

## 一、选择题

- 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）。  
 A. request.getParameter (name)  
**B. request.getParameter ( " name" )**  
 C. request.getParameterValues (name)  
 D. request.getParameterValues ( "name" )
- 使用 response 对象进行重定向时，使用的方法是（ ）。  
 A. getAttribute      B. setContentType  
**C. sendRedirect**      D. setAttribute
- session 对象中用于设定指定名字的属性值，并且把它存储在 session 对象中的方法是（ ）。  
**A. setAttribute**      B. getAttributeNames  
 C. getValue      D. getAttribute
- 在 application 对象中用（ ）方法可以获得 application 对象中的所有变量名。  
 A. getServerInfo      **B. nextElements ()**  
 C. removeAttribute      D. getRealPath
- 要使用 JSTL 的核心标签库，需要在 JSP 源文件的首部加入如下声明语句（ ）  
**A. <%@ taglib prefix="c"**  
**uri="http://java.sun.com/jsp/jstl/core" %>**  
 B. <%@ taglib prefix=" x"  
 uri=" http://java.sun.com/jsp/jstl/xml " %>  
 C. <%@ taglib prefix=" fmt "  
 uri=" http://java.sun.com/jsp/jstl/fmt " %>  
 D. <%@ taglib prefix=" sql "  
 uri=" http://java.sun.com/jsp/jstl/sql " %>
- 以下（ ）标签用于实现循环功能，类似与 Java 语句中的 for 循环。  
 A. <c:set>      **B. <c:forEach>**  
 C.<c:Tokens>      D. <c:import>
- 在编写 Servlet 时，需要继承\_\_HttpServlet\_\_类，在 Servlet 中声明 doGet () 和 doPost () 需要 request 和 response 类型的两个参数。
- jsp 主要内置对象有： application 、 exception 、 pageContext 、 request 、 response 、 session 、 out、config、page。
- 下列关于 HTTP 协议说法正确的是。ABCDE  
 A) HTTP 是一种请求/响应式的协议  
 B) HTTP 请求消息中 Accept 表示浏览器可接受的 MIME 类型  
 C) HTTP 请求消息中 Accept-Encoding 表示浏览器能够进行解码的数据编码方式  
 D) HTTP 请求消息中 Accept-Language 表示浏览器所希望的语言种类  
 E) HTTP 请求消息中 Host 表示初始 URL 中的主机和端口。

10. 下面对 out 对象说法错误的是 。 B

- A) out 对象用于输出数据
- B) out 对象的范围是 application。
- C) 如果 page 指令选择了 autoflush="true"，那么当出现由于当前的操作不清空缓存而造成缓冲区溢出的情况时，这个类的所有 I/O 操作会自动清空缓冲区的内容。
- D) out.newLine() 方法用来输出一个换行符
- E) out.close() 方法用来关闭输出流

11. 下面关于 request 对象说法错误的是 。 E

- A) request 对象是 ServletRequest 的一个实例
- B) 当客户端请求一个 JSP 网页时，JSP 引擎会将客户端的请求信息包装在这个 request 对象中
- C) getParameterName() 方法返回本次请求的参数名字
- D) getParameter() 方法返回包含指定参数的单独值的字符串
- E) getServerName() 返回接收请求的服务器的主机名和端口号

12. 下面关于 session 对象说法中正确的是 。 ABCD

- A) session 对象的类是 HttpSession。HttpSession 由服务器的程序实现
- B) session 对象提供 HTTP 服务器和 HTTP 客户端之间的会话
- C) session 可以用来储存访问者的一些特定信息
- D) session 可以创建访问者信息容器。
- E) 当用户在应用程序的页之间跳转时，存储在 session 对象中的变量会清除

13. 下面关于 pageContext 对象说法中正确的是 。 ABCDE

- A) pageContext 对象为 JSP 页面包装页面的上下文。
- B) pageContext 对象创建和初始化都是由容器来完成的
- C) getRequest() 方法返回当前的 request 对象
- D) getSession() 方法返回当前页面的 session 对象。
- E) removeAttribute() 方法用来删除默认页面范围或特定范围之中的已命名对象。

14. 下列关于 Application 对象说法中错误的是 。 B

- A) Application 对象用于在多个程序中保存信息
- B) Application 对象用来在所有用户间共享信息，但不可以在 Web 应用程序运行期间持久地保持数据。
- C) getAttribute(String name) 方法返回由 name 指定的名字 application 对象的属性的值。
- D) getAttributeNames() 方法返回所有 application 对象的属性的名字。
- E) setAttribute(String name , Object object) 方法设置指定名字 name 的 application 对象的属性值  
object

