**Java中的易混问题收集  
  
第一，final, finally, finalize的区别.  
  
final 修饰符（关键字）如果一个类被声明为final，意味着它不能再派生出新的子类，不能作为父类被继承。因此一个类不能既被声明为 abstract的，又被声明为final的。将变量或方法声明为final，可以保证它们在使用中不被改变。被声明为final的变量必须在声明时给定初值，而在以后的引用中只能读取，不可修改。被声明为final的方法也同样只能使用，不能重载   
  
finally 再异常处理时提供 finally 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 catch 子句就会执行，然后控制就会进入 finally 块（如果有的话）。   
finalize 方法名。Java 技术允许使用 finalize() 方法在垃圾收集器将对象从内存中清除出去之前做必要的清理工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 Object 类中定义的，因此所有的类都继承了它。子类覆盖 finalize() 方法以整理系统资源或者执行其他清理工作。finalize() 方法是在垃圾收集器删除对象之前对这个对象调用的。   
  
第二，Anonymous Inner Class (匿名内部类) 是否可以extends(继承)其它类，是否可以implements(实现)interface(接口)?   
  
匿名的内部类是没有名字的内部类。不能extends(继承) 其它类，但一个内部类可以作为一个接口，由另一个内部类实现。   
  
第三，Static Nested Class 和 Inner Class的不同，说得越多越好  
Nested Class （一般是C++的说法），Inner Class (一般是JAVA的说法)。Java内部类与C++嵌套类最大的不同就在于是否有指向外部的引用上。具体可见http: //www.frontfree.net/articles/services/view.asp?id=704&page=1   
注： 静态内部类（Inner Class）意味着1创建一个static内部类的对象，不需要一个外部类对象，2不能从一个static内部类的一个对象访问一个外部类对象   
  
第四，&和&&的区别。  
  
&是位运算符。&&是布尔逻辑运算符。   
  
第五，HashMap和Hashtable的区别。  
  
都属于Map接口的类，实现了将惟一键映射到特定的值上。   
HashMap 类没有分类或者排序。它允许一个 null 键和多个 null 值。   
Hashtable 类似于 HashMap，但是不允许 null 键和 null 值。它也比 HashMap 慢，因为它是同步的。   
  
第六，Collection 和 Collections的区别。   
Collections是个java.util下的类，它包含有各种有关集合操作的静态方法。   
Collection是个java.util下的接口，它是各种集合结构的父接口。   
  
第七，什么时候用assert。   
断言是一个包含布尔表达式的语句，在执行这个语句时假定该表达式为 true。如果表达式计算为 false，那么系统会报告一个 Assertionerror。它用于调试目的：   
assert(a > 0); // throws an Assertionerror if a <= 0   
断言可以有两种形式：   
assert Expression1 ;   
assert Expression1 : Expression2 ;   
Expression1 应该总是产生一个布尔值。   
Expression2 可以是得出一个值的任意表达式。这个值用于生成显示更多调试信息的 String 消息。   
断言在默认情况下是禁用的。要在编译时启用断言，需要使用 source 1.4 标记：   
javac -source 1.4 Test.java   
要在运行时启用断言，可使用 -enableassertions 或者 -ea 标记。   
要在运行时选择禁用断言，可使用 -da 或者 -disableassertions 标记。   
要系统类中启用断言，可使用 -esa 或者 -dsa 标记。还可以在包的基础上启用或者禁用断言。   
可 以在预计正常情况下不会到达的任何位置上放置断言。断言可以用于验证传递给私有方法的参数。不过，断言不应该用于验证传递给公有方法的参数，因为不管是否 启用了断言，公有方法都必须检查其参数。不过，既可以在公有方法中，也可以在非公有方法中利用断言测试后置条件。另外，断言不应该以任何方式改变程序的状 态。   
  
第八，GC是什么? 为什么要有GC? (基础)。  
  
GC是垃圾收集器。Java 程序员不用担心内存管理，因为垃圾收集器会自动进行管理。要请求垃圾收集，可以调用下面的方法之一：   
System.gc()   
Runtime.getRuntime().gc()   
  
第九，String s = new String("xyz");创建了几个String Object?   
  
