****1、说出ArrayList,Vector, LinkedList 的存储性能和特性****

ArrayList 和Vector 都是使用数组方式存储数据,可以直接按序号索引，遍历效率高，但插入元素不如LinkedList快。Vector使用了synchronized 方法，性能较ArrayList差。LinkedList使用双向链表实现存储，插入和删除效率高，遍历慢。

****2、String 是最基本的数据类型吗？****

不是。基本数据类型包括：byte/short/int/long/float/double/char/boolean

**3、int 和Integer 有什么区别？**

int是原始数据类型，Integer是原始类型封装类，即引用类型。原始数据类型在堆栈上进行内存分配，存储的是这些变量的值，引用类型对象的具体信息存储在堆上，在栈上给其引用，存储的是指向存数据对象的内存地址。对象引用实例变量的缺省值为null，而原始类型实例变量的缺省值与它们的类型有关。

**4、String、StringBuffer、StringBuilder的区别？**

String：字符串常量，操作少量数据用

Stringbuffer：字符串变量，多线程操作字符串缓冲区，线程安全

StringBuilder：字符串变量，单线程操作字符串缓冲区，线程不安全但效率更高

**5、运行时异常与一般异常有何异同？**

异：运行时异常runtime表示虚拟机的通常操作中可能遇到的异常,编译时不会报错。非运行时异常checked也就是我们经常遇到的IO异常，以及SQL异常都是这种异常，在编译时就必须声明或抛出。

同：都是程序运行过程中可能出现的非正常状态。

**6、HashMap 和Hashtable 的区别？**

HashMap 是Hashtable都实现了Map 接口，主要区别在于HashMap 允许空键值，Hashtable 的方法是Synchronize 的，而HashMap 不是。Hashtable 继承自Dictionary 类，而HashMap 是Map interface 的一个实现。

**7、throw和throws的区别。**

Throws用于声明方法可能抛出的异常，写在方法名的后面，由该方法的调用者来处理。Throw用在方法体中，指语句抛出一个异常，由方法体内的语句处理。

**8、final, finally, finalize 的区别？**

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

**9、forward 和redirect 的区别？**

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。

**10、abstract class 和interface 有什么区别？**

1.接口中所有属性都是pubic static final修饰的，方法都是public abstract修饰的

2.抽象类中有构造方法，接口没有

3.抽象类中可以有具体实现的方法

4.一个类可以实现多个接口，但只能继承一个抽象类

**11、给我一个你最常见到的runtime exception。**

ArithmeticException, ArrayStoreException, BufferOverflowException,

BufferUnderflowException, CannotRedoException, CannotUndoException,

ClassCastException, CMMException, ConcurrentModificationException,

DOMException, EmptyStackException, IllegalArgumentException,

IllegalMonitorStateException, IllegalPathStateException,

IllegalStateException, ImagingOpException, IndexOutOfBoundsException,

MissingResourceException, NegativeArraySizeException,

NoSuchElementException, NullPointerException, ProfileDataException,

ProviderException, RasterFormatException, SecurityException,

SystemException, UndeclaredThrowableException,

UnmodifiableSetException, UnsupportedOperationException

ClassNotFoundException SQLException NullPointerException

**12、什么是事务，事物的特性？**

JAVA事务分三种：JDBC事务，JTA事务，容器事务。

是指作为单个逻辑工作单元执行的一系列操作，要么完全地执行，要么完全地不执行。

原子性（atomic）:一组操作不可分割

一致性(consistent):操作之前与操作之后状态不变

隔离性（isolation）:事务和事务之间相互隔离，互不干扰

持久性（durable）:事务一旦提交，其对数据库的改变是持久性的

**13、抽象关键字abstract和哪些不可以共存？**

final、private、static、native、synchronized

1. **多态的定义及实现方式。**

定义：同一引用，使用不同的对象，执行不同的方法

实现方式：将父类类型作为方法的参数；将父类类型作为方法的返回值类型

**15、Super()和this()是否可以同时出现在构造函数中？为什么一定要定义在第一行？**

不可以。Super()和this()都只能写在第一行，只能出现一个。

Super()和this()都是调用构造函数，构造函数用于初始化，初始化动作要先完成。

**16、手写一个单例模式例子**

public class Single {

// 构造方法私有化

private Single() { }

// 创建私有静态的类对象

private static Single s;

// 定义公有静态方法，返回该对象

public static Single getSingle() {

if (null == s) {

s = new Single();

}

return s;

}

}

**17、集合（List）和数组（Array）的区别**

数组是固定长度的，集合是可变长度的 。

数组可以存储基本数据类型和引用数据类型，集合只能存储引用数据类型

数组存储的元素必须是同一个数据类型，集合存储的对象可以是不同数据类型

1. **List、Set、Map的区别**

1.List和Set是Collection接口的实现，都是单列元素的集合。

2.List表示有先后顺序的集合，里面的元素可以重复，Set无序不能重复

3.Map是以键值对存在的双列的集合，无序不能重复。

**19、接口的优点**

对于修改关闭，对于扩展开放，便于协同开发

**20、当类被加载时，请简述类中的静态声明执行顺序。**

首先初始化静态属性，然后执行静态块，最后执行类加载时的构造函数。

**21、char 型变量中能不能存贮一个中文汉字?为什么?**

能够定义成为一个中文的，因为java 中以unicode 编码，一个char占2 个字节(Byte)共16 位（bit），所以放一个中文是没问题的。

**22、启动一个线程是用run()还是start()?**

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。

**23、什么情况下调用doGet()和doPost()？**

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()。

**24、short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?**

short s1 = 1; s1 = s1 + 1; （s1+1 运算结果是int 型，需要强制转换类型）short s1 = 1; s1 += 1;（可以正确编译）

**25、String s = new String("xyz");创建了几个String Object?**

两个，一个是编译时决定的，最后放在常量池中。一个是运行时放在堆里面的。两个都是"xyz"，s确实只是一个引用，本身不是对象，常量池中只会有一个“xyz”，堆中可以有多个。

**26、数组有没有length()这个方法? String 有没有length()这个方法？**

数组没有length()这个方法，有length 的属性。String 有有length()这个方法。

**27、构造器Constructor 是否可被override?**

构造器Constructor 不能被继承，因此不能重写Overriding，但可以被重载Overloading。

**28、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?**

分几种情况：

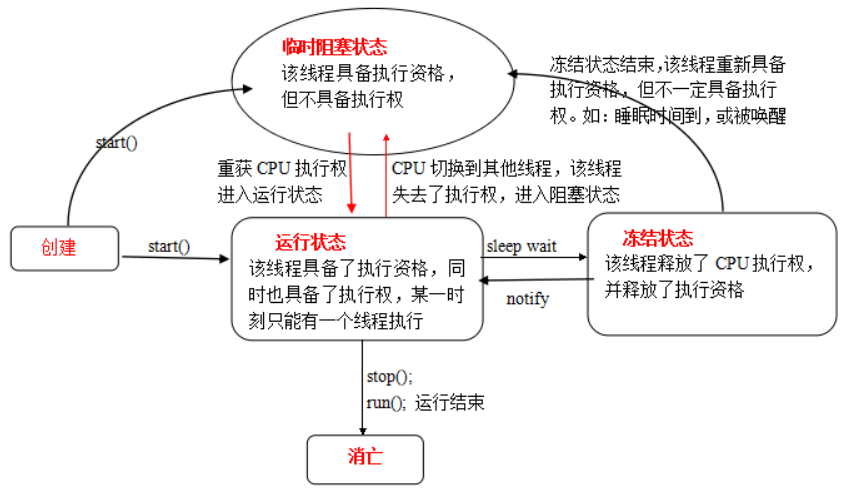
1. 其他方法前是否加了synchronized关键字，如果没加，则能。
2. 如果这个方法内部调用了wait，则可以进入其他synchronized方法。
3. 如果其他方法都加了synchronized关键字，并且内部没有调用wait，则不能。
4. 如果其他方法是static，它用的同步锁是当前类的字节码，与非静态的方法不能同步，因为非静态的方法用的是this。



**29、线程的基本概念、线程的基本状态以及状态之间的关系**

线程是CPU调度和分配的最小单位，是进程执行运算的基本单位。

基本状态：就绪 运行 等待 睡眠 阻塞 死亡（结束）

状态之间的关系：调用线程的start方法后线程进入就绪状态，线程调度系统将就绪状态的线程转为运行状态，遇到synchronized语句时，由运行状态转为阻塞，当synchronized获得锁后，由阻塞转为运行，在这种情况可以调用wait方法转为挂起状态，当线程关联的代码执行完后，线程变为结束状态。

**30、什么是多线程？什么是进程？什么是线程安全？线程同步的实现方式？同步和异步的区别？**

**多线程：**一个应用程序有一个以上的线程我们称之为多线程；

**进程：**进程是一个正在执行的程序

**线程安全：**多线程访问同一代码，不会产生不确定的结果。编写线程安全的代码是低依靠线程同步

**线程同步的7种方式：** 1、同步方法：public synchronized void save(){} 2、同步代码块：synchronized(object){} 3、使用特殊域变量(volatile)实现线程同步 4、使用重入锁实现线程同步 5、使用局部变量实现线程同步 6、使用阻塞队列实现线程同步 7、使用原子变量实现线程同步

**同步和异步：**同步就是顺序执行，执行完一个再执行下一个，需要等待、协调运行。异步就是彼此独立,在等待某事件的过程中继续做自己的事，不需要等待这一事件完成后再工作。

**31、有三种方式可以用来创建线程：继承Thread类，实现Runnable接口，应用程序可以使用Executor框架来创建线程池，你更喜欢哪一种？** 实现Runable接口，因为java只支持单继承，却可以多实现

**32、什么是死锁？如何确保N个线程可以访问N个资源同时又不导致死锁？ 死锁：**两个线程相互等待互不相让（个线程想先锁对象1，再锁对象2，恰好另外有一个线程先锁对象2，再锁对象1。在这个过程中，当线程1把对象1锁好以后，就想去锁对象2，但是不巧，线程2已经把对象2锁上了，也正在尝试去锁对象1） **解决办法：**指定获取锁的顺序，并强制线程按照指定的顺序获取锁。因此，如果所有的线程都是以同样的顺序加锁和释放锁，就不会出现死锁了。

**33、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，而Servlet主要用于控制逻辑。

**34、JSP四种注释的方法：**

答：(1)<%-- --%> (2)// (3)/\* \*/ (4)<!-- -->

**35、JSP中的九种内置对象和作用分别是什么？** 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：网页的属性在这里管理Session：与请求有关的会话期，存储特定用户会话所需的属性及配置信息 Out：用来传送回应的输出 Config：servlet的架构部件 Page：JSP网页本身 Exception：针对错误的网页，未捕捉的异常 Application：正在执行的内容

**36、JAVA中四大域有哪些？生命周期？作用范围大小？**

pageContext--->request--->session---->context pageContext：生命周期：当对JSP的请求时开始，当响应结束时销毁。作用范围：（页面范围）整个JSP页面。 request（ServletRequest）：生命周期：在service 方法调用前由服务器创建，传入service方法。整个请求结束，request生命结束。作用范围：（请求范围）整个请求链（请求转发也存在）。  session（HttpSession）：生命周期：在第一次调用 request.getSession() 方法时，服务器会检查是否已经有对应的session,如果没有就在内存中创建一个session并返回，默认生命周期30min，调用invalidate（）关闭session。作用范围：一次会话。 context（ServletContext）：生命周期：当Web应用被加载进容器时创建代表整个web应用的ServletContext对象，当服务器关闭或Web应用被移除时，ServletContext对象跟着销毁。作用范围：整个Web应用。

1. **El表达式的作用和语法是什么？**

作用：表达式语言，用来取域中的数据 ，配合JSTL可在jsp页面省略大量<% %>脚本片段，使程序更整洁

语法：${参数名字}如果浏览器不支持EL表达式需手动开启：isELIgnored=“false”

**38、简述MVC设计模式：**

在实际开发中jsp擅长做数据的显式，Servlet擅长请求控制，JavaBean擅长封装数据。M，代表的是Model(javabean);v，代表的是view(jsp)；c，代表的是ctroll(Servlet)。

**39、什么是javabean？EJB？POJO？ORM？JSON？JDO**

**Javabean：**只是一种规范，一个类中的字段必须全部用private修饰符来修饰，必须为每一个属性提供一个访问器(get+属性名，属性名首字母大写)和修改器(set+属性名，属性名首字母必须大写)，必须为这个类提供一个无参的构造方法。满足这些条件的类，我们就称这个类为JavaBean。

**EJB：**它是一组"功能"JavaBean的集合。上面说了JavaBean是实现了一种规范的Java对象。这里说EJB是一组JavaBean，的意思是这一组JavaBean组合起来实现了某个企业组的业务逻辑。这里的一组JavaBean不是乱组合的，它们要满足能实现某项业务功能的搭配。找个比方，对于一身穿着来说，包括一顶帽子，一件衣服，一条裤子，两只鞋。这穿着就是EJB，其它的就是一个JavaBean。**POJO：**指那些没有从任何类继承、也没有实现任何接口，更没有被其它框架侵入的java对象，格式是用于数据的临时传递，它只能装载数据， 作为数据存储的载体，而不具有业务逻辑处理的能力，如将jsp页面多个字段封装到一个对象会中去。而javabean虽然数据的获取与pojo一样，但是javabean当中可以有其它的方法,封装一些简答逻辑。

**ORM：**对象关系映射，一种为了解决面向对象与关系数据库存在的互不匹配的现象的技术，如将数据库某条信息封装映射到某个对象中去。

**Json：**只是将数据以某一种规范进行展示，{key1:value1,key2,value2}

步骤：导入json.jar；创建一个JsonObject对象；将实体类添加到该对象中去。**JDO**：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化API。

**40、通过Java代码来连接数据库，从而对数据库进行CRUD的步骤有哪些？**

导入第三方jar包--加载数据库驱动-->创建连接对象（Connection）-->创建代表发送sql语句的Statement对象-->向数据库发送sql语句，获取代表结果集的ResultSet对象-->释放数据库资源

