

《Java Web - 就业技能》

章节	Ch01 - 搭建 Web 环境初识 JSP
题目 1)	BS/CS 架构的区别?
	cs 是客户端服务器模式; bs 是浏览器服务器模式。它们最主要的区别是 cs 架构的应用程序需要在用户的计算机上进行安装。如我们常用的 word,excel 等应用软件属于 cs 架构; 而 bs 架构的应用程序不需要在用户的计算机上进行安装, 只需要有一个浏览器就可以运行 bs 架构的应用程序。如现在网上一般的电子商务网站即属于此类型。
题目 2)	Tomcat 目录下有那些文件和文件夹
	bin 目录 存放各种平台下用于启动和停止 Tomcat 的脚本文件 conf 目录 存放 Tomcat 服务器的各种配置文件 lib 目录 存放 Tomcat 服务器所需的各种 JAR 文件 logs 目录 存放 Tomcat 的日志文件 temp 目录 Tomcat 运行时用于存放临时文件 webapps 目录 当发布 Web 应用时, 默认情况下会将 Web 应用的文件存放于此目录中 work 目录 Tomcat 把由 JSP 生成的 Servlet 放于此目录下
题目 3)	JSP 页面元素的组成?
	静态内容 HTML 静态文本 指令 以 "<%@" 开始, 以 "%>" 结束 小脚本 <%Java 代码 %> 表达式 <%=Java 表达式 %> 声明 <%! 方法 %> 注释 <!-- 客户端可以查看到 --> <%-- 客户端无法查看到 --%>
题目 4)	Myeclipse 如何部署安装 tomcat?
	上机操作题 1. Window -> Preferences -> MyEclipse -> Servers -> Tomcat 2. 选择 Tomcat 版本及安装路径 3. 设置为可用状态 (Enable) 4. 指定 Tomcat 运行 Java 的运行环境
题目 5)	Web 项目常见异常
	未启动 Tomcat 未部署 Web 应用 URL 输入错误 目录不能被引用

章节	Ch02 - JSP 实现数据传递和保存
题目 1)	四个作用域
	application 在所有应用程序中有效 session 在当前会话中有效

	request 在当前请求中有效 page 在当前页面有效
题目 2)	如何解决乱码问题?
	<p>第一种: 在 Tomcat 目录结构\conf\server.xml 中设置字符集 <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" URIEncoding="UTF-8"/></p> <p>第二种: String name = request.getParameter("name"); // 读取用户名和密码 name = new String(name.getBytes("ISO-8859-1"), "UTF-8");// 对请求数据进行字符编码</p> <p>第三种: request.setCharacterEncoding("UTF-8");// 设置读取请求信息的字符编码为 UTF-8 String name = request.getParameter("name"); // 读取用户名和密码 String pwd = request.getParameter("pwd");</p>
题目 3)	转发重定向的区别?
	<p>转发的作用: 在服务器端, 将请求发送给服务器上的其他资源, 以共同完成一次请求的处理 request.getRequestDispatcher("welcome.jsp").forward(request, response);</p> <p>重定向: 客户端将重新发送请求到指定的 URL response.sendRedirect("welcome.jsp");</p>
题目 4)	Cookie 的作用
	<p>cookie 作用:</p> <ol style="list-style-type: none"> 1.可以在客户端上保存用户数据, 起到简单的缓存和用户身份识别等作用。 2.保存用户的登陆状态, 用户进行登陆, 成功登陆后, 服务器生成特定的 cookie 返回给客户端, 客户端下次访问该域名下的任何页面, 将该 cookie 的信息发送给服务器, 服务器经过检验, 来判断用户是否登陆。 3.记录用户的行为。 cookie 以字符串的形式保存在浏览器中。
题目 5)	JSP 内置对象
	<p>request 请求对象 类型 javax.servlet.HttpServletRequest 作用域 Request response 响应对象 类型 javax.servlet.HttpServletResponse 作用域 Page pageContext 页面上下文对象 类型 javax.servlet.jsp.PageContext 作用域 Page session 会话对象 类型 javax.servlet.http.HttpSession 作用域 Session application 应用程序对象 类型 javax.servlet.ServletContext 作用域 Application out 输出对象 类型 javax.servlet.jsp.JspWriter 作用域 Page config 配置对象 类型 javax.servlet.ServletConfig 作用域 Page page 页面对象 类型 javax.lang.Object 作用域 Page</p>

