题目：115 个 Java 面试题和答案——终极（下）

第一篇讨论了面向对象编程和它的特点，关于 Java 和它的功能的常见问题，Java 的集合类，垃圾收集器，本章主要讨论异常处理，Java 小应用程序，Swing，JDBC，远程方法调用(RMI)，Servlet 和 JSP。

异常处理

Java 小应用程序(Applet)

Swing

JDBC

远程方法调用（RMI）

Servlet

JSP

异常处理

43.Java 中的两种异常类型是什么？他们有什么区别？

Java 中有两种异常：受检查的(checked)异常和不受检查的(unchecked)异常。不受检查的异常不需要在方法或者是构造函数上声明，就算方法或者是构造函数的执行可能会抛出这样的异常，并且不受检查的异常可以传播到方法或者是构造函数的外面。相反，受检查的异常必须要用 throws 语句在方法或者是构造函数上声明。这里有 Java 异常处理的一些小建议。

44.Java 中 Exception 和 Error 有什么区别？

Exception 和 Error 都是 Throwable 的子类。Exception 用于用户程序可以捕获的异常情况。Error 定义了不期望被用户程序捕获的异常。

45.1 throw 和 throws 有什么区别？

throw 关键字用来在程序中明确的抛出异常，相反，throws 语句用来表明方法不能处理的异常。每一个方法都必须要指定哪些异常不能处理，所以方法的调用者才能够确保处理可能发生的异常，多个异常是用逗号分隔的。

45.2 异常处理的时候，finally 代码块的重要性是什么？

无论是否抛出异常，finally 代码块总是会被执行。就算是没有 catch 语句同时又抛出异常的

情况下，finally 代码块仍然会被执行。最后要说的是，finally 代码块主要用来释放资源，比如：I/O 缓冲区，数据库连接。

46.异常处理完成以后，Exception 对象会发生什么变化？

Exception 对象会在下一个垃圾回收过程中被回收掉。

47.finally 代码块和 finalize()方法有什么区别？

无论是否抛出异常，finally 代码块都会执行，它主要是用来释放应用占用的资源。finalize()方法是 Object 类的一个 protected 方法，它是在对象被垃圾回收之前由 Java 虚拟机来调用的。

Java 小应用程序(Applet)

48.什么是 Applet？

java applet 是能够被包含在 HTML 页面中并且能被启用了 java 的客户端浏览器执行的程序。Applet 主要用来创建动态交互的 web 应用程序。

49.解释一下 Applet 的生命周期

applet 可以经历下面的状态：

Init：每次被载入的时候都会被初始化。
Start：开始执行 applet。
Stop：结束执行 applet。
Destroy：卸载 applet 之前，做最后的清理工作。

50.当 applet 被载入的时候会发生什么？

首先，创建 applet 控制类的实例，然后初始化 applet，最后开始运行。

51.Applet 和普通的 Java 应用程序有什么区别？

applet 是运行在启用了 java 的浏览器中，Java 应用程序是可以在浏览器之外运行的独立的 Java 程序。但是，它们都需要有 Java 虚拟机。

进一步来说，Java 应用程序需要一个有特定方法签名的 main 函数来开始执行。Java applet 不需要这样的函数来开始执行。

最后，Java applet 一般会使用很严格的安全策略，Java 应用一般使用比较宽松的安全策略。

52.Java applet 有哪些限制条件？

主要是由于安全的原因，给 applet 施加了以下的限制：

applet 不能够载入类库或者定义本地方法。
applet 不能在宿主机上读写文件。
applet 不能读取特定的系统属性。
applet 不能发起网络连接，除非是跟宿主机。
applet 不能够开启宿主机上其他任何的程序。