15. 在 JSP 文件中加载动态页面可以用 指令。 AB

- A) <%@ include file="fileName" %>指令

- B) <jsp:include>指令
- C) page 指令
- D) <jsp:forward>指令
- E) Taglib 指令

16. 如果想在页面中使用一个 JavaBean 我们可以使用 指令。 BD

- A) <%@ include file="fileName" %>
- B) page
- C) <jsp:forward>
- D) <jsp:useBean>
- E) 以上选项全都正确

17. Servlet 的生命周期包括下列哪几个阶段 。ABCDE

- A) 装载 Servlet
- B) 创建一个 Servlet 实例
- C) 调用 init() 方法
- D) 激活 Service() 方法，并传递请求和响应对象
- E) 调用 destroy() 方法来销毁 Servlet

18. 下列对 HttpServlet 类描述错误的是 。 E

- A) HttpServlet 类是针对使用 Http 协议的 Web 服务器的 Servlet 类
- B) HttpServlet 类通过执行 Servlet 借口，能够提供 Http 协议的功能
- C) HttpServlet 的子类实现了 doGet() 方法去响应 HTTP 的 Get 请求
- D) HttpServlet 的子类实现了 doPost() 方法去响应 HTTP 的 Post 请求
- E) HttpServlet 类通过 init() 方法和 destory() 方法管理 Servlet 自身的资源

19. 下列对于 web.xml 的配置说法错误的是 。 E

- A) 在 web.xml 描述中，首先要声明 Servlet
- B) 在 web.xml 描述中，要指定这个 Servlet 的名字
- C) 在 web.xml 描述中，要指定这个 Servlet 的类
- D) 在 web.xml 描述中，要为 Servlet 做 URI 映射
- E) 在 web.xml 中不可同时指定多个 Servlet

20. Servlet 通过什么接口可以实现请求转发机制 。 B

- A) HttpServletRequest 接口
- B) RequestDispatcher 接口
- C) HttpServletResponse 接口
- D) ServletConfig 接口
- E) Filter 接口

21. 下面关于 RequestDispatcher.forward() 方法说法正确的是 。 ABC

- A) Servlet 类使用 RequestDispatcher.forward() 方法来转发他所收到的 HTTP 请求

- B) 在调用 forward() 方法之前，不能有内容已经被实际输出到了客户端
- C) 该方法允许一个 Servlet 对一个请求作初步处理，而另一个资源将产生响应
- D) 该方法能包含服务器端的程序
- E) 以上都正确

22. 下列那种方法能实现请求转发 。 ABC

- A) 通过 ServletRequest 接口提供的 getRequestDispatcher(String path) 方法
- B) 通过 ServletContext 接口提供的 getRequestDispatcher(String path) 方法
- C) 通过 ServletContext 接口提供的 getNamedDispatche (String name) 方法
- D) 通过 HttpServletResponse 接口提供的 sendRedirect (String url) 方法
- E) 以上都可以

23. 下列说法正确的有 。 ADE

- A) forward 方法请求转发的组件只能是同一个 WEB 应用中
- B) RequestDispatcher 对象指定的 URL 是相对于整个 WEB 站点的根目录
- C) sendRedirect 方法中的 URL 是相对于当前 WEB 应用程序的根目录
- D) RequestDispatcher 对象指定的 URL 是相对于当前 WEB 应用程序的根目录
- E) sendRedirect 方法中的 URL 是相对于整个 WEB 站点的根目录

24. 关于请求转发和请求重定向之间的区别有哪些 。 ABCDE

- A) RequestDispatcher 对象指定的 URL 是相对于当前 WEB 应用程序的根目录，而 sendRedirect 方法中的 URL 是相对于整个 WEB 站点的根目录。
- B) 调用 sendRedirect 方法重定向的访问过程结束后，浏览器地址栏中显示的 URL 会改变；而调用 forward 方法就不会改变
- C) sendRedirect 方法响应的结果就是告诉浏览器去重新发出对另外一个 URL 的访问请求。forward 方法在服务器端内部将请求转发给另外一个资源。
- D) forward 方法的 request 对象和 response 对象属于同一个访问请求和响应过程；而 sendRedirect 方法的 request 对象和 response 对象属于两个独立的访问请求和响应过程
- E) 无论是 forward 方法，还是 sendRedirect 方法，在调用它们之前，都不能有内容 已经被实际输出到了客户端

## 二、简答题

1. 简述 JSP 中动态 include 与静态 include 的区别？

答：动态 INCLUDE 用 jsp:include 动作实现

```
<jsp:include page="included.jsp" flush="true" />
```

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

```
<%@ include file="included.htm" %>
```