章节	Ch03 - Servlet
题目 1)	Servlet 生命周期?
	<p>Servlet 生命周期包括三部分:</p> <p>初始化: Web 容器加载 servlet, 调用 init()方法</p>

	<p>处理请求：当请求到达时，运行其 service()方法。service()自动派遣运行与请求相对应的 doXXX (doGet 或者 doPost) 方法。</p> <p>销毁：服务结束，web 容器会调用 servlet 的 destroy()方法销毁 servlet。</p>
题目 2)	get 提交 post 提交有何区别
	<p>(1) get 一般用于从服务器上获取数据，post 一般用于向服务器传送数据</p> <p>(2) 请求的时候参数的位置有区别，get 的参数是拼接在 url 后面，用户在浏览器地址栏可以看到。post 是放在 http 包的包体中。比如说用户注册，你不能把用户提交的注册信息用 get 的方式吧，那不是说把用户的注册信息都显示在 Url 上了吗，是不安全的。</p> <p>(3) 能提交的数据有区别，get 方式能提交的数据只能是文本，且大小不超过 1024 个字节，而 post 不仅可以提交文本还有二进制文件。所以说想上传文件的话，我们就需要使用 post 请求方式</p> <p>(4) servlet 在处理请求的时候分别对应使用 doGet 和 doPost 方式进行处理请求</p>
题目 3)	JSP 与 Servlet 有什么区别
	<p>Servlet 是服务器端的程序，动态生成 html 页面发送到客户端，但是这样程序里会有很多 out.println().java 与 html 语言混在一起很乱，所以后来 sun 公司推出了 JSP。其实 JSP 就是 Servlet，每次运行的时候 JSP 都首先被编译成 servlet 文件，然后再被编译成.class 文件运行。有了 jsp，在 MVC 项目中 servlet 不再负责动态生成页面，转而去负责控制程序逻辑的作用，控制 jsp 与 javabean 之间的流转。</p>
题目 4)	request.getAttribute()和 request.getParameter
	<p>(1) 有 setAttribute，没有 setParameter 方法</p> <p>(2) getParameter 获取到的值只能是字符串，不可以是对象，而 getAttribute 获取到的值是 Object 类型的。</p> <p>(3) 通过 form 表单或者 url 来向另一个页面或者 servlet 传递参数的时候需要用 getParameter 获取值；getAttribute 只能获取 setAttribute 的值</p> <p>(4) setAttribute 是应用服务器把这个对象放到该页面所对应的一块内存当中，当你的页面服务器重新定向到另一个页面的时候，应用服务器会把这块内存拷贝到另一个页面对应的内存当中。通过 getAttribute 可以取得你存下的值，当然这种方法可以用来传对象。用 session 也是一样的道理，这是说 request 和 session 的生命周期不一样而已。</p>
题目 5)	servlet 涉及到那些接口和类
	Collection List set arraylist linklist hashset treeset

章节	Ch04 - EL 与 JSTL
题目 1)	EL 表达式的语法是什么？
	美元符号加大括号 \${ }
题目 2)	为什么要用 EL 表达式，有什么好处
	<p>1.在页面中用 js 脚本和 jsp 表达式获取数据显示比较麻烦</p> <p>2.好处是 EL 表达式简单易用，默认可访问所有的 JSP 隐藏对象</p>
题目 3)	JSTL 常实现网页逻辑判断和迭代显示的，需要引入？
	<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
题目 4)	JSTL 里面的判断标签是什么？
	<pre><c:if test=""> </c:if> <c:choose> <c:when test="\${}"> </c:when></pre>

