

# 模拟试题

## 一 选择题 （50题）

针对以下题目请选择正确的答案（每道题目有一个或多个正确的答案）。

1) 在JSP中，要定义一个方法，需要用到以下（\_\_）元素。

- a) `<%= %>`
- b) `<% %>`
- c) `<%! %>`
- d) `<%@ %>`

2) JSP页面经过编译之后，将创建一个（\_\_）。

- a) applet
- b) servlet
- c) application
- d) exe文件

3) 当JSP页面执行自定义标签，遇到结束标签时将调用（\_\_）方法执行处理。

- a) `doStartTag()`
- b) `doEndTag()`
- c) `doInitBody()`
- d) `doAfterBody()`

4) 在J2EE中，test.jsp文件中有如下一行代码：

```
<jsp:useBean id="user" scope="____" type="com.UserBean" />
```

要使user对象中一直存在于对话中，直至其终止或被删除为止，下划线中应填入（\_\_）。

- a) page
- b) request
- c) session
- d) application

5) 在J2EE中，以下不是JSP隐式对象的是（\_\_）。

- a) `pageContext`
- b) `context`
- c) `application`
- d) `out`

6) Servlet中，`HttpServletResponse`的（\_\_）方法用来把一个Http请求重定向到另外的URL。

- a) `sendURL()`
- b) `redirectURL()`
- c) `sendRedirect()`
- d) `redirectResponse()`

7) 在JSP中，`page`指令的（\_\_）属性用来引入需要的包或类。

- a) `extends`
- b) `import`
- c) `language`
- d) `contentType`

8) 在Servlet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（\_\_）方法。

- a) init
- b) service
- c) doFilter
- d) destroy

9) 在JSP中，只有一行代码：<%= ' A' + ' B' %>，运行将输出（\_\_）。

- a) A+B
- b) AB
- c) 131
- d) 错误信息，因为表达式是错误的

10) 给定一个Servlet的代码片段如下：

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

```
_____  
out.println("hi kitty!");  
out.close();  
}
```

运行次Servlet时输出如下：

hi kitty!

则应在此Servlet下划线处填充如下代码。

- a) PrintWriter out = response.getWriter();
- b) PrintWriter out = request.getWriter();
- c) OutputStream out = response.getOutputStream();
- d) OutputStream out = request.getWriter();

11) 给定如下JSP代码，假定在浏览器中输入

URL: http://localhost:8080/web/jsp1.jsp，可以调用这个JSP，那么这个JSP的输出是（\_\_）。

```
<%@ page contentType="text/html; charset=GBK" %>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
<html>  
<body>  
<% int counter = 10; %>  
<c:if test=" ${counter%2==1} " >  
<c:set var=" isOdd" value=" true" ></c:set>  
</c:if>
```



```
<c:choose>
<c:when test=" ${isOdd==true}" >it' s an odd </c:when>
<c:otherwise>it' s an even </c:otherwise>
</c:choose>
</body>
</html>
```

- a) 一个HTML页面，页面上显示it's an odd
- b)** 一个HTML页面，页面上显示it's an even
- c) 一个空白的HTML页面
- d) 错误信息

12) 给定java程序User.java,代码如下:

```
package user;

public class User{
    private String password;
    private String loginName;
    public String getLoginName(){
        return loginName;
    }
    public String getPassword(){
        return password;
    }
    public void setLoginName(String loginName){
        this.loginName=loginName;
    }
    public void setPassword(String password){
        this.password=password;
    }
}
```

给定user.jsp,代码片断如下:

---

loginName: <jsp:getProperty name="user" property="loginName"/>

password: <jsp:getProperty name="user" property="password"/>

在浏览器上输入[URL:http://localhost/user.jsp?loginName=sa&password=sa](http://localhost/user.jsp?loginName=sa&password=sa)并回车

要使浏览器上显示:

**loginName: sa          password: sa**

在user.jsp中的下划线处应填入以下( )代码.



- a) `<jsp:useBean id="user" class="user.User" scope="page"/>`  
`<jsp:setProperty name="user" property="loginName" value="*" />`  
`<jsp:setProperty name="user" property="password" value="*" />`
- b) `<jsp:useBean id="user" class="user.User" scope="application"/>`  
`<jsp:setProperty name="user" property="loginName" value="*" />`  
`<jsp:setProperty name="user" property="password" value="*" />`
- c) `<jsp:useBean id="user" class="user.User" scope=" application"/>`  
`<jsp:setProperty name="user" property="*" />`
- d) `<jsp:useBean id="user" class="user.User" scope=" page"/>`  
`<jsp:setProperty name="user" property="*" />`

13) 某JSP程序中声明使用javaBean的语句如下:

`<jsp:useBean id="user" class="mypackage.User" scope="page"/>`

要取出该javaBean的loginName属性值,以下语句正确的是( )。(选择两项)

- a) `<jsp:getProperty name="user" property="loginName"/>`
- b) `<jsp:getProperty id="user" property="loginName"/>`
- c) `<%=user.getLoginName()%>`
- d) `<%=user.getProperty("loginName")%>`

14) J2EE中, 以下关于HttpServlet的说法错误的是( )。

- a) HttpServlet是一个抽象类
- b) HttpServlet类扩展了GenericServlet类
- c) HttpServlet类的子类必须至少重写service方法
- d) HttpServlet位于javax.servlet.http包中

15) 给顶一个JSP程序源代码, 如下:

```
<jsp:include page="two.jsp" flush="true">
<jsp:param name="location" value="beijing"/>
</jsp:include>
```

在two.jsp中使用( ) 代码片段可以输出参数location的值。

- a) `<jsp:getParam name="location">`
- b) `<jsp:getParameter name="location">`
- c) `<%=request.getAttribute("location")%>`
- d) `<%=request.getParameter("location")%>`

16) J2EE中, JSP EL表达式: `s{(10*10) ne 100}`的值是( )。

- a) 0
- b) true
- c) false
- d) 1



17) 在J2EE中, test.jsp文件如下:

```
<html>
<%{%>
<jsp:useBean id="buffer" scope="page"
type="java.lang.StringBuffer"/>
<%    buffer.append("ABC");%>
<%}%> buffer is <%=buffer%>
</html>
```

运行时, 将发生( )。

- a) 转译期错误
- b) 运行期错误
- c) 运行后, 浏览器上显示: buffer is null
- d) 运行后, 浏览器上显示: buffer is ABC

18) 如JSP页面执行时遇到自定义标签, doStartTag()方法将被调用( )次。

- a) 1
- b) 1或多
- c) 0或1
- d) 0或多

19) 给定include1.jsp文件代码片断, 如下:

```
<% pageContext.setAttribute("User","HAHA");%>
_____//此处填写代码
```

给定include2.jsp文件代码片断, 如下:

```
<%=pageContext.getAttribute("User")%>
```

要求运行include1.jsp时, 浏览器上输出: HAHA

要满足以上条件, jsp1.jsp中下划线处应填入以下( )语句。

- a) <jsp:include page="include2.jsp" flush="true"/> null
- b) <%@ include file="include2.jsp"%> HAHA
- c) <jsp:forward page="include2.jsp"/> null
- d) <% response.sendRedirect("include2.jsp");%> null

20) 在J2EE中, 在web.xml中, 有如下代码:

```
<session - config>
    <session - timeout>30</session - timeout>
</session - config>
```

上述代码定义了默认的会话超时时长, 时长为30( )。

- a) 毫秒
- b) 秒
- c) 分钟
- d) 小时



21) 在inc.jsp中包含数行代码与一张图片，现在要创建一个home.jsp，每次浏览home.jsp时都能够以最快的速度显示inc.jsp中的所有内容以及home.jsp自身要显示的内容，应该在home.jsp中使用的语句是( )。

- a) `<%@include file = "inc.jsp"%>`                      b) `<%@include page="inc.jsp"flush="true">`  
c) `<jsp:include file = "inc.jsp"flush = "false">`      d) `<jsp:forward page = "inc.jsp"/>`

22) 给定一个JSP程序源码如下：

```
<jsp:include page = "two.jsp"flush = "true">
<jsp:param name = "location"value = "beijing"/>
</jsp:include>
```

在two.jsp中加入( )代码片断可以输出参数location的值。

- a) `<jsp:getParam name = "location">`                      b) `<jsp:getParameter name = "location">`  
c) `<% = request.getAttribute("location")%>`      d) `<% = request.getParameter("location")%>`

23) 使用自定义标签时，要为JSP引擎提供有关自定义标签的相关信息，下列信息描述语句正确的是( )。

- a) `<tag>`  
    `<name>hello</name>`  
    `<tag-class>com.tagdemo.HelloTag</tag-class>`  
    `</tag>`  
b) `<taglib>`  
    `<name>hello</name>`  
    `<tag-class>com.tagdemo.HelloTag</tag-class>`  
    `</taglib>`  
c) `<tag-info>`  
    `<name>hello</name>`  
    `<tagclass>com.tagdemo.HelloTag</tagclass>`  
    `</tag-info>`  
d) `<tag-desc>`  
    `<name>hello</name>`  
    `<tagclass>com.tagdemo.HelloTag</tagclass>`  
    `</tag-desc>`

