

线程编程方面

60、java中有几种方法可以实现一个线程？用什么关键字修饰同步方法？stop()和suspend()方法为何不推荐使用？

答：有两种实现方法，分别是继承Thread类与实现Runnable接口

用synchronized关键字修饰同步方法

反对使用stop()，是因为它不安全。它会解除由线程获取的所有锁定，而且如果对象处于一种不连贯状态，那么其他线程能在那种状态下检查和修改它们。结果很难检查出真正的问题所在。suspend()方法容易发生死锁。调用suspend()的时候，目标线程会停下来，但却仍然持有在这之前获得的锁定。此时，其他任何线程都不能访问锁定的资源，除非被“挂起”的线程恢复运行。对任何线程来说，如果它们想恢复目标线程，同时又试图使用任何一个锁定的资源，就会造成死锁。所以不应该使用suspend()，而应在自己的Thread类中置入一个标志，指出线程应该活动还是挂起。若标志指出线程应该挂起，便用wait()命其进入等待状态。若标志指出线程应当恢复，则用一个notify()重新启动线程。

61、sleep() 和 wait() 有什么区别？

答：sleep是线程类(Thread)的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep不会释放对象锁。

wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

62、同步和异步有何异同，在什么情况下分别使用他们？举例说明。

答：如果数据将在线程间共享。例如正在写的以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。

当应用程序在对象上调用了需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

63、启动一个线程是用run()还是start()？

答：启动一个线程是调用start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由JVM调度并执行。这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

64、当一个线程进入一个对象的一个synchronized方法后，其它线程是否可进入此对象的其它方法？

答：不能，一个对象的一个synchronized方法只能由一个线程访问。

65、请说出你所知道的线程同步的方法。

答：wait():使一个线程处于等待状态，并且释放所持有的对象的lock。

sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉InterruptedException异常。

notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由JVM确定唤醒哪个线程，而且不是按优先级。

Allnotity():唤醒所有处于等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

66、多线程有几种实现方法,都是什么?同步有几种实现方法,都是什么?

答：多线程有两种实现方法，分别是继承Thread类与实现Runnable接口

同步的实现方面有两种，分别是synchronized,wait与notify

67、线程的基本概念、线程的基本状态以及状态之间的关系

答：线程指在程序执行过程中，能够执行程序代码的一个执行单位，每个程序至少都有一个线程，也就是程序本身。

Java中的线程有四种状态分别是：运行、就绪、挂起、结束

68、简述synchronized和java.util.concurrent.locks.Lock的异同？

答：主要相同点：Lock能完成synchronized所实现的所有功能

主要不同点：Lock有比synchronized更精确的线程语义和更好的性能。synchronized会自动释放锁，而Lock一定要求程序员手工释放，并且必须在finally从句中释放。

Jsp方面

69、forward 和redirect的区别

答: 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, 浏览器根本不知道服务器发送的内容是从哪儿来的, 所以它的地址栏中还是原来的地址。

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, 一般来说浏览器会用刚才请求的所有参数重新请求, 所以session, request参数都可以获取。

70、jsp有哪些内置对象?作用分别是什么?

答: JSP共有以下9种基本内置组件(可与ASP的6种内部组件相对应):

request 用户端请求, 此请求会包含来自GET/POST请求的参数

response 网页传回用户端的回应

pageContext 网页的属性是在这里管理

session 与请求有关的会话期

application servlet 正在执行的内容

out 用来传送回应的输出

config servlet的构架部件

page JSP网页本身

exception 针对错误网页, 未捕捉的例外

71、jsp有哪些动作?作用分别是什么?

答: 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

72、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

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

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

73、两种跳转方式分别是什么?有什么区别?

答: 有两种, 分别为:

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

```
<jsp:forward page="nextpage.jsp"/>
```

前者页面不会转向include所指的页面, 只是显示该页的结果, 主页面还是原来的页面。执行完后还会回来, 相当于函数调用。并且可以带参数。后者完全转向新页面, 不会再回来。相当于go to 语句。

74、JSP的内置对象及方法。

答: 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

Servlet方面

75、说一说Servlet的生命周期?

答: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被服务器实例化后,容器运行其init方法,请求到达时运行其service方法,service方法自动派遣运行与请求对应的doXXX方法(doGet, doPost)等,当服务器决定将实例销毁的时候调用其destroy方法。

与cgi的区别在于servlet处于服务器进程中,它通过多线程方式运行其service方法,一个实例可以服务于多个请求,并且其实例一般不会销毁,而CGI对每个请求都产生新的进程,服务完成后就销毁,所以效率上低于servlet。

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

答:前者仅是容器中控制权的转向,在客户端浏览器地址栏中不会显示出转向后的地址;后者则是完全的跳转,浏览器将会得到跳转的地址,并重新发送请求链接。这样,从浏览器的地址栏中可以看到跳转后的链接地址。所以,前者更加高效,在前者可以满足需要时,尽量使用forward()方法,并且,这样也有助于隐藏实际的链接。在有些情况下,比如,需要跳转到一个其它服务器上的资源,则必须使用sendRedirect()方法。

77、Servlet的基本架构

答:

```
public class ServletName extends HttpServlet {  
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws  
        ServletException, IOException {  
    }  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws  
        ServletException, IOException {  
    }  
}
```

78、什么情况下调用doGet()和doPost()?