	<pre><c:otherwise> </c:otherwise> </c:choose></pre>
题目 5)	JSTL 里面的循环标签怎么用的?
	<pre><c:forEach items="\${list}" var="v"> 用于迭代数字或集合 \${v.id} </c:forEach></pre>

章节	Ch05 - 数据源及分层开发 (MVC)
题目 1)	什么是 MVC (模型 视图 控制器)?
	<p>MVC 是一个架构模式，它分离了表现与交互。它被分为三个核心部件：模型、视图、控制器。下面是每一个部件的分工：</p> <p>视图：是用户看到并与之交互的界面。</p> <p>模型：表示业务数据，并提供数据给视图。</p> <p>控制器：接受用户的输入并调用模型和视图去完成用户的需求。</p>
题目 2)	你能解释下 MVC 的完整流程吗?
	<p>下面是 MVC（模型、视图、控制器）架构的控制流程：</p> <p>所有的终端用户请求被发送到控制器。</p> <p>控制器依赖请求去选择加载哪个模型，并把模型附加到对应的视图。</p> <p>附加了模型数据的最终视图做为响应发送给终端用户。</p>
题目 3)	使用 MVC 有哪些好处?
	<p>分离了关注点。后台代码被移到单独的类文件，我们可以最大限度的重复利用代码。</p> <p>自动化 UI 测试成为可能，因为后台代码移到了.NET 类。这让我们更容易做单元测试和自动化测试。</p>
题目 4)	Java 中经典 MVC 使用框架
	<p>Spring Web MVC 是一种基于 Java 的实现了 Web MVC 设计模式的请求驱动类型的轻量级 Web 框架。使用了 MVC 架构模式的思想，将 web 层进行职责解耦，基于请求驱动指的就是使用请求-响应模型。框架的目的就是帮助我们简化开发。</p>
题目 5)	提交 FORM 表单的方式
	<p>Submit 类型的 button jquery \$("form").submit() ajax</p>

章节	Ch06 - 综合练习 (MVC)
题目 1)	什么情况下调用 doGet()和 doPost()?
	<p>默认情况是调用 doGet()方法，</p> <p>JSP 页面中的 Form 表单的 method 属性设置为 post 的时候，调用的为 doPost()方法；为 get 的时候，调用 deGet()方法。</p>
题目 2)	JSP 常用指令有哪些?
	<p>page、include、taglib</p> <p>a、page 指令：定义页面的一些属性，常用属性：</p> <p>contentType="text/html;charset=utf-8";向浏览器端输出数据的编码</p> <p>pageEncoding="utf-8"; JSP 编译成 java 文件时所用的编码</p> <p>b、include 指令：引入一个静态的 JSP 页面</p>

	c、taglib 指令：引入一个标签库
题目 3)	JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？
	JSP 是 Servlet 的扩展，本质上是 Servlet 的简易方式，更强调应用的外表表达。 JSP 编译后是"类 Servlet"。Servlet 和 JSP 最主要的不同点在于,Servlet 的应用逻辑是在 java 文件中， 并且完全从表示层中的 HTML 里分离开来。而 JSP 的情况是 java 和 HTML 可以组合成一个扩展名为.jsp 的文件。 JSP 偏重于视图，Servlet 偏重于业务逻辑。
题目 4)	在 web 应用开发过程中经常遇到输出某种编码的字符，如 iso-8859-1 等，如何输出一个某种编码的字符串？
	下面这个例子是将 iso-8859-1 转换成 utf-8 编码方式。 <pre>public String Translate(String str) { String s=""; try{ s=new String(str.getBytes("iso-8859-1"),"UTF-8"); s=str.trim(); }catch(Exception e){ System.err.println(e.getMessage()); } return s; }</pre>
题目 5)	常用的 web 容器和开发工具？
	容器：Tomcat、Weblogic 工具：Eclipse、Jbuilder

