**中软国际考题-笔试**

1. 单选题
2. Servlet程序的入口点是(A )。
3. init（）  
   B. main（）  
   C. service（）  
   D. doGet（）

2、下列选项中，可以准确地获取请求页面的一个文本框的输入（文本框的名称为 name）。（ B ）

A.request.getParameter（name）

B.request.getParameter（"name"）

C.request.getParameterValues(name)

D.request.getParameterValues("name")

3、HttpServletRequest对象是由谁创建的? （A）

A、由Servlet容器负责创建，对于每个HTTP请求， Servlet容器都会创建一个HttpServletRequest对象

B、由JavaWeb应用的Servlet或JSP组件负责创建， 当Servlet或JSP组件响应HTTP请求时，先创建 HttpServletRequest对象

C 由程序员自己创建

D 由Servlet创建，于每个HTTP请求， Servlet都会创建一个HttpServletRequest对象

4、下面哪一项对Servlet描述错误？（C）

A. Servlet是一个特殊的Java类，它必须直接或间接实现Servlet接口

B. Servlet接口定义了Servelt的生命周期方法

C. 当多个客户请求一个Servlet时，服务器为每一个客户启动一个进程

D. Servlet客户线程调用service方法响应客户的请求

5、在Java EE的Web应用中，编译后的class文件存放的目录（A）

A. /WEB-INF/classes/ B. /WEB-INF/libs/ C. /WEB-INF/src/ D. 根目录

6、在Servlet生命周期中，对应服务阶段的方法是（C）

A. init() B. destroy() C. service() D. requestHandler()

7、通过response的（ A ）方法可以设置响应所采用的字符编码类型。

A. setCharacterEncoding()

B. setCharset()

C. setEncoding()

D. setContentType()

8、编写一个Filter，需要（ B ）

A. 继承Filter 类

B. 实现Filter 接口

C. 继承HttpFilter 类

D. 实现HttpFilter接口

9.自定义标签的配置文件放在(D)

A. WebRoot

B. lib

C. classes

D. WEB-INF

10.自定义标签的作用是 ( C )

A. 编写和使用方便

B. 规定是这样的，如果不用，别人会说我们不专业

C. 可以减少jsp中的java代码,将代码与界面标签分离,简化前台开发

D. 连数据库

11． 在J2EE中，重定向到另一个页面，以下（ C ）语句是正确的

A. request . sendRedirect(“http :// www . svse . com . cn”);

B. request . sendRedirect();

C. response . sendRedirect(“http: // www . svse . com . cn”);

D. response .sendRedirect();

12． EL表达式，${10 mod3}，执行结果为：(B)

A. 10 mod 3

B. 1

C. 3

D. null

13、给定一个Servlet的doGet方法中的代码片断，如下

request,setAttribute(“name”,”zhang”);

response.sendRedirect(“http://localhost:8080/servlet/MyServlet”);

那么在MyServlet中可以使用( D )方法把属性的值取出来。

A. String str=request.getAttribute(“name”);

B. String str =(String)request.getAttribute(“name”);

C. Object str=request.getAttribute(“name”);

D. 无法取出

14、在Severlet过滤器的生命周期方法中，每当传递请求或响应时，web容器会调用( C )方法.

A. init B. Service C. doFilter D. destroy

15、给定一个Servlet的代码片断，如下：

public void doGet(HttpServerletRequest request,HttpServletResponse response) thows ServletException,IOException{

\_\_\_\_\_\_\_\_\_\_\_

out.println(“hi kitty!”);

out.close();

}

运行此Servlet时输出如下：

hi kitty!

则应在此Servlet下划线处填充的代码是( A )。

A. PrintWriter out = response.getWriter();

B. PrintWriter out = request.getWriter();

C. OutputStream out = response.getOutputStream();

D. OutputStream out = request.getWriter();

16、Servlet 中HttpServletRequest中（A）方法用于返回应用程序上下文路径

A、getContextPath() B、getPath()

1. getRequestURL() D、getRequestURI

17、在J2EE中，转发到另一个页面，以下（ B ）语句是正确的。

A. request.getRequestDispatcher("/index.jsp");

B．request.getRequestDispatcher("/index.jsp").forward(request,response);

C. request.forward("/index.jsp");

D.request.getRequestDispatcher("/index.jsp").forward();