两个对象，一个是“xyx”,一个是指向“xyx”的引用对象s。   
  
第十，Math.round(11.5)等於多少? Math.round(-11.5)等於多少?   
  
Math.round(11.5)返回（long）12，Math.round(-11.5)返回（long）-11;   
  
第十一，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;有错，s1是short型，s1+1是int型,不能显式转化为short型。可修改为s1 =(short)(s1 + 1) 。short s1 = 1; s1 += 1正确。   
  
第十二，sleep() 和 wait() 有什么区别? 搞线程的最爱   
sleep()方法是使线程停止一段时间的方法。在sleep 时间间隔期满后，线程不一定立即恢复执行。这是因为在那个时刻，其它线程可能正在运行而且没有被调度为放弃执行，除非(a)“醒来”的线程具有更高的优先级   
(b)正在运行的线程因为其它原因而阻塞。   
wait()是线程交互时，如果线程对一个同步对象x 发出一个wait()调用，该线程会暂停执行，被调对象进入等待状态，直到被唤醒或等待时间到。   
  
第十三，Java有没有goto?   
Goto java中的保留字，现在没有在java中使用。   
  
第十四，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。   
  
第十五，Overload和Override的区别。Overloaded的方法是否可以改变返回值的类型?   
  
方 法的重写Overriding和重载Overloading是Java多态性的不同表现。重写Overriding是父类与子类之间多态性的一种表现，重 载Overloading是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方 法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded的方法是可以改变返回值的类型。   
  
第十六，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。   
  
第十七，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   
  
第十八，error和exception有什么区别?   
error 表示恢复不是不可能但很困难的情况下的一种严重问题。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问题。也就是说，它表示如果程序运行正常，从不会发生的情况。   
  
第十九，List, Set, Map是否继承自Collection接口?   
List，Set是   
  
Map不是   
  
第二十，abstract class和interface有什么区别?   
  
声 明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。   
接 口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有 程序体。接口只可以定义static final成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。 然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到 接口类型或从接口类型转换，instanceof 运算符可以用来决定某对象的类是否实现了接口。   
  
第二十一，abstract的method是否可同时是static,是否可同时是native，是否可同时是synchronized?   
  
都不能   
  
第二十二，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，但前提是实体类必须有明确的构造函数。   
  
第二十三，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。   
  
第二十四，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。   
  
第二十五，是否可以继承String类?   
String类是final类故不可以继承。   
  
第二十六，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?   
不能，一个对象的一个synchronized方法只能由一个线程访问。   
  
第二十七，try {}里有一个return语句，那么紧跟在这个try后的finally {}里的code会不会被执行，什么时候被执行，在return前还是后?   
  
会执行，在return前执行。   
  
第二十八，编程题: 用最有效率的方法算出2乘以8等於几?   
  
有C背景的程序员特别喜欢问这种问题。   
  
2 << 3   
  
第二十九，两个对象值相同(x.equals(y) == true)，但却可有不同的hash code，这句话对不对?   
不对，有相同的hash code。   
  
第三十，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，并可返回变化后的结果，那么这里到底是值传递还是引用传递?   
  
是值传递。Java 编程语言只由值传递参数。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的方法中改变，但对象的引用是永远不会改变的。   
  
第三十一，swtich是否能作用在byte上，是否能作用在long上，是否能作用在String上?   
  
switch（expr1）中，expr1是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、 short、 char 或者 byte。long,string 都不能作用于swtich。   
  
第三十二，编程题: 写一个Singleton出来。  
  
Singleton模式主要作用是保证在Java应用程序中，一个类Class只有一个实例存在。   
一般Singleton模式通常有几种种形式:   
第一种形式: 定义一个类，它的构造函数为private的，它有一个static的private的该类变量，在类初始化时实例话，通过一个public的getInstance方法获取对它的引用,继而调用其中的方法。   
public class Singleton {   
　　private Singleton(){}   
　　//在自己内部定义自己一个实例，是不是很奇怪？   
　　//注意这是private 只供内部调用   
　　private static Singleton instance = new Singleton();   
　　//这里提供了一个供外部访问本class的静态方法，可以直接访问　　   
　　public static Singleton getInstance() {   
　　　　return instance; 　　   
　　 }   
}   
第二种形式:   
public class Singleton {   
　　private static Singleton instance = null;   
　　public static synchronized Singleton getInstance() {   
　　//这个方法比上面有所改进，不用每次都进行生成对象，只是第一次　　　 　   
　　//使用时生成实例，提高了效率！   
　　if (instance==null)   
　　　　instance＝new Singleton();   
return instance; 　　}   
}   
其他形式:   
定义一个类，它的构造函数为private的，所有方法为static的。   
一般认为第一种形式要更加安全些   
  