章节	Ch07 - 分页查询
题目 1)	分页的作用
	在项目开发中，数据很多，web 项目中一张网页不可能显示太多的数据，所以要用分页。 分页分为真分页，假分页
题目 2)	分页类大该包含那些属性
	上一页、下一页、首页、尾页、当前页、每页显示条数
题目 3)	有那些注意事项？
	第一步 要设置每页显示条数 第二步 设置总页数 第三步 设置当前页
题目 4)	当前页代码？
	<pre>public void setCurrentPage(Integer currentPage) { this.currentPage = currentPage; pre = currentPage <= 1 ? 1 : currentPage - 1; next = currentPage >= pageCount ? pageCount : currentPage + 1; }</pre>
题目 5)	总条数代码？

	<pre> public void setTotalCount(Integer totalCount) { this.totalCount = totalCount; totalPage = totalCount%pageCount==0?totalCount/pageCount:totalCount/pageCount+1; } </pre>
--	---

章节	Ch08 - 监听器，过滤器
题目 1)	监听器有哪些作用和用法?
	<p>Java Web 开发中的监听器 (listener) 就是 application、session、request 三个对象创建、销毁或者往其中添加修改删除属性时自动执行代码的功能组件，如下所示：</p> <p>①ServletContextListener：对 Servlet 上下文的创建和销毁进行监听。</p> <p>②ServletContextAttributeListener：监听 Servlet 上下文属性的添加、删除和替换。</p> <p>③HttpSessionListener：对 Session 的创建和销毁进行监听。</p> <p>补充：session 的销毁有两种情况：</p> <p>session 超时（可以在 web.xml 中通过<session- config>/<session-timeout>标签配置超时时间）；</p> <p>通过调用 session 对象的 invalidate()方 法使 session 失效。</p> <p>④HttpSessionAttributeListener：对 Session 对象中属性的添加、删除和替换进行监听。</p> <p>⑤ServletRequestListener：对请求对象的初始化和销毁进行监听。</p> <p>⑥ServletRequestAttributeListener：对请求对象属性的添加、删除和替换进行监听。常见的监听器用途主要包括：网站在线人数技术、监听用户的行为(管理员踢人)。</p>
题目 2)	过滤器有哪些作用和用法?
	<p>Java Web 开发中的过滤器 (filter) 是从 Servlet 2.3 规范开始增加的功能，并在 Servlet 2.4 规范中得到增强。</p> <p>对 Web 应用来说，过滤器是一个驻留在服务器端的 Web 组件，它可以截取客户端和服务端之间的请求与响应信息，并对这些信息进行过 滤。</p> <p>当 Web 容器接受到一个对资源的请求时，它将判断是否有过滤器与这个资源相关联。如果有，那么容器将把请求交给过滤器进行处理。</p> <p>在过滤器中，你可以改 变请求的内容，或者重新设置请求的报头信息，然后再将请求发送给目标资源。</p> <p>当目标资源对请求作出响应时候，容器同样会将响应先转发给过滤器，再过滤器中， 你可以对响应的内容进行转换，然后再将响应发送到客户端。</p> <p>常见的过滤器用途主要包括：对用户请求进行统一认证、对用户的访问请求进行记录和审核、对用户发送的数据进行过滤或替换、转换图象格式、对响应内容进行压缩以减少传输量、对请求或响应进行加解密处理、触发资源访问事件、对 XML 的输出应用 XSLT 等。</p> <p>和过滤器相关的接口主要有：Filter、FilterConfig、FilterChain</p>
题目 3)	java 中过滤器和监听器都有什么区别?
	<p>一、过滤器是一个程序，它先于与之相关的 servlet 或 JSP 页面运行在服务器上。过滤器可附加到一个或多个 servlet 或 JSP 页面上，并且可以检查进入这些资源的请求信息。在这之后，过滤器可以作如下的选择：</p> <p>①以常规的方式调用资源（即，调用 servlet 或 JSP 页面）。</p> <p>②利用修改过的请求信息调用资源。</p>

