1、J2EE是什么？

2、介绍JAVA中的Collection FrameWork(包括如何写自己的数据结构)?

如COLLECTION中遗留类(HASHTABLE、VECTOR)和现有类的区别？(同步)

3、Java中异常处理机制，事件机制？

4、EJB与JAVA BEAN的区别？

EJB与JAVA BEAN是SUN的不同组件规范，EJB是在容器中运行的，分步式的，而JAVA BEAN主要是一种可利用的组件，主要在客户端UI表现上。

5、JAVA中的多形与继承？

6、抽象类与接口？

接口的好处是将实现与表现分离。

抽象类与接口都用于抽象，但是抽象类(JAVA中)可以有自己的部分实现，而接口则完全是一个标识(同时有多重继承的功能)。

7、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？

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

forward()方法可以使用原来的对象，而且速度较快。

9、JAVA解析XML的方式？

SAX、DOM

10、用JAVA实现数据库查询，这是一道编程题。

11、用JAVA实现一种排序 ，比较，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。

12、JSP中动态INCLUDE与静态INCLUDE的区别？

老一套，动态可以带参数，静态相当于一段静态HTML文件。

13、应用服务器与WEB SERVER的区别？

14、设计模式与UML方面。

如工厂模式，解决多数据库支持问题。

15、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”。

16、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。

17、WEB SERVICE名词解释。JSWDL开发包的介绍。JAXP、JAXM的解释。SOAP、UDDI解释。

18、BS与CS的联系与区别。

19、LINUX下线程，GDI类的解释。

20、JAVA多线程编程。

用JAVA写一个多线程程序，如写四个线程，二个加1，二个对一个变量减一，输出。

21。JAVA的基本功： STRING与STRINGBUFFER的区别。

22。STRUTS的应用(如STRUTS架构)

23，Collection 和 Collections的区别。   
24，什么时候用assert。   
25，GC是什么? 为什么要有GC?   
26，String s = new String(“xyz”);创建了几个String Object?   
27，Math.round(11.5)等於多少? Math.round(-11.5)等於多少?   
28，short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?   
29，sleep() 和 wait() 有什么区别?   
30，Java有没有goto?   
31，数组有没有length()这个方法? String有没有length()这个方法?   
32，Overload和Override的区别。Overloaded的方法是否可以改变返回值的类型?   
33，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?   
34，给我一个你最常见到的runtime exception。   
35，error和exception有什么区别?   
36，List, Set, Map是否继承自Collection接口?   
37，abstract class和interface有什么区别?   
38，abstract的method是否可同时是static,是否可同时是native，是否可同时是synchronized?   
39，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)?   
40，启动一个线程是用run()还是start()?   
41，构造器Constructor是否可被override?   
42，是否可以继承String类?   
43，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?   
44，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?   
45，编程题: 用最有效率的方法算出2乘以8等於几?   
46，两个对象值相同(x.equals(y) == true)，但却可有不同的hash code，这句话对不对?   
47，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?   
48，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?   
49，编程题: 写一个Singleton出来。

50，JAVA

      MVC表示什么意思？

      JAVA Bean的用途？并在什么情况下用EJB什么情况下不用？

51．J2EE是什么？包含了哪些技术？

52．EJB与JAVA BEAN的区别？

53．面向对象的特征有哪些方面

54简述抽象类与接口的区别

55．说出一些常用的类，包,接口，请各举5个

56谈谈final, finally, finalize的区别

57请问下面的代码输出什么？

父类：

package test;

public class FatherClass

{

  public FatherClass()

{

System.out.println(“FatherClass Create”);

}

}

子类:

package test;

import test.FatherClass;

public class ChildClass extends FatherClass

{

public ChildClass()

{

System.out.println(“ChildClass Create”);

}

public static void main(String[] args)

{

FatherClass fc = new FatherClass();

ChildClass cc = new ChildClass();

}

}

59int 和 Integer 有什么区别是什么？

60.String 和StringBuffer的区别是什么？什么场合用StringBuffer?

61.Static Nested Class 和 Inner Class的不同

62.JAVA SERVLET API中forward() 与redirect()的区别

63.JAVA解析XML的方式有哪些？他们的区别是什么？

64.笔试可能的形式如下：

给你一个:驱动程序A,数据源名称为B,用户名称为C,密码为D,数据库表为T，请用JDBC检索出表T的所有数据。

65.JSP中动态INCLUDE与静态INCLUDE的书写方式以及他们之间的区别

66.说出在JSP页面里是怎么分页的（要说出的方式越多越好）?

67.用socket通讯写出客户端和服务器端的通讯，要求客户发送数据后能够回显相同的数据。

68.&和&&的区别

69.Java中异常处理机制是什么，事件机制是什么？

70.写一个单子模式出来。

71.HashMap和Hashtable的区别

72.Collection 和 Collections的区别

73.说出ArrayList,Vector, LinkedList的存储性能和特性

74.GC是什么? 为什么要有GC?

