**中软国际考题-笔试**

1. 单选题

1、在JSP 中使用<jsp:getProperty> 标记时，不会出现的属性是(C )。

A．property   
B．name   
C．value   
D．以上皆不会出现

2、请问以下哪个选项是正确的脚本元素？（A）

A、<% int x;%>

B、<% private int x;%>

C、<%=x%>

<%=x;%>

3、当多个用户请求同一个JSP页面时，Tomcat服务器为每个客户启动一个（B）

A）进程  **B）线程** C）程序 D服务。

4、当用户请求jsp页面时，jsp引擎就会执行该页面的字节码文件响应客户的请求，执行字节码文件的结果是：（C）

A. 发送一个jsp源文件到客户端

B. 发送一个java文件到客户端

C. 发送一个html页面到服务端

D. 什么都不做

5、下列哪一个不是EL定义的隐式对象（C）

A. cookie B. pageContext C. attributes D. initParam

6、Http请求模式中缺省的模式是（B）

A. get B. post C. put D. delete

B. post C. put D. delete

7、在JSP页面中，能够完成输出操作的内置对象是  ( A )

   A.  out

   B.  response

   C.  request

   D.  config

8、给定以下JSP代码片段，有2个客户依次浏览该JSP;且每个客户只浏览一次，第2个客户会看到浏览器显示（ A ）。

<%int x = 1; %>

<%!int x = 10; %>

X =<%=x%>

   A.  x=1

   B.  x=2

   C.  x=10

   D.  x=11

9． 在JSP中，若要在JSP正确使用标签：<x:getKing/>，在jsp中声明的taglib指令为：<%@tagliburi = “/WEB-INF/myTags.tld” prefix = “\_\_A\_\_\_”%>，下划线处应该是

   A.  x

   B.  getKing

   C.  myTags

   D.  king

10． 能够保留请求作用域中数据的转向方式是\_\_\_\_\_\_\_\_\_D\_\_\_\_\_

   A.  response.forward()

   B.  response.sendRedirect()

   C. session.getRequestDispatcher().forward(request,response)

   D. request.getRequestDispatcher().forward(request,response)

11.某JSP中有如下代码，显示结果为（B ）

<%

         int a = 5;

         request.setAttribute(“a”,”123”);

         session.setAttribute(“a”,”456”);

%>

<c:out value="${a}"/>

   A.  5

   B.  123

   C.  456

   D.  null

12、下列选项中关于HttpSession描述错误的是（ D ）。

A. HttpSession通过HttpServletRequest对象获得

B.HttpSession可以用来保存数据，并实现数据的传递

C.HttpSession被创建后，将始终存在，直到服务停止

D.调用HttpSession的invalidate()方法，可以删除创建的HttpSession对象及数据

13、如果某一JSP页面的表单中，有几个复选框，name为“habit”,则该jsp提交后，通过下面（ C ）语句去获取用户选中复选框的值。

A. request.getAttribute(“habit”);

B. request.getParameter(“habit”);

C. request.getParameterValues(“habit”);

D. request.getHabit();

14、假设在helloapp应用中有一个hello.jsp，文件路径为C:/Tomcat/webapps/helloapp/hello/hello.jsp那么在浏览器端访问hello.jsp的URL是( C )。