**41、jstl与传统jsp开发手段的区别**

传统的jsp开发会用脚本元素在页面中嵌入程序源代码，这样的jsp页面，维护很复杂 jstl使jsp作者可以减少对编制脚本元素的需求，甚至可以不需要他们，并避免了相关的维护成本

**42、过滤器的作用是什么**

截取从客户端发过来请求并进行处理，对数据进行重新编码，过滤掉客户的某些不应出现的词汇，可以验证用户是否已经登陆

**43、java中有几种类型的流？**

字节流：InputStream/OutputStream

字符流：InputStreamReader/OutputStreamWriter

**44、如何实现servlet的单线程模式**

<%@ page isThreadSafe="false"%>

**45、四种会话跟踪技术？页面间对象传递的方法？**

cookies，url重写，session，隐藏表单域

request，session，application，cookie

**46、java中会存在内存泄漏吗，请简单描述。**

Java的确存在内存泄漏，并且事态可以变的相当严重java garbage collector 自动释放那些内存里面程序不需要的对象，以避免大多数的其它程序上下文的内存泄漏，但是java应用程序依旧会有相当的内存泄漏，查找原因会十分困难，有两类主要的java内存泄漏：

1. 不再需要的对象引用
2. 未释放的系统资源

**47、静态变量和实例变量的区别** 静态变量属于类，不用实例化就已经存在，所有的实例共享同一个静态变量，可以通过类名和实例名来访问。

实例变量属于实例，每个实例都有自己的这个变量，只能通过实例名来访问。

**48、名词解释：数组**

数组是一种数据结构，用来存储同一类型值的集合，通过一个整形下标可以访问数组中的每一个值，一旦创建了数组就不能改变它的大小了。

**49、访问控制符的各自作用域** Public：公共的，所有类都可访问 Protected：受保护的，当前包所有类及其子类均可访问缺省：默认，只有当前包可以访问Private：私有的，只有同一类可以访问

**50、request对象的常用方法及作用（写三种）**

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

**51、如何实现线程之间的通信**

使用Object类提供的wait()、notify()、notifyAll()三个方法实现线程同步、共享内存、管道流pipes实现了UI线程和工作线程之间的数据通信、while轮询的方式（浪费CPU资源）

1. **简述反射技术** 反射就是动态加载一个指定的类，并获取该类中的所有内容，且将字节码文件封装成对象，并将字节码文件中的内容也都封装成了对象，这样便于操作这些成员。

简单说：反射技术可以对一个类进行解剖

反射的好处：大大增强了程序的扩展性

**53、请写出以下运算符的优先级别：**1.() 2.> 3.++ 4.+= 5.&&

1>3>2>5>4（算>关>逻>赋）

**54、是否可以在静态（static）方法内部调用非静态方法？**static方法内部不能直接调用非静态方法，可以在调用之前实例化非静态方法所在的类，再用类点方法来调用。

**55、heap 和stack 有什么区别。**

栈是一种线性集合，遵循LIFO，可基于数组或者链表来实现，采用一级缓存，速度快，存放基本数据类型，可共享，但是栈中的数据大小与生存期必须是确定的，缺乏灵活性，内存由虚拟机释放。堆是一个有序集合，一般通过二叉树来实现，采用二级缓存，速度慢，一个对象如数组，他的实体存储在堆里面，他的引用存储在栈里面，一般由程序员释放。

**56、抽象类为什么不能实例化？**

抽象类就是类里面含有了抽象的方法。抽象的方法就是没有实现的方法。有抽象的方法当然不能实例化，如果它能实例那么这个实例，调用那个没有实现的抽象类会有什么结果。所以抽象类不能实例化。

**57、java 中有几种类型的流？**

字节流继承于InputStream OutputStream

字符流继承于InputStreamReader OutputStreamWriter。

**58、Class.forName()的作用?**

调用该方法返回一个以字符串指定类名的类的对象。此对象用来与特定的数据库连接。

**59、XML 与HTML 的主要差异**

XML 不是HTML 的替代。XML 和HTML 为不同的目的而设计：

XML 被设计为传输和存储数据，其焦点是数据的内容。

HTML 被设计用来显示数据，其焦点是数据的外观。

**60、说一下什么是spring？**

a、Spring的核心是一个轻量级（Lightweight）的容器（Container）。

b、Spring是实现IoC（Inversion of Control）容器和非入侵性（No intrusive）的框架。

c、Spring提供AOP（Aspect-oriented programming）概念的实现方式

d、Spring提供对持久层（Persistence）、事物（Transcation）的支持。

e、Spring供MVC Web框架的实现，并对一些常用的企业服务API（Application Interface）提供一致的模型封装。

f、Spring提供了对现存的各种框架（Structs、JSF、Hibernate、Ibatis、Webwork等）相整合的方案。

总之，Spring是一个全方位的应用程序框架。

**61、简述 spring 的优缺点？**

优点：

a.使用Spring的IOC容器，将对象之间的依赖关系交给Spring，降低组件之间的耦合性，让我们更专注于应用逻辑

b.可以提供众多服务，事务管理，WS等。

c.AOP的很好支持，方便面向切面编程。

d.对主流的框架提供了很好的集成支持，如hibernate,Struts2,JPA等

e.Spring DI机制降低了业务对象替换的复杂性。

f.Spring属于低侵入，代码污染极低。

g.Spring的高度可开放性，并不强制依赖于Spring，开发者可以自由选择Spring部分或全部

缺点:

a.jsp中要写很多代码、控制器过于灵活，缺少一个公用控制器

b.Spring不支持分布式，这也是EJB仍然在用的原因之一。

**62、解释一下什么是控制反转？** 其实就是将对象的初始化创建任务交给spring容器依赖注入(DI)和控制反转(IOC)是从不同的角度的描述的同一件事情，就是指通过引入IOC容器，利用依赖关系注入的方式，实现对象之间的解耦。

**63、spring 有哪几种注入方式？**

a、实例工厂的方法注入;b、静态工厂的方法注入;c、构造器注入;d、Set注入;

**64、名词解释AJAX**

AJAX 全称为“Asynchronous JavaScript and XML”（异步JavaScript 和 XML），是一种创建交互式网页应用的网页开发技术。

**65、什么是AOP？**  通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一 添加功能的一种技术。是对OOP的一种横向补充。

**66、简述AJAX 应用和传统 WEB 应用的区别。**

传统 WEB 应用是页面完全刷新, 简单操作也导致重新加载页面全部数据。AJAX 应用是部分刷新页面，只作必要的数据交换，并且可以异步访问服务器端。

**67、简述 MVC 的优缺点。**

优点：（1）开发效率高。（2）松耦合，程序灵活性和扩展性好。

（3）代码重用度更高。（4）易于人员分工。

缺点:（1）代码复杂度增加。（2）代码数量增加。（3）不适合开发小型项目。

**68、jsp 有哪些动作?作用分别是什么?**

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

**69、java中常用的设计模式，说出五种**

Factory（工厂模式）、Singleton（单例模式）、Facade（门面模式）、

Bridge（桥梁模式）、Composite（合成模式）

**70、存储过程、函数的区别**

存储过程是用户定义的一系列sql语句的集合，返回参数，涉及特定表或其他对象的任务，用户可以调用存储过程。 函数通常是数据库已经定义得方法，它接收参数并返回某种类型的值或表对象，并不涉及特定用户表。

**71、J2EE 是技术还是平台还是框架？**  
J2EE 本身是一个标准，一个为企业分布式应用的开发提供的标准平台。J2EE 也是一个框架，包JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

**72、Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢? 是用==还是equals()? 它们有何区别?** Set 里的元素是不能重复的，那么用iterator()方法来区分重复与否。equals()是判读两个Set是否相等。equals()和==方法决定引用值是否指向同一对象equals()在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值。

**73、如何启动时不需输入用户名与密码?**

修改服务启动文件，增加 WLS\_USER和WLS\_PW项。也可以在boot.properties文件中增加加密过的用户名和密码.

1. **写clone()方法时，通常都有一行代码，是什么？**

Clone有缺省行为，super.clone();他负责产生正确大小的空间，并逐位复制。

**75、sleep()和wait()有什么区别**

Sleep()是线程类（Thread）的方法，导致此线程暂停执行指定的时间，把执行机会给其他线程，但是监控状态依然保持，到时候会自动回复，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或notifyAll方法后，本线程才进入对象锁定池准备获得对象锁进入运行状态。

**76、名词解释：web容器** 给处于其中的应用程序组件（JSP、SERVLET）提供一个环境，使JSP、SERVLET直接和容器中的环境变量接口交互，不必关注其他系统问题。

**77、当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并返回变化后的结果，那么这到底是值传递还是引用传递？**

是值传递。Java只有值传递参数，当一个对象实例作为一个参数被传递到方法中时，参数的值就是对该对象的引用。对象的内容可以在被调用那个的方法中改变，但对象的引用是永远不会改变的

1. **介绍一下XMLHttpRequest对象**

Ajax的核心是javascript对象XmlHttpRequest。该对象在IE5中首次引入，它是一种支持异步请求的技术。简而言之，XmlHttpRequest使你可以用javascript向服务器提出请求并处理响应，而不阻塞用户。通过XmlHttpRequest对象，web开发人员可以在页面加载以后进行页面的局部更新

**79、C/S和B/S的区别：**

C/S:客户端/服务器端 一般使用于软件开发，对客户端的计算机配置要求较高，客户端也需要装软件。

B/S：浏览器/服务器 一般使用于网站开发，客户端只要安装浏览器就可以完成操作客户端的基本上不需要加载什么，但是对于服务器端的配置要求比较高

**80、spring中bean有哪几中作用域**

1、[singleton](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-singleton" \t "_blank): 2、[prototype](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-prototype" \t "_blank)：3、[request](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-request" \t "_blank)：4、[session](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-global-session" \t "_blank)：5、[global session](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-global-session" \t "_blank)

**81、解释seralizable的含义及作用**

实现java.io.seralizable接口，使其序列化。可以实现数据网络传输简单说就是为了保存在内存中的各种对象的状态，并且可以把保存的对象状态再读出来。把的内存中的对象保存到一个文件中或者数据库中时候；用套接字在网络上传送对象的时候；通过RMI传输对象的时候需要使用序列化。

**82、JSP中动态include和静态include的区别？** 动态include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,它不会检查所含文件的变化，适用于包含静态页面

**83、排序有哪几种方法？** 冒泡排序、快速排序、选择排序、插入排序、希尔排序、基数排序、鸡尾酒排序、桶排序、鸽巢排序、归并排序等。

1. **请说出delete，truncate和drop的区别？**
2. Drop和truncate是DDL语句，执行后会自动提交，delete是DML语句，不能自动提交，事务提交后才生效。
3. Delete语句每次删除一行，从数据库的缓存区域清除该数据，保留标识计数值，truncate删除表中所有行，释放储存数据所占用的空间，重置标识计数值。3、Delete和truncate只删除数据不删除表的结构，drop删除表的结构，整个对象都会被删掉

从速度上来看，drop>truncate>delete

**85、数据库中varchar和char的区别？**

都是字符串类型，char是Unicode编码，最大长度为8000字符，所占空间为定义时的大小；varchar是非Unicode编码，最大长度为8000字符，所占空间为实际大小；数据量大、需要经常修改的时候用char查询速度快，存储长度不是固定的时候用varchar。

**86、GC 是什么? 为什么要有GC?** GC 是垃圾收集的意思（Gabage Collection）对象的存储是要占用内存空间的如果不回收不会再次使用的内存空间那会造成资源浪费甚至导致系统崩溃而无法正常执行程序

**87、请列举当前一些当前流行的数据库引擎**

SQL SERVER,ORACLE,BDE,Microsoft，Jet。

1. **请列举你知道的前端框架**

Bootstrap（twitter）、Fbootstrapp、BootMetro、Kickstrap、Foundation、GroundworkCSS、Gumby、HTML KickStart、IVORY

**89、TCP和UDP的区别** TCP是面向连接的通信协议，TCP提供两台计算机之间的可靠的无差别错的数据传输。 UDP是无连接通信协议，UDP不保证可靠的数据的传输，但能够向若干个目标发送数据，接受发自若干个源的数据。

**90、在applicationContext.xml文件中的<bean>标签中有一个scope属性,scope属性有4个值** Singleton（单例模式，默认）>prototype（多例模式，开发常用）>request>session

1. **懒加载和非懒加载的对比**

懒加载：只有当使用某一对象的时候才创建对象：消耗内存小一些

非懒加载的方式：不管使不使用对象，都提前创建好对象：比较消耗内存；提前帮助程序员发现错误

**92、ajax请求总共有多少种callback**

onSuccess onFailure onUninitialized onLoading

onLoading onInteractive onComplete onException

**93、dom里面的添加、删除、替换、创建节点的方法**

1、创建元素节点：createElement

用法：document.createElement("p");//创建p标签；

    创建文本节点：createTextNode;

用法：document.createTextNode("txt");//创建文本标签；

2、插入节点：appendchild:在要插入的元素节点上调用，他插入指定的节点使其成为那个节点的最后一个子节点。

                   insertBefore:在已有的元素前插入一个新元素；

                   insertAfter:在现有的元素后面插入一个新元素；

3、删除节点：removeChild;从文档树中删除一个节点（先找到要删除节点的父节点，用父节点删除）；

                  用法：var para1=document.getElementById("p1");

                               para1.parentNode.removeChild(para1);

4、替换节点：replaceChild;将一个节点替换为另一个节点（先找到要替换的节点的父节点，用父节点替换）；

**94、什么是jquery对象和dom对象**

1，DOM对象是我们用传统的方法（[JavaScript](http://lib.csdn.net/base/javascript" \t "_blank" \o "JavaScript知识库)）获得的对象，Query对象就是用jQuery的类库选择器获得的对象。

2，JQuery对象就是通过jQuery包装DOM对象后产生的对象。JQuery对象是jQuery独有的，其可以使用jQuery里的方法，但是不能使用DOM的方法

3，DOM对象就是javascript固有的一些对象。DOM对象能使用javascript固有的方法，但是不能使用jQuery里面的方法。

相互转化：var$cr=$("#cr"); //jquery对象  
 var cr = $cr[0]; //dom对象 也可写成 var cr=$cr.get(0);