1. 多选
2. 下面关于Servlet的陈述正确的是？（BC）

A 我们在浏览器的地址栏直接输入要请求的Servlet，该Servlet默认会使用doPost方法处理请求

B Servlet运行在服务器端

C Servlet的生命周期概括为以下几个阶段：装载Servlet、创建一个Servlet实例、调用Servlet的init()方法、服务、销毁

D Servlet不能向浏览器发送Html标签

2、下面关于servlet的功用说法正确的有哪些？( A B C D E )

A、读取客户程序发送来的显式数据 B、读取客户程序发送来的隐式数据

C、生成相应的结果 D、发送显式的数据给客户程序

E、发送隐式的数据给客户程序

3、定义一个Servlet需要实现哪些方法？（A B C）

A）init(ServletConfig config) B）service(ServletRequest request,ServletResponse response)

C）ServletConfig getServletConfig() D）getAttribute(String args);

4、在Servlet里，能实现跳转的方法有（BC）

A. 运用javax.servlet.http.HttpServletRequest接口的sendRedirect方法

B. 运用javax.servlet.http.HttpServletResponse接口的sendRedirect方法

C. 运用javax.servlet.RequestDispatcher接口的forword方法

D. 运用javax.servlet.ResponseDispatcher接口的forword方法

4、对于servlet在程序中的作用，以下说法正确的是( AB )

A. 分离业务逻辑代码与表现层代码

B. 让程序执行流程更加清晰

C. 加快了程序访问速度

D. 大大减少了编码工作量

5、有关会话跟踪技术描述正确的是 A,B,C (多选)

A.  Cookie是Web服务器发送给客户端的一小段信息，客户端请求时，可以读取该信 息发送到服务器端

B.  关闭浏览器意味着会话ID丢失，但所有与原会话关联的会话数据仍保留在服务器 上，直至会话过期

C.  在禁用Cookie时可以使用URL重写技术跟踪会话

D.  隐藏表单域将字段添加到HTML表单并在客户端浏览器中显示

6、在web.xml中有如下代码：

<web-app>

<servlet>

<servlet-name>LoginServlet</servlet-name>

<servlet-class>jbit.servlet.LoginServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name> LoginServlet </servlet-name>

<url-pattern>/ LoginServlet </url-pattern>

</servlet-mapping>

</web-app>

下列选项描述不正确的是（ BC ）。 【选两项】

A. 在<servlet-mapping>中的<url-patten>表示用户请求访问Servlet的URL

B. Servlet容器会根据<servlet-class>查找到与其对应的<servlet-name>

C. <servlet>中的<servlet-name>内容与<servlet-mapping>中的<servlet-name>内容可以不一致

D. 以上描述都正确

7、对于ServletRequest接口的获取请求参数的方法的描述那一项是正确的（B、C）

A、getParameter方法只用于接收POST请求参数，接收GET请求参数需要使用getQueryString方法

B、如果一个参数key有多个值，那么getParameter(key)方法会返回一个值

C、如果一个参数key有多个值，那么getParameterValues(key)方法会返回一个包含所有值的字符串数组

D、 getParameter方法返回Object对象，使用前要强制类型转换：如String str=(String) request.getParameter(key)