53.什么是不受信任的 applet？

不受信任的 applet 是不能访问或是执行本地系统文件的 Java applet，默认情况下，所有下载的 applet 都是不受信任的。

54.从网络上加载的 applet 和从本地文件系统加载的 applet 有什么区别？

当 applet 是从网络上加载的时候，applet 是由 applet 类加载器载入的，它受 applet 安全管理器的限制。

当 applet 是从客户端的本地磁盘载入的时候，applet 是由文件系统加载器载入的。

从文件系统载入的 applet 允许在客户端读文件，写文件，加载类库，并且也允许执行其他程序，但是，却通不过字节码校验。

55.applet 类加载器是什么？它会做哪些工作？

当 applet 是从网络上加载的时候，它是由 applet 类加载器载入的。类加载器有自己的 java 名称空间等级结构。类加载器会保证来自文件系统的类有唯一的名称空间，来自网络资源的类有唯一的名称空间。

当浏览器通过网络载入 applet 的时候，applet 的类被放置于和 applet 的源相关联的私有的名称空间中。然后，那些被类加载器载入进来的类都是通过了验证器验证的。验证器会检查类文件格式是否遵守 Java 语言规范，确保不会出现堆栈溢出(stack overflow)或者下溢(underflow)，传递给字节码指令的参数是正确的。

56.applet 安全管理器是什么？它会做哪些工作？

applet 安全管理器是给 applet 施加限制条件的一种机制。浏览器可以只有一个安全管理器。安全管理器在启动的时候被创建，之后不能被替换覆盖或者是扩展。

Swing

57.弹出式选择菜单(Choice)和列表(List)有什么区别

Choice 是以一种紧凑的形式展示的，需要下拉才能看到所有的选项。Choice 中一次只能选中

一个选项。List 同时可以有多个元素可见，支持选中一个或者多个元素。

58.什么是布局管理器？

布局管理器用来在容器中组织组件。

59.滚动条(Scrollbar)和滚动面板(JScrollPane)有什么区别？

Scrollbar 是一个组件，不是容器。而 ScrollPane 是容器。ScrollPane 自己处理滚动事件。

60.哪些 Swing 的方法是线程安全的？

只有 3 个线程安全的方法： repaint(), revalidate(), and invalidate()。

61.说出三种支持重绘(painting)的组件。

Canvas, Frame, Panel,和 Applet 支持重绘。

62.什么是裁剪(clipping)？

限制在一个给定的区域或者形状的绘图操作就做裁剪。

63.MenuItem 和 CheckboxMenuItem 的区别是什么？

CheckboxMenuItem 类继承自 MenuItem 类，支持菜单选项可以选中或者不选中。

64.边缘布局(BorderLayout)里面的元素是如何布局的？

BorderLayout 里面的元素是按照容器的东西南北中进行布局的。

65.网格包布局(GridBagLayout)里面的元素是如何布局的？

GridBagLayout 里面的元素是按照网格进行布局的。不同大小的元素可能会占据网格的多于 1 行或一列。因此，行数和列数可以有不同的大小。

66.Window 和 Frame 有什么区别？

Frame 类继承了 Window 类，它定义了一个可以有菜单栏的主应用窗口。

67.裁剪(clipping)和重绘(repainting)有什么联系？

当窗口被 AWT 重绘线程进行重绘的时候，它会把裁剪区域设置成需要重绘的窗口的区域。

68.事件监听器接口(event-listener interface)和事件适配器(event-adapter)有什么关系？

事件监听器接口定义了对特定的事件，事件处理器必须要实现的方法。事件适配器给事件监听器接口提供了默认的实现。

69.GUI 组件如何来处理它自己的事件？

GUI 组件可以处理它自己的事件，只要它实现相对应的事件监听器接口，并且把自己作为事件监听器。

70.Java 的布局管理器比传统的窗口系统有哪些优势？

Java 使用布局管理器以一种一致的方式在所有的窗口平台上摆放组件。因为布局管理器不会和组件的绝对大小和位置相绑定，所以他们能够适应跨窗口系统的特定平台的不同。

71.Java 的 Swing 组件使用了哪种设计模式？

Java 中的 Swing 组件使用了 MVC(视图-模型-控制器)设计模式。

JDBC

72.什么是 JDBC？

JDBC 是允许用户在不同数据库之间做选择的一个抽象层。JDBC 允许开发者用 JAVA 写数据库应用程序，而不需要关心底层特定数据库的细节。

73.解释下驱动(Driver)在 JDBC 中的角色。

JDBC 驱动提供了特定厂商对 JDBC API 接口类的实现，驱动必须要提供 java.sql 包下面这些类的实现：Connection, Statement, PreparedStatement,CallableStatement, ResultSet 和 Driver。