A. http://localhost:8080/hello.jsp

B. http://localhost:8080/helloapp/hello.jsp

C. http://localhost:8080/helloapp/hello/hello.jsp

D. http://localhost:8080/helloapp/hello

15、要使用request获取所有同名的参数值，则需要调用（C）函数：

A. getParameterValue B. getParameterNames

C. getParameterValues D. getAttribute

16、page、request、session、application的生命期从长到短为（C）

A. page、request、session、application

B. session、page、request、 application

C. application、session、request、page

D. request、session、 page、application

17、page指令用于定义JSP文件中的全局属性，下列关于该指令用法的描述不正确的是（D）。

A. <%@ page %>作用于整个JSP页面。

B. 可以在一个页面中使用多个<%@ page %>指令。

C. 为增强程序的可读性，建议将<%@ page %>指令放在JSP文件的开头，但不是必须的。

D. <%@ page %>指令中的属性只能出现一次。

1. 多选题
2. 对于<jsp:param>动作，描述正确的是？（AB）

A、< jsp:param >是<jsp:include>、<jsp:forward>标记的子标记

B、如果有aa.jsp代码：

<jsp:forward page="next.jsp">

<jsp:param name="name" value="jb-aptech"/>

</jsp:forward>,则在next.jsp中可以使用request.getParameter(“name”);把属性name的值取出来

C、如果有aa.jsp代码：

<jsp:forward page="next.jsp">

<jsp:param name="name" value="jb-aptech"/>

</jsp:forward>,则在next.jsp中可以使用request.geAttribute(“name”);把属性name的值取出来

D、如果<jsp:param>标记不放在<jsp:forward>标记内，也就是不作为 <jsp:forwrd>

的子标记，则使用浏览器查看时会显示错误页面

2、请问以下哪些选项是JSP的内置对象？（A、D）

A、response

B、req

C、HttpServletRequest

D、request

3、下列哪一项属于JSP动作指令标记？（A B C）

A）<jsp:param> B）<jsp:plugin> C）<jsp:useBean> D）<jsp:javaBean>

4、下列说法正确的是：（A、B、D、E）

A. include指令通知容器将当前的jsp页面中内嵌的、在指定位置上的资源内容包含

B. include指令中file属性指定要包含的文件名

C. include指令只允许你包含动态页面

D. taglib指令允许页面使用自定义标签

E. 你必须在使用自定义标签之前使用<% @taglib %>指令

4、下列属于JSP脚本元素的表示方式的是（ACD）

A. <% %> B. <%@ %> C. <%! %> D. <%= %>

5、下列属于JSP的标准指令的是（BCD）

A. import B. include C. page D. taglib

6、Login.jsp为登录页面，表单代码如下：

<form action="index.jsp"method="post">

<input type="text" name="name"/>

<input type="submit" value="login"/>

</form>

Index.jsp中直接显示用户名，以下哪种代码正确（ C、D）

A. ${requestScope.name}

B. <%=name%>

C. ${param.name}

D. <%=request.getParameter(“name”)%>

7、下面关于转发与重定向错误的是( AD )。【选两项】

A. 转发后地址栏中会显示新地址

B. 转发后地址栏中会显示旧地址

C. 重定向会产生一个新的请示

D. 转发会产生一个新的请求

8、page指令元素的特点有（BC）

A.page指令元素中的属性都只能出现一次。

B.通过page指令元素可以导入需要的包。

C.通过page指令元素可以设置session内置对象是否有效。

D.通过page指令元素可以指定JSP页面所使用的标签库。

9、在J2EE中，以下是JSP隐式对象的是( ACD )。

A. pageContext B. context

C. application D. out

1. 填空题
2. JSP有三种指令，它们分别是：\_\_\_\_\_include \_\_\_\_\_、\_\_\_\_\_ page\_\_\_\_\_、\_\_\_\_\_ taglib\_\_\_\_\_
3. 表达式用于向页面输出信息，其使用格式是以\_\_\_\_\_\_\_\_\_\_标记开始，以\_\_\_\_\_\_\_\_\_标记结束。(<%=、%>)
4. JSP开发网站的两种模式分别为 JSP+JavaBean 和 JSP+Servlet+JavaBean
5. 在jsp中为内置对象定义了四种作用范围，即\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_四个作用范围。（page、request、session、application）
6. 对于每一个客户端都要访问的变量，应该将它设为\_\_\_\_\_\_\_\_\_\_\_变量。(application范围)
7. request.setAttribute("user","svse"); session.setAttribute("user","bob");

使用{user}则显示\_\_\_\_\_\_\_\_\_\_ 。(svse)

1. 一个完整的JSP页面是由普通的HTML标记、JSP指令标记、JSP动作标记、变量和方法的声明、程序片、表达式、注释 7种要素构成。
2. JSP中提供的内置对象有out、page、request、response、session、application、