第三十三 Hashtable和HashMap   
Hashtable继承自Dictionary类，而HashMap是Java1.2引进的Map interface的一个实现   
  
HashMap允许将null作为一个entry的key或者value，而Hashtable不允许   
  
还有就是，HashMap把Hashtable的contains方法去掉了，改成containsvalue和containsKey。因为contains方法容易让人引起误解。   
  
最大的不同是，Hashtable的方法是Synchronize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。**

1. ArrayList和Vector的区别,HashMap和Hashtable的区别   
   答：就ArrayList与Vector主要从二方面来说.   
   一.同步性:Vector是线程安全的，也就是说是同步的，而ArrayList是线程序不安全的，不是同步的   
   二.数据增长:当需要增长时,Vector默认增长为原来一培，而ArrayList却是原来的一半   
   就HashMap与HashTable主要从三方面来说。   
   一.历史原因:Hashtable是基于陈旧的Dictionary类的，HashMap是Java 1.2引进的Map接口的一个实现   
   二.同步性:Hashtable是线程安全的，也就是说是同步的，而HashMap是线程序不安全的，不是同步的   
   三.值：只有HashMap可以让你将空值作为一个表的条目的key或value
2. char型变量中能不能存贮一个中文汉字?为什么?   
   答：是能够定义成为一个中文的，因为java中以unicode编码，一个char占16个字节，所以放一个中文是没问题的
3. 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
4. 、介绍JAVA中的Collection FrameWork(包括如何写自己的数据结构)?   
   答：Collection FrameWork如下：   
   Collection   
   ├List   
   │├LinkedList   
   │├ArrayList   
   │└Vector   
   │　└Stack   
   └Set   
   Map   
   ├Hashtable   
   ├HashMap   
   └WeakHashMap   
   Collection是最基本的集合接口，一个Collection代表一组Object，即Collection的元素（Elements）   
   Map提供key到value的映射
5. 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:"+RemoteIP+RemotePort);   
   in = new BufferedReader(new   
     
   InputStreamReader(socket.getInputStream()));   
   String line = in.readLine();   
   System.out.println("Cleint send is :" + line);   
   out = new PrintWriter(socket.getOutputStream(),true);   
   out.println("Your Message Received!");   
   out.close();   
   in.close();   
   socket.close();   
   }   
   }catch (IOException e)   
   {   
   out.println("wrong");   
   }   
   }   
   public static void main(String[] args)   
   {   
   new Server();   
   }   
   };   
   Client端程序:   
   package test;   
   import java.io.\*;   
   import java.net.\*;   
     
   public class Client   
   {   
   Socket socket;   
   BufferedReader in;   
   PrintWriter out;   
   public Client()   
   {   
   try   
   {   
   System.out.println("Try to Connect to 127.0.0.1:10000");   
   socket = new Socket("127.0.0.1",10000);   
   System.out.println("The Server Connected!");   
   System.out.println("Please enter some Character:");   
   BufferedReader line = new BufferedReader(new   
     
   InputStreamReader(System.in));   
   out = new PrintWriter(socket.getOutputStream(),true);   
   out.println(line.readLine());   
   in = new BufferedReader(new InputStreamReader(socket.getInputStream()));   
   System.out.println(in.readLine());   
   out.close();   
   in.close();   
   socket.close();   
   }catch(IOException e)   
   {   
   out.println("Wrong");   
   }   
   }   
   public static void main(String[] args)   
   {   
   new Client();   
   }   
   };   
     