74.Class.forName()方法有什么作用？

这个方法用来载入跟数据库建立连接的驱动。

75.PreparedStatement 比 Statement 有什么优势？

PreparedStatements 是预编译的，因此，性能会更好。同时，不同的查询参数值，PreparedStatement 可以重用。

76.什么时候使用 CallableStatement？用来准备 CallableStatement 的方法是什么？

CallableStatement 用来执行存储过程。存储过程是由数据库存储和提供的。存储过程可以接受输入参数，也可以有返回结果。非常鼓励使用存储过程，因为它提供了安全性和模块化。准备一个 CallableStatement 的方法是：

CallableStament.prepareCall();

77.数据库连接池是什么意思？

像打开关闭数据库连接这种和数据库的交互可能是很费时的，尤其是当客户端数量增加的时候，会消耗大量的资源，成本是非常高的。可以在应用服务器启动的时候建立很多个数据库连接并维护在一个池中。连接请求由池中的连接提供。在连接使用完毕以后，把连接归还到池中，以用于满足将来更多的请求。

远程方法调用(RMI)

78.什么是 RMI？

Java 远程方法调用(Java RMI)是 Java API 对远程过程调用(RPC)提供的面向对象的等价形式，支持直接传输序列化的 Java 对象和分布式垃圾回收。远程方法调用可以看做是激活远程正在运行的对象上的方法的步骤。RMI 对调用者是位置透明的，因为调用者感觉方法是执行在本地运行的对象上的。看下 RMI 的一些注意事项。

79.RMI 体系结构的基本原则是什么？

RMI 体系结构是基于一个非常重要的行为定义和行为实现相分离的原则。RMI 允许定义行为的代码和实现行为的代码相分离，并且运行在不同的 JVM 上。

80.RMI 体系结构分哪几层？

RMI 体系结构分以下几层：

存根和骨架层(Stub and Skeleton layer)：这一层对程序员是透明的，它主要负责拦截客户端发出的方法调用请求，然后把请求重定向给远程的 RMI 服务。

远程引用层(Remote Reference Layer)：RMI 体系结构的第二层用来解析客户端对服务端远程对象的引用。这一层解析并管理客户端对服务端远程对象的引用。连接是点到点的。

传输层(Transport layer)：这一层负责连接参与服务的两个 JVM。这一层是建立在网络上机器间的 TCP/IP 连接之上的。它提供了基本的连接服务，还有一些防火墙穿透策略。

81.RMI 中的远程接口(Remote Interface)扮演了什么样的角色？

远程接口用来标识哪些方法是可以被非本地虚拟机调用的接口。远程对象必须要直接或者是间接实现远程接口。实现了远程接口的类应该声明被实现的远程接口，给每一个远程对象定义构造函数，给所有远程接口的方法提供实现。

82.java.rmi.Naming 类扮演了什么样的角色？

java.rmi.Naming 类用来存储和获取在远程对象注册表里面的远程对象的引用。Naming 类的

每一个方法接收一个 URL 格式的 String 对象作为它的参数。

83.RMI 的绑定(Binding)是什么意思？

绑定是为了查询找远程对象而给远程对象关联或者是注册以后会用到的名称的过程。远程对象可以使用 Naming 类的 bind()或者 rebind()方法跟名称相关联。

84.Naming 类的 bind()和 rebind()方法有什么区别？

bind()方法负责把指定名称绑定给远程对象，rebind()方法负责把指定名称重新绑定到一个新的远程对象。如果那个名称已经绑定过了，先前的绑定会被替换掉。

85.让 RMI 程序能正确运行有哪些步骤？

为了让 RMI 程序能正确运行必须要包含以下几个步骤：

编译所有的源文件。
使用 rmic 生成 stub。
启动 rmiregistry。
启动 RMI 服务器。
运行客户端程序。

86.RMI 的 stub 扮演了什么样的角色？

远程对象的 stub 扮演了远程对象的代表或者代理的角色。调用者在本地 stub 上调用方法，它负责在远程对象上执行方法。当 stub 的方法被调用的时候，会经历以下几个步骤：