config、exception、pageContext

1. 要在JSP文件中使用标准标签库，需要在开始位置添加<%@ taglib prefix=”c” uri=”http://java.sun.com/jsp/jstl/core”%>
2. 要设置某个JSP页面为错误处理页面的page指令是 。（<%@ page isErrorPage="true"%>）
3. 判断题
4. jsp:include动作标记与include指令标记包含文件的处理时间和方式不同（T）
5. 程序片变量不同于在“<%!”和“%>”之间声明的页面成员变量，不能在不同客户访问页面的线程之间共享。（T）
6. 标记的page属性值是相对的URL地址，只能静态的URL。（F）（可以不是）
7. 页面指令的import属性可出现多次。（ T ）
8. 当用户请求JSP页面时，JSP引擎就会执行该页面的字节码文件响应客户的请求，执行字节码文件的结果是发送一个HTML页面到客户端。（T）
9. JSP的运行需要依赖于操作系统，JRE以及web服务器软件，例如：Tomcat. ( T )
10. Tomcat服务器的默认端口80 ( F )
11. JSP可以通过request.getParameter(“参数名”) 获取参数的值，返回值为String类型。 （ T ）
12. JSP中可以在<%%>中插入java程序片段，例如：对象的创建等. ( T )
13. JSP允许使用plugin指令来自定义标签库。 ( F )
14. 在page指令中，import参数允许多次重复使用。（T）
15. JSP被称为Java服务器页面，其根本是一个简化的Servlet设计。（ T ）
16. JSP文件其实就是html5文件。（ F ）
17. HttpServletRequest接口的putSession方法用于创建会话。（ X ）
18. cookie对象是由服务器端创建的，由客户端存储的。（正确）
19. <%@%>是jsp中指令标记。（ T ）
20. JSP中导入某个类的指令是<%@ page import=”包名.类名”%>。( T )
21. JSP中有个隐含的默认对象request，它的类型 是RequestDipatcher。（ F ）
22. <%@ page %>指令中的属性只能出现一次。（ F ）
23. 访问JSP页面，需要预编译成Servlet。( T )
24. <jsp:foward>标签可以转发request。 ( T )
25. response.sendRedirect()和<jsp:foward>都不可以转发request。 （ F ）
26. jsp:forward地址栏变化，可以跳转到任何页面和机器。（ F ）

五、简答题

# 1、jsp有哪些动作?作用分别是什么?？

答案：JSP共有以下6种基本动作

jsp:include：在页面被请求的时候引入一个文件。

jsp:useBean：寻找或者实例化一个JavaBean。

jsp:setProperty：设置JavaBean的属性。

jsp:getProperty：输出某个JavaBean的属性。

jsp:forward：把请求转到一个新的页面。

jsp:plugin：根据浏览器类型为Java插件生成OBJECT或EMBED标记。

2、简要说明Servlet和JSP分别有什么作用？（5分）

答案：JSP的本质就是Servlet，JSP和Servlet都可以生成动态页面。然而，Servlet生成动态页面比较复杂，不管静态还是动态的内容都需要用代码输出。JSP相对简单很多，JSP中静态内容使用HTML及文本即可，只有动态 内容使用Java代码生成。Servlet往往用来接收JSP的请求，获取请求数据，进行业务处理。

3、简要说明Servlet和JSP之间跳转有哪两种方式？有什么区别？（5分）

答案：有两种跳转方式，一种是响应重定向，即调用响应接口的sendRedirect方法，一种是请求转发，即调用请求转发器的forward方法。其中响应重定向相当于重新发出一次请求，之前的请求对象不会发到下一个资源；请求转发会于当前的请求对象转发到下一个资源，实际中请求转发使用较多。

4、简要说明如果要在请求范围内共享数据，如何实现？（5分）

答案：可以使用请求属性实现。请求接口中有setAttribute方法，可以将任何类型对象作为属性存储，同时提供方法getAttribute可以读取请求属性。往往会在一个组件中存属性，在另一个组件中获取使用属性，那么二者跳转的时候一定要用请求转发跳转，否则将生成一个新的请求对象，就无法获取之前存储的属性。

5、JSP的内置对象及方法。

request表示HttpServletRequest对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取cookie, header, 和session数据的有用的方法。

response表示HttpServletResponse对象，并提供了几个用于设置送回浏览器的响应的方法（如cookies,头信息等）

out对象是javax.jsp.JspWriter的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

pageContext表示一个javax.servlet.jsp.PageContext对象。它是用于方便存取各种范围的名字空间、servlet相关的对象的API，并且包装了通用的servlet相关功能的方法。

session表示一个请求的javax.servlet.http.HttpSession对象。Session可以存贮用户的状态信息

applicaton 表示一个javax.servle.ServletContext对象。这有助于查找有关servlet引擎和servlet环境的信息

config表示一个javax.servlet.ServletConfig对象。该对象用于存取servlet实例的初始化参数。

page表示从该页面产生的一个servlet实例

6、JSP中动态INCLUDE与静态INCLUDE的区别？

答：动态INCLUDE用jsp:include动作实现,它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数;静态INCLUDE用include伪码实现,定不会检查所含文件的变化，适用于包含静态页面.