   14、用JAVA实现一种排序，JAVA类实现序列化的方法(二种)？ 如在COLLECTION框架中，实现比较要实现什么样的接口？   
   答:用插入法进行排序代码如下   
   package test;   
   import java.util.\*;   
   class InsertSort   
   {   
   ArrayList al;   
   public InsertSort(int num,int mod)   
   {   
   al = new ArrayList(num);   
   Random rand = new Random();   
   System.out.println("The ArrayList Sort Before:");   
   for (int i=0;i<num ;i++ )   
   {   
   al.add(new Integer(Math.abs(rand.nextInt()) % mod + 1));   
   System.out.println("al["+i+"]="+al.get(i));   
   }   
   }   
   public void SortIt()   
   {   
   Integer tempInt;   
   int MaxSize=1;   
   for(int i=1;i<al.size();i++)   
   {   
   tempInt = (Integer)al.remove(i);   
   if(tempInt.intValue()>=((Integer)al.get(MaxSize-1)).intValue())   
   {   
   al.add(MaxSize,tempInt);   
   MaxSize++;   
   System.out.println(al.toString());   
   } else {   
   for (int j=0;j<MaxSize ;j++ )   
   {   
   if   
     
   (((Integer)al.get(j)).intValue()>=tempInt.intValue())   
   {   
   al.add(j,tempInt);   
   MaxSize++;   
   System.out.println(al.toString());   
   break;   
   }   
   }   
   }   
   }   
   System.out.println("The ArrayList Sort After:");   
   for(int i=0;i<al.size();i++)   
   {   
   System.out.println("al["+i+"]="+al.get(i));   
   }   
   }   
   public static void main(String[] args)   
   {   
   InsertSort is = new InsertSort(10,100);   
   is.SortIt();   
   }   
   }   
   JAVA类实现序例化的方法是实现java.io.Serializable接口   
   Collection框架中实现比较要实现Comparable 接口和 Comparator 接口
6. 编程：编写一个截取字符串的函数，输入为一个字符串和字节数，输出为按字节截取的字符串。 但是要保证汉字不被截半个，如“我ABC”4，应该截为“我AB”，输入“我ABC汉DEF”，6，应该输出为“我ABC”而不是“我ABC+汉的半个”。   
   答：代码如下：   
   package test;   
     
   class SplitString   
   {   
   String SplitStr;   
   int SplitByte;   
   public SplitString(String str,int bytes)   
   {   
   SplitStr=str;   
   SplitByte=bytes;   
   System.out.println("The String is:´"+SplitStr+"´;SplitBytes="+SplitByte);   
   }   
   public void SplitIt()   
   {   
   int loopCount;   
     
   loopCount=(SplitStr.length()%SplitByte==0)?(SplitStr.length()/SplitByte):(SplitStr.length()/Split   
     
   Byte+1);   
   System.out.println("Will Split into "+loopCount);   
   for (int i=1;i<=loopCount ;i++ )   
   {   
   if (i==loopCount){   
     
   System.out.println(SplitStr.substring((i-1)\*SplitByte,SplitStr.length()));   
   } else {   
     
   System.out.println(SplitStr.substring((i-1)\*SplitByte,(i\*SplitByte)));   
   }   
   }   
   }   
   public static void main(String[] args)   
   {   
   SplitString ss = new SplitString("test中dd文dsaf中男大3443n中国43中国人   
     
   0ewldfls=103",4);   
   ss.SplitIt();   
   }   
   }

Jsp方面   
  
**1、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）：   
1) pageContext 表示页容器，网页的属性是在这里管理 –>E、 标签、 上传  
2) request 服务器端取得客户端的信息：头信息、Cookie、请求参数、MVC设计模式  
3) response 服务器端回应给客户端信息：Cookie、重定向  
4) session 表示每一个用户，用于登录验证上  
5) application 表示整个服务器，getRealPath()  
6) config 去的初始化参数，初始化参数在web.xml中配置  
7) exception 表示的是错误页的处理操作  
8) page 如同this一样，表示整个JSP页面  
9) out 输出，但是尽量使用表达式输出**  
2、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   
  
3、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" %>   
  
4、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。   
  
Servlet方面   
  
1、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。   
  
2、Servlet版本间(忘了问的是哪两个版本了)的不同?   
希望大家补上，谢谢   
  
3、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。   
  
4、Servlet的基本架构   
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 {   
}   
}   
  