8、以下哪些是Java EE标准组件？它们是（A 、D ）。

A.JSP B.JDK C.JFrame D.Servlet

1. 填空题
2. Servlet不同于\_\_\_ HttpServlet\_\_\_\_\_,后者是前者的一个常用扩展。Servlet一旦载入它就常驻在\_\_\_ 内存\_\_\_\_\_中，这样加快了响应的速度。
3. HttpServlet处理客户端请求的方法是：\_\_\_\_\_\_\_\_\_\_ 和\_\_\_\_\_\_\_\_\_。( doGet()、doPost() )
4. Servlet中的doGet和doPost方法的参数为：HttpServletRequest和HttpservletResponse。
5. HttpServletRequest对象是由\_Servlet容器\_\_\_\_\_\_\_创建的？
6. 过滤器Filter接口中最主要的方法是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_方法。(doFilter)
7. Servlet的生命周期有\_\_\_\_\_\_\_\_\_\_ ,\_\_\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_。(实例化,初始化,service服务,destroy销毁)
8. 在Servlet中用来接收客户表单数据的两个常用方法为doGet()、doPost()。
9. Servlet 中由\_\_\_\_\_\_\_\_\_方法决定调用doGet还是doPost (service)。
10. Servlet中，使用 类中定义的 方法来处理客户端发出的超链接请求 。(HttpServlet doGet)
11. 判断题
12. HttpServlet封装了编写基于Http协议Servlet的大部分功能（T）
13. Servlet不可以在不同应用服务器平台下进行移植（F）(可以)
14. 同一个项目中的两个Servlet之间是不可以相互交流的（F）（可以）
15. 设置Cookie可通过response对象的响应方法实现。（ T ）
16. Servlet程序运行在服务器Web容器中。（ T ）
17. Tomcat时一款免费、开源而且全面支持Java EE规范的应用服务器。（ F ）
18. Servlet的init方法是线程安全的。 （T）
19. Web [3.0可以使用@WebServlet](mailto:3.0可以使用@WebServlet)注解来配置Servlet ( T )
20. HttpSession对象不可以用来进行用户的访问控制 ( F )
21. ServletConfig对象可以获取初始化参数的值。 （ T ）
22. 调用getCreationTime（）可以获取session对象创建的时间，该时间的单位是秒 ( F )
23. 在MVC设计模式体系结构中，Servlet是实现控制器的首选方案 ( T )
24. [我们自己编写的Servlet继承了HttpServlet类，一定需覆盖doPost或者doGet](http://ptr.fy.chaoxing.com/exam/test/javascript:void(0);)（F）
25. 利用response对象的sendRedirect方法只能实现本网站内的页面跳转，但不能传递参数。（ F ）
26. Servlet的GET方法和POST方法没什么区别。（ F ）
27. 可通继承HttpServlet类的方式创建Servlet。（ √ ）
28. Servlet 创建是单例模式的并且默认创建时间是服务器启动加载项目时。（ F ）
29. 在JavaEE的过滤器中，完成过滤业务的是doFilter()方法。（ T ）
30. Servlet 可以在以下请求、会话和响应作用域存储数据。 ( F )
31. 在servlet中，下列request.getParameter(“title”);语句可以获取页面请求中一个文本框的输入（文本框的名称为title）。（ T ）
32. 每个Servlet都必须实现javax.servlet.HttpServlet接口。（ T ）
33. Cookie是一个文本文件。（ T ）
34. Cookie以Key、Value的形式记录内容。（ T ）
35. Cookie一般存储于服务器中。( F )
36. Cookie一般也有有效期。（ T ）

五、简答题

# 1、简单说一下Servlet的生命周期?？

答案：servlet有良好的生存期的定义，包括加载和实例化、初始化、处理请求以及服务结束。这个生存期由javax.servlet.Servlet接口的init,service和destroy方法表达。 Servlet的生命周期是当服务器装载运行servlets，接收来自客户端的多个请求并且返回数据给客户端，然后再删除移开servlets的时间。

2、请求接口中的哪个方法可以返回请求参数的值？哪个方法可以把请求参数的多个值同时返回？（5分）

答案：请求接口中的getParameter 方法可以返回摸一个请求参数的值，getParaneterValues方法可以把多个请求参数的多个值同时返回，封装到一个数组中，例如复选框的多个值。

3、简单介绍你了解的MVC

MVC的全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，是一种软件设计典范。它是用一种业务逻辑、数据与界面显示分离的方法来组织代码，将众多的业务逻辑聚集到一个部件里面，在需要改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑，达到减少编码的时间。

V即View视图是指用户看到并与之交互的界面。比如由html元素组成的网页界面，或者软件的客户端界面。MVC的好处之一在于它能为应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，它只是作为一种输出数据并允许用户操纵的方式。

M即model模型是指模型表示业务规则。在MVC的三个部件中，模型拥有最多的处理任务。被模型返回的数据是中立的，模型与数据格式无关，这样一个模型能为多个视图提供数据，由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

C即controller控制器是指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，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示返回的数据。

三者之间的调用关系。

4、JAVA SERVLET API中forward() 与redirect()的区别？

forward是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，其实客户端浏览器只发了一次请求，所以它的地址栏中还是原来的地址，session,request参数都可以获取。

redirect就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，相当于客户端浏览器发送了两次请求。

5、如何处理Web应用中的中文乱码

**出现中文乱码问题的种类：**

**1. JSP页面显示乱码**   
JSP页面显示乱码是由于页面未设置编码方案，通过页面的page指令的contextType属性和pageEncoding属性以及HTML META标签的charset属性中都指定使用中文编码即可。代码举例如下。