7、两种跳转方式分别是什么有什么区别

答：在JSP中跳转有两种方式

forward跳转：<jsp:forward page ="跳转页面地址">

response跳转：response.sendRedirect("跳转页面地址");

两种跳转的区别如下:

forward跳转：服务器端跳转，地址栏不改变。执行到跳转语句后马上执行跳转，之后的代码不再执行（跳转之前一定要释放全部资源）,request设置的属性在跳转后的页面仍然可以使用。

response跳转：客户端跳转，地址栏改变。,所有代码执行完毕后跳转。

跳转后页面不能使用上一个页面的request.使用地址重写传递参数（response.sendRedirect("URL?参数名=参数值")）。

8、Request对象的主要方法：[列举6个]setAttribute(String name,Object)：设置名字为name的request的参数值  
getAttribute(String name)：返回由name指定的属性值  
getCookies()：返回客户端的所有Cookie对象，结果是一个Cookie数组  
getCharacterEncoding()：返回请求中的字符编码方式  
getHeader(String name)：获得HTTP协议定义的文件头信息  
getHeaders(String name)：返回指定名字的request Header的所有值，结果是一个枚举的实例  
getHeaderNames()：返回所以request Header的名字，结果是一个枚举的实例  
getParameter(String name)：获得客户端传送给服务器端的有name指定的参数值  
getParameterNames()：获得客户端传送给服务器端的所有参数的名字，结果是一个枚举的实例  
getParameterValues(String name)：获得有name指定的参数的所有值  
getRequestURI()：获取发出请求字符串的客户端地址  
getRemoteAddr()：获取客户端的IP地址  
getRemoteHost()：获取客户端的名字  
getSession([Boolean create])：返回和请求相关Session  
getServerName()：获取服务器的名字  
getServletPath()：获取客户端所请求的脚本文件的路径  
getServerPort()：获取服务器的端口号  
removeAttribute(String name)：删除请求中的一个属性

9、描述Cookie和Session的作用，区别和各自的应用范围，Session工作原理。

Cookie:主要用在保存客户端，其值在客户端与服务端之间传送，不安全，存储的数据量有限。Session:保存在服务端，每一个session在服务端有一个sessionID作一个标识。存储的数据量大，安全性高。占用服务端的内存资源。

10、简述JSP的生命周期（5分）

答案：

编译阶段：servlet容器编译servlet源文件，生成servlet类

初始化阶段：加载与JSP对应的servlet类，创建其实例，并调用它的初始化方法

执行阶段：调用与JSP对应的servlet实例的服务方法

销毁阶段：调用与JSP对应的servlet实例的销毁方法，然后销毁servlet实例。

11、列举常用的JSP内置对象，描述它们的作用。

答案：Request：表示HttpServletRequest对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取cookie,header,和session数据的有用的方法。用户端请求，此请求会包含来自GET/POST请求的参数

Respons：表示HttpServletResponse对象，并提供了几个用于设置送回浏览器的响应的方法（如cookies,头信息等）网页传回用户端的回应

Out：对象是javax.jsp.JspWriter的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。用来传送回应的输出

pageContext：表示一个javax.servlet.jsp.PageContext对象。它是用于方便存取各种范围的名字空间、servlet相关的对象的API，并且包装了通用的servlet相关功能的方法。网页的属性是在这里管理

Session：表示一个请求的javax.servlet.http.HttpSession对象。Session可以存贮用户的状态信息。与请求有关的会话期

Application：表示一个javax.servle.ServletContext对象。这有助于查找有关servlet引擎和servlet环境的信息。 servlet正在执行的内容

Config：表示一个javax.servlet.ServletConfig对象。该对象用于存取servlet实例的初始化参数。servlet的构架部件

Page：表示从该页面产生的一个servlet实例。JSP网页本身

Exception:针对错误网页，未捕捉的例外

12、JSP中指令元素共有几个

答案： page指令：设置一些JSP页面的信息（比如编码，缓存大小，导包等）

include指令：静态包含其他资源

taglib指令：引入第三方的标签库

13、include标准动作与include指令的区别

答案：include标准动作与include指令都是实现包含其他页面的功能，但是include标准动作的属性是page，实现动态包含，发生在请求阶段；而include指令的属性是file，实现静态包含，发生在翻译阶段。