Jdbc、Jdo方面   
  
1、可能会让你写一段Jdbc连Oracle的程序,并实现数据查询.   
答:程序如下：   
package hello.ant;   
import java.sql.\*;   
public class jdbc   
{   
String dbUrl="jdbc:oracle:thin:@127.0.0.1:1521:orcl";   
String theUser="admin";   
String thePw="manager";   
Connection c=null;   
Statement conn;   
ResultSet rs=null;   
public jdbc()   
{   
try{   
Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();   
c = DriverManager.getConnection(dbUrl,theUser,thePw);   
conn=c.createStatement();   
}catch(Exception e){   
e.printStackTrace();   
}   
}   
public boolean executeUpdate(String sql)   
{   
try   
{   
conn.executeUpdate(sql);   
return true;   
}   
catch (SQLException e)   
{   
e.printStackTrace();   
return false;   
}   
}   
public ResultSet executeQuery(String sql)   
{   
rs=null;   
try   
{   
rs=conn.executeQuery(sql);   
}   
catch (SQLException e)   
{   
e.printStackTrace();   
}   
return rs;   
}   
public void close()   
{   
try   
{   
conn.close();   
c.close();   
}   
catch (Exception e)   
{   
e.printStackTrace();   
}   
}   
public static void main(String[] args)   
{   
ResultSet rs;   
jdbc conn = new jdbc();   
rs=conn.executeQuery("select \* from test");   
try{   
while (rs.next())   
{   
System.out.println(rs.getString("id"));   
System.out.println(rs.getString("name"));   
}   
}catch(Exception e)   
{   
e.printStackTrace();   
}   
}   
}   
  
2、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。   
  
3、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化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。   
  
4、在ORACLE大数据量下的分页解决方法。一般用截取ID方法，还有是三层嵌套方法。   
答:一种分页方法   
<%   
int i=1;   
int numPages=14;   
String pages = request.getParameter("page") ;   
int currentPage = 1;   
currentPage=(pages==null)?(1):{Integer.parseInt(pages)}   
sql = "select count(\*) from tables";   
ResultSet rs = DBLink.executeQuery(sql) ;   
while(rs.next()) i = rs.getInt(1) ;   
int intPageCount=1;   
intPageCount=(i%numPages==0)?(i/numPages):(i/numPages+1);   
int nextPage ;   
int upPage;   
nextPage = currentPage+1;   
if (nextPage>=intPageCount) nextPage=intPageCount;   
upPage = currentPage-1;   
if (upPage<=1) upPage=1;   
rs.close();   
sql="select \* from tables";   
rs=DBLink.executeQuery(sql);   
i=0;   
while((i<numPages\*(currentPage-1))&&rs.next()){i++;}   
%>   
//输出内容   
//输出翻页连接   
合计:<%=currentPage%>/<%=intPageCount%><a href="List.jsp?page=1">第一页</a><a   
  
href="List.jsp?page=<%=upPage%>">上一页</a>   
<%   
for(int j=1;j<=intPageCount;j++){   
if(currentPage!=j){   
%>   
<a href="list.jsp?page=<%=j%>">[<%=j%>]</a>   
<%   
}else{   
out.println(j);   
}   
}   
%>   
<a href="List.jsp?page=<%=nextPage%>">下一页</a><a href="List.jsp?page=<%=intPageCount%>">最后页   
  
</a>   
  
Xml方面   
  
1、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)   
  
2、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。   
  
3、用jdom解析xml文件时如何解决中文问题?如何解析?   
答:看如下代码,用编码方式加以解决   
package test;   
import java.io.\*;   
public class DOMTest   
{   
private String inFile = "c:\\people.xml";   
private String outFile = "c:\\people.xml";   
public static void main(String args[])   
{   
new DOMTest();   
}   
public DOMTest()   
{   
try   
{   
javax.xml.parsers.DocumentBuilder builder =   
  
javax.xml.parsers.DocumentBuilderFactory.newInstance().newDocumentBuilder();   
org.w3c.dom.Document doc = builder.newDocument();   
org.w3c.dom.Element root = doc.createElement("老师");   
org.w3c.dom.Element wang = doc.createElement("王");   
org.w3c.dom.Element liu = doc.createElement("刘");   
wang.appendChild(doc.createTextNode("我是王老师"));   
root.appendChild(wang);   
doc.appendChild(root);   
javax.xml.transform.Transformer transformer =   
javax.xml.transform.TransformerFactory.newInstance().newTransformer();   
transformer.setOutputProperty(javax.xml.transform.OutputKeys.ENCODING, "gb2312");   
transformer.setOutputProperty(javax.xml.transform.OutputKeys.INDENT, "yes");   
  
transformer.transform(new javax.xml.transform.dom.DOMSource(doc),   
new   
  
javax.xml.transform.stream.StreamResult(outFile));   
}   
catch (Exception e)   
{   
System.out.println (e.getMessage());   
}   
}   
}   
  