	<p>③调用资源，但在发送响应到客户机前对其进行修改。</p> <p>④阻止该资源调用，代之以转到其他的资源，返回一个特定的状态代码或生成替换输出。</p> <p>在 Servlet 作为过滤器使用时，它可以对客户请求进行处理。处理完成后，它会交给下一个过滤器处理，这样，客户的请求在过滤链里逐个处理，直到请求发送到目标为止。</p> <p>例如，某网站里有提交“修改的注册信息”的网页，当用户填写完修改信息并提交后，服务器在进行处理时需要做两项工作：判断客户端的会话是否有效；对提交的数据进行统一编码。这两项工作可以在由两个过滤器组成的过滤链里进行处理。当过滤器处理成功后，把提交的数据发送到最终目标；如果过滤器处理不成功，将把视图派发到指定的错误页面。</p> <p>二、监听器是实现了 javax.servlet.ServletContextListener 接口的服务器端程序，它也是随 web 应用的启动而启动，只初始化一次，随 web 应用的停止而销毁。</p> <p>主要作用是：做一些初始化的内容添加工作、设置一些基本的内容、比如一些参数或者是一些固定的对象等等</p>
题目 4)	监听器有哪些类型
	<p>按监听对象划分：servlet2.4 规范定义的事件有三种：</p> <ol style="list-style-type: none"> 1、用于监听应用程序环境对象 (ServletContext) 的事件监听器 2、用于监听用户会话对象 (HttpSession) 的事件监听器 3、用于监听请求消息对象 (ServletRequest) 的事件监听器 <p>按监听的时间类项划分</p> <ol style="list-style-type: none"> 1、用于监听域对象自身的创建和销毁的事件监听器 2、用于监听域对象中的属性的增加和删除的事件监听器 3、用于监听绑定到 HttpSession 域中的某个对象的状态的事件监听器
题目 5)	监听器常用的用途
	<ol style="list-style-type: none"> 1.通常使用 Web 监听器做以下的内容： 2.统计在线人数，利用 HttpSessionListener 3.加载初始化信息：利用 ServletContextListener 4.统计网站访问量

章节	Ch09 - Ajax 与 jQuery (一)
题目 1)	如何解决跨域问题？
	<p>理解跨域的概念：协议、域名、端口都相同才同域，否则都是跨域</p> <p>出于安全考虑，服务器不允许 ajax 跨域获取数据，但是可以跨域获取文件内容，所以基于这一点，可以动态创建 script 标签，使用标签的 src 属性访问 js 文件的形式获取 js 脚本，并且这个 js 脚本中的内容是函数调用，该函数调用的参数是服务器返回的数据，为了获取这里的参数数据，需要事先在页面中定义回调函数，在回调函数中处理服务器返回的数据，这就是解决跨域问题的主流解决方案</p>
题目 2)	简述 ajax 的过程。
	<ol style="list-style-type: none"> 1. 创建 XMLHttpRequest 对象,也就是创建一个异步调用对象 2. 创建一个新的 HTTP 请求,并指定该 HTTP 请求的方法、URL 及验证信息 3. 设置响应 HTTP 请求状态变化的函数 4. 发送 HTTP 请求 5. 获取异步调用返回的数据 6. 使用 JavaScript 和 DOM 实现局部刷新
题目 3)	一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？

	分为 4 个步骤： 1. 当发送一个 URL 请求时，不管这个 URL 是 Web 页面的 URL 还是 Web 页面上每个资源的 URL，浏览器都会开启一个线程来处理这个请求，同时在远程 DNS 服务器上启动一个 DNS 查询。这能使浏览器获得请求对应的 IP 地址。 2. 浏览器与远程 Web 服务器通过 TCP 三次握手协商来建立一个 TCP/IP 连接。该握手包括一个同步报文，一个同步-应答报文和一个应答报文，这三个报文在浏览器和服务器之间传递。该握手首先由客户端尝试建立起通信，而后服务器应答并接受客户端的请求，最后由客户端发出该请求已经被接受的报文。 3. 一旦 TCP/IP 连接建立，浏览器会通过该连接向远程服务器发送 HTTP 的 GET 请求。远程服务器找到资源并使用 HTTP 响应返回该资源，值为 200 的 HTTP 响应状态表示一个正确的响应。 4. 此时，Web 服务器提供资源服务，客户端开始下载资源。
题目 4)	ajax 请求时，如何解释 json 数据
	使用 eval() 或者 JSON.parse() 鉴于安全性考虑，推荐使用 JSON.parse() 更靠谱，对数据的安全性更好。
题目 5)	http 常见的状态码有那些？分别代表是什么意思？
	200 - 请求成功 301 - 资源（网页等）被永久转移到其它 URL 404 - 请求的资源（网页等）不存在 500 - 内部服务器错误