**95、spring中bean有哪几中作用域**

1、[singleton](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-singleton" \t "_blank): 2、[prototype](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-prototype" \t "_blank)：3、[request](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-request" \t "_blank)：4、[session](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-global-session" \t "_blank)：

5、[global session](http://student.csdn.net/source/csdn/fckeditor/editor/ch03s04.html" \l "beans-factory-scopes-global-session" \t "_blank)

**96、ssh理解**

Struts就是一个MVC的架构,在控制器,视图和模型方面都有其特定的组件，在SSH框架中，更多的是利用了Controller这个组件。

Spring实现了反转控制（Inversion of Control）和面向方面编程(Achievement-Oriented Programming)。Spring框架的能实现应用的分层架构,在应用中要使用的对象可以被一个简单的XML文件连在一起，这个XML文件含有到像事务管理器、对象工厂、包含业务逻辑的服务对象、和数据存取对象这些对象的引用。

Hibernate是一种ORM映射工具，它不仅提供了从Java类到数据表之间的映射，也提供了数据查询和恢复机制。相对于使用JDBC和SQL来手工操作数据库，使用Hibernate，可以大大减少操作数据库的工作量。

**96、cookie和session的区别**

1、session保存在服务器，客户端不知道其中的信息；cookie保存在客户端，服务器能够知道其中的信息。

2、session中保存的是对象，cookie中保存的是字符串。

3、session不能区分路径，同一个用户在访问一个网站期间，所有的session在任何一个地方都可以访问到。而cookie中如果设置 了路径参数，那么同一个网站中不同路径下的cookie互相是访问不到的。

4、session默认需要借助cookie才能正常工作。如果客户端完全禁止cookie，session，这种方法将失效。但是如果服务器端启用了url编码，也就是用URLEncoder.encode("index.jsp?id=3","UTF-8");把所有的url编码了，则会在url后面出现如下类似的东西index.jsp:jsessionid=fdsaffjdlksfd124324lkdjsf?id=3服务器通过这个进行session的判断。

5 session在用户会话结束后就会关闭了，但cookie因为保存在客户端，可以长期保存

6 cookie:是服务端向客户端写入的小的片段信息。cookie信息保存在服务器缓存区，不会在客户端显现。当你第一次登陆一个网站，服务器向你的机器写得片段信息。你可以在Internet选项中找到存放cookie的文件夹。如果不删除，cookie就一直在这个文件夹中。

**97、迭代器的使用**

迭代器是一种设计模式，它是一个对象，它可以遍历并选择序列中的对象，而开发人员不需要了解该序列的底层结构。迭代器通常被称为“轻量级”对象，因为创建它的代价小。

Java中的Iterator功能比较简单，并且只能单向移动：

(1) 使用方法iterator()要求容器返回一个Iterator。第一次调用Iterator的next()方法时，它返回序列第一个元素。注意：iterator()方法是java.lang.Iterable接口,被Collection继承。

(2) 使用next()获得序列中的下一个元素。

(3) 使用hasNext()检查序列中是否还有元素。

(4) 使用remove()将迭代器新返回的元素删除。

**99、进程和线程的区别**

进程和线程的主要差别在于它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程

**100、ioc、di和aop**

Ioc—Inversion of Control，即“控制反转”，不是什么技术，而是一种设计思想。在Java开发中，Ioc意味着将你设计好的对象交给容器控制，而不是传统的在你的对象内部直接控制。

　DI—Dependency Injection，即“依赖注入”：组件之间依赖关系由容器在运行期决定，形象的说，即由容器动态的将某个依赖关系注入到组件之中。依赖注入的目的并非为软件系统带来更多功能，而是为了提升组件重用的频率，并为系统搭建一个灵活、可扩展的平台。通过依赖注入机制，我们只需要通过简单的配置，而无需任何代码就可指定目标需要的资源，完成自身的业务逻辑，而不需要关心具体的资源来自何处，由谁实现。

　　理解DI的关键是：“谁依赖谁，为什么需要依赖，谁注入谁，注入了什么”，那我们来深入分析一下：

　　●谁依赖于谁：当然是应用程序依赖于IoC容器；

　　●为什么需要依赖：应用程序需要IoC容器来提供对象需要的外部资源；

　　●谁注入谁：很明显是IoC容器注入应用程序某个对象，应用程序依赖的对象；

　　●注入了什么：就是注入某个对象所需要的外部资源（包括对象、资源、常量数据）。

IoC和DI由什么关系呢？其实它们是同一个概念的不同角度描述，由于控制反转概念比较含糊（可能只是理解为容器控制对象这一个层面，很难让人想到谁来维护对象关系），所以2004年大师级人物Martin Fowler又给出了一个新的名字：“依赖注入”，相对IoC 而言，“依赖注入”明确描述了“被注入对象依赖IoC容器配置依赖对象”。

AOP：即面向切面编程，可以说是OOP（Object Oriented Programming，面向对象编程）的补充和完善。OOP引入封装、继承、多态等概念来建立一种对象层次结构，用于模拟公共行为的一个集合。

AOP、OOP在字面上虽然非常类似，但却是面向不同领域的两种设计思想。OOP（面向对象编程）针对业务处理过程的实体及其属性和行为进行抽象封装，以获得更加清晰高效的[逻辑单元](http://baike.so.com/doc/922770.html" \t "_blank)划分。

而AOP则是针对业务处理过程中的切面进行提取，它所面对的是处理过程中的某个步骤或阶段，以获得逻辑过程中各部分之间低[耦合性](http://baike.so.com/doc/5725915.html" \t "_blank)的隔离效果。这两种设计思想在目标上有着本质的差异。

动态的将代码块切入到制定的位置和方法中

**101、在jsp中，我们可以怎样获取来自客户端请求的ip地址？**

Request.getRemoteAddr();

JSP的include指令和include操作有什么区别?  
 1) 语法结构不同  
 2) include指令更适合于包含静态页面， include操作更适合于包含动态页面  
 3)include操作还带有参数 flush=”true|false”

**102、简述一下ajax的原理及实现步骤？**

原理： HTTP协议的异步通信  
get 请求  
1) 创建一个XMLHttpRequest对象  
2) 调用该对象的open方法  
3) 如果是get请求，设置回调函数onreadystatechange = callback  
4) Send  
如果是post 请求  
5) 创建一个XMLHttpRequest对象  
6) 调用该对象的open方法  
7) 调用setRequestHeader(“Content-Type”, “application/x-www-form-urlencoded”);  
8) 设置回调函数onreadystatechange = callback

9) Send

AJAX全称为“Asynchronous JavaScript and XML"(异步JavaScript和XML），是指一种创建交互式网页应用的网页开发技术，它使得浏览器可以为用户提供更为自然的浏览体验。AJAX不是一种技术，它实际上是集中技术的集合体，每种技术都其独特的地方，合成在一起就成为了一个功能强大的技术。AJAX包括：

1.XHTML和CSS

2.使用文档对象模型DOM（Document Object Model)做动态显示和交互。

3.使用XML和XSLY做数据交互和操作。

4.使用XMLHttpRequest将它们进行绑定在一起。

AJAX的工作原理相当于在用户和服务器之间加了一个中间层，使用户操作与服务器相应异步化。并不是所有的用户请求都提交给服务器，像一些数据验证和数据处理等都交给AJAX引擎自己来做，只有确定需要从服务器读取新数据时再由AJAX引擎代为向服务器提交请求。XMLHttpRequest是ajax的核心机制，它是在IE5中首先引入的，是一种支持异步请求的技术。简单的说，也就是javascript可以及时向服务器提出请求和处理响应，而不阻塞用户。达到无刷新的效果。

**103、什么是反向代理？反向代理用到了哪些服务器？**

  反向代理（Reverse Proxy）方式是指以代理服务器来接受Internet上的连接请求，然后将请求转发给内部网络上的服务器；并将从服务器上得到的结果返回给Internet上请求连接的客户端，此时代理服务器对外就表现为一个服务器。

1，作内容服务器的替身

2，作为内容服务器的负载均衡器

通常的代理服务器，只用于代理内部网络对Internet的连接请求，客户机必须指定代理服务器,并将本来要直接发送到Web服务器上的http请求发送到代理服务器中。当一个代理服务器能够代理外部网络上的主机，访问内部网络时，这种代理服务的方式称为反向代理服务。

**104、java类的种类？**

Class interface package

**105、你用过哪些服务器？如何管理服务器？他们的区别？**

Web服务器是运行及发布Web应用的容器，只有将开发的Web项目放置到该容器中，才能使网络中的所有用户通过浏览器进行访问。开发Java Web应用所采用的服务器主要是与JSP/Servlet兼容的Web服务器，比较常用的有Tomcat、Resin、JBoss、WebSphere 和 WebLogic 等

**106、 java分布式开发？**

分布式服务下的关键技术：消息队列架构

消息对列通过消息对象分解系统耦合性，不同子系统处理同一个消息

分布式服务下的关键技术：消息队列原理

分布式服务下的关键技术：服务框架架构  
服务框架通过接口分解系统耦合性，不同子系统通过相同的接口描述进行服务启用  
服务框架是一个点对点模型  
服务框架面向同构系统  
适合：移动应用、互联网应用、外部系统

分布式服务下的关键技术：服务框架原理

分布式服务下的关键技术：服务总线架构

服务总线同服务框架一样，均是通过接口分解系统耦合性，不同子系统通过相同的接口描述进行服务启用  
服务总线是一个总线式的模型  
服务总线面向同构、异构系统  
适合：内部系统

分布式服务下的关键技术：服务总线原理

分布式架构下系统间交互的5种通信模式  
request/response模式（同步模式）：客户端发起请求一直阻塞到服务端返回请求为止。  
Callback（异步模式）：客户端发送一个RPC请求给服务器，服务端处理后再发送一个消息给消息发送端提供的callback端点，此类情况非常合适以下场景：A组件发送RPC请求给B，B处理完成后，需要通知A组件做后续处理。  
Future模式：客户端发送完请求后，继续做自己的事情，返回一个包含消息结果的Future对象。客户端需要使用返回结果时，使用Future对象的.get(),如果此时没有结果返回的话，会一直阻塞到有结果返回为止。  
Oneway模式：客户端调用完继续执行，不管接收端是否成功。  
Reliable模式：为保证通信可靠，将借助于消息中心来实现消息的可靠送达，请求将做持久化存储，在接收方在线时做送达，并由消息中心保证异常重试。

五种通信模式的实现方式-同步点对点服务模式

五种通信模式的实现方式-异步点对点消息模式1

五种通信模式的实现方式-异步点对点消息模式2

五种通信模式的实现方式-异步广播消息模式

**107、用选择排序是先对一个数组的排序？**

原理：每一趟从待排序的记录中选出最小的元素，顺序放在已排好序的序列最后，直到全部记录排序完毕。

static void BubbleSort(int a []){

　　int temp=0;

　　for (int i = 0; i < a.length ; i++) {

　　for (int j = 0; j < a.length - i - 1; j++){

　　if (a[j]>a[j + 1]){ //把这里改成大于，就是升序了

　　temp=a[j];

　　a[j]=a[j + 1];

　　a[j + 1]=temp;

　　} }}}

**108、如何使用Oracle游标**

Oracle游标分为显示游标和隐式游标。   
显示游标（Explicit Cursor）:在PL/SQL程序中定义的、用于查询的游标称作显示游标。   
隐式游标（Implicit Cursor）:是指非PL/SQL程序中定义的、而且是在PL/SQL中使用UPDATE/DELETE语句时，Oracle系统自动分配的游标。

**109、你知道的java开发模式，简述在什么样的情况下使用该模式？**

MVC是一个很常用的程序开发[设计模式](http://www.so.com/s?q=%E8%AE%BE%E8%AE%A1%E6%A8%A1%E5%BC%8F&ie=utf-8&src=wenda_link" \t "_blank)，M-Model(模型):封装应用程序的状态;V-View(视图):表示用户界面;C-Controller(控制器):对用户的输入作出反应，创建并设置模型

**110、面向对象的优缺点？**

优: 1.易维护 2.质量高 3.效率高 4.易扩展  
(1)是一种全新的系统分析设计方法(对象、类、结构属性、方法)  
(2)适用于各类信息系统的开发。  
(3)实现了对客观世界描述到[软件结构](http://www.so.com/s?q=%E8%BD%AF%E4%BB%B6%E7%BB%93%E6%9E%84&ie=utf-8&src=wenda_link" \t "_blank)的直接转换 ，大大减少后续软件开发量。  
(4)开发工作的重用性、继承性高，降低重复工作量。  
(5)缩短了开发周期

缺：类和继承等特点使得程序会多很多指针操作，程序的处理效率相对要低

1.需要一定的软件支持环境 2.不太适宜大型的MIS开发，若缺乏整体系统设计划分，易造成系统结构不合理、各部分关系失调等问题 3.只能在现有业务基础上进行分类整理，不能从科学管理角度进行理顺和优化 4.初学者不易接受、难学

**111、谈谈你对servlet的理解？**Serverlet 就是一个Java类，Web中应用的应该是HttpServerlet ，Serverlet类最大的好处就是能够提供request/response的服务器功能，当有请求提交到Serverlet 时，他执行它自身的service（request,response)方法。