六、程序实现题

1、编写程序reg.htm和reg.jsp，reg.html做一用户注册界面，包括：用户名，年龄，性别。然后提交到reg.jsp进行注册检验，若用户名为admin，就提示“欢迎你，管理员”，否则，显示“注册成功”并显示出注册信息

|  |
| --- |
| Reg.htm参考代码  <%@ page contentType="text/html;charset=GBK" %>  <html>  <body>  <form method=post action="reg.jsp">  用户名：  <input type=text name="name"><br>  年龄：  <input type=text name="age"><br>  性别：  <input type=text name="sex"><br>  <input type=submit value="submit">  </form>  </body>  </html>  Reg.jsp参考代码  <%@ page contentType="text/html;charset=GBK" %>  <html>  <head>  <title>注册检验</title>  </head>  <body>  <%  if(request.getParameter("name").equals("admin"))  out.println("欢迎你，管理员");  else  {  out.println("注册成功"+"<br>");  out.println("用户名:"+ request.getParameter("name")+"<br>");  out.println("年龄:"+ request.getParameter("age")+"<br>");  out.println("性别:"+ request.getParameter("sex")+"<br>");  }  %>  </body>  </html> |

2、使用JSP来计算特定页面访问的总人数。

（10分）

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ page import="java.io.\*,java.util.\*" %>  <html>  <html>  <head>  <title>访问量统计</title>  </head>  <body>  <%  Integer hitsCount = (Integer)application.getAttribute("hitCounter");  if( hitsCount ==null || hitsCount == 0 ){  /\* 第一次访问 \*/  out.println("欢迎访问XX网站!");  hitsCount = 1;  }else{  /\* 返回访问值 \*/  out.println("欢迎再次XX网站!");  hitsCount += 1;  }  application.setAttribute("hitCounter", hitsCount);  %>  <p>页面访问量为: <%= hitsCount%></p>  </body>  </html> |

3、已知JavaBean定义如下，假设已经在请求范围内存在一个泛型集合为list，存储的数据为Account，请使用jstl+el编辑jsp页面代码实现显示集合数据:

**package** com.etc.bean;

**import** java.io.Serializable;

/\*\*

\* Account bean定义

\* **@author** knowno

\*

\*/

**public** **class** Account **implements** Serializable{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 5659939915458771174L;

**private** String accountName;

**private** String accountPwd;

**public** String getAccountName() {

**return** accountName;

}

**public** **void** setAccountName(String accountName) {

**this**.accountName = accountName;

}

**public** String getAccountPwd() {

**return** accountPwd;

}

**public** **void** setAccountPwd(String accountPwd) {

**this**.accountPwd = accountPwd;

}

}

参考答案：

<%@ page language="java" import="java.util.\*" pageEncoding="utf-8"%>

<%@ taglib prefix="c" uri="<http://java.sun.com/jsp/jstl/core>"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

  <head>

  </head>

  <body>

      <c:forEach items="${list}"  var="account">

            ${account.accountName},${account.accountPwd}<br/>

      </c:forEach>

  </body>

</html>

4、使用application内置对象实现一个计数器功能（程序的访问人数）

（10分）

|  |
| --- |
| <@% page import=”java.util.\*” contentType=”text/html;charset=utf-8” pageEncoding=”utf-8”%>  <html>  <body>  <% int n=0;  String counter=(String)application.getAttribute(“counter”);  If(counter!=null)  n=Integer.parseInt(counter);  n+=1;  out.print(“you are the”+n+”vister”);  counter=String.valueOf(n);  application.setAttribute(“counter”,counter);  %>  </body>  </html> |

5、、程序实现题，使用JSTL标签与EL表达式提取请求中商品数据，并对其数据进行分页显示。（10分）

已知数据格式：序号、商品、单价、数量、生产日期

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <html>  <head>  <title>c:forEach 标签实例</title>  </head>  <body>  <table>  <c:forEach var="goods" varStatus=”st” items=”${goodList}” begin="${start}" end="${end}">  <tr>  <td>${st.count}</td>  <td>${goods.name}</td>  <td>${goods.price}</td>  <td>${goods.total}</td>  <td>${goods.productionTime}</td>  </tr>  </c:forEach>  <table  </body>  </html> |