章节	Ch10 - Ajax 与 jQuery (二)
题目 1)	AJAX 应用和传统 Web 应用有什么不同？
	在传统的 Javascript 编程中，如果想得到服务器端数据库或文件上的信息，或者发送客户端信息到服务器，需要建立一个 HTML form 然后 GET 或者 POST 数据到服务器端。用户需要点击“Submit”按钮来发送或者接受数据信息，然后等待服务器响应请求，页面重新加载。因为服务器每次都会返回一个新的页面，所以传统的 web 应用有可能很慢而且用户交互不友好。 使用 AJAX 技术，就可以使 Javascript 通过 XMLHttpRequest 对象直接与服务器进行交互。通过 HTTP Request，一个 web 页面可以发送一个请求到 web 服务器并且接受 web 服务器返回的信息(不用重新加载页面)，展示给用户的还是通一个页面，用户感觉页面刷新，也看不到到 Javascript 后台进行的发送请求和接受响应。
题目 2)	介绍一下 XMLHttpRequest 对象的常用方法和属性。
	open(“method”, “URL”) 建立对服务器的调用，第一个参数是 HTTP 请求 方式可以为 GET, POST 或任何服务器所支持的您想调用的方式。 第二个参数是请求页面的 URL。 send()方法，发送具体请求 abort()方法，停止当前请求 readyState 属性 请求的状态 有 5 个可取值 0=未初始化，1=正在加载 2=以加载，3=交互中，4=完成 responseText 属性 服务器的响应，表示为一个串 reponseXML 属性 服务器的响应，表示为 XML status 服务器的 HTTP 状态码，200 对应 ok 400 对应 not found

题目 3)	AJAX 请求总共有多少种 CALLBACK。
	<p>Ajax 请求总共有八种 Callback</p> <p>onSuccess</p> <p>onFailure</p> <p>onUninitialized</p> <p>onLoading</p> <p>onLoaded</p> <p>onInteractive</p> <p>onComplete</p> <p>onException</p>
题目 4)	介绍一下 Prototype 的\$()函数, \$F()函数, \$A()函数都是什么作用
	<p>\$() 方法是在 DOM 中使用过于频繁的 document.getElementById() 方法的一个便利的简写, 就像这个 DOM 方法一样, 这个方法返回参数传入的 id 的那个元素。</p> <p>\$F()函数是另一个大受欢迎的“快捷键”, 它能用于返回任何表单输入控件的值, 比如 textbox, drop-down list。这个方法也能用元素 id 或元素本身做为参数。</p> <p>\$A()函数能把它接收到的单个的参数转换成一个 Array 对象。</p>
题目 5)	AJAX 的全称是什么? 介绍一下 AJAX?
	<p>AJAX 的全称是 Asynchronous JavaScript And XML.</p> <p>AJAX 是 2005 年由 Google 发起并流行起来的编程方法, AJAX 不是一个新的编程语言, 但是它是一个使用已有标准的新的编程技术。</p> <p>使用 AJAX 可以创建更好, 更快, 更用户界面友好的 Web 应用。</p> <p>AJAX 技术基于 Javascript 和 HTTP Request.</p>