4、编程用JAVA解析XML的方式.   
答:用SAX方式解析XML，XML文件如下：   
<?xml version="1.0" encoding="gb2312"?>   
<person>   
<name>王小明</name>   
<college>信息学院</college>   
<telephone>6258113</telephone>   
<notes>男,1955年生,博士，95年调入海南大学</notes>   
</person>   
事件回调类SAXHandler.java   
import java.io.\*;   
import java.util.Hashtable;   
import org.xml.sax.\*;   
public class SAXHandler extends HandlerBase   
{   
private Hashtable table = new Hashtable();   
private String currentElement = null;   
private String currentValue = null;   
public void setTable(Hashtable table)   
{   
this.table = table;   
}   
public Hashtable getTable()   
{   
return table;   
}   
public void startElement(String tag, AttributeList attrs)   
throws SAXException   
{   
currentElement = tag;   
}   
public void characters(char[] ch, int start, int length)   
throws SAXException   
{   
currentValue = new String(ch, start, length);   
}   
public void endElement(String name) throws SAXException   
{   
if (currentElement.equals(name))   
table.put(currentElement, currentValue);   
}   
}   
JSP内容显示源码,SaxXml.jsp:   
<HTML>   
<HEAD>   
<TITLE>剖析XML文件people.xml</TITLE>   
</HEAD>   
<BODY>   
<%@ page errorPage="ErrPage.jsp"   
contentType="text/html;charset=GB2312" %>   
<%@ page import="java.io.\*" %>   
<%@ page import="java.util.Hashtable" %>   
<%@ page import="org.w3c.dom.\*" %>   
<%@ page import="org.xml.sax.\*" %>   
<%@ page import="javax.xml.parsers.SAXParserFactory" %>   
<%@ page import="javax.xml.parsers.SAXParser" %>   
<%@ page import="SAXHandler" %>   
<%   
File file = new File("c:\\people.xml");   
FileReader reader = new FileReader(file);   
Parser parser;   
SAXParserFactory spf = SAXParserFactory.newInstance();   
SAXParser sp = spf.newSAXParser();   
SAXHandler handler = new SAXHandler();   
sp.parse(new InputSource(reader), handler);   
Hashtable hashTable = handler.getTable();   
out.println("<TABLE BORDER=2><CAPTION>教师信息表</CAPTION>");   
out.println("<TR><TD>姓名</TD>" + "<TD>" +   
(String)hashTable.get(new String("name")) + "</TD></TR>");   
out.println("<TR><TD>学院</TD>" + "<TD>" +   
(String)hashTable.get(new String("college"))+"</TD></TR>");   
out.println("<TR><TD>电话</TD>" + "<TD>" +   
(String)hashTable.get(new String("telephone")) + "</TD></TR>");   
out.println("<TR><TD>备注</TD>" + "<TD>" +   
(String)hashTable.get(new String("notes")) + "</TD></TR>");   
out.println("</TABLE>");   
%>   
</BODY>   
</HTML>   
  
EJB方面   
  
1、EJB2.0有哪些内容?分别用在什么场合? EJB2.0和EJB1.1的区别?   
答： 规范内容包括Bean提供者，应用程序装配者，EJB容器，EJB配置工具，EJB服务提供者，系统管理员。这里面，EJB容器是EJB之所以能够运行的 核心。EJB容器管理着EJB的创建，撤消，激活，去活，与数据库的连接等等重要的核心工作。 JSP,Servlet,EJB,JNDI,JDBC,JMS.....   
  
