应用-2
题目 1)	简述异常处理器
	springmvc 在处理请求过程中出现异常信息交由异常处理器进行处理, 自定义异常处理器可以实现一个系统的异常处理逻辑。 系统的 dao、service、controller 出现都通过 throws Exception 向上抛出, 最后由 springmvc 前端控制器交由异常处理器进行异常处理
题目 2)	如何从 url 上获取参数(@PathVariable)
	@RequestMapping("item/{id}") public Item queryItemById(@PathVariable() Integer id) {} {xxx}叫做占位符, 请求的 URL 可以是 "item /1" 或 "item/2" 使用(@PathVariable() Integer id)获取 url 上的数据 如果@RequestMapping 中表示为"item/{id}", id 和形参名称一致, @PathVariable 不用指定名称。如果不一致, 例如"item/{itemId}"则需要指定名称@PathVariable("itemId")
题目 3)	如何使用 InitBinder 来处理 Date 类型的参数
	@InitBinder

	<pre>public void initBinder(ServletRequestDataBinder binder){ binder.registerCustomEditor(Date.class, new CustomDateEditor(new SimpleDateFormat("yyyy-MM-dd"), true)); }</pre>
题目 4)	如何实现文件上传
	<p>1.导入 commons-fileupload.jar commons-io.jar</p> <p>2.配置</p> <pre><bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"> <property name="maxUploadSize" value="102400000"></property></bean></pre> <p>3.编写控制器代码</p> <p>4.表单</p> <pre><form action="mvc/upload" method="post" enctype="multipart/form-data"> <input type="file" name="file">
 <input type="submit" value="submit"></form></pre>
题目 5)	简述自定义拦截器
	<p>1.创建一个 MyInterceptor 类，并实现 HandlerInterceptor 接口</p> <p>2.在 SpringMVC 的配置文件中配置</p> <pre><mvc:interceptors> <mvc:interceptor> <mvc:mapping path="/mvc/**"/> <bean class="test.SpringMVC.Interceptor.MyInterceptor"></bean> </mvc:interceptor> </mvc:interceptors></pre>

章节	Ch06 - Spring MVC 扩展和 SSM 框架整合
题目 1)	SSM 整合步骤
	<p>1.Mybatis 和 spring 整合，通过 spring 管理 mapper 接口 使用 mapper 的扫描器自动扫描 mapper 接口在 spring 中进行注册</p> <p>2.通过 spring 管理 service，实现事务控制</p> <p>3.springMVC 是 spring 模块，无缝整合</p>
题目 2)	Mybatis 和 hibernate 比较
	<p>1. Mybatis 和 hibernate 不同，它不完全是个 ORM 框架，因为 MyBatis 需要程序员自己编写 Sql 语句，不过 mybatis 可以通过 XML 或注解方式灵活配置要运行的 sql 语句，并将 java 对象和 sql 语句映射生成最终执行的 sql，最后将 sql 执行的结果再映射生成 java 对象。</p> <p>2. Mybatis 学习门槛低，简单易学，程序员直接编写原生态 sql，可严格控制 sql 执行性能，灵活度高，非常适合对关系数据模型要求不高的软件开发，例如互联网软件、企业运营类软件等，因为这类软件需求变化频繁，一旦需求变化要求成果输出迅速。但是灵活的前提是 mybatis 无法做到数据库无关性，如果需要进行支持多种数据库的软件则需要自定义多套 sql 映射文件，工作量大。</p> <p>3. Hibernate 对象/关系映射能力强，数据库无关性好，对于关系模型要求高的软件（例如需</p>

	<p>求固定的定制化软件) 如果用 hibernate 开发可以节省很多代码, 提高效率。但是 Hibernate 的学习门槛高, 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, 以及怎样用好 Hibernate 需要具有很强的经验和能力才行。</p> <p>4. 总之, 按照用户的需求在有限的资源环境下只要能做出维护性、扩展性良好的软件架构都是好架构, 所以框架只有适合才是最好。</p>
题目 3)	Maven 是什么及其优点
	<p>1.项目非常大时, 可借助 Maven 将一个项目拆分成多个工程, 最好是一个模块对应一个工程, 利于分工协作</p> <p>2.借助 Maven, 可将 jar 包仅仅保存在“仓库”中, 有需要该文件时, 就引用该文件接口, 不需要复制文件过来占用空间</p> <p>3.借助 Maven 可以以规范的方式下载 jar 包, 因为所有的知名框架或第三方工具的 jar 包已经按照统一的规范存放到了 Maven 的中央仓库中</p> <p>4.Maven 会自动将你要加入到项目中的 jar 包导入, 不仅导入, 而且还会将该 jar 包所依赖的 jar 包都自动导入进来</p>
题目 4)	git 的作用
	<p>他的作用有两个, 第一个就是版本控制, 第二个就是作为开源代码库, 他的第一个作用就类似与在软件开发中使用的其他版本控制软件类似与 SVN、VSS、CVS 等等, 第二个作用主要用于开源项目的维护, 而且现在的 IT 界越来越具有开源精神, 程序员之间可以通过 git 共同来维护一个项目, 贡献自己的聪明才智</p>
题目 5)	ssm 和 ssh 的优势
	<p>两者的相同点:</p> <p>Hibernate 与 MyBatis 都可以是通过 SessionFactoryBuider 由 XML 配置文件生成 SessionFactory, 然后由 SessionFactory 生成 Session, 最后由 Session 来开启执行事务和 SQL 语句。其中 SessionFactoryBuider, SessionFactory, Session 的生命周期都是差不多的。</p> <p>Hibernate 和 MyBatis 都支持 JDBC 和 JTA 事务处理。</p> <p>两者各自优势:</p> <p>MyBatis 可以进行更为细致的 SQL 优化, 可以减少查询字段。</p> <p>MyBatis 容易掌握, 而 Hibernate 门槛较高。</p> <p>Hibernate 的 DAO 层开发比 MyBatis 简单, Mybatis 需要维护 SQL 和结果映射。</p> <p>Hibernate 对对象的维护和缓存要比 MyBatis 好, 对增删改查的对象的维护要方便。</p> <p>Hibernate 数据库移植性很好, MyBatis 的数据库移植性不好, 不同的数据库需要写不同 SQL。</p> <p>Hibernate 有更好的二级缓存机制, 可以使用第三方缓存。MyBatis 本身提供的缓存机制不佳, 更新操作不能指定刷新指定记录, 会清空整个表, 但是也可以使用第三方缓存。</p> <p>Hibernate 封装性好, 屏蔽了数据库差异, 自动生成 SQL 语句, 应对数据库变化能力较弱, SQL 语句优化困难。</p> <p>MyBatis 仅实现了 SQL 语句和对象的映射, 需要针对具体的数据库写 SQL 语句, 应对数据库变化能力较强, SQL 语句优化较为方便。</p> <p>总结:</p> <p>SSM 和 SSH 不同主要在 MVC 实现方式, 以及 ORM 持久化方面不同 (Hiibernate 与 Mybatis)。SSM 越来越轻量级配置, 将注解开发发挥到极致, 且 ORM 实现更加灵活, SQL 优化更简便; 而 SSH 较注重配置开发, 其中的 Hiibernate 对 JDBC 的完整封装更面向对象, 对增删改查的数据维护更自动化, 但 SQL 优化方面较弱, 且入门门槛稍高。</p>