章节	Ch11 - 投票系统 (一) MVVM
题目 1)	什么是 MVVM?
	<p>MVVM 是 Model-View-ViewModel 的简写。</p> <p>相对于 MVC 的历史来说, MVVM 是一个相当新的架构, MVVM 最早于 2005 年被微软的 WPF 和 Silverlight 的架构师 John Gossman 提出, 并且应用在微软的软件开发中。</p>
题目 2)	MVVM 有哪些优点?
	<p>MVVM 模式和 MVC 模式一样, 主要目的是分离视图(View)和模型(Model), 有几大优点</p> <ol style="list-style-type: none"> 1. 低耦合。视图(View)可以独立于 Model 变化和修改, 一个 ViewModel 可以绑定到不同的 "View" 上, 当 View 变化的时候 Model 可以不变, 当 Model 变化的时候 View 也可以不变。 2. 可重用性。你可以把一些视图逻辑放在一个 ViewModel 里面, 让很多 view 重用这段视图逻辑。 3. 独立开发。开发人员可以专注于业务逻辑和数据的开发(ViewModel), 设计人员可以专注于页面设计, 使用 Expression Blend 可以很容易设计界面并生成 xml 代码。 4. 可测试。界面素来是比较难于测试的, 而现在测试可以针对 ViewModel 来写。
题目 3)	MVVM 与 MVP 有什么区别?
	<p>MVVM 模式将 Presenter 改名为 ViewModel, 基本上与 MVP 模式完全一致。</p> <p>唯一的区别是, 它采用双向绑定 (data-binding): View 的变动, 自动反映在 ViewModel, 反之亦然。Angular 和 Ember 都采用这种模式。</p>
题目 4)	在 MVVM 设计模式中, 组件变成了 Model-View-ViewModel。MVVM 有哪两个规则?

	View 持有 ViewModel 的引用, 反之没有 ViewModel 持有 Model 的引用, 反之没有
题目 5)	MVVM 模式的组成
	<p>Model Model 是指任何一个领域模型(domain model), 它代表了真实情况的内容 (一个面向对象的方法), 或表示内容 (以数据为中心的方法) 的数据访问层。</p> <p>View 就像在 MVC 和 MVP 模式下, View 就是用户界面 (UI) 。</p> <p>Viewmodel ViewModel 是一个公开公共属性和命令的抽象的 view。取代了 MVC 模式的 controller, 或 MVP 模式的任命者(presenter), MVVM 有一个驱动。在 viewmodel 中, 这种驱动传达视图和数据绑定的通信。此 viewmodel 已被描述为该 model 中的数据的状态。</p> <p>Binder 机制 MVVM 模式中包含了说明性数据和命令绑定机制。微软的做法中, binder 是一种叫 XAML 的标记语言。binder 使得开发人员不需要去编写样板化的代码来同步视图模型和视图。如果在微软解决方案库外执行实施成功了, 那么这种数据绑定机制就是促成这个模式得以发展的关键性因素。</p>

章节	Ch12 - 投票系统 (二) MVVM
题目 1)	实例化 servlet 有几种方式?
	<p>有两种方式</p> <p>a、第一次请求时, 实例化 servlet 对象</p> <p>b、在 web.xml 文件中, 在 servlet 标签中添加 loadon-startup 标签, tomcat 启动时就会实例化 servlet 对象。</p>
题目 2)	JSP 乱码如何解决?
	<p>a、JSP 页面乱码 <%@page contentType="text/html;charset=utf-8" %></p> <p>b、表单提交时出现乱码 request.setCharacterEncoding("utf-8");</p> <p>c、数据库出现乱码 jdbc:mysql://localhost:3306:/user?useSSL=false&useUnicode=true&characterEncoding=utf-8;</p> <p>其实我一般的处理的方法就是配置一个过滤器对每个 JSP 页面进行字符集处理。</p>
题目 3)	session 和 application 的区别?
	<p>a、两者的作用范围不同 session 是用户级别的, application 是 web 应用程序级别的, 是一个全局作用域。 一个用户一个 session, 每个用户的 session 不同, 用户所访问的网站多个页面共享同一个 session。 一个 web 应用程序一个 application 对象, 每个 web 应用程序的 application 对象不同, 但是一个 web 应用程序的多个用户之间共享一个 application。</p> <p>b、生命周期不同 session: 用户首次在网站中发出请求创建 session, 用户注销、离开网站或者关闭浏览器消</p>