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

## 2. 简述 servlet 的工作原理和生命周期

生命周期：Servlet 是一种可以在 Servlet 容器中运行的组件，那么理所当然就应该有一个从创建到销毁的过程，这个过程我们可以称之为 Servlet 生命周期。Servlet 的生命周期可以分为加载、实例化、初始化、处理客户请求和卸载五个阶段，体现在方法上主要是 `init()`、`service()` 和 `destroy()` 三个方法。生命周期的具体说明如下：

- Servlet 容器完成加载 Servlet 类和实例化一个 Servlet 对象
- `init()` 方法完成初始化工作，该方法由 Servlet 容器调用完成
- `service()` 方法处理客户端请求，并返回响应结果
- `destroy()` 方法在 Servlet 容器卸载 Servlet 之前被调用，释放一些资源

工作原理：Servlet 是运行在 Servlet 容器中的，由 Servlet 容器来负责 Servlet 实例的查找、创建以及整个生命周期的管理，Servlet 整个生命周期可以分为四个阶段：类装载及实例创建阶段、实例初始化阶段、服务阶段以及实例销毁阶段。

类装载及实例创建阶段

默认情况下，Servlet 实例是在接受到第一个请求时进行创建并且以后的请求进行复用，如果有 Servlet 实例需要进行一些复杂的操作，需要在初始化时就完成，比如打开文件、初始化网络连接等，可以配置在服务器启动时就创建实例，具体配置方法为在声明 `servlet` 标签中添加 `<load-on-startup>1</load-on-startup>` 标签。

初始化 `init(ServletConfig config)`

一旦 Servlet 实例被创建，将会调用 Servlet 的 `init` 方法，同时传入 `ServletConfig` 实例，传入 Servlet 的相关配置信息，`init` 方法在整个 Servlet 生命周期中只会调用一次。

服务 `service()`

为了提高效率，Servlet 规范要求一个 Servlet 实例必须能够同时服务于多个客户端请求，即 `service()` 方法运行在多线程的环境下，Servlet 开发者必须保证该方法的线程安全性。

销毁 `destroy()`

当 Servlet 容器将决定结束某个 Servlet 时，将会调用 `destroy()` 方法，在 `destroy` 方法中进行资源释放，一旦 `destroy` 方法被调用，Servlet 容器将不会再发送任何请求给这个实例，若 Servlet 容器需再次使用该 Servlet，需重新再实例化该 Servlet 实例。

### 3. (1) 这 4 个对象的生命周期？

生命周期就是指对象的创建到销毁的期间

**page: jsp** 页面被执行，生命周期开始，**jsp** 页面执行完毕，生命周期结束

**request**：用户发送一个请求，开始，服务器返回响应，请求结束，生命周期结束

**session**：用户打开浏览器访问，创建 **session**(开始)，**session** 超时或被声明失效，该对象生命周期结束

**application**：**web** 应用加载的时候创建(开始)，**web** 应用被移除或服务器关闭，对象销毁(结束)

## （2）四个域的作用范围

什么是域？为什么把这 4 个对象叫做域对象呢？

域：即范围的意思

**web** 中的域对象，可以存储对象，在作用范围内都可以取到

内部是 **Map** 集合的实现 **Map<String, Object>**

**page**：只在当前 **jsp** 页面有效

**request**：只在当前请求有效，每次请求分别对应不同的 **request** 域对象

// **session**：默认情况下，同一个浏览器来访问有效(发送同一个 **sessionid**)

**session**：只在一次会话中有效，会话结束就无法取到数据了（特殊情况,发送 **Cookie**）

**application**：在一个 **web** 应用中有效（只要服务器不关,web 应用不移除就可以取数据）

四个域对象的范围由小到大排列依次为：**page request session application**

## （3）哪种情况下用哪种域对象。

原则：四个域对象在选择的时候，能用范围小的绝不用范围大的

**page**：数据只是暂时存在集合，在 **jsp** 页面的其他地方要用，用 **page**(页面中自定义的 **map**)

什么时候需要用 **map** 了，就用 **page**

**request**：数据只是做显示的，看完了就没用了，就存 **request** 域

请求转发，**Servlet** 产生的处理结果(数据)交给 **jsp** 显示，

**session**：数据给用户看完了，一会还要用，会话结束了就没用了

用户登陆，用户信息发给客户端看，看完了，一会访问别的页面还要看用户信息

购物车，购物成功了，给用户看购物车，待会随时可以查看购物车

请求重定向，因为是两次请求，第一次请求的数据，第二次请求还要看

**application**：数据给一个用户用完了，别人还要用

聊天室，聊天记录，需要给所有的用户看

统计网站在线人数，所有人看到的应该是一个数

总结： 需要定义 Map 不如用 page，请求转发 Servlet 带给 jsp 的数据存 request

请求重定向带过去的数据存 Session，全局的数据存 application