初始化到包含了远程对象的 JVM 的连接。
序列化参数到远程的 JVM。
等待方法调用和执行的结果。
反序列化返回的值或者是方法没有执行成功情况下的异常。
把值返回给调用者。

87.什么是分布式垃圾回收(DGC)？它是如何工作的？

DGC 叫做分布式垃圾回收。RMI 使用 DGC 来做自动垃圾回收。因为 RMI 包含了跨虚拟机的远程对象的引用，垃圾回收是很困难的。DGC 使用引用计数算法来给远程对象提供自动内存管理。

88.RMI 中使用 RMI 安全管理器(RMISecurityManager)的目的是什么？

RMISecurityManager 使用下载好的代码提供可被 RMI 应用程序使用的安全管理器。如果没有设置安全管理器，RMI 的类加载器就不会从远程下载任何的类。

89.解释下 Marshalling 和 demarshalling。

当应用程序希望把内存对象跨网络传递到另一台主机或者是持久化到存储的时候，就必须要把对象在内存里面的表示转化成合适的格式。这个过程就叫做 Marshalling，反之就是 demarshalling。

90.解释下 Serialization 和 Deserialization。

Java 提供了一种叫做对象序列化的机制，他把对象表示成一连串的字节，里面包含了对象的数据，对象的类型信息，对象内部的数据的类型信息等等。因此，序列化可以看成是为了把对象存储在磁盘上或者是从磁盘上读出来并重建对象而把对象扁平化的一种方式。反序列化是把对象从扁平状态转化成活动对象的相反的步骤。

Servlet

91.什么是 Servlet？

Servlet 是用来处理客户端请求并产生动态网页内容的 Java 类。Servlet 主要是用来处理或者是存储 HTML 表单提交的数据，产生动态内容，在无状态的 HTTP 协议下管理状态信息。

92.说一下 Servlet 的体系结构。

所有的 Servlet 都必须要实现的核心的接口是 javax.servlet.Servlet。每一个 Servlet 都必须要直接或者是间接实现这个接口，或者是继承 javax.servlet.GenericServlet 或者 javax.servlet.http.HTTPServlet。最后，Servlet 使用多线程可以并行的为多个请求服务。

93.Applet 和 Servlet 有什么区别？

Applet 是运行在客户端主机的浏览器上的客户端 Java 程序。而 Servlet 是运行在 web 服务器上的服务端的组件。applet 可以使用用户界面类，而 Servlet 没有用户界面，相反，Servlet 是等待客户端的 HTTP 请求，然后为请求产生响应。

94.GenericServlet 和 HttpServlet 有什么区别？

GenericServlet 是一个通用的协议无关的 Servlet，它实现了 Servlet 和 ServletConfig 接口。继承自 GenericServlet 的 Servlet 应该要覆盖 service()方法。最后，为了开发一个能用在网页上服务于使用 HTTP 协议请求的 Servlet，你的 Servlet 必须要继承自 HttpServlet。这里有 Servlet 的例子。

95.解释下 Servlet 的生命周期。

对每一个客户端的请求，Servlet 引擎载入 Servlet，调用它的 init()方法，完成 Servlet 的初始化。然后，Servlet 对象通过为每一个请求单独调用 service()方法来处理所有随后来自客户端的请求，最后，调用 Servlet(译者注：这里应该是 Servlet 而不是 server)的 destroy()方法把 Servlet

删除掉。

96.doGet()方法和 doPost()方法有什么区别？

doGet：GET 方法会把名值对追加在请求的 URL 后面。因为 URL 对字符数目有限制，进而限制了用在客户端请求的参数值的数目。并且请求中的参数值是可见的，因此，敏感信息不能用这种方式传递。

doPOST：POST 方法通过把请求参数值放在请求体中来克服 GET 方法的限制，因此，可以发送的参数的数目是没有限制的。最后，通过 POST 请求传递的敏感信息对外部客户端是不可见的。