	亡。 application: 启动 web 服务器创建, 关闭 web 服务器销毁。
题目 4)	什么是重量级? 什么是轻量级?
	轻量级是指它的创建和销毁不需要消耗太多的资源, 意味着可以在程序中经常创建和销毁 session 对象。 重量级意味着不能随意的创建和销毁它的实例, 会占用太多的资源。
题目 5)	过滤器有哪些作用?
	a、可以验证客户是否来自可信的网络 b、可以对客户提交的数据进行重写编码 c、可以从系统里获得配置的信息 d、可以过滤掉客户某些不应该出现的词汇 e、验证客户是否登录 f、验证客户的浏览器是否支持当前的应用 g、可以记录系统的日志

章节	Ch13 - 投票系统 (三) MVVM
题目 1)	开发中都用到了哪些设计模式? 用在什么场合?
	每个模式都描述了一个在我们的环境中不断出现的问题, 然后描述了该问题的解决方案的核心。 通过这种方式, 你可以无数次的使用 哪些已有的解决方案, 无需再重复相同的工作。主要用到了 MVC 设计模式。 用来开发 JSP/Servlet 或者 J2EE 的相关应用。简单工厂模式等。
题目 2)	J2EE 常用的设计模式? 说明工厂模式。
	能够说出五个就好) Factory (工厂模式) Singleton (单例模式) Adapter (适配器模式) Proxy (代理模式) Bridge (桥梁模式) 答:Java 中一共有 23 中设计模式: Factory (工厂模式)、Builder (建造模式)、Factory Method (工厂方法模式)、ProtoType (原始模型模式)、Strategy (策略模式) Singleton (单例模式)、Facade (门面模式)、Adapter (适配器模式)、Bridge (桥梁模式)、Composite (合成模式)、 Decorator (装饰模式)、FlyWeight (享元模式)、Proxy (代理模式)、Command (命令模式)、Interpreter (解释器模式)、 Visitor (访问者模式)、Iterator (迭代子模式)、Mediator (调停者模式)、Memento (备忘录模式)、Observer (观察者模式)、 State (状态模式)、Strategy (策略模式)、Template Method (模板方法模式)、Chain Of Responsibility (责任链模式)、 工厂模式: 工厂模式是一种经常被使用到的模式, 根据工厂模式实现的类可以根据提供的数据生成一组类中某个类的实例, 通常一组类中有一个公共的抽象父类并且实现了相同的方法, 但是这些方法针对不同的数据进行了不同的操作。 首先需要定义一个基类, 该类的子类通过不同的方法实现了基类中的方法。然后定义一个工厂类, 工厂类可以根据条件

	生成不同的子类实例。当得到子类的实例后，开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子类的实例。
题目 3)	说一下 MVVM 设计模式
	<p>MVVM (Model-View-ViewModel)</p> <p>MVVM 它是将“数据模型数据双向绑定”的思想作为核心，因此在 View 和 Model 之间没有联系，通过 ViewModel 进行交互，</p> <p>而且 Model 和 ViewModel 之间的交互是双向的，因此视图的数据的变化会同时修改数据源，而数据源数据的变化也会立即反应到 View 上。</p> <p>这方面典型的应用有.NET 的 WPF，js 框架 Knockout、AngularJS 等。</p>
题目 4)	web.xml 的作用？
	用于配置 Web 应用的相关信息，如：监听器 (listener)、过滤器 (filter)、Servlet、相关参数、会话超时时间、安全验证方式、错误页面等
题目 5)	双向数据绑定是什么意思
	<p>双向数据绑定指的就是，绑定对象属性的改变到用户界面的变化的能力，反之亦然。换种说法，如果我们有一个 user 对象和一个 name 属性，一旦我们赋了一个新值给 user.name,在 UI 上就会显示新的姓名了。同样地，如果 UI 包含了一个输入用户姓名的输入框，输入一个新值就应该会使 user 对象的 name 属性做出相应的改变。</p>