**112、创建对象的几种方式？**

(1) 用new语句创建对象，这是最常见的创建对象的方法。

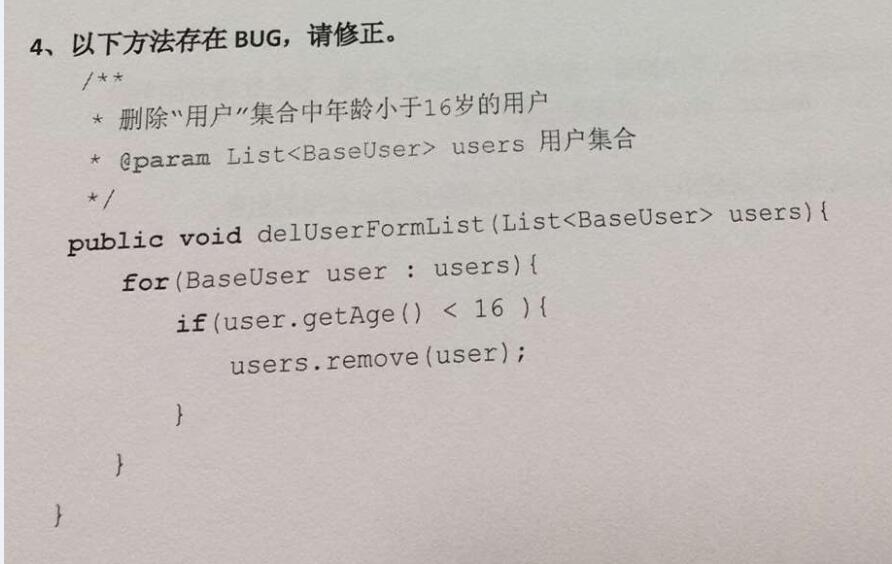
(2) 运用反射手段,调用java.lang.Class或者java.lang.reflect.Constructor类的newInstance()实例方法。

(3) 调用对象的clone()方法。

(4) 运用反序列化手段，调用java.io.ObjectInputStream对象的 readObject()方法。

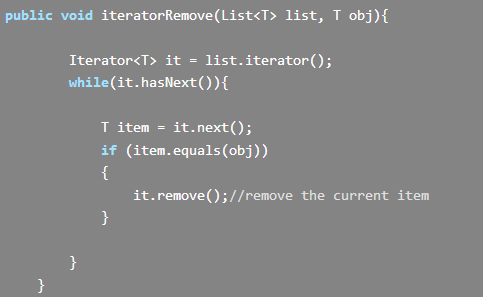
**113、不通过构造函数也能创建对象吗？**

(1)和(2)都会明确的显式的调用构造函数 ；(3)是在内存上对已有对象的影印，所以不会调用构造函数 ；(4)是从文件中还原类的对象，也不会调用构造函数。

**114 、**

Java中的For each实际上使用的是iterator进行处理的。而iterator是不允许集合在iterator使用期间删除的。所以导致了iterator抛出了ConcurrentModificationException 。对Iterator的remove()方法调用必须在Iterator的next()方法之后。调用next()方法后只能执行一次remove()方法。

总结：普通for循环是根据索引删，可能会删不干净，迭代器是根据集合数据的值来删



**115、什么是策略模式？什么情况下使用？**

策略模式的设计原则如下：  
1. 将应用中需要经常变化的代码独立出来，应和那些不需要经常变化的代码分开。  
2. 应针对接口，而不是类进行编程。  
3. 在类中应多用组合，少用继承。  
例子：  
我们要实现一个鸭子模拟器，这个鸭子模拟器由Duck类描述，而Duck类有如下4个行为：  
1. display  
2. swim  
3. fly（飞）  
4. quack（叫）  
其中swim是所有鸭子都具有的特性，而且所有鸭子的这些特性都相同，因此，这个方法可以直接在Duck类中实现。display方法也是所有鸭子具有的特性，但随着鸭子的种类不同，display也有所不同，因此，display方法应为Duck类的抽象方法。fly和quack并不是所有鸭子的特性，如橡皮鸭子即不会飞，也不会叫。因此，可以将这两个方法看作是两个行为，可将每一个行为设计成一个接口。这样可以和Duck类完全脱离。因为，fly和quack与Duck一点关系都没有（别的东西也可能fly和quack），然后不同的fly和quack分别用实现相应接口的类表示。

**116、常用算法归类**

常用的对称加密算法有：DES、3DES、RC2、RC4、AES

常用的非对称加密算法有：RSA、DSA、ECC

使用单向散列函数的加密算法：MD5、SHA

**117、java中数据类型分类和转换问题？**

原生类是指Java中，数据类型分为基本数据类型（或叫做原生类、内置类型）和引用数据类型。那么原生类为基本数据类型，有八种，这样转换的时候就有表达范围问题。

a所占位数少的可以转换为所占位数多的类型，比如byte转char,char转int等；

b而所占位数多的转为所占位数少的默认情况下不能实现转换，需要强制类型转换，这样可能会丢失一部分原始数据；

c此外，boolean类型数据和其他七种不能互相转换。

**118、ejb中的sessionbean和entitybean的区别和含义？**

对于客户机，SessionBean是一种非持久性对象，它实现某些在服务器上运行的业务逻辑。

对于客户机，EntityBean是一种持久性对象，它代表一个存储在持久性存储器中的实体的对象视图，或是一个由现有企业应用程序实现的实体。

**119、使用final关键字修饰一个变量时,是引用不能变,还是引用的对象不能变**

使用final关键字修饰一个变量时，是指引用变量不能变，引用变量所指向的对象中的内容还是可以改变的。



**120、描述jvm加载class文件的原理机制？**

类装载器就是寻找类或接口字节码文件进行解析并构造JVM内部对象表示的组件，在java中类装载器把一个类装入JVM，经过以下步骤：

1、装载：查找和导入Class文件

2、链接：其中解析步骤是可以选择的

（a）检查：检查载入的class文件数据的正确性

（b）准备：给类的静态变量分配存储空间

（c）解析：将符号引用转成直接引用

3、初始化：对静态变量，静态代码块执行初始化工作

**121、某一培训学校要做一个网站，其中有一个招生报名功能模块，这件事情安排给你做，你描述一下工作如何开展，具体的实施步骤。**

首先需要了解的有：  
1、网站的建站目的；（招生）  
2、客户以及网站功能需求；（学习内容与招生的流程）  
3、制定项目执行计划；  
4、需求分析后根据实际情况设定功能属性（数据库表建立）；  
5、根据计划完成编码、测试（其中包含反馈实际执行进度）；

**122、Java遵循标准I/O的模型，提供了哪三种标准输入输出？**

Syetem.in，System.out，以及System.err。

**123、如果需要对对象的序列化进行深度的控制，需要实现什么接口和方法？**

Serializable接口；writeObject()与readObject()

**124、HQL和SQL语句有什么区别?**

sql 面向数据库表查询

hql 面向对象查询

hql ： from 后面跟的 类名＋类对象 where 后 用 对象的属性做条件

sql： from 后面跟的是表名         where 后 用表中字段做条件

**125、分别写出http https ftp telnet ssh的默认端口**

FTP就是专门用来传输文件的协议 默认端口21

SSH 为建立在应用层和传输层基础上的安全协议 默认端口22

HTTP超文本传输协议是互联网上应用最为广泛的一种网络协议 默认端口80

Telnet协议是TCP/IP之一，是Internet远程登陆服务的标准协议和主要方式。默认端口23

HTTPS，是以安全为目标的HTTP通道，简单讲是HTTP的安全版 默认端口443

TOMCAT 8080

**126、hibernate和 mybatis的优缺点**

Mybatis框架相对简单，很容易上手，但也相对简陋些，Hibernate有良好的映射机制，无需关心sql的生成与结果映射，可以更专注于业务流程，而Mybatis需要手动编写sql语句。Hibernate与具体数据库的关联只需在XML文件中配置即可，所有的HQL语句与具体使用的数据库无关，移植性很好。而MyBatis项目中所有的sql语句都是依赖所用的数据库的，数据库类型的支持不好。

**127、简述ssm和ssh框架各部分在项目中的作用**

Ssm框架中，spring是一个轻量级的控制反转(IOC)和面向切面(AOP);springMVC分离了控制器、模型对象、分派器以及处理程序对象的角色；mybatis使用简单的XML或注解用于配置和原始映射，将接口和普通的Java对象映射成数据库中的记录。

ssh框架中Struts是前台框架，主要负责前后台交互的逻辑处理以及前台组件;Hibernate是后台框架，主要负责和数据库的交互，属于ORM;spring是嵌入MVC的框架，主要是代码注入。

**128、谈谈对jQuery的理解**

jQuery是封装的js包，相当于把一些琐碎的功能、方法封装起来，给一个入口让他人调用，可以减少代码量。

**129、你常用的几个前端框架的优缺点有哪些？**

我常用的前端框架有easyUI和jQuery。

easyUI优点：

1.easyUI是一个基于jQuery的集成了各种用户界面的插件

1. 使用easyUI不需要太多javascript代码，一般只需要使用一些html标记定义用户接口
2. easyUI非常简单，但是功能很强大。

easyUI缺点：对服务器负载性能的影响，越多的效果就会使服务器需要更多的CPU资料来调配所有。

jQuery优点：实现脚本与页面的分离；最少的代码做最多的事情；性能支持比较好；插件分开

jQuery缺点：不能向后兼容；多个插件冲突；对动画和特效的支持差

**130、服务器如何防止重复提交**

方法一：利用javascript阻止其它不变

方法二：利用服务器端session防止重复提交

**131、编写触发器，当A表al字段变化时，将变化前的记录值插入到B表的al字段**

CREATE TRIGGER [dbo].UpdateDepartmentid  
ON [dbo].[zdyemployee]  
AFTER Update  
AS   
BEGIN  
update zdyemployee   
set depid= e.depid ,depname=e.depname  
from Update e INNER join zdyemployee z   
on z.empid=e.empid   
END

**132、数据库连接池的工作机制**

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表记为忙。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表记为空闲，其他调用就可以使用这个连接

**133、Hibernate缓存机制**

[缓存](http://baike.so.com/doc/4597671-4809507.html" \t "_blank)是介于[应用程序](http://baike.so.com/doc/3417785-3597266.html" \t "_blank)和物理数据源之间，其作用是为了降低应用程序对物理数据源访问的频次，从而提高了应用的运行性能。缓存内的数据是对物理数据源中的数据的复制，[应用程序](http://baike.so.com/doc/3417785-3597266.html" \t "_blank)在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据

**134、懒汉式和饿汉式的优缺点**

所谓“懒汉式”与“饿汉式”的区别，是在与建立单例对象的时间的不同。

“懒汉式”是在你真正用到的时候才去建这个单例对象

“饿汉式”是在不管你用的用不上，一开始就建立这个单例对象；

它有以下几个要素：私有的构造方法；指向自己实例的私有静态引用；以自己实例为返回值的静态的公有的方法

**135、一个参数既可以是const还可以是volatile吗？**

可以 就是说 这个参数程序不能改 但是外部硬件可以改

所以使用const volatile修饰变量表明变量值只由外部条件改变，不会出现其它的副作用。比如说 外部的一个寄存器 对于CPU来说是只读的 它的值就可以设置成这种类型

**136、java中常用的组件有哪些？**

1.标识(Label) ：语法格式: Label(String s),功能: 构造一个显示字符串s的标识。  
2.按钮(Button) ： 语法格式:Button (String s) 功能:构造一个以字符串s为标识的按钮。  
3.复选框(Checkbox)：语法格式: Checkbox (String s) 功能:构造一个以字符串s为标识的复选框条目，未被选中。  
4.[单选按钮](https://www.baidu.com/s?wd=%E5%8D%95%E9%80%89%E6%8C%89%E9%92%AE&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4njfsnWf3rjm4rH6vuhNb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En10dnHndn1c1n1RsnWcLPHcY" \t "_blank)(Radio Buttons)：语法格式: CheckboxGroup( ) 功能: 构造一个条目组。  
5.选择菜单(Choice Menu) ： 语法格式: Choice ( ) 功能:构造一个选择菜单。说明:构造完之后，再使用Choice类中的addItem方法加入菜单的条目。  
6.列表框(Scrolling List) ：语法格式:List (int n, boolean b)功能:构造一个有n个列表项的列表框，并根据b的值决定是否允许多选。  
7.单行文本输入框(TextField)：语法格式:TextField (int i) 功能:构造一个字符串长度为i单行文本输入框  
8.多行文本输入框(TextArea)：语法格式: TextArea (int i, int j)功能: 构造一个多行文本输入框,行数为i,列数为j

**137、Linux常用的操作命令** shutdown -h now：关机命pwd：查看当前所在目录su:启用超级用户 d：打开目录 cd：命令用来改变所在目录 cd /：转到根目录中 cd ~：转到/home/user用户目录下 cd /usr： 转到根目录下的usr目录中--[绝对路径](https://www.baidu.com/s?wd=%E7%BB%9D%E5%AF%B9%E8%B7%AF%E5%BE%84&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9PjTYuHnvnW6knH-9mvf40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1bkrjT4n1TkrHDknjTznj0d" \t "_blank) cd test： 转到[当前目录](https://www.baidu.com/s?wd=%E5%BD%93%E5%89%8D%E7%9B%AE%E5%BD%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9PjTYuHnvnW6knH-9mvf40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1bkrjT4n1TkrHDknjTznj0d" \t "_blank)下的test子目录中--[相对路径](https://www.baidu.com/s?wd=%E7%9B%B8%E5%AF%B9%E8%B7%AF%E5%BE%84&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9PjTYuHnvnW6knH-9mvf40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1bkrjT4n1TkrHDknjTznj0d" \t "_blank)ls –a：查看目录下文件 mkdir：创建文件目录 rm –rf：删除文件或目录 rm ri：删除文件前询问用户 cp：复制文件命令 源文件 目标文件夹 mv：移动文件或修改文件名 history：查看当前用户执行过哪些命令(默认是记录1000条)! 历史编号：重新执行某一个历史命令 Pwd：查看当前用户停留在哪个目录中 Top：查看任务管理器： ps -au:静态查看任务管理器 ifconfig：查看ip地址 mkdir 文件名()：创建单个文件夹 mkdir -p /xx/xx/xx/xx：创建多级目录 touch 文件名：创建文件 file 文件名：查看某一文件时文件夹还是普通文件

**138、 Spring的事务有哪几种方式，隔离级别？**

原子性（一组操作不可分割），一致性（操作之前和操作之后状态不变），隔离性（事务和事务之间相互隔离互不干扰），持久性（当数据保存到数据库中之后，即使程序崩溃数据还是完好无损）

修改数据库的级别语句为：set global transaction isolation level (事务级别)