24) 要设置某个JSP页面为错误处理页面，以下page指令正确的是( )。

- a) `<%@ page errorPage="true"%>`                      b) `<%@ page`  
`isErrorPage="true"%>`  
c) `<%@ page extends="javax.servlet.jsp.JspErrorPage"%>`



d) `<%@ page info="error"%>`

25) **JavaMail**是一组用于发送和接收邮件消息的**API**。发送邮件使用（ ）协议，接收邮件使用（\_\_）协议。

a) POP3 SMTP

b) POP3 TCP

c) SMTP TCP

d) SMTP

POP3

26) 在**Servlet**里，能正确获取**session**的语句是（\_\_）。

a) `HttpSession session = request.getSession(true);`

b) `HttpSession session = request.getSession(true);`

c) `HttpSession session = response.getSession(true);`

d) `HttpSession session = response.getSession(true);`

27) 在**JSP**中，（\_\_）动作用于将请求转发给其他**JSP**页面。

a) forward

b) include

c) useBean

d) setProperty

28) 在**J2EE**中，在一个**JSP**文件中，有表达式`<%=2+3 %>`，它将输出（\_\_）。

a) 2+3

b) 5

c) 23

d) 不会输出，因为表达式是错误的

29) 在**J2EE**实现企业级应用开发中，（\_\_）是描述标记库的**XML**文档。

a) TLD文件

b) DTD文件

c) WAR文件

d) EAR文件

30) 给定**JSP**程序源码如下，可以在下划线处插入并且能够正确输出**WELCOME, JACK**的语句是（\_\_）。（选择二项）

`<html>`

`<body>`

`<% String name="JACK"; %>`

`WELCOME, _____`

`</body></html>`

a) name

b) `<%=name%>`

c) `out.println(name);`

d) `<% out.println(name); %>`

31) 在**J2EE**中，重定向到另一个页面，以下（\_\_）语句是正确的。



- a) request.sendRedirect("http://www.jb-aptech.com.cn");
- b) request.sendRedirect();
- c) response.sendRedirect("http://www.jb-aptech.com.cn");
- d) response.sendRedirect();

32) 在JSP中，使用<jsp:useBean>动作可以将javaBean嵌入JSP页面，对JavaBean的访问范围不能是（\_\_）。

- a) page
- b) request
- c) response
- d) application

33) 在J2EE中，使用Servlet过滤器时，可以在web.xml文件的（\_\_）元素中包括<init-param>元素。

- a) <filter>
- b) <filter-mapping>
- c) <filter-name>
- d) <filter-class>

34) Servlet中，使用（\_\_）接口或类中定义的（\_\_）方法来处理客户端发出的表单数据请求

- a) HttpServlet doHead
- b) HttpServlet doPost
- c) ServletRequest doGet
- d) ServletRequest doPost

35) 给定JSP程序源码如下，该JSP运行后输出的结果是（\_\_）。

```
<html>  
<% int count =1;%> Count: <% ++count; %>  
</html>
```

- a) Count:1
- b) Count:2
- c) 1:2
- d) Count:

36) 给定如下Servlet代码，假定在浏览器中输入URL:

http://localhost:8080/servlet/HelloServlet，可以调用这个servlet，那么这个Servlet的输出是（\_\_）。

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
public class HelloServlet extends HttpServlet{  
    public void service(HttpServletRequest req, HttpServletResponse res)  
        throws ServletException, IOException{  
    }  
    public void doGet(HttpServletRequest req, HttpServletResponse res)
```



```

throws ServletException, IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    out.println("<html>");
    out.println("<body>");
    out.println("doGet Hello World!");
    out.println("</body>");
    out.println("</html>");
    out.close();
}

public void doPost(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    out.println("<html>");
    out.println("<body>");
    out.println("doPost Hello World!");
    out.println("</body>");
}
}

```

- a) 一个HTML页面，页面上显示doGet Hello World!
- b) 一个HTML页面，页面上显示doPost Hello World!
- c) 一个空白的HTML页面
- d) 错误信息

37) 在J2EE中，request对象的（\_\_）方法可以获取页面请求中一个表单组件对应多个值时的用户的请求数据。

- a) String getParameter(String name)
- b) String[] getParameter(String name)
- c) String getParameterValues(String name)
- d) String[] getParameterValues(String name)

38) （\_\_）是发送邮件的协议。

- a) SMTP
- b) LAMP
- c) POP3
- d) HTTP

39) WEB应用中，常用的会话跟踪方法不包括（\_\_）。

- a) URL重写
- b) Cookie



c) 隐藏表单域

d) 有状态HTTP协议

40) 在J2EE中,  $\${2 + "4"}$ 将输出 ( )。

a) 2 + 4

b) 6

c) 24

d) 不会输出, 因为表达式是错误的

41) 在J2EE的Model II 模式中, 模型层对象被编写为 ( )。

a) Applet

b) JSP

c) Server

d) JavaBean

42) J2EE中, JSP EL 表达式:  $\${user.loginName}$ 执行效果等同于 ( )。

a) `<% = user.getLoginName()%>`

b) `<%user.getLoginName();%>`

c) `<% = user.loginName%>`

d) `<% user.loginName;%>`

43) J2EE中, ( ) 类 ( ) 方法用于返回应用程序的上下文路径。

a) `HttpServletRequest`、`getContextPath()`

b) `HttpServletRequest`、`getPathInfo()`

c) `ServletContext`、`getContextPath()`

d) `ServletContext`、`getPathInfo()`

44) 在J2EE中, 使用Servlet过滤器时, 需要在web.xml通过 ( ) 元素将过滤器映射到Web资源。

a) `<filter>`

b) `<filter-mapping>`

c) `<servlet>`

d) `<servlet-mapping>`

45) 给定JSP程序源码如下:

```
<html>
```

```
    <% int count =1;%> _____
```

```
</html>
```

以下 ( ) 语句可以在下划线处插入, 并且运行后输出结果是: 1。

a) `<%=++count %>`

b) `<% ++count; %>`

c) `<% count++; %>`

d) `<% =count++ %>`

46) 在J2EE中, 在web.xml中定义过滤器时可以指定初始化参数, 以下定义正确的是 ( )

a) `<filter>`

```
    <filter-name>someFilter</filter-name>
```

```
    <filter-class>filters.SomeFilter</filter-class>
```

```
    <init-param>
```

```
        <param-name>encoding</param-name>
```



```

        <param-value>EUC_JP</param-value>
    </init-param>
</filter>
b) <filter>
    <filter-name>someFilter</filter-name>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>EUC_JP</param-value>
    </init-param>
</filter>
c) <filter-mapping>
    <filter-name>someFilter</filter-name>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>EUC_JP</param-value>
    </init-param>
</filter-mapping>
d) <filter-mapping>
    <filter-name>someFilter</filter-name>
    <filter-class>filters.SomeFilter</filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>EUC_JP</param-value>
    </init-param>
</filter-mapping>

```

47) **servlet** 的生命周期又一系列事件组成，把这些事件按照先后顺序排序，以下正确的是（\_\_）

- a) 加载类，实例化，请求处理，初始化，销毁
- b) 加载类，实例化，初始化，请求处理，销毁**
- c) 实例化，加载类，初始化，请求处理，销毁
- d) 加载类，初始化，实例化，请求处理，销毁

48) 在J2EE中，给定某Servlet的代码如下，编译运行该文件，以下陈述正确的是（\_\_）。

```

Public class Servlet1 extends HttpServlet{
    Public void init() throws ServletException{
    }

```



```

        Public void service(HttpServletRequest
request,HttpServletResponse response)
            throws ServletException,IOException{
        PrintWriter out = response.getWriter();
        out.println("hello!");
    }
}

```

- a) 编译该文件时会提示缺少doGet()或者doPost()方法，编译不能够成功通过
- b)** 编译后，把Servlet1.class放在正确位置，运行该Servlet，在浏览器中会看到输出文字：hello!
- c) 编译后，把Servlet1.class放在正确位置，运行该Servlet，在浏览器中看不到任何输出的文字
- d) 编译后，把Servlet1.class放在正确位置，运行该Servlet,在浏览器中会看到运行期错误信息

49) 某web应用的上下文路径是root，配置文件web.xml中有如下片段：

```

<servlet>
    <servlet-name>process</servlet-name>
    <servlet-class>slt.ProcessServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>process</servlet-name>
    <url-pattern>/proc</url-pattern>
</servlet-mapping>

```

以下说法正确的是（\_\_）。（选择二项）

- a)** 在包slt中，有一个名为ProcessServlet.class的文件
- b)** 该servlet的访问路径是http://机器IP:端口号/root/proc
- c) 该servlet的访问路径是http://机器IP:端口号/root/ProcessServlet
- d) 该servlet的访问路径是http://机器IP:端口号/root/ProcessServlet

50) 在J2EE中，test.jsp文件中有如下一行代码：

```
<jsp:useBean id="user" scope="__" type="com.UserBean">
```

要使user对象可以作用于整个应用程序，下划线中应添入（\_\_）。

- a) page
- b) request
- c) session
- d) application**

## 二 简述题 （9题）

1. jsp有哪些内置对象？作用分别是什么？



2. forward 和redirect的区别
3. jsp有哪些动作?作用分别是什么?
4. 两种跳转方式分别是什么?有什么区别?
5. 说一说Servlet的生命周期?
6. JAVA SERVLET API中forward() 与redirect()的区别?
7. 什么情况下调用doGet()和doPost()?
8. 如何实现servlet的单线程模式
9. JSP和Servlet有哪些相同点和不同点,他们之间的联系是什么?