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8" import="java.sql.\*"%>

<metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">

**2. 表单提交中文时出现乱码**   
JSP获取页面参数时，一般是采用默认的编码方案(ISO8859-1)，如果页面参数的编码和页面的编码不一致时，很有肯能出现乱码。如果确定乱码问题是出于此，解决方案如下。

在获取表单数据时，首先使用request的setCharacterEncoding()方法强行设置获取表单数据的编码方案，但是要注意**这种方法仅仅适用于表单的method属性为post时**。代码举例如下。

//假设页面要获得的参数名称为tname

request.setCharacterEncoding("UTF-8");

String topic=request.getParameter("tname");

* 如果表单的method属性为get方式，使用手动转换的方式，代码举例如下。

//假设页面要获得的参数名称为tname

String topic=request.getParameter("tname");

topic=newString(topic.getBytes("ISO-8859-1"),"UTF-8");

* 变量输出到JSP页面上出现了乱码，通过response的setContentType()方法设置。

response.setContentType("text/html;charset=utf-8");

**3. 数据库读取数据显示乱码**   
当程序向数据库保存信息或者读取数据库的信息时会出现中文乱码现象，要保证数据库和jsp页面的编码一致，建议统一使用utf-8编码方案。本文以mysql为例。   
第一步：更改数据库编码方式   
首先更改表的编码方式，alter table 表名 default character set utf8;   
再更改列的编码方式，alter table 表名 change 列名列名数据结构 character set utf-8 ；   
第二步：在连接数据库时

//要输入你自己的用户名、密码、数据库名字

connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/database?user=root&password=root&userUnicode=true&characterEncoding=utf-8");

6、每次请求Servlet都会创建一个Servlet对象吗？为什么？

服务器只会创建每个Servlet的单一实例，每个用户请求都会引发新的线程，将用户请求交给相应的doGet或者doPost进行处理。  
首次创建Servlet时，它的init方法会得到调用，因此init方法是放置一次性代码的地方，之后针对每个用户请求，都会创建一个线程，该线程调用前面创建的实例的Service方法。多个并发请求回导致多个线程同时调用Service（尽管可以实现特殊的接口（SingleThreadModel）——规定任何时间只允许单个线程运行）。之后Service方法依据接受到得[HTTP请求](https://www.baidu.com/s?wd=HTTP%E8%AF%B7%E6%B1%82&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9nv79mH6kP1b3mh7hnym10ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHfzPjbLnHD4" \t "_blank)类型调用doGet，doPost，或者其他的doXXX方法。

7、JSP和Servlet有哪些相同点和不同点，他们之间的联系是什么？  
是Servlet技术的扩展，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侧重于视图，Servlet主要用于控制逻辑。

8、简述请求转发与重定向的区别（5分）

答案： 请求转发是服务器内部进行跳转，请求服务器中的另一个文件或资源；重定向是客户端跳转，服务器将请求的路径发送到客户端，在由客服端去请求资源，所以可以请求服务器或其他服务的资源。

9、Servlet执行时一般实现哪几个方法

答案

public void init(ServletConfig config)

public ServletConfig getServletConfig()

public String getServletInfo()

public void service(ServletRequest request,ServletResponse response)

public void destroy()。

10、请描述JavaWeb的四大作用域为：PageContext，ServletRequest，HttpSession，ServletContext；

答案：PageContext域：作用范围是整个JSP页面，是四大作用域中最小的一个；生命周期是当对JSP的请求时开始，当响应结束时销毁。

ServletRequest域：作用范围是整个请求链（请求转发也存在）；生命周期是在service方法调用前由服务器创建，传入service方法。整个请求结束，request生命结束。

HttpSession域：作用范围是一次会话。生命周期是在第一次调用request.getSession()方法时，服务器会检查是否已经有对应的session,如果没有就在内存中创建一个session并返回。当一段时间内session没有被使用（默认为30分钟），则服务器会销毁该session。如果服务器非正常关闭（强行关闭），没有到期的session也会跟着销毁。如果调用session提供的invalidate（） ，可以立即销毁session。