75.数组有没有length()这个方法? String有没有length()这个方法？他们都是干什么的？

76.Overload和Override的区别。Overloaded的方法是否可以改变返回值的类型?

77.error和exception有什么区别?

78.List, Set, Map是否继承自Collection接口?

79.String是最基本的数据类型吗?

80.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)?

81.启动一个线程是用run()还是start()?

82.父类的构造函数是否可被子类重写?

83.是否可以继承String类?为什么？

84.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?

85.两个对象值相同(x.equals(y) == true)，但却可有不同的hash Code，这句话对不对?

86.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?

87.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?

88.写出程序:设计4个线程，其中两个线程每次对j增加1，另外两个线程对j每次减少1。

89.JSP的内置对象及其作用和主要方法

90.说出Servlet的生命周期，并说出Servlet和CGI的区别。

91.EJB是基于哪些技术实现的?并说出SessionBean和EntityBean的区别，StatefulBean和StatelessBean的区别。

92.说出数据连接池的工作机制是什么?

93.在JSP中可以写哪些内容？

94.如果在SERVLET中要用到数据库连接，那么这个连接写到哪里合适？

95.你知道什么是J2ME吗？和J2SE、J2EE有什么联系？

96.请问下面的代码的输出结果是什么？

package test;

public class OuterClass

{

private class InterClass

{

public InterClass()

{

System.out.println(“InterClass Create”);

}

}

public OuterClass()

{

InterClass ic = new InterClass();

System.out.println(“OuterClass Create”);

}

public static void main(String[] args)

{

OuterClass oc = new OuterClass();

}

}

97.请选择两相正确答案

class OuterClass

{

 private double d1 = 1.0;

//在此处插入选择的代码

}

1. class InnerOne

{

  public static double methoda() {return d1;}

  }

1. public class InnerOne

{

  static double methoda() {return d1;}

  }

1. private class InnerOne

{

  double methoda() {return d1;}

  }

1. static class InnerOne

{

  protected double methoda() {return d1;}

  }

1. abstract class InnerOne

{

  public abstract double methoda();

  }

98.char型变量中能不能存贮一个中文汉字?为什么?

99.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?

100.float型float f=3.4是否正确?

101.jsp有哪些动作?作用分别是什么?

102.请书写一个Servlet并输出HelloWorld字符串

103.Class.forName的作用是什么?

104.你在项目中用到了xml技术的哪些方面?如何实现的?

105.在输出xml文件时如何解决中文问题?

Java基础方面:

106.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.  
107.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  
108.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。  
109.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  
110.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  
111.值：只有HashMap可以让你将空值作为一个表的条目的key或value

112.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

113.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

114.继承时候类的执行顺序问题,一般都是选择题,问你将会打印出什么?  
答:父类：  
package test;  
public class? FatherClass  
{  
 public FatherClass()  
{  
System.out.println(“FatherClass Create”);  
}  
}  
子类:  
package test;  
import test.FatherClass;  
public class? ChildClass extends FatherClass  
{  
public ChildClass()  
{  
System.out.println(“ChildClass Create”);  
}  
public static void main(String[] args)   
{  
FatherClass fc = new FatherClass();  
ChildClass cc = new ChildClass();  
}  
}  
输出结果：  
C:\>java test.ChildClass  
FatherClass Create  
FatherClass Create  
ChildClass Create

114.内部类的实现方式?  
答：示例代码如下：  
package test;  
public class? OuterClass  
{  
private class InterClass  
{  
public InterClass()  
{  
System.out.println(“InterClass Create”);  
}  
}  
public OuterClass()  
{  
InterClass ic = new InterClass();  
System.out.println(“OuterClass Create”);  
}  
public static void main(String[] args)   
{  
OuterClass oc = new OuterClass();  
}  
}  
输出结果:  
C:\>java test/OuterClass  
InterClass Create  
OuterClass Create  
再一个例题：  
public class OuterClass {   
 private double d1 = 1.0;   
 //insert code here   
}   
You need to insert an inner class declaration at line 3. Which two inner class declarations are

valid(Choose two.)   
A. class InnerOne{  
 public static double methoda() {return d1;}  
 }   
B. public class InnerOne{  
 static double methoda() {return d1;}  
 }   
 private class InnerOne{  
 double methoda() {return d1;}  
 }   
D. static class InnerOne{  
 protected double methoda() {return d1;}  
 }   
E. abstract class InnerOne{  
 public abstract double methoda();  
 }   
说明如下：  
一.静态内部类可以有静态成员，而非静态内部类则不能有静态成员。 故 A、B 错  
二.静态内部类的非静态成员可以访问外部类的静态变量，而不可访问外部类的非静态变量；return d1 出错。

故 D 错  
三.非静态内部类的非静态成员可以访问外部类的非静态变量。 故 C 正确  
四.答案为C、E

115、垃圾回收机制,如何优化程序?