答:Jsp页面中的form标签里的method属性为get时调用doGet(),为post时调用doPost()。

79、servlet的生命周期

答:web容器加载servlet,生命周期开始。通过调用servlet的init()方法进行servlet的初始化。通过调用service()方法实现,根据请求的不同调用不同的do***()方法。结束服务,web容器调用servlet的destroy()方法。

80、如何现实servlet的单线程模式

答:<%@ page isThreadSafe="false"%>

81、页面间对象传递的方法

答:request, session, application, cookie等

82、JSP和Servlet有哪些相同点和不同点,他们之间的联系是什么?

答:JSP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。

83、四种会话跟踪技术

答:会话作用域ServletsJSP 页面描述

page否是代表与一个页面相关的对象和属性。一个页面由一个编译好的Java servlet类(可以带有任何的include指令,但是没有include动作)表示。这既包括servlet又包括被编译成servlet的JSP页面

request是代表与 Web 客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面，涉及多个 Web 组件（由于 forward 指令和 include 动作的关系）

session是代表与用于某个 Web 客户机的一个用户体验相关的对象和属性。一个 Web 会话可以也经常跨越多个客户机请求

application是代表与整个 Web 应用程序相关的对象和属性。这实质上是跨越整个 Web 应用程序，包括多个页面、请求和会话的一个全局作用域

84、Request对象的主要方法

答：

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

getAttributeNames(): 返回request对象所有属性的名字集合，结果是一个枚举的实例

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

getContentLength(): 返回请求的Body的长度

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

getInputStream(): 返回请求的输入流，用于获得请求中的数据

getMethod(): 获得客户端向服务器端传送数据的方法

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

getProtocol(): 获取客户端向服务器端传送数据所依据的协议名称

getQueryString(): 获得查询字符串

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

85、我们在web应用开发过程中经常遇到输出某种编码的字符，如iso8859-1等，如何输出一个某种编码的字符串？

答：

```
Public String translate (String str) {  
    String tempStr = "";  
    try {  
        tempStr = new String(str.getBytes("ISO-8859-1"), "GBK");  
        tempStr = tempStr.trim();  
    }  
    catch (Exception e) {  
        System.err.println(e.getMessage());  
    }  
    return tempStr;  
}
```


86、Servlet执行时一般实现哪几个方法？

答：

```
public void init(ServletConfig config)
public ServletConfig getServletConfig()
public String getServletInfo()
public void service(ServletRequest request,ServletResponse response)
public void destroy()
```

Jdbc、Jdo方面

87、Class.forName的作用？为什么要用？

答：调用该访问返回一个以字符串指定类名的类的对象。

88、Jdo是什么？

答：JDO是Java对象持久化的新的规范，为java data

object的简称,也是一个用于存取某种数据仓库中的对象的标准API。JDO提供了透明的对象存储，因此对开发人员来说，存储数据对象完全不需要额外的代码（如JDBC

API的使用）。这些繁琐的例行工作已经转移到JDO产品提供商身上，使开发人员解脱出来，从而集中时间和精力在业务逻辑上。另外，JDO很灵活，因为它可以在任何数据底层上运行。JDBC只是面向关系数据库（RDBMS）JDO更通用，提供到任何数据底层的存储功能，比如关系数据库、文件、XML以及对象数据库（ODBMS）等等，使得应用可移植性更强。

89、说出数据连接池的工作机制是什么？

答：J2EE服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其标记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接标记为空闲，其他调用就可以使用这个连接。

90、Jdo是什么？

答：JDO是Java对象持久化的新的规范，为java data object的简称,也是一个用于存取某种数据仓库中的对象的标准API。JDO提供了透明的对象存储，因此对开发人员来说，存储数据对象完全不需要额外的代码（如JDBC API的使用）

。这些繁琐的例行工作已经转移到JDO产品提供商身上，使开发人员解脱出来，从而集中时间和精力在业务逻辑上。另外，JDO很灵活，因为它可以在任何数据底层上运行。JDBC只是面向关系数据库（RDBMS）JDO更通用，提供到任何数据底层的存储功能，比如关系数据库、文件、XML以及对象数据库（ODBMS）等等，使得应用可移植性更强。

Xml方面

91、xml有哪些解析技术？区别是什么？

答：有DOM,SAX,STAX等

DOM:处理大型文件时其性能下降的非常厉害。这个问题是由DOM的树结构所造成的，这种结构占用的内存较多，而且DOM必须在解析文件之前把整个文档装入内存,适合对XML的随机访问SAX:不现于DOM,SAX是事件驱动型的XML解析方式。它顺序读取XML文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理XML文件，适合对XML的顺序访问

STAX:Streaming API for XML (StAX)

92、你在项目中用到了xml技术的哪些方面？如何实现的？

答：用到了数据存贮，信息配置两方面。在做数据交换平台时，将不能数据源的数据组装成XML文件，然后将XML文件压缩打包加密后通过网络传送给接收者，接收解密与解压缩后再同XML文件中还原相关信息进行处理。在做软件配置时，利用XML可以很方便的进行，软件的各种配置参数都存贮在XML文件中。

93、XML文档定义有几种形式？它们之间有何本质区别？解析XML文档有哪几种方式？

答：a: 两种形式 dtd schema，b: 本质区别:schema本身是xml的，可以被XML解析器解析(这也是从DTD上发展schema的根本目的)，c:有DOM,SAX,STAX等

DOM:处理大型文件时其性能下降的非常厉害。这个问题是由DOM的树结构所造成的，这种结构占用的内存较多，而且DOM必须在解析文件之前把整个文档装入内存,适合对XML的随机访问

SAX:不现于DOM,SAX是事件驱动型的XML解析方式。它顺序读取XML文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理XML文件，适合对XML的顺序访问

STAX:Streaming API for XML (StAX)

动力节点