ServletContext域：作用范围是整个Web应用。当Web应用被加载进容器时创建代表整个web应用的ServletContext对象，当服务器关闭或Web应用被移除时，ServletContext对象跟着销毁。

11、会话跟踪的方式有哪些

答案：url重写：利用？传递上次请求的数据到新的资源

隐藏域：<input type="hidden">

Cookie：客户端存储技术

Session：服务器端会话技术

12、过滤器常见的用途：

答案：统一设置编码

登录和权限拦截

统一关闭数据库连接

六、程序实现题

1、有一个jsp页面向后端名字为MyServlet的Servlet使用发起请求(post)，并且传递参数username,myServlet接收到参数之后，放入HttpSession中，并且跳转到a.jsp页面(**Forward**)，请写出MyServlet中doPost方法中的代码

|  |
| --- |
| **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  String username=request.getParameter("username");  HttpSession session=request.getSession();  session.setAttribute("username", username);  RequestDispatcher dispatcher = request.getRequestDispatcher("a.jsp");  dispatcher.forward(request, response);  } |

2、请实现登录某商城的功能。

（10分）

|  |
| --- |
| 登录页面代码：login.html  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>登录</title>  </head>  <body>  <fieldset>  <legend>用户登录</legend> <br />  <form action="loginServlet" method="post" name="login">  用户名：<input type="text" name="username" /> <br /> <br />  密&nbsp;码：<input type="password" name="password" /> <br />  <input type="submit" value="登录" />  </form>  </fieldset>  </body>  </html>  创建Servlet：LoginServlet.java  public class LoginServlet extends HttpServlet {  private static final long serialVersionUID = 1L;  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  doPost(request, response);  }  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  response.setContentType("text/html; charset=UTF-8");  response.setCharacterEncoding("utf-8");  request.setCharacterEncoding("utf-8");    String username = request.getParameter("username");  String password = request.getParameter("password");    System.out.println(username+":"+password);  //获取输出流，用户显示信息，也可以用转发或者重定向  PrintWriter pw = response.getWriter();  if("admin".equals(username) && "123".equals(password)){  pw.write("login success");  //转发---需要编写success.jsp页面  request.getRequestDispatcher("success.jsp").forward(request, response);  }else{  pw.write("login fail");    //转发---需要编写fail.jsp页面  request.getRequestDispatcher("fail.jsp").forward(request, response);;  }    }  } |

3、已知JavaBean定义如下，假设已经存在一个登录的表单，文本框名字为 accountName，密码框名字为accountPwd，将该表单提交至AccountServlet，请根据注释填写代码

说明：红色部分为答案部分。

**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;

private String state;//getter setter省略

private String score;//getter setter省略

**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;

}

}

package com.etc.controller;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* 配置servlet，访问路径为／AccountServlet

\*/

**@WebServlet("/AccountServlet")**

public class AccountServlet **extends HttpServlet** {

    private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AccountServlet () {

super();

// TODO Auto-generated constructor stub

}

    /\*\*

     \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

     \*/

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        // 设置请求编码格式为utf－8

        request.setCharacterEncoding("utf-8");

//获取请求参数accountName accountPwd

String accountName=null,accountPwd=null;

if(request.getParameter(“accountName”)!=null)

{

accountName=request.getParameter(“accountName”);

}

if(request.getParameter(“accountpwd”)!=null)

{

accountPwd =request.getParameter(“accountPwd”);

}

//调用Service获取相应结果

Account account=AccountService.login(accountName,accountPwd);

//判断account

//如果account不为null，将account对象存储到request范围,转发到pinfo.jsp

if（account！＝null）

｛

request.setAttribute(“account”,account);

request.getRequestDispatcher(“pinfo.jsp”).forward(request,response);

｝

    }

    /\*\*

     \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

     \*/

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        // TODO Auto-generated method stub

        doGet(request, response);

    }

}

4、前后端分离的程序实现中遇到跨域访问问题，Servlet中如何解决。（10分）

|  |
| --- |
| @WebFilter  public class SimpleCORSFilter implements Filter{  @Override  public void destroy() {  }  @Override  public void doFilter(ServletRequest req, ServletResponse res,  FilterChain chain) throws IOException, ServletException {  HttpServletResponse response = (HttpServletResponse) res;  response.setHeader("Access-Control-Allow-Origin", "\*");  response.setHeader("Access-Control-Allow-Methods", "POST, GET, OPTIONS, DELETE");  response.setHeader("Access-Control-Max-Age", "3600");  response.setHeader("Access-Control-Allow-Headers", "x-requested-with");  chain.doFilter(req, res);  }  @Override  public void init(FilterConfig arg0) throws ServletException {  }  } |