2、EJB与JAVA BEAN的区别？   
答: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。   
  
3、EJB的基本架构   
答:一个EJB包括三个部分:   
Remote Interface 接口的代码   
package Beans;   
import javax.ejb.EJBObject;   
import java.rmi.RemoteException;   
public interface Add extends EJBObject   
{   
//some method declare   
}   
Home Interface 接口的代码   
package Beans;   
import java.rmi.RemoteException;   
import jaax.ejb.CreateException;   
import javax.ejb.EJBHome;   
public interface AddHome extends EJBHome   
{   
//some method declare   
}   
EJB类的代码   
package Beans;   
import java.rmi.RemoteException;   
import javax.ejb.SessionBean;   
import javx.ejb.SessionContext;   
public class AddBean Implements SessionBean   
{   
//some method declare   
}   
  
J2EE,MVC方面   
  
1、MVC的各个部分都有那些技术来实现?如何实现?   
答:MVC 是Model－View－Controller的简写。"Model" 代表的是应用的业务逻辑（通过JavaBean，EJB组件实现）， "View" 是应用的表示面（由JSP页面产生），"Controller" 是提供应用的处理过程控制（一般是一个Servlet），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。   
  
2、应用服务器与WEB SERVER的区别？   
希望大家补上，谢谢   
  
3、J2EE是什么？   
答:Je22 是Sun公司提出的多层(multi-diered),分布式(distributed),基于组件(component-base)的企业级应用模型 (enterpriese application model).在这样的一个应用系统中，可按照功能划分为不同的组件，这些组件又可在不同计算机上，并且处于相应的层次(tier)中。所属层次包括客户 层(clietn tier)组件,web层和组件,Business层和组件,企业信息系统(EIS)层。   
  
4、WEB SERVICE名词解释。JSWDL开发包的介绍。JAXP、JAXM的解释。SOAP、UDDI,WSDL解释。   
答：Web Service描述语言WSDL   
SOAP即简单对象访问协议(Simple Object Access Protocol)，它是用于交换XML编码信息的轻量级协议。   
UDDI 的目的是为电子商务建立标准；UDDI是一套基于Web的、分布式的、为Web Service提供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的Web Service注册，以使别的企业能够发现的访问协议的实现标准。   
  
5、BS与CS的联系与区别。   
希望大家补上，谢谢   
  
6、STRUTS的应用(如STRUTS架构)   
答：Struts 是采用Java Servlet/JavaServer Pages技术，开发Web应用程序的开放源码的framework。 采用Struts能开发出基于MVC(Model-View-Controller)设计模式的应用构架。 Struts有如下的主要功能：   
一.包含一个controller servlet，能将用户的请求发送到相应的Action对象。   
二.JSP自由tag库，并且在controller servlet中提供关联支持，帮助开发员创建交互式表单应用。   
三.提供了一系列实用对象：XML处理、通过Java reflection APIs自动处理JavaBeans属性、国际化的提示和消息。   
  
设计模式方面   
  
1、开发中都用到了那些设计模式?用在什么场合?   
答： 每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心。通过这种方式，你可以无数次地使用那些已有的解决方案，无需在重 复相同的工作。主要用到了MVC的设计模式。用来开发JSP/Servlet或者J2EE的相关应用。简单工厂模式等。   
  
2、UML方面   
答：标准建模语言UML。用例图,静态图(包括类图、对象图和包图),行为图,交互图(顺序图,合作图),实现图,   
  
JavaScript方面   
  
1、如何校验数字型?   
var re=/^\d{1,8}$|\.\d{1,2}$/;   
var str=document.form1.all(i).value;   
var r=str.match(re);   
if (r==null)   
{   
sign=-4;   
break;   
}   
else{   
document.form1.all(i).value=parseFloat(str);   
}   
  
CORBA方面   
  
1、CORBA是什么?用途是什么?   
答：CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture)，由对象管理组织 (Object Management Group，缩写为 OMG)标准化。它的组成是接口定义语言(IDL), 语言绑定(binding:也译为联编)和允许应用程序间互操作的协议。 其目的为：   
用不同的程序设计语言书写   
在不同的进程中运行   
为不同的操作系统开发   
  
LINUX方面   
  
1、LINUX下线程，GDI类的解释。   
答：LINUX实现的就是基于核心轻量级进程的"一对一"线程模型，一个线程实体对应一个核心轻量级进程，而线程之间的管理在核外函数库中实现。   
GDI类为图像设备编程接口类库。