数据库中4中隔离级别优先级大小

Read unmmitted (可以引发脏度，张三读取了李四还未提交的数据，造成人才两空) < read Committed(可以避免脏读，但会产生不可重复读，多次读取结果不一样) < repeatable read （会引发虚读）< serializable（可以解决虚读）

**1****39、Adriod常用的数据库及框架**

在项目开发中一般会用数据库来进行数据持久化存储，常用的有两种方式：   
1、SQLite

SQLite，是一款轻型的数据库，是遵守[ACID](http://baike.baidu.com/item/ACID/10738" \t "_blank)的关系型[数据库管理系统](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F" \t "_blank)，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目。它的设计目标是[嵌入式](http://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F" \t "_blank)的，而且目前已经在很多[嵌入式产品](http://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F%E4%BA%A7%E5%93%81" \t "_blank)中使用了它，它占用资源非常的低，在[嵌入式设备](http://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F%E8%AE%BE%E5%A4%87" \t "_blank)中，可能只需要几百K的内存就够了  
2、ORMLite框架

 ORMlite是类似hibernate的对象映射框架，主要面向java语言，同时，是时下最流行的android面向数据库的的编程工具。

**140、什么是jquery对象和dom对象**

1，DOM对象是我们用传统的方法（[JavaScript](http://lib.csdn.net/base/javascript" \t "_blank" \o "JavaScript知识库)）获得的对象，Query对象就是用jQuery的类库选择器获得的对象。

2，JQuery对象就是通过jQuery包装DOM对象后产生的对象。JQuery对象是jQuery独有的，其可以使用jQuery里的方法，但是不能使用DOM的方法

3，DOM对象就是javascript固有的一些对象。DOM对象能使用javascript固有的方法，但是不能使用jQuery里面的方法。

相互转化：var$cr=$("#cr"); //jquery对象  
 var cr = $cr[0]; //dom对象 也可写成 var cr=$cr.get(0);

**141、什么是java Hotspot？**

HotSpot是较新的Java虚拟机，用来代替JIT(Just in Time)，可以大大提高Java运行的性能。 HotSpot包括一个解释器和两个编译器（client 和 server，二选一的），解释与编译混合执行模式，默认启动解释执行。编译器：java源代码被编译器编译成class文件（字节码），java字节码在运行时可以被动态编译（JIT）成本地代码(前提是解释与编译混合执行模式且虚拟机不是刚启动时)。   解释器： 解释器用来解释class文件（字节码），java是解释语言（书上这么说的）。

    server启动慢，占用内存多，执行效率高，适用于服务器端应用；

    client启动快，占用内存小，执行效率没有server快，默认情况下不进行动态编译，适用于桌面应用程序

**142、cookie和session的区别：**

1、session保存在服务器，客户端不知道其中的信息；cookie保存在客户端，服务器能够知道其中的信息。

2、session中保存的是对象，cookie中保存的是字符串。

3、session不能区分路径，同一个用户在访问一个网站期间，所有的session在任何一个地方都可以访问到。而cookie中如果设置 了路径参数，那么同一个网站中不同路径下的cookie互相是访问不到的。

4、session默认需要借助cookie才能正常工作。如果客户端完全禁止cookie，session，这种方法将失效。但是如果服务器端启用了url编码，也就是用URLEncoder.encode("index.jsp?id=3","UTF-8");..把所有的url编码了，则会在url后面出现如下类似的东西index.jsp:jsessionid=fdsaffjdlksfd124324lkdjsf?id=3服务器通过这个进行session的判断。

5 session在用户会话结束后就会关闭了，但cookie因为保存在客户端，可以长期保存

6 cookie:是服务端向客户端写入的小的片段信息。cookie信息保存在服务器缓存区，不会在客户端显现。当你第一次登陆一个网站，服务器向你的机器写得片段信息。你可以在Internet选项中找到存放cookie的文件夹。如果不删除，cookie就一直在这个文件夹中。

**143、请写出以下缩写的意义，JSSE、HTTPS、HTTP、SSL?**

JSSE是基于安全算法和握手机制之上的合成体。JSSE将危险的安全弱点降到最低点，并且它减轻了开发者的负担，使得开发者可以很轻松的整合到程序中。

HTTP是客户端浏览器或其他程序与[Web服务](http://baike.baidu.com/item/Web%E6%9C%8D%E5%8A%A1" \t "_blank)器之间的应用层通信协议。

HTTPS（全称：Hyper Text Transfer Protocol over Secure Socket Layer），是以安全为目标的[HTTP](http://baike.baidu.com/item/HTTP" \t "_blank)通道，简单讲是HTTP的安全版

HTTPS和HTTP的区别主要为以下四点：

一、https协议需要到ca申请证书，一般免费证书很少，需要交费。

二、http是[超文本传输协议](http://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE" \t "_blank)，信息是明文传输，https 则是具有[安全性](http://baike.baidu.com/item/%E5%AE%89%E5%85%A8%E6%80%A7" \t "_blank)的[ssl](http://baike.baidu.com/item/ssl" \t "_blank)加密传输协议。

三、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。

四、http的连接很简单，是无状态的；HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的[网络协议](http://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%8D%8F%E8%AE%AE" \t "_blank)，比http协议安全。

SSL(Secure Sockets Layer ,及其继任者[传输层安全](http://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%B1%82%E5%AE%89%E5%85%A8" \t "_blank)（Transport Layer Security，TLS）是为[网络通信](http://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E9%80%9A%E4%BF%A1" \t "_blank)提供安全及[数据完整性](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AE%8C%E6%95%B4%E6%80%A7" \t "_blank)的一种安全协议。TLS与SSL在[传输层](http://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%B1%82" \t "_blank)对网络连接进行加密

**144、迭代器的使用：**

　　迭代器是一种设计模式，它是一个对象，它可以遍历并选择序列中的对象，而开发人员不需要了解该序列的底层结构。迭代器通常被称为“轻量级”对象，因为创建它的代价小。

　　Java中的Iterator功能比较简单，并且只能单向移动：

　　(1) 使用方法iterator()要求容器返回一个Iterator。第一次调用Iterator的next()方法时，它返回序列的第一个元素。注意：iterator()方法是java.lang.Iterable接口,被Collection继承。

　　(2) 使用next()获得序列中的下一个元素。

　　(3) 使用hasNext()检查序列中是否还有元素。

(4) 使用remove()将迭代器新返回的元素删除。

**145、ssh理解**

Struts就是一个MVC的架构,在控制器,视图和模型方面都有其特定的组件，在SSH框架中，更多的是利用了Controller这个组件。

Spring实现了反转控制（Inversion of Control）和面向方面编程(Achievement-Oriented Programming)。Spring框架的能实现应用的分层架构,在应用中要使用的对象可以被一个简单的XML文件连在一起，这个XML文件含有到像事务管理器、对象工厂、包含业务逻辑的服务对象、和数据存取对象这些对象的引用。

Hibernate是一种ORM映射工具，它不仅提供了从Java类到数据表之间的映射，也提供了数据查询和恢复机制。相对于使用JDBC和SQL来手工操作数据库，使用Hibernate，可以大大减少操作数据库的工作量。

**146、ssm的理解**

Spring就像是整个项目中装配bean的大工厂，在配置文件中可以指定使用特定的参数去调用实体类的构造方法来实例化对象。

Spring的核心思想是IoC（控制反转），即不再需要程序员去显式地`new`一个对象，而是让Spring框架帮你来完成这一切。

SpringMVC在项目中拦截用户请求，它的核心Servlet即DispatcherServlet承担中介或是前台这样的职责，将用户请求通过HandlerMapping去匹配Controller，Controller就是具体对应请求所执行的操作。SpringMVC相当于SSH框架中struts。

mybatis是对jdbc的封装，它让数据库底层操作变的透明。mybatis的操作都是围绕一个sqlSessionFactory实例展开的。mybatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。

**147、Adriod的四大组件及作用**

**1、Activity 2、 service 3、 content provider 4、broadcast receiver**

1、activity

（1）一个Activity通常就是一个单独的屏幕（窗口）。

（2）Activity之间通过Intent进行通信。

（3）android应用中每一个Activity都必须要在AndroidManifest.xml配置文件中声明，否则系统将不识别也不执行该Activity。

2、service

（1）service用于在后台完成用户指定的操作。service分为两种：

（a）started（启动）：当应用程序组件（如activity）调用startService()方法启动服务时，服务处于started状态。

（b）bound（绑定）：当应用程序组件调用bindService()方法绑定到服务时，服务处于bound状态。

(2)startService()与bindService()区别：

(a)started service（启动服务）是由其他组件调用startService()方法启动的，这导致服务的onStartCommand()方法被调用。当服务是started状态时，其生命周期与启动它的组件无关，并且可以在后台无限期运行，即使启动服务的组件已经被销毁。因此，服务需要在完成任务后调用stopSelf()方法停止，或者由其他组件调用stopService()方法停止。

(b)使用bindService()方法启用服务，调用者与服务绑定在了一起，调用者一旦退出，服务也就终止，大有“不求同时生，必须同时死”的特点。

(3)开发人员需要在应用程序配置文件中声明全部的service，使用<service></service>标签。

(4)Service通常位于后台运行，它一般不需要与用户交互，因此Service组件没有图形用户界面。Service组件需要继承Service基类。Service组件通常用于为其他组件提供后台服务或监控其他组件的运行状态。

3、content provider

（1）android平台提供了Content Provider使一个应用程序的指定数据集提供给其他应用程序。其他应用可以通过ContentResolver类从该内容提供者中获取或存入数据。

（2）只有需要在多个应用程序间共享数据是才需要内容提供者。例如，通讯录数据被多个应用程序使用，且必须存储在一个内容提供者中。它的好处是统一数据访问方式。

（3）ContentProvider实现数据共享。ContentProvider用于保存和获取数据，并使其对所有应用程序可见。这是不同应用程序间共享数据的唯一方式，因为android没有提供所有应用共同访问的公共存储区。

（4）开发人员不会直接使用ContentProvider类的对象，大多数是通过ContentResolver对象实现对ContentProvider的操作。

（5）ContentProvider使用URI来唯一标识其数据集，这里的URI以content://作为前缀，表示该数据由ContentProvider来管理。

4、broadcast receiver

（1）你的应用可以使用它对外部事件进行过滤，只对感兴趣的外部事件(如当电话呼入时，或者数据网络可用时)进行接收并做出响应。广播接收器没有用户界面。然而，它们可以启动一个activity或serice来响应它们收到的信息，或者用NotificationManager来通知用户。通知可以用很多种方式来吸引用户的注意力，例如闪动背灯、震动、播放声音等。一般来说是在状态栏上放一个持久的图标，用户可以打开它并获取消息。

（2）广播接收者的注册有两种方法，分别是程序动态注册和AndroidManifest文件中进行静态注册。

（3）动态注册广播接收器特点是当用来注册的Activity关掉后，广播也就失效了。静态注册无需担忧广播接收器是否被关闭，只要设备是开启状态，广播接收器也是打开着的。也就是说哪怕app本身未启动，该app订阅的广播在触发时也会对它起作用。

## **148、String s = new String("xyz");创建了几个String Object?**

2个，1个在字符串常量池中，一个在堆内存中。

## **149、是否可以在静态（static）方法内部调用非静态（static）方法？**

不可以,如果其中包含对象的method()；不能保证对象初始化。

## **150、heap 和stack 有什么区别。**

（栈）stack的空间由操作系统自动分配和释放，空间小，读写速度快，（堆）heap的空间是手动申请和释放的，空间大，速度相对栈来说较慢。在Java中声明一个对象，会先在栈内存中为其分配地址空间，若再new一下，实例化它，则在堆内存中为其分配地址空间。

## **151、抽象类为什么不能实例化？**

答：抽象类里面可能含有抽象方法。抽象方法是不能被实现的。如果一个实例调用了抽象方法就会出错。为了安全，抽象类和接口都不能被实例化，这是java的规范。

**152、ioc、di和aop**

**Ioc**：Ioc—Inversion of Control，即“控制反转”，不是什么技术，而是一种设计思想。在Java开发中，Ioc意味着将你设计好的对象交给容器控制，而不是传统的在你的对象内部直接控制。

**DI：**DI—Dependency Injection，即“依赖注入”：组件之间依赖关系由容器在运行期决定，形象的说，即由容器动态的将某个依赖关系注入到组件之中。依赖注入的目的并非为软件系统带来更多功能，而是为了提升组件重用的频率，并为系统搭建一个灵活、可扩展的平台。通过依赖注入机制，我们只需要通过简单的配置，而无需任何代码就可指定目标需要的资源，完成自身的业务逻辑，而不需要关心具体的资源来自何处，由谁实现。 理解DI的关键是：“谁依赖谁，为什么需要依赖，谁注入谁，注入了什么”，那我们来深入分析一下：

　　●谁依赖于谁：当然是应用程序依赖于IoC容器；

　　●为什么需要依赖：应用程序需要IoC容器来提供对象需要的外部资源；

　　●谁注入谁：很明显是IoC容器注入应用程序某个对象，应用程序依赖的对象；

　　●注入了什么：就是注入某个对象所需要的外部资源（包括对象、资源、常量数据）。

IoC和DI由什么关系呢？其实它们是同一个概念的不同角度描述，由于控制反转概念比较含糊（可能只是理解为容器控制对象这一个层面，很难让人想到谁来维护对象关系），所以2004年大师级人物Martin Fowler又给出了一个新的名字：“依赖注入”，相对IoC 而言，“依赖注入”明确描述了“被注入对象依赖IoC容器配置依赖对象”。

**AOP：**即面向切面编程，可以说是OOP（Object Oriented Programming，面向对象编程）的补充和完善。OOP引入封装、继承、多态等概念来建立一种对象层次结构，用于模拟公共行为的一个集合。

AOP、OOP在字面上虽然非常类似，但却是面向不同领域的两种设计思想。OOP（面向对象编程）针对业务处理过程的实体及其属性和行为进行抽象封装，以获得更加清晰高效的[逻辑单元](http://baike.so.com/doc/922770.html" \t "_blank)划分。

而AOP则是针对业务处理过程中的切面进行提取，它所面对的是处理过程中的某个步骤或阶段，以获得逻辑过程中各部分之间低[耦合性](http://baike.so.com/doc/5725915.html" \t "_blank)的隔离效果。这两种设计思想在目标上有着本质的差异。