5、完整书写com.handson.servlet.HelloServlet类代码，要求用户通过Get方式访问该Servlet对象后，HelloServlet向浏览器输出“中国，您好！”这句中文，并将这句话保存在session的“title”属性中以备日后使用。

|  |
| --- |
| Package com.handson.servlet;  Public class HelloServlet extends HttpServlet{   Public void doGet(HttpServletRequest request , HttpServletResponse response) throws IOException{   String msg = “中国，你好！”;  HttpSession session = request.getSession();    Session.setAttribute(“title” , msg);   Response.setContentType(“text/html;charset=utf-8”);    Response.setCharacterEncoding(“UTF-8”);    PrintWriter out = Response.getWrite();    out.println(msg);   out.flush();  }  Public void doPost(HttpServletRequest request , HttpServletResponse response) throws IOException{  This.doGet(request , response); } |

6、在程序里怎样注册servlet。

|  |
| --- |
| <servlet>  <servlet-name>displayChart</servlet-name>  <servlet-class>com.free.servlet.DisplayChart</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>displayChart</servlet-name>  <url-pattern>/displayChart</url-pattern>  </servlet-mapping> |

7、Servlet实现用户登录程序实现，进行用户数据存储在session中 并且实现cookie保存用户账号、要求：登录账号不改变cookie中不重复存储数据

（10分）

|  |
| --- |
| package com.web0605.servlet;  import java.io.IOException;  import javax.servlet.ServletContext;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.Cookie;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import javax.servlet.http.HttpSession;  import com.web0602.dao.UserDao;  import com.web0602.entry.UserInfo;  /\*\*  \* Servlet implementation class LoginServlet  \*/  @WebServlet("/LoginServlet")  public class LoginServlet extends HttpServlet {  private static final long serialVersionUID = 1L;  /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public LoginServlet() {  super();  }    protected void doGet(HttpServletRequest request,  HttpServletResponse response) throws ServletException, IOException {  doPost(request, response);  }  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse  \* response)  \*/  protected void doPost(HttpServletRequest request,  HttpServletResponse response) throws ServletException, IOException {  // 设置字符集  String account = request.getParameter("account");  String pass = request.getParameter("pass");  //用户登录判断  //调用用户数据库操作层的方法判断用户是否存在  UserInfo user= UserDao.selectByCondition(account, pass);    if (user!=null) {// 账号存在  Cookie[] cooks= request.getCookies();  boolean isExit=false;  if(cooks!=null){  for(Cookie co:cooks){  if(co.getName().equals("account") && co.getValue().equals(account) ){  isExit=true;  break;  }  }}  if(!isExit){  System.out.println("执行。。。。。。。。。。。。。");  // 创建Cookie对象  Cookie co = new Cookie("account", account);  co.setMaxAge(60\*60\*2);//有效时长2个小时  // 存储Cookie对象  response.addCookie(co);  //获得session对象、  }  HttpSession se= request.getSession();  se.setAttribute("user", user);//将登录成功的用户存储在会话中  System.out.println("--------------------");  /\* se.setMaxInactiveInterval(60\*30);\*/  //获得上下文对象  ServletContext con=this.getServletContext();  Integer number=(Integer) con.getAttribute("number");  /\*int number=1;\*/  if(number==null){  number=1;  }else{  number++;  }  con.setAttribute("number",number);  /\* request.getRequestDispatcher("index.jsp").forward(request, response);\*/  response.sendRedirect("index.jsp");  }else {  request.getRequestDispatcher("login.jsp")  .forward(request, response);  }  }  } |

8、编写login.jsp，提交到LoginServlet，在LoginServlet中获得输入的用户名和密码，如果用户名或密码为空，则跳转到LoginFailServlet,否则跳转到LoginSuccessServlet （10分）

|  |
| --- |
| String username=request.getParameter("username");  String pwd=request.getParameter("pwd");  **if(username==null||username.equals("")||pwd==null||pwd.equals("")){**  response.sendRedirect("LoginFailServlet");  }**else{**  response.sendRedirect("LoginSuccessServlet");  } |