6、实现用户登录功能，三个页面登录页面login.jsp,登录逻辑处理页面loginSubmit.jsp，欢迎页面welcome.jsp.用户再登录页面输入用户名和密码，前台页面使用js进行验证，如果用户名或密码为空，则提示用户输入用户名和密码;如果用户名为”admin”，密码为”123”,则登录成功跳转至欢迎页面,欢迎页面显示当前用户登录的用户名，例如:“欢迎admin登录”,如果失败，停留再登录页面,继续登录。

|  |
| --- |
| login.jsp:  <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <base href=*"*${pageContext.request.contextPath}*/"*/>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>Insert title here</title>  <script type=*"text/javascript"*>  **function** validForm() {  **var** username=document.getElementById("username").value;  **var** password=document.getElementById("password").value;  alert("username="+username);  alert("password="+password);  **if**(username==""){  document.getElementById("err\_username").innerHTML="用户名不能为空";  **return** **false**;  }  **else** **if**(password==""){  document.getElementById("err\_username").innerHTML=""  document.getElementById("err\_password").innerHTML="密码不能为空";  **return** **false**;  }**else**{  **return** **true**;  }  }  </script>  </head>  <body>  <form action=*"loginSubmit.jsp"* method=*"post"* onsubmit="return validForm()">  用户名:<input type=*"text"* name=*"username"* id=*"username"*/>  <font id=*"err\_username"* color=*"red"*></font><br/>  密码:<input type=*"password"* name=*"password"* id=*"password"*/>  <font id=*"err\_password"* color=*"red"*></font>  <br/>  <input type=*"submit"* value=*"提交"*/><br/>  </form>  </body>  </html> |
| loginSubmit.jsp:  <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <base href=*"*${pageContext.request.contextPath}*/"*/>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>Insert title here</title>  </head>  <body>  <%  String username=request.getParameter("username");  String password=request.getParameter("password");  **if**(username.equals("admin") && password.equals("123")){  session.setAttribute("username", username);  response.sendRedirect(request.getContextPath()+"/welcome.jsp");  }**else**{  response.sendRedirect(request.getContextPath()+"/login.jsp");  }  %>  </body>  </html> |
| welcome.jsp:  <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <base href=*"*${pageContext.request.contextPath}*/"*/>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>Insert title here</title>  </head>  <body>  欢迎${username}登录  </body>  </html> |

7、编写一个JSP页面，要求用户输入自己的身份证号，提交后在页面上输出该用户身份证号

|  |
| --- |
| <%@ page language=”java” contentType="text/html; charset=UTF-8"%> <html>  <head><title>请入身份证号码 </title></head>  <form name="form1" method="post" action="card.jsp">  请输入您的身份证号码：<input type="text" name="IDCard" size="18"/>  <input type="submit" value="提交">  <form>  </html>  ----------------------------------------------------------------------  card.jsp:  <%@ page language="java" contentType="text/html; charset=UTF-8"%> <html>  <head><title> 身份证提交 </title></head>  <body>  <%  String IDCard=request.getParameter("IDCard");  %>  您的身份证号码是:<%=IDCard%>  </body>  </html> |

8、编写过滤器，实现设置编码格式为utf-8。

答案：

1、定义过滤器类：

public class EncodingFilter implements Filter {

private String encode;

public void destroy() {

}

public void doFilter(ServletRequest arg0, ServletResponse arg1,

FilterChain arg2) throws IOException, ServletException {

arg0.setCharacterEncoding(encode);

arg1.setContentType("text/html; charset="+encode);

arg2.doFilter(arg0, arg1);

}

public void init(FilterConfig arg0) throws ServletException {

encode = arg0.getInitParameter("encode");

}

}

2、在web.xml中配置过滤器

<filter>

<filter-name>encodFilter</filter-name>

<filter-class>com.rn.filter.EncodingFilter</filter-class>

<init-param>

<param-name>encode</param-name>

<param-value>utf-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

9、编写一段代码 读取Cookie,以获取account的值，并输出account的值（10分）

|  |
| --- |
| Cookie[] cookies = request.getCookies(); if(cookies != null) { for(int i = 0 ; i < cookies.length ; i++) { if(cookies[i].getName().equals("account")){ out.println(cookies[i].getValue()); } } |