动态的将代码块切入到制定的位置和方法中

## **153、抽象类的方法是否可同时是static,是否可同时是native，是否可同时是synchronized?**

答：抽象方法不可被上述三个中的任意一个修饰，抽象方法只能被public 或protected 修饰，其他修饰符皆不可以。抽象类就是没有什么具体的实现,主要是让子类来实现。如果在再abstract 之前加static,native,synchronized 这些修饰符没有任何意义。

## **154、java 中有几种类型的流？JDK 为每种类型的流提供了一些抽象类以供继承，请说出他们分别是哪些类？**

答：java中有字节流，字符流。字节流继承于InputStream OutputStream，字符流继承于InputStreamReader OutputStreamWriter。在java.io包中还有许多其他的流，主要是为了提高性能和使用方便。

## **155、Class.forName()的作用?**

按参数中指定的字符串形式的类名去搜索并加载相应的类，如果该类字节码已经被加载过，则返回代表该字节码的 Class 实例对象。在jdbc中，调用该方法加载数据库驱动器类的对象。此对象用来与特定的数据库连接

## **156、XML 与HTML 的主要差异()**

答：XML 不是HTML 的替代。XML 和HTML 为不同的目的而设计：

XML 被设计为传输和存储数据，其焦点是数据的内容。

HTML 被设计用来显示数据，其焦点是数据的外观。

HTML 旨在显示信息，而XML 旨在传输信息。

## **157、HashMap 和Hashtable 的区别。**

HashMap 是Hashtable 的轻量级实现（非

线程安全的实现），他们都完成了Map 接口，主要区别在于HashMap 允许null作为键和值，非线程安全，效率上可能高于Hashtable。

Hashtable 不允许null作为键和值。

Hashtable 的方法是Synchronized 的，而HashMap 不是，在多个线程访问Hashtable 时，不需要自己为它的方法实现同步，而HashMap 就必须为之提供外同步。Hashtable 和HashMap 采用的hash/rehash 算法都大概一样，所以性能不会有很大的差异。

## **158、abstract class 和interface 有什么区别?**

抽象类和接口都不能直接实例化，接口可以多实现，类只能单继承，

接口只能声明公共的静态常量和抽象方法，抽象类可以声明普通变量和实现方法，接口用于设计，他的关注点是一系列操作，抽象类用于重构，他主要关注一些共同特征。

**159、forward 和redirect 的区别**

forward转发是服务器直接访问目标地址的URL，把响应内容读取过来，然后发给浏览器，地址栏不改变，以前的request中存放的变量不会失效，但是如果刷新浏览器会转到开始的页面。

redirect重定向就是服务端重新去请求一个地址，浏览器地址栏会刷新，重定向比转发的速度慢，因为浏览器还要发送一个新的请求，而且经过重定向后，request内的变量全部失效，而进入一个新的request作用域。

**160、线程和进程有什么区别？**

线程是进程的子集，一个进程可以有很多线程，每条线程并行执行不同的任务。不同的进程使用不同的内存空间，而所有的线程共享一片相同的内存空间。

161、面向接口编程的好处

接口只显示入口参数和返回参数，而具体的实现类，和方法实现都可以完全隐藏起来，这样做就可以制定一个统一的标准，即使因为开发人员人多脑杂，模块间的代码风格和思路不同，也完全不影响整个系统的架构，这样使得系统的复杂性得到有效控制，降低开发的难度和成本。

## **162、Thread 类中的start() 和 run() 方法有什么区别？**

start()方法被用来启动新创建的线程，而且start()内部调用了run()方法，这和直接调用run()方法的效果不一样。Run()只会在原来的线程中运行，start()方法才会启动新线程。

## **163、什么是线程？**

线程是程序进程中的最小运作单位。线程有新建、就绪、运行、阻塞和死亡几种基本状态。

1.新建状态(New)：

        当用new操作符创建一个线程时， 例如new Thread(r)，线程还没有开始运行，此时线程处在新建状态。

2.就绪状态(Runnable)

        一个新创建的线程并不自动开始运行，要执行线程，必须调用线程的start()方法。当线程对象调用start()方法即启动了线程，start()方法创建线程运行的系统资源，并调度线程运行run()方法。当start()方法返回后，线程就处于就绪状态。

        处于就绪状态的线程并不一定立即运行run()方法，线程还必须同其他线程竞争CPU调度时间，只有获得CPU调度时间才可以运行线程。因为在单CPU的计算机系统中，不可能同时运行多个线程，一个时刻仅有一个线程处于运行状态。因此此时可能有多个线程处于就绪状态。

3.运行状态(Running)

        当线程获得CPU调度时间后，它才进入运行状态，真正开始执行run()方法.

4. 阻塞状态(Blocked)

   线程运行过程中，可能由于各种原因进入阻塞状态:

        1>线程通过调用sleep方法进入睡眠状态；

        2>线程调用一个在I/O上被阻塞的操作，即该操作在输入输出操作完成之前不会返回到它的调用者；

        3>线程试图得到一个锁，而该锁正被其他线程持有；

        4>线程在等待某个触发条件；

        ......

        所谓阻塞状态是正在运行的线程没有运行结束，暂时让出CPU，这时其他处于就绪状态的线程就可以获得CPU调度，进入运行状态。

 5. 死亡状态(Dead)

        有两个原因会导致线程死亡：

        1) run方法正常退出，自然死亡，

        2) 一个未捕获的异常终止了run方法而使线程猝死。

        为了确定线程在当前是否存活着，需要使用isAlive方法。如果是可运行或被阻塞，这个方法返回true； 如果线程仍旧是new状态且不是可运行的， 或者线程死亡了，则返回false.

## **164、如何在Java中实现线程？**

可以继承Thread 类或者直接调用Runnable接口来重写run()方法实现线程。

## **165、用Runnable还是Thread？**

Java是单继承，多实现。如果要继承其他类，实现Runnable接口比较好。

## **166、如何处理并发和同步**

一种是java中的同步锁，synchronized，用于线程同步。

另外一种是数据库的悲观锁和乐观锁。如果是分布式的系统，只能利用数据库的锁来实现。

## **167、什么是线程安全？Vector是一个线程安全类吗？**

如果代码所在的进程中有多个线程在同时运行，而这些线程可能会同时运行这段代码。如果每次运行结果和单线程运行的结果是一样的，而且其他的变量的值也和预期的是一样的，就是线程安全的。一个线程安全的计数器类的同一个实例对象在被多个线程使用的情况下也不会出现计算失误。可以将集合类分成两组，线程安全和非线程安全的。Vector 是用同步方法来实现线程安全的, 而和它相似的ArrayList不是线程安全的

## **168、 Java多线程中调用wait() 和 sleep()方法有什么不同？**

Java程序中wait 和 sleep都会造成某种形式的暂停。wait()方法用于线程间通信，如果等待条件为真且其它线程被唤醒时它会释放锁，而sleep()方法仅仅释放CPU资源或者让当前线程停止执行一段时间，但不会释放锁

## **169、同步和异步的区别和联系**

同步关键字synchronized，假如这个同步的监视对象是类的话，那么如果当一个对象访问类里面的同步方法的话，其它的对象如果想要继续访问类里面的这个同步方法，就会进入阻塞，只有等前一个对象执行完该同步方法后当前对象才能够继续执行该方法。这就是同步。相反，如果方法前没有同步关键字修饰的话，那么不同的对象可以在同一时间访问同一个方法，这就是异步

## **170、常见并发同步案例分析**

火车票订票系统。

同时存在多人抢购一张票，多个线程进来，要保证多个用户能同时看到有票，一个种是对象锁，用于线程同步；另外一种是数据库的锁；如果是分布式的系统，只能利用数据库的锁来实现。

## **171、常见的提高并发下访问的效率的手段**

1.服务器网络带宽不够可以增加网络带宽，DNS域名解析分发多台服务器。

2.负载均衡，前置代理服务器nginx、apache等等

3.数据库查询优化，读写分离，分表等等

172、什么是AJAX为什么要用ajax：  
AJAX是一种交互式网页开发技术。  
1. 通过异步模式，提升了用户体验  
2. 优化了浏览器和服务器之间的传输，减少不必要的数据往返

3. Ajax引擎在客户端运行，承担了一部分本来由服务器承担的工作，从而减少了大用户量下的服务器负载

173、AJAX最大的特点是什么。  
Ajax可以实现动态不刷新（局部刷新）  
就是能在不更新整个页面的前提下维护数据。这使得Web应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变过的信息。

## **174、Java 中的构造器链是什么？**

答：当你从一个构造器中调用另一个构造器，就是Java 中的构造器链。这种情况只在重载了类的构造器的时候才会出现。

## **175、32 位和 64 位的 JVM中，int 类型变量的长度是多大？最大堆内存分别是多数？**

答：32 位和 64 位的 JVM 中，int 类型变量的长度是相同的，都是 32 位或者 4 个字节。

## **176、System.out.println()里面， 三个单词分别代表什么意思？**

System是系统提供的预定义的final类。

out是一个打印流（PrintStream）对象。

println是out对象里面一个重载的方法。

177、AJAX应用和传统Web应用有什么不同。  
 在传统的Javascript编程中，如果想得到服务器端数据库或文件上的信息，或者发送客户端信息到服务器，需要建立一个HTML form然后GET或者POST数据到服务器端。用户需要点击”Submit”按钮来发送或者接受数据信息，然后等待服务器响应请求，页面重新加载。因为服务器每次都会返回一个新的页面， 所以传统的web应用有可能很慢而且用户交互不友好。  
使用AJAX技术， 就可以使Javascript通过XMLHttpRequest对象直接与服务器进行交互。 一个web页面可以发送一个请求到web服务器并且接受web服务器返回的信息(不用重新加载页面)，展示给用户的还是同一个页面，也看不到到Javascript后台进行的发送请求和接受响应。

178、AJAX都有哪些优点和缺点？  
1、最大的一点是页面无刷新，用户的体验非常好。  
2、使用异步方式与服务器通信，具有更加迅速的响应能力。  
3、可以把以前一些服务器负担的工作转嫁到客户端，ajax的原则是“按需取数据”，可以最大程度的减少冗余请求和响应，减少对服务器造成的负担。  
4、已经被广泛支持，不需要下载插件或者小程序。  
ajax的缺点  
1、ajax不支持浏览器back按钮。  
2、安全问题 AJAX暴露了与服务器交互的细节。  
5、不容易调试。

## **179、Mybatis缓存？**

一级缓存：Mybatis的一级缓存的作用域是session，当openSession()后，如果执行相同的SQL（相同语句和参数），Mybatis不执行SQL，而是直接从缓存中返回查询结果。

二级缓存：Mybatis的二级缓存的作用域是一个mapper的namespace，同一个namespace中查询sql可以从缓存中获取结果。二级缓存是可以跨session的。

## **180、Java程序怎么优化？**

（1）程序设计的方法和模式（2）java布署的环境

## **181、数据连接池的工作机制是什么?**

客户端程序需要连接时，连接池驱动程序会返回一个未使用的连接池连接并将其表记为忙。如果当前没有空闲连接，连接池驱动程序就新建一定数量的连接，新建连接的数量由配置参数决定。当使用的连接池连接调用完成后，连接池驱动程序将此连接表记为空闲，其他程序就可以使用这个连接。

## **182、继承的原则？**

继承使得一个对象可以获取另一个对象的属性。使用继承可以让已经测试完备的功能得以复用，并且可以一次修改，所有继承的地方都同时生效。

## **183、Iterator和ListIterator的区别是什么？**

iterator用来遍历Set和List集合，ListIterator只能用来遍历List。

Iterator对集合只能是前向遍历，ListIterator既可以前向也可以后向。

ListIterator实现了Iterator接口，并包含其他的功能，比如：增加元素，替换元素，获取前一个和后一个元素的索引，等等。

## **184、什么是递归，什么时候使用递归？**

递归就是函数调用函数本身，当一个功能被重复使用，而每一次使用该功能时的参数不确定，都由上次的功能元素结果来确定。简而言之：功能内部又用到该功能，但是传递的参数值不确定。

## **185、springMVC工作原理?**

spring mvc核心是DispatcherServlet,spring将请求交给DispatcherServlet，他去查询HandlerMapping,找到处理请求的Controller.将请求提交到Controller进行业务逻辑处理后，会返回一个ModelAndView对象，Dispathcher查询ViewResolver视图解析器,找到ModelAndView对象指定的页面返回给客户端。

## **186、正则表达式的好处和弊端？常用操作分为哪几种？**

好处：对字符串的复杂操作变得更为简单

弊端：可阅读性差

比较常用的有：匹配，切割，替换

## **187、简述一下JVM加载class文件的原理机制 ？**

JVM中类的装载是由ClassLoader类加载器来实现的，他是一个Java系统组件，它负责在运行时查找和装入类文件的类

## **188、你所知道的应用服务器合 web server？**

应用服务器：Weblogic、Tomcat、Jboss

web server：IIS、Apache

## **189、什么是存储过程及其优点？**

[存储过程](https://www.baidu.com/s?wd=%E5%AD%98%E5%82%A8%E8%BF%87%E7%A8%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9nW7-rAf1uHbknhDLrHRv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1bznjfsPjRsPj0vPHc3rHcd" \t "_blank)是一组预先编绎好的SQL语句。将其放在服务器上，由用户通过指定[存储过程](https://www.baidu.com/s?wd=%E5%AD%98%E5%82%A8%E8%BF%87%E7%A8%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9nW7-rAf1uHbknhDLrHRv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1bznjfsPjRsPj0vPHc3rHcd" \t "_blank)的名字来执行它。

优点：（1）执行速度快（2）[模块化程序设计](https://www.baidu.com/s?wd=%E6%A8%A1%E5%9D%97%E5%8C%96%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9nW7-rAf1uHbknhDLrHRv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1bznjfsPjRsPj0vPHc3rHcd" \t "_blank)（3）减少网络通信量（4）保证系统的安全性

## **190、动态查询如何实现？表的结构变化后，如果不修改程序，如何设计和实现查询？**

将查询封装到存储过程中，通过调用存储过程实现动态查询；

表的结构发生变化后修改相应的存储过程即可在不修改程序的情况下实现查询

191、什么是Java虚拟机？为什么Java被称作是“平台无关的编程语言”？  
java的跨平台不是java源程序的跨平台 ， java源程序先经过javac编译器编译成二进制的.class字节码文件（java的跨平台指的就是.class字节码文件的跨平台，.class字节码文件与平台无关），.class文件再运行在jvm上，java解释器会将其解释成对应平台的机器码执行，所以java所谓的跨平台就是在不同平台上安装了不同的jvm，而在不同平台上生成的.class文件都是一样的，而.class文件再由对应平台的jvm解释成对应平台的机器码执行。

## **192、”static”关键字是什么意思？Java中是否可以覆盖(override)一个private或者是static的方法？**

“static”用来声明变量或者方法，可以在没有所属类的实例下被访问。  
Java中static方法不能被覆盖，因为方法覆盖是基于运行时动态绑定的，而static方法是编译时静态绑定的。Private修饰的方法是私有的，override重写必须是继承关系，父类私有的方法对子类是不可见的，所以不能。

## **193、什么是值传递和引用传递？**

值传递是对基本型变量而言的,传递的是该变量的一个副本,改变副本不影响原变量.