97.什么是 Web 应用程序？

Web 应用程序是对 Web 或者是应用服务器的动态扩展。有两种类型的 Web 应用：面向表现的和面向服务的。面向表现的 Web 应用程序会产生包含了很多种标记语言和动态内容的交互的 web 页面作为对请求的响应。而面向服务的 Web 应用实现了 Web 服务的端点(endpoint)。一般来说，一个 Web 应用可以看成是一组安装在服务器 URL 名称空间的特定子集下面的 Servlet 的集合。

98.什么是服务端包含(Server Side Include)？

服务端包含(SSI)是一种简单的解释型服务端脚本语言，大多数时候仅用在 Web 上，用 servlet 标签嵌入进来。SSI 最常用的场景把一个或多个文件包含到 Web 服务器的一个 Web 页面中。当浏览器访问 Web 页面的时候，Web 服务器会用对应的 servlet 产生的文本来替换 Web 页面中的 servlet 标签。

99.什么是 Servlet 链(Servlet Chaining)？

Servlet 链是把一个 Servlet 的输出发送给另一个 Servlet 的方法。第二个 Servlet 的输出可以发送给第三个 Servlet，依次类推。链条上最后一个 Servlet 负责把响应发送给客户端。

100.如何知道是哪一个客户端的机器正在请求你的 Servlet？

ServletRequest 类可以找出客户端机器的 IP 地址或者是主机名。getRemoteAddr()方法获取客户端主机的 IP 地址，getRemoteHost()可以获取主机名。看下这里的例子。

101.HTTP 响应的结构是怎么样的？

HTTP 响应由三个部分组成：

状态码(Status Code)：描述了响应的状态。可以用来检查是否成功的完成了请求。请求失败的情况下，状态码可用来找出失败的原因。如果 Servlet 没有返回状态码，默认会返回成功的状态码 HttpServletResponse.SC_OK。

HTTP 头部(HTTP Header)：它们包含了更多关于响应的信息。比如：头部可以指定认为响应过期的过期日期，或者是指定用来给用户安全的传输实体内容的编码格式。如何在 Serlet 中检索 HTTP 的头部看这里。

主体(Body)：它包含了响应的内容。它可以包含 HTML 代码，图片，等等。主体是由传输在 HTTP 消息中紧跟在头部后面的数据字节组成的。

102.什么是 cookie？session 和 cookie 有什么区别？

cookie 是 Web 服务器发送给浏览器的一块信息。浏览器会在本地文件中给每一个 Web 服务器存储 cookie。以后浏览器在给特定的 Web 服务器发请求的时候，同时会发送所有为该服务器存储的 cookie。下面列出了 session 和 cookie 的区别：

无论客户端浏览器做怎么样的设置，session 都应该能正常工作。客户端可以选择禁用 cookie，但是，session 仍然是能够工作的，因为客户端无法禁用服务端的 session。
在存储的数据量方面 session 和 cookies 也是不一样的。session 能够存储任意的 Java 对象，cookie 只能存储 String 类型的对象。

103.浏览器和 Servlet 通信使用的是什么协议？
浏览器和 Servlet 通信使用的是 HTTP 协议。

104.什么是 HTTP 隧道？

HTTP 隧道是一种利用 HTTP 或者是 HTTPS 把多种网络协议封装起来进行通信的技术。因此，HTTP 协议扮演了一个打通用于通信的网络协议的管道的包装器的角色。把其他协议的请求掩盖成 HTTP 的请求就是 HTTP 隧道。

105.sendRedirect()和 forward()方法有什么区别？

sendRedirect()方法会创建一个新的请求，而 forward()方法只是把请求转发到一个新的目标上。重定向(redirect)以后，之前请求作用域范围以内的对象就失效了，因为会产生一个新的请求，而转发(forwarding)以后，之前请求作用域范围以内的对象还是能访问的。一般认为 sendRedirect()比 forward()要慢。

