



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0  
公众号后台回复“面试手册”

### 1.关于 Web 应用程序，下列说法错误的是（ b ）。

- a) WEB-INF 目录存在于 web 应用的根目录下
- b) WEB-INF 目录与 classes 目录平行
- c) web.xml 在 WEB-INF 目录下
- d) Web 应用程序可以打包为 war 文件

解释：classes 目录位于 WEB-INF 目录之下，而不是平行的关系。

### 2.有关 Servlet 的生命周期说法正确的有（ cd ）

- a) Servlet 的生命周期由 Servlet 实例控制
- b) init()方法在创建完 Servlet 实例后对其进行初始化，传递的参数为实现 ServletContext 接口的对象
- c) service()方法响应客户端发出的请求
- d) destroy()方法释放 Servlet 实例

解释：Servlet 的生命周期是由 Servlet 容器（Tomcat 就是常见的 Servlet 容器）管理的，因此 a 不对。Servlet 中的 init（）方法有两个重载，一个是空参的，另外一个带 ServletConfig 形参的，而不是

ServletContext，因此 b 不对。

关于 d 选项，说法其实并不好，正确的表达应该是因为 Servlet 实例要释放（销毁）了，才会先调用 destroy（）方法

### 3.有关会话跟踪技术描述正确的是（ abc ）。

- 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

### 4.以下 web.xml 片断（ d ）正确地声明 servlet 上下文参数

```
a) <init-param>
  <param-name>MAX</param-name>
  <param-value>100</param-value>
</init-param>

b) <context-param>
  <param name="MAX" value="100" />
```

```
<context-param>
```

c) `<context>`

```
<param name="MAX" value="100" />
```

```
<context>
```

d) `<context-param>`

```
<param-name>MAX</param-name>
```

```
<param-value>100</param-value>
```

```
<context-param>
```

补充：关于 context-param 和 init-param 的作用，请参考下文

<https://www.cnblogs.com/hzj-/articles/1689836.html>

## 5.以下 ( a ) 可用于检索 session 属性 userid 的值

a) session.getAttribute ("userid" );

b) session.setAttribute ("userid" );

c) request.getParameter ("userid" );

d) request.getAttribute ("userid" );

## 6.下列 JSP 代码，以下 ( cd ) 可放置在//1 处，不会发生编译错误。

```
<html>
<body>
<%
for(int i = 0; i < 10; i++) {
//1
}
%>
</body>
</html>
```

a) `<%= i %>`

b) `i`

c) `%><%= i %><%`

d) 不写任何内容

## 7.考虑下面两个 JSP 文件代码片断

test1.jsp:

```
HTML>
<BODY>
<% pageContext.setAttribute(" ten" ,new Integer(10));%>
//1
</BODY>
</HTML>
```

test2.jsp:

数字为：<%= pageContext.getAttribute(" ten" )%>

以下 ( c ) 放置在 test1.jsp 中的 //1 处，当请求 test1.jsp 时正确输出 test2.jsp 中的内容。

a) <jsp:include page=" test2.jsp" />

b) <jsp:forward page=" test2.jsp" />

c) <%@ include file=" test2.jsp" %>

d) 由于 pageContext 对象的 scope 属性为 page, 所以 test2.jsp 不能访问 test1.jsp 定义的属性

解释：JSP 中 include 的两种方法，1、<jsp:include page=" test2.jsp" /> 属于动态引入/2、<%@ include

file=" test2.jsp" %> 属于静态引入。两者的区别如下：

1. 执行时间上的区别

<%@ include file=" test2.jsp" %> 是在翻译阶段执行（将 JSP 页面转换成 servlet 的阶段）。通俗的话讲

就是先合并在一起，然后再编译成一个 Servlet 文件。

<jsp:include page=" test2.jsp" /> 在请求处理阶段执行。通俗的话讲就是先各自编译，然后在处理请求的

时候结果再合并到一起。

2. 引入内容的方式区别

<%@ include file=" test2.jsp" %> 是纯粹的把部分代码写到了另一页面（或者说是共享），而那另一页面

中不能有相同的变量名，但可以借用主页面的内容。

<jsp:include page=" test2.jsp" /> 引入执行页面或 servlet 所生成的应答文本。

<jsp:forward page=" test2.jsp" /> 属于请求转发，由于 pageContext 对象的 scope 属性为 page, 所以如果

请求转发了，那么 test2.jsp 就不能访问 test1.jsp 定义的属性了。所以该题选 C。

## 8. 有关 JSP 隐式对象，以下 ( acd ) 描述正确。

a) 隐式对象是 WEB 容器加载的一组类的实例，可以直接在 JSP 页面使用

b) 不能通过 config 对象获取 ServletContext 对象

c) response 对象通过 sendRedirect 方法实现重定向

d) 只有在出错处理页面才有 exception 对象

解释：jsp 的九大内置对象分别是：config、request、response、out、page、pageContext、session、exception、application。其中 exception 是特殊的内置对象，只有当在 jsp 中添加 isErrorPage="true" 属性时如下配置时才可以使用。该属性一般出现在设定的错误界面。

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" isErrorPage="true" %>
```

## 9. 考虑下面 JSP 文件代码片断：