引用传递一般是对于引用型变量而言的,传递的是该对象地址的一个副本, 并不是原对象本身 。 所以对引用对象进行操作会同时改变原对象.

java内的传递都是值传递.将对象作为参数传给方法，方法接受的这是这个对象的副本，改变副本不会影响原来的对象。

## **194、如何确保N个线程可以访问N个资源同时又不导致死锁？**

使用多线程的时候，一种非常简单的避免死锁的方式就是：指定获取锁的顺序，并强制线程按照指定的顺序获取锁。因此，如果所有的线程都是以同样的顺序加锁和释放锁，就不会出现死锁了。

## **195、Java中垃圾回收有什么目的？什么时候进行垃圾回收？**

垃圾回收的目的是识别并且丢弃应用不再使用的对象来释放和重用资源。

触发GC（Garbage Collector，垃圾回收）的条件：

（1）当应用程序空闲时，即没有应用线程在运行时，GC会被调用。

（2）Java堆内存不足时，GC会被调用

## **196、什么是Applet？**

Java应用小程序，可以直接嵌入到网页中，并能够产生特殊的效果，applet经编译后会产生.class文件，把.class文件嵌入到html页面中，用户在链接网页时，applet便会伴随网页一起下载到用户计算机运行。

applet主要用来创建动态交互的web应用程序。

197、什么是JDBC？  
JDBC（Java DataBase Connectivity）,是一种java链接数据库的技术。通过JDBC可以用纯Java语言和标准的SQL语句编写完整的数据库应用程序。

## **198、什么是JSP页面？**

JSP页面包含了静态HTML、jsp元素和java代码。他是一种动态页面技术。

## **199、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，不用明确的声明就可以直接使用他们。有九种：

application

page

request

response

session

exception

out

config

pageContext

## **200、JSP和Servlet有哪些相同点和不同点，他们之间的联系是什么？**

JSP是用于视图展示。JSP页面最终还是被服务器编译成Servlet。Servlet主要用于控制逻辑。

## **201、JSP中的九种内置对象和作用分别是什么？**

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：网页的属性在这里管理

Session：与请求有关的会话期，存储特定用户会话所需的属性及配置信息

Application：正在执行的内容

Out：用来传送回应的输出

Config：servlet的架构部件

Page：JSP网页本身

Exception：针对错误的网页，未捕捉的异常

## **202、Servlet的生命周期过程：**

当客户端第一次请求一个servlet时，服务器会创建这个servlet的实例，执行init()方法，然后根据请求方式执行doget/dopost方法，一个servlet类在服务器只会有一个实例。

## **203、El表达式的作用是什么？**

用来取域(pageContext、request、session、context)中的数据

## **204、简述MVC设计模式：**

在实际开发中jsp擅长做数据的显式，Servlet擅长请求控制，JavaBean擅长封装数据。Model是指javabean;view用jsp实现；Servlet充当ctroll。

## **205、什么是javabean？**

只是一种规范，一个类中的字段必须全部用private修饰符来修饰，必须为每一个属性提供一个访问器和修改器，就是get set方法，必须为这个类提供一个无参的构造方法。满足这些条件的类，我们就称这个类为JavaBean。

## **206、通过Java代码来连接数据库，从而对数据库进行CRUD的步骤有哪些？**

导入第三方jar包--加载数据库驱动-->创建连接对象（Connection）-->创建代表发送sql语句的Statement对象-->向数据库发送sql语句，获取代表结果集的ResultSet对象-->释放数据库资源

## **207、Json规范是什么样的，如何将一个实体类转化成JSon格式的数据？**

Json只是将数据以某一种规范进行展示，{“key1”:value1,“key2”,value2}。

## **208、TCP和UDP的区别**

TCP必须建立连接后才能通信，通信更加可靠，传送数据量大，速度慢。

UDP无须建立连接而直接发送数据包，数据的传输不可靠，但能够向若干个目标发送数据，接受来自若干个源的数据。

## **209、spring中创建对象时的初始化与销毁方法**

init-method、destroy-method属性

## **210、同时编译java两个类应带什么参数？**

CLASSPATH

## **211、ajax请求总共有多少种callback（回掉函数）**

Success、Failure、Loading、Complete、Exception

## **212、Try{}里面有一个return语句，那么紧跟在这个try后面的finally{}里的code会不会被执行**

会执行，在return前执行

## **213、当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并返回变化后的结果，那么这到底是值传递还是引用传递？**

是值传递。Java只有值传递参数，当一个对象实例作为一个参数被传递到方法中时，参数的值就是对该对象的引用。被调用那个的方法中改变的是对象的副本，但对象的引用是永远不会改变的

## **214、类被加载时，类中的静态声明执行的顺序**

静态属性初始化--->静态代码块---->类的构造函数

## **215、接口里的成员变量为什么都必须是final static修饰的**

接口只有抽象方法，不能通过行为来修改属性，所以接口的属性都是全局常量

## **216、jsp四大域及作用范围**

Page 当前页面有效

Request 当前请求中有效

Session 当前回话中有效

Application 当前应用程序中有效

## **217、什么是多线程，什么是线程安全，如何解决？**

多线程：在一个应用程序中，同时有多个不同的执行路径

线程安全：就是在多个线程共享一个数据会受到其他线程的干扰

如何解决：使用线程同步，给线程上锁。

## **218、java中会存在内存泄漏吗，请简单描述。**

Java存在内存泄漏，GC会自动释放内存中程序不需要的对象，但是对不需要的对象引用、没有及时释放系统资源会有内存泄漏。

## **219、静态变量和实例变量的区别**

静态变量属于类，不用实例化就已经存在，所有的实例共享同一个静态变量，可以通过类名和实例名来访问。实例变量属于实例，每个实例都独享自己的变量，只能通过实例名来访问。

## **220、访问控制符的各自作用域**

Public：所有类都可访问

Protected：当前包所有类及其子类均可访问

缺省：默认，只有当前包可以访问

Private：私有的，只有同一类可以访问

## **221、request对象的常用方法及作用（写三种）**

getAttribute(String name)：返回由name指定的属性值

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

getInputStream()：返回请求的输入流，用于获得请求中的数据

getParameter(String name)：获得客户端传送给服务器端的有name指定的参数值

getSession([Boolean create])：返回和请求相关Session

## **222、spring的DI两种给对象赋值的方法：**

通过set方法赋值，前提是实体类必须满足javabean规范

通过构造方法赋值

## **223、什么是过滤器**

过滤器，通常是指在java web工程中，在你传入request、response信息前，提前过滤掉一些信息，或者提前设置一些参数，然后再传入servlet进行业务逻辑的处理。

## **224、如何实现线程之间的通信**

使用Object类提供的wait()、notify()、notifyAll()三个方法。

## **225、简述反射技术**

反射就是动态加载一个指定的类，并获取该类中的所有内容，且将字节码文件和文件中的内容都封装成对象，便于操作。

## **226、关系数据库中的几种设计范式**

第一范式1NF是无重复的列; 第二范式2NF是实体的属性完全依赖于主关键字,即有自增主键；第三范式3NF就是属性不依赖于其它非主属性，即表之间通过主键关联。

## **227、常用的几种数据库建模工具**

powerDesign、together、Rational Rose

## **228、什么是缓存？常用的缓存技术？分布式缓存的实现方式？**

缓存是复制了频繁使用的数据以利于快速访问的技术；常用的有redis；实现方式：通过key做一致性哈希，实现key对应redis结点的分布。

## **229、说说你知道的HTML5技术**

Email Inputs：输入框的邮箱验证

Placeholder：这个input属性的意义就是不必通过javascript来做placeholder的效果了输入框的提示信息；

Required：输入框必填验证；

Autofocus：是聚焦到输入框里面；

Video支持；

Pattern属性:我们可以在markup里面直接使用正则表达式;

## **230、说说你知道的CSS3技术**

transform：旋转、缩放和移动元素；

transition: 渐变属性；

animation: 动画属性；

shadow：阴影效果；

opacity：透明效果；

## **231、lazy-init的属性和对应的特点**

bean标签的懒加载属性；

true：第一次索引bean的时候加载；

false：非懒加载

## **232、什么是aop和其作用？**

面向切面编程，是对oop的一种补充；作用：动态的将代码切入到类的指定方法的指定位置。

## **233、Dom，jdom，dom4j，sax的区别？**

Dom：丰富的API集合，方便导航，适合对xml修改；

Jdom：速度快，可选择性好，不支持dom遍历包；

Dom4j：灵活性高，移植性差，可选择性很好；

Sax：无需将内容加载到内存中因此内存消耗少，适合读取xml

## **234、java程序的种类**

内嵌于网页的Applet；可独立运行的Application；服务器端的Servlets

## **235、get和post的区别**

GET请求的数据会附在URL之后，POST不会；

POST的安全性要比GET的安全性高；

GET后退按钮/刷新无害，POST数据会被重新提交；

提交数据的编码类型不同。

## **236、构造器是否可以被重写？**

构造器不能被继承，因此不能重写,但可以被重载Overload

## **237、能不能自己写个类，也叫java.lang.String**

可以，但在应用的时候，需要用自己的类加载器去加载，否则，系统的类加载器永远只是去加载jre.jar包中的那个java.lang.String。

## **238、java中常用的设计模式，说出五种**

工厂模式、单例模式、策略模式、代理模式、桥接模式、装饰模式、观察者模式、访问者模式、迭代器模式

## **239、MVC 的各个部分都有那些技术来实现?**

用JSP技术实现视图的功能，用Servlet技术实现控制器的功能，用JavaBean技术实现模型的功能，通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

## **240、什么是策略模式？什么情况下使用？**

策略模式定义了一系列的算法，并将每一个算法封装起来，而且使它们还可以相互替换。策略模式让算法独立于使用它的客户而独立变化。

应用场景：

1、 多个类只区别在表现行为不同,可以使用Strategy模式,在运行时动态选择具体要执行的行为.(例如FlyBehavior和QuackBehavior)

2、 需要在不同情况下使用不同的策略(算法),或者策略还可能在未来用其它方式来实现.

## **241、Jdo 是什么?**

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化 API

## **242、页面间对象传递的方法**

request，session，application，cookie

## **243、请列举当前一些当前流行的数据库引擎,**

SQL SERVER,ORACLE,BDE,Microsoft，Jet。

## **244、GC 是什么? 为什么要有GC?**

GC 是垃圾收集的意思（Gabage Collection）,内存处理是编程人

员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统

的不稳定甚至崩溃，Java 提供的GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，Java 语言没有提供释放已分配内存的显示操作方法。

## **245、线程的基本概念、线程的基本状态以及状态之间的关系**

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。

## **246、list、set、map的区别？**

list、set 都是Collection的实现接口，list 元素有序可重复、set 无序不能重复、map 是键值对集合。

## **247、重载与重写区别？**

重载是同一个类中，方法名称相同， 但是参数或个数不同。与返回值没有关系。

重写是基于继承关系。子类与父类方法的返回值，方法名，形参列表形同。

## **248、什么是死锁(deadlock)？**

两个进程都在等待对方执行完毕才能继续往下执行的时候就发生了死锁。结果就是两个进程都陷入了无限的等待中

## **249、数组(Array)和列表(ArrayList)有什么区别？什么时候应该使用Array**

Array可以包含基本类型和对象类型，ArrayList只能包含对象类型。

Array大小是固定的，ArrayList的大小是动态变化的。

ArrayList操作方法更多。

对于基本类型数据，集合使用自动装箱来减少编码工作量。但是，当处理固定大小的基本数据类型的时候，这种方式相对比较慢。

## **250、有三种方式可以用来创建线程：继承Thread类，实现Runnable接口，应用程序可以使用Executor框架来创建线程池，你更喜欢哪一种？**

实现Runnable接口这种方式更受欢迎，因为这不需要继承Thread类。在应用设计中已经继承了别的对象的情况下，只能实现接口。同时，线程池也是非常高效的，很容易实现和使用。

## **251、在JAVA 中，如何跳出当前的多重嵌套循环？**

使用return break

## **252、一个“.java”源文件中是否可以包括多个类（非内部类）？有什么限制？**

一个源文件可以存在多个类，但是只能又一个和文件名同名的类

## **253、说一下什么是spring？**

Spring的核心是IOC控制反转和AOP面向切面。

将对象创建过程交给spring容器，通过容器管理对象的创建销毁， 将对象创建过程从编译时延期到运行时，通过注解注入依赖，实现针对接口编程。

Spring还提供了对现存的各种框架整合的方案。

## **254、简述 spring 的优缺点？**

优点：

a.使用Spring的IOC容器，将对象之间的依赖关系交给Spring，降低组件之间的耦合性，让我们更专注于应用逻辑

b.可以提供众多服务，事务管理，WS等。

c.AOP支持，方便面向切面编程。

d.对主流的框架提供了很好的集成支持，如hibernate,Struts2,mybatis等

e.Spring DI机制降低了业务对象替换的复杂性。

缺点:

b.Spring不支持分布式，这也是EJB仍然在用的原因之一。

## **255、spring 有哪几种注入方式？**

a、实例工厂的方法注入;

b、静态工厂的方法注入;

c、构造器注入;

d、Set注入;