106.什么是 URL 编码和 URL 解码？

URL 编码是负责把 URL 里面的空格和其他的特殊字符替换成对应的十六进制表示，反之就是解码。

JSP

107.什么是 JSP 页面？

JSP 页面是一种包含了静态数据和 JSP 元素两种类型的文本的文本文档。静态数据可以用任何基于文本的格式来表示，比如：HTML 或者 XML。JSP 是一种混合了静态内容和动态产生的内容的技术。这里看下 JSP 的例子。

108.JSP 请求是如何被处理的？

浏览器首先要请求一个以.jsp 扩展名结尾的页面，发起 JSP 请求，然后，Web 服务器读取这个请求，使用 JSP 编译器把 JSP 页面转化成一个 Servlet 类。需要注意的是，只有当第一次请求页面或者是 JSP 文件发生改变的时候 JSP 文件才会被编译，然后服务器调用 servlet 类，处理浏览器的请求。一旦请求执行结束，servlet 会把响应发送给客户端。这里看下如何在 JSP 中获取请求参数。

109.JSP 有什么优点？

下面列出了使用 JSP 的优点：

JSP 页面是被动态编译成 Servlet 的，因此，开发者可以很容易的更新展现代码。
JSP 页面可以被预编译。
JSP 页面可以很容易的和静态模板结合，包括：HTML 或者 XML，也可以很容易的和产生动态内容的代码结合起来。
开发者可以提供让页面设计者以类 XML 格式来访问的自定义的 JSP 标签库。
开发者可以在组件层做逻辑上的改变，而不需要编辑单独使用了应用层逻辑的页面。

110.什么是 JSP 指令(Directive)？JSP 中有哪些不同类型的指令？

Directive 是当 JSP 页面被编译成 Servlet 的时候，JSP 引擎要处理的指令。Directive 用来设置页面级别的指令，从外部文件插入数据，指定自定义的标签库。Directive 是定义在<%@ 和 %>之间的。下面列出了不同类型的 Directive：

包含指令(Include directive)：用来包含文件和合并文件内容到当前的页面。
页面指令(Page directive)：用来定义 JSP 页面中特定的属性，比如错误页面和缓冲区。
Taglib 指令： 用来声明页面中使用的自定义的标签库。

111.什么是 JSP 动作(JSP action)？

JSP 动作以 XML 语法的结构来控制 Servlet 引擎的行为。当 JSP 页面被请求的时候，JSP 动作会被执行。它们可以被动态的插入到文件中，重用 JavaBean 组件，转发用户到其他的页面，或者是给 Java 插件产生 HTML 代码。下面列出了可用的动作：

jsp:include-当 JSP 页面被请求的时候包含一个文件。
jsp:useBean-找出或者是初始化 Javabean。
jsp:setProperty-设置 JavaBean 的属性。
jsp:getProperty-获取 JavaBean 的属性。
jsp:forward-把请求转发到新的页面。
jsp:plugin-产生特定浏览器的代码。

112.什么是 Scriptlets？

JSP 技术中，scriptlet 是嵌入在 JSP 页面中的一段 Java 代码。scriptlet 是位于标签内部的所有的东西，在标签与标签之间，用户可以添加任意有效的 scriplet。

113.声明(Decalaration)在哪里？

声明跟 Java 中的变量声明很相似，它用来声明随后要被表达式或者 scriptlet 使用的变量。添加的声明必须要用开始和结束标签包起来。

114.什么是表达式(Expression)？

JSP 表达式是 Web 服务器把脚本语言表达式的值转化成一个 String 对象，插入到返回给客户端的数据流中。表达式是在<%=和%>这两个标签之间定义的。

115.隐含对象是什么意思？有哪些隐含对象？

JSP 隐含对象是页面中的一些 Java 对象，JSP 容器让这些 Java 对象可以为开发者所使用。开发者不用明确的声明就可以直接使用他们。JSP 隐含对象也叫做预定义变量。下面列出了 JSP 页面中的隐含对象：

application
page
request
response
session
exception
out
config
pageContext

祝：编程快乐！