```
<HTML>
<BODY>
<jsp:include page=" test2.jsp" >
<jsp:param name=" username" value=" zhangsan" />
</jsp:include>
</BODY>
</HTML>
```

以下 ( c ) 代码片断放置在 test2.jsp 中不会导致错误。

- a) <jsp:getParam name=" username" />
- b) <jsp:include param =" username" />
- c) <%=request.getParameter("username") %>
- d) <%=request.getAttribute("username") %>

解释：<jsp:include page=" test2.jsp" >属于动态调用 test2.jsp 界面，相当于动态去请求 test2.jsp 所生成的Servlet，在请求的同时携带了请求参数“ username”，我们知道在 Servlet 中获取请求携带的参数就是通过request.getParameter(key)来获取的，因此 C 正确。

## 10.以下是 login.jsp 文件的代码片断：

```
<%@ page isELIgnored="false"%>
<html>
<body>
<FORM action="login.jsp" method="GET">
<input type="text" name="name" value="${param['name']}">
<input type="submit" value="提交">
</FORM>
<P>
用户名为： ${param.name}
</body>
</html>
```

以下 ( C ) 描述正确。

- a) 发生运行错误
- b) 页面会出现一文本框，并且文本框中内容为\${param['name']}
- c) 当用户输入名字并单击“提交”按钮时，在同一页面中的“用户名为：”字样后面会显示用户输入的内容
- d) 当用户输入名字并单击“提交”按钮时，在同一页面中的“用户名为：”字样后面会显示\${param.name}

## 11.doAfterBody()方法是在 ( b ) 接口中定义的。

- a) Tag
- b) IterationTag
- c) BodyTag
- d) TagSupport

解释：该知识点属于 jsp 自定义标签中的知识，比较生僻。如果了解如何在 jsp 中使用自定义标签的可以参

考下文 <https://www.cnblogs.com/flying607/p/5063207.html>

## 12.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

## 13.有关 MVC 设计模式 ( b ) 描述不正确

- a) 使用 Servlet 作为控制器
- b) MVC 设计模式增大了维护难度
- c) MVC 设计模式属于 Model 2
- d) 模型对象向客户端显示应用程序界面

补充知识：

#### Model 1

Model 1 的基础是 JSP 文件，它由一些相互独立的 JSP 文件，和其他一些 Java Class 组成（不是必须的）。

这些 JSP 从 HTTP Request 中获得所需要的数据，处理业务逻辑，然后将结果通过 Response 返回前端浏览器。

#### Model 2

采用面向对象技术实现 MVC 模式从而扩展 JSP/Servlet 的模式被成为是 Model 2 模式。Apache Jakarta 项

目中 Struts 是一个实现 Model 2 的很好的框架，它通过一些 Custom Tag Lib 处理表现层，用 ActionFrom Bean

表示数据，用自己提供的一个 ActionServlet 作为控制器实现页面的流转的控制功能。

说的直白一些，model1 即为单纯的 jsp+java，没有框架参与，通过 response 和 request 对象传送值域，

而 model2,则使用较为流行的 struts2 框架。

### 14.在 Linux 中，可以使用命令（ c ）加挂计算机上的非 Linux 文件系统。

- a) cat /proc/filesystems
- b) ln
- c) mount
- d) df

### 15.下面关于 Linux 中 shell 的说法错误的是（ d ）。

- a) shell 是解释用户在终端键入的命令的一种中间程序
- b) shell 可以读取并执行脚本文件中的命令
- c) 用户可以使用参数将命令行的参数传递给 shell 脚本，从而实现在 Linux 中的交互式编程
- d) 默认情况下，Linux 中创建的所有文件都具有执行权限

### 16.在 Oracle 中，当需要使用显式游标更新或删除游标中的行时，UPDATE 或 DELETE 语句必须使用（ c ）子句

- a) WHERE CURRENT OF
- b) WHERE CURSOR OF
- c) FOR UPDATE
- d) FOR CURSOR OF

### 17.在 Oracle 中，使用下列的语句 CREATE PUBLIC SYNONYM parts FOR Scott.inventory;完成的任务是（ d ）

- a) 将 Scott.inventory 对象的访问权限赋予所有用户
- b) 指定了新的对象权限
- c) 指定了新的系统权限
- d) 给 Scott.inventory 对象创建一个公用同义词 parts

## 18.在 Oracle 中，执行如下 PL/SQL 语句后

```
CREATE TYPE car AS OBJECT ( id NUMBER, model VARCHAR2(25), color VARCHAR2(15) );  
...  
DECLARE  
myvar car.model%TYPE;  
BEGIN  
...  
END;
```

变量 myvar 的数据类型为 ( c )。

- a) NUMBER
- b) car 类型
- c) VARCHAR2
- d) OBJECT

解释：该知识点属于 PL/SQL 高级编程，中的对象类型 ( OBJECT TYPE )

## 19.List、 Map、 Set 三个接口存储元素时各有什么特点？

解答：

1 ) List是有序的Collection，使用此接口能够精确的控制每个元素插入的位置。用户能够使用索引（元素在List中的位置，类似于数组下标）来访问List中的元素，这类似于Java的数组。

2 ) Set是一种不包含重复的元素的Collection，即任意的两个元素e1和e2都有e1.equals(e2)=false，Set最多有一个null元素。

3 ) Map接口：请注意，Map没有继承Collection接口，Map提供key到value的映射

## 20.在项目中用过 Spring 的哪些方面？及用过哪些 Ajax 框架？

...



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Q Java研发军团

Java研发军团《Java面试手册》V1.0  
公众号后台回复“面试手册”

Java研发军团  
<https://www.ycbbs.vip>