## **256、解释一下什么是控制反转？**

Spring 通过一种称作控制反转（IoC）的技术促进了松耦合。当应用了 IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。

## **257、抽象关键字abstract和哪些不可以共存？**

final、private、static

## **258、简述 MVC 的优缺点。**

优点：

开发效率高。

松耦合，程序灵活性和扩展性好。

代码重用度更高。

易于人员分工。

缺点:

代码复杂度增加。

代码数量增加。

不适合开发小型项目。

## **259、jsp 有哪些动作?作用分别是什么?**

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。

## **260、char 型变量中能不能存贮一个中文汉字?为什么?**

能够存一个中文的，因为java 中以unicode 编码，一个char

占2 个字节(Byte)共16 位（bit），所以放一个中文是没问题的。

## **261、启动一个线程是用run()还是start()?**

启动一个线程是调用start()方法，使线程处于可运行状态。

## **262、error 和exception 有什么区别?**

error 表示JVM处理不了的严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。exception 表示一种设计或实现的问题，是可以用程序处理的。

## **263、数组有没有length()这个方法? String 有没有length()这个方法？**

数组没有length()这个方法，有length 的属性。String 有有length()这个方法。

## **264、多态的定义及实现方式。**

定义：同一引用，使用不同的对象，执行不同的方法

实现方式：将父类类型作为方法的参数；将父类类型作为方法的返回值类型

## **265、接口的优点**

对于修改关闭，对于扩展开放，便于协同开发

## **266、如何提高编码效率**

编码效率是一个长期积累的过程，工作中灵活运用工具类，将重用资源抽取成一个公共资源，多写文档和注释

## **267、EJB与JAVA BEAN的区别？**

Java Bean 是可复用的组件，对Java Bean并没有严格的规范，理论上讲，任何一个Java类都可以是一个Bean。但通常情况下，由于Java Bean是被容器所创建（如Tomcat）的，所以Java Bean应具有一个无参的构造器，另外，通常Java Bean还要实现Serializable接口用于实现Bean的持久性。Java Bean实际上相当于微软COM模型中的本地进程内COM组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于DCOM，即分布式组件。它是基于Java的远程方法调用（RMI）技术的，所以EJB可以被远程访问（跨进程、跨计算机）。但EJB必须被布署在诸如Webspere、WebLogic这样的容器中，EJB客户从不直接访问真正的EJB组件，而是通过其容器访问。EJB容器是EJB组件的代理，EJB组件由容器所创建和管理。客户通过容器来访问真正的EJB组件。

## **268、JSP中动态INCLUDE与静态INCLUDE的区别？**

动态INCLUDE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不会检查所含文件的变化，适用于包含静态页面

## **269、什么时候用assert。**

assertion(断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制。在实现中，assertion就是在程序中的一条语句，它对一个boolean表达式进行检查，一个正确程序必须保证这个boolean表达式的值为true；如果该值为false，说明程序已经处于不正确的状态下，系统将给出警告或退出。一般来说，assertion用于保证程序最基本、关键的正确性。assertion检查通常在开发和测试时开启。为了提高性能，在软件发布后，assertion检查通常是关闭的。

## **270、接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类(concrete class)?**

接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，可以继承，但是和实体类的继承一样，也要求父类可继承，并且拥有子类可访问到的构造器。

## **271、Set里的元素是不能重复的，那么用什么方法来区分重复与否呢? 是用==还是equals()? 它们有何区别?**

Set里的元素是不能重复的，那么用iterator()方法来区分重复与否。equals()是判读两个Set是否相等。

equals()和==方法决定引用值是否指向同一对象equals()在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值。

## **272、switch语句能否作用在byte上,能否作用在long上,能否作用在String上**

switch可作用于char byte short int  
 switch可作用于char byte short int对应的包装类

switch不可作用于long double float boolean，包括他们的包装类

switch中可以是字符串类型,String(jdk1.7之后才可以作用在string上)  
 switch中可以是枚举类型

## **273、J2EE是技术还是平台还是框架？**

J2EE本身是一个标准，一个为企业分布式应用的开发提供的标准平台。

J2EE也是一个框架，包括JDBC、JNDI、RMI、JMS、EJB、JTA等技术。

## **274、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？**

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)

## **275、说说你所熟悉或听说过的j2ee中的几种常用模式?及对设计模式的一些看法**

Session Facade Pattern：使用SessionBean访问EntityBean

Message Facade Pattern：实现异步调用

EJB Command Pattern：使用Command JavaBeans取代SessionBean，实现轻量级访问

Data Transfer Object Factory：通过DTO Factory简化EntityBean数据提供特性

Generic Attribute Access：通过AttibuteAccess接口简化EntityBean数据提供特性

Business Interface：通过远程（本地）接口和Bean类实现相同接口规范业务逻辑一致性

ＥＪＢ架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂，项目队伍越庞大则越能体现良好设计的重要性。

## **276、说出一些常用的类，包，接口，请各举5个**

常用的类：BufferedReader BufferedWriter FileReader FileWirter String Integer

常用的包：java.lang java.awt java.io java.util java.sql

常用的接口：Remote List Map Document NodeList

## **278、如何现实servlet的单线程模式**

<%@ page isThreadSafe=”false”%>

## **279、j2ee常用的设计模式？说明工厂模式。**

Java中的23种设计模式：

Factory（工厂模式）， Builder（建造模式）， Factory Method（工厂方法模式），

Prototype（原始模型模式），Singleton（单例模式）， Facade（门面模式），

Adapter（适配器模式）， Bridge（桥梁模式）， Composite（合成模式），

Decorator（装饰模式）， Flyweight（享元模式）， Proxy（代理模式），

Command（命令模式）， Interpreter（解释器模式）， Visitor（访问者模式），

Iterator（迭代子模式）， Mediator（调停者模式）， Memento（备忘录模式），

Observer（观察者模式）， State（状态模式）， Strategy（策略模式），

Template Method（模板方法模式）， Chain Of Responsibleity（责任链模式）

工厂模式：工厂模式是一种经常被使用到的模式，根据工厂模式实现的类可以根据提供的数据生成一组类中某一个类的实例，通常这一组类有一个公共的抽象父类并且实现了相同的方法，但是这些方法针对不同的数据进行了不同的操作。首先需要定义一个基类，该类的子类通过不同的方法实现了基类中的方法。然后需要定义一个工厂类，工厂类可以根据条件生成不同的子类实例。当得到子类的实例后，开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子类的实例。

## **280、为什么放到HttpSession中的对象必须要是可序列化的？**

没必须,不过session反序列化过程会导致对象不可用.

## **281、排序都有哪几种方法？请列举。用JAVA实现一个快速排序。**

排序的方法有：插入排序（直接插入排序、希尔排序），交换排序（冒泡排序、快速排序），选择排序（直接选择排序、堆排序），归并排序，分配排序（箱排序、基数排序）

快速排序的伪代码。

使用快速排序方法对a[ 0 :n- 1 ]排序

从a[ 0 :n- 1 ]中选择一个元素作为m i d d l e，该元素为支点把余下的元素分割为两段left 和r i g h t，使得l e f t中的元素都小于等于支点，而right 中的元素都大于等于支点

递归地使用快速排序方法对left 进行排序

递归地使用快速排序方法对right 进行排序

所得结果为l e f t + m i d d l e + r i g h t

## **282、请对以下在J2EE中常用的名词进行解释(或简单描述)**

web容器：给处于其中的应用程序组件（JSP，SERVLET）提供一个环境，使JSP,SERVLET直接更容器中的环境变量接口交互，不必关注其它系统问题。主要有WEB服务器来实现。例如：TOMCAT,WEBLOGIC,WEBSPHERE等。该容器提供的接口严格遵守J2EE规范中的WEB APPLICATION 标准。我们把遵守以上标准的WEB服务器就叫做J2EE中的WEB容器。

EJB容器：Enterprise java bean 容器。更具有行业领域特色。他提供给运行在其中的组件EJB各种管理功能。只要满足J2EE规范的EJB放入该容器，马上就会被容器进行高效率的管理。并且可以通过现成的接口来获得系统级别的服务。例如邮件服务、事务管理。

## **283、JAVA语言如何进行异常处理，关键字：throws,throw,try,catch,finally分别代表什么意义？在try块中可以抛出异常吗？**

每个异常都是一个对象，它是Throwable类或其它子类的实例。当一个方法出现异常后便抛出一个异常对象，该对象中包含有异常信息，调用这个对象的方法可以捕获到这个异常并进行处理。Java的异常处理是通过5个关键词来实现的：try、catch、throw、throws和finally。

用try来指定一块预防所有“异常”的程序。紧跟在try程序后面，应包含一个catch子句来指定你想要捕捉的“异常”的类型。

throw语句用来明确地抛出一个“异常”。

throws用来标明一个成员函数可能抛出的各种“异常”。

Finally为确保一段代码不管发生什么“异常”都被执行一段代码。

Try是可以嵌套的

## **284、ejb中的sessionbean和entitybean的区别和含义？。**

EJB包括Session Bean、Entity Bean、Message Driven Bean，

SessionBean在J2EE应用程序中被用来完成一些服务器端的业务操作，例如访问数据库、调用其他EJB组件。EntityBean被用来代表应用系统中用到的数据。

对于客户机，SessionBean是一种非持久性对象，它实现某些在服务器上运行的业务逻辑。

对于客户机，EntityBean是一种持久性对象，它代表一个存储在持久性存储器中的实体的对象视图，或是一个由现有企业应用程序实现的实体。

Session Bean用于实现业务逻辑，它可以是有状态的，也可以是无状态的。每当客户端请求时，容器就会选择一个Session Bean来为客户端服务。SessionBean可以直接访问数据库，但更多时候，它会通过Entity Bean实现数据访问。

Entity Bean是域模型对象，用于实现O/R映射，负责将数据库中的表记录映射为内存中的Entity对象，事实上，创建一个Entity Bean对象相当于新建一条记录，删除一个Entity Bean会同时从数据库中删除对应记录，修改一个EntityBean时，容器会自动将Entity Bean的状态和数据库同步

## **sql语句怎么走**

From-on（联合条件）-left join（联合）-while（条件）-group by（分组）-with-having（筛选）-select（查询）-distinct（去重）-order by（排序）-top（选取）

## **285、Java 的通信编程，编程题(或问答)，用JAVA SOCKET编程，读服务器几个字符，再写入本地显示？**

答:Server端程序:

package test;

import java.net.\*;

import java.io.\*;

public class Server {

private ServerSocket ss;

private Socket socket;

private BufferedReader in;

private PrintWriter out;

public Server() {

try {

ss=new ServerSocket(10000);

while(true) {

socket = ss.accept();

String RemoteIP = socket.getInetAddress().getHostAddress();

String RemotePort = ":"+socket.getLocalPort();

System.out.println("A client come in!IP:"+Remo

## **286、Servlet执行时一般实现哪几个方法？**

public void init(ServletConfig config)

public ServletConfig getServletConfig()

public String getServletInfo()

public void service(ServletRequest request,ServletResponse response)

public void destroy()

## **287、BS与CS的联系与区别。**

C/S是Client/Server的缩写。服务器通常采用高性能的PC、工作站或小型机，并采用大型数据库系统，如Oracle、Sybase、Informix或 SQL Server。客户端需要安装专用的客户端软件。

B/Ｓ是Brower/Server的缩写，客户机上只要安装一个浏览器（Browser），如Netscape Navigator或Internet Explorer，服务器安装Oracle、Sybase、Informix或 SQL Server等数据库。在这种结构下，用户界面完全通过WWW浏览器实现，一部分事务逻辑在前端实现，但是主要事务逻辑在服务器端实现。浏览器通过Ｗeb Server 同数据库进行数据交互。

C/S 与 B/S 区别：

１．硬件环境不同:

C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务.

B/S 建立在广域网之上的, 不必是专门的网络硬件环境,例与电话上网, 租用设备. 信息自己管理. 有比C/S更强的适应范围, 一般只要有操作系统和浏览器就行

２．对安全要求不同

C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用C/S 结构适宜. 可以通过B/S发布部分可公开信息.

B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。

３．对程序架构不同

C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑.

B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上. 比C/S有更高的要求 B/S结构的程序架构是发展的趋势, 从MS的.Net系列的BizTalk 2000 Exchange 2000等, 全面支持网络的构件搭建的系统. SUN 和IBM推的JavaBean 构件技术等,使 B/S更加成熟.

４．软件重用不同

C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在B/S要求下的构件的重用性好.

B/S 对的多重结构,要求构件相对独立的功能. 能够相对较好的重用.就入买来的餐桌可以再利用,而不是做在墙上的石头桌子

５．系统维护不同

C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统

B/S 构件组成,方面构件个别的更换,实现系统的无缝升级. 系统维护开销减到最小.用户从网上自己下载安装就可以实现升级.

６．处理问题不同

C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统

B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是C/S无法作到的. 与操作系统平台关系最小.

７．用户接口不同

C/S 多是建立的Window平台上,表现方法有限,对程序员普遍要求较高

B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低,减低开发成本.

８．信息流不同

C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低

B/S 信息流向可变化, B-B B-C B-G等信息、流向的变化, 更像交易中心。

## **288、什么是java序列化，如何实现java序列化？**

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

序列化的实现：将需要被序列化的类实现Serializable接口，该接口没有需要实现的方法，implements Serializable只是为了标注该对象是可被序列化的，然后使用一个输出流(如：FileOutputStream)来构造一个ObjectOutputStream(对象流)对象，接着，使用ObjectOutputStream对象的writeObject(Object obj)方法就可以将参数为obj的对象写出(即保存其状态)，要恢复的话则用输入流。

## **289、请问你在什么情况下会在你的JAVA代码中使用可序列化？**

cluster中session复制,缓存persist与reload