116、float型float f=3.4是否正确?  
答:不正确。精度不准确,应该用强制类型转换，如下所示：float f=(float)3.4

117、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

118、Java中异常处理机制，事件机制？

118、JAVA中的多形与继承？

119、抽象类与接口？  
答：抽象类与接口都用于抽象，但是抽象类(JAVA中)可以有自己的部分实现，而接口则完全是一个标识(同时有多重继承的功能)。

120.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();  
}  
};

121、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  
{  
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  
{  
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

122、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();  
}  
}

123、JAVA多线程编程。 用JAVA写一个多线程程序，如写四个线程，二个加1，二个对一个变量减一，输出。   
希望大家补上，谢谢

124、STRING与STRINGBUFFER的区别。   
答：STRING的长度是不可变的，STRINGBUFFER的长度是可变的。如果你对字符串中的内容经常进行操作，特别是内容要修改时，那么使用StringBuffer，如果最后需要String，那么使用StringBuffer的toString()方法

Jsp方面

125、jsp有哪些内置对象?作用分别是什么?  
答:JSP共有以下9种基本内置组件（可与ASP的6种内部组件相对应）：   
　request 用户端请求，此请求会包含来自GET/POST请求的参数   
 response 网页传回用户端的回应   
 pageContext 网页的属性是在这里管理   
 session 与请求有关的会话期   
 application servlet 正在执行的内容   
 out 用来传送回应的输出  
 config servlet的构架部件   
 page JSP网页本身   
 exception 针对错误网页，未捕捉的例外

126、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

127、JSP中动态INCLUDE与静态INCLUDE的区别？   
答：动态INCLUDE用jsp:include动作实现  
 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

128、两种跳转方式分别是什么?有什么区别?  
答：有两种，分别为：  
   
   
 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

129、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。

130、Servlet版本间(忘了问的是哪两个版本了)的不同?  
131、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。

132、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

133、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();  
}  
}  
}

134、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。

135、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。

136、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

//输出内容  
//输出翻页连接  
合计:/[第一页](http://blog.csdn.net/zlsunnan/archive/2004/08/22/List.jsp?page=1)<>

href=”List.jsp?page=”>上一页

[[]](http://blog.csdn.net/zlsunnan/archive/2004/08/22/list.jsp?page=%3c%25=j%25%3e)

[下一页](http://blog.csdn.net/zlsunnan/archive/2004/08/22/List.jsp?page=%3c%25=nextPage%25%3e)[最后页](http://blog.csdn.net/zlsunnan/archive/2004/08/22/List.jsp?page=%3c%25=intPageCount%25%3e)

Xml方面

137、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)

138、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。

139、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());  
}  
 }  
}

140、编程用JAVA解析XML的方式.  
答:用SAX方式解析XML，XML文件如下：

141.每家公司无一例外的都要问到如下问题：“你曾经做过哪些项目？”

142.在用友公司的问题是“如果给你一个方法（函数），你来测试它的各种性能，你该如何做？大概内容是：String ArrangeNumber(String str) str是一个只包含数字的字符串，然后按从小到大进行排列再输出，这是黑箱测试”

143.在用友公司的第二个问题是“你认为DOT NET是什么？同时你又是如何理解的？”

144.在东方环讯的问题比较杂，从JAVA基础到J2EE、J2ME、JSP、SERVLET、数据库全都问到了，这些问题不是很难，课本上和课堂上都提到过，只要平时理解了，就应该没问题。记得问到了如下问题：

a) 在JAVA中的可视范围有哪些？（要把每个访问修饰符的作用讲清楚即可：PUBLIC、PRIVATE、PROTECTED、（DEFAULT））

b) SERVLET的周期中包括哪几个部分？（void init()、void service(…)、void destroy()）

c) JSP中的内置对象有哪些？（request、response、out、application、session、pageContext、page、config、exception）

d) 在JSP中可以写哪些内容？（脚本语言、指令元素、动作元素、注释、声明、代码段、表达式等等）

e) 如果在SERVLET中要用到数据库连接，那么这个连接写到哪里合适？（在void init()方法里）

f) INCLUDE指令和标记有区别吗？（前者包括源码，后者包括生成结果）

g) 如何理解数据库中的触发器？（好比是一个事件处理程序。）

h) 在数据库中写存储过程的意义是什么？（运行时速度快）

i) 你是如何理解JAVA中的对象机制？（自己发挥吧，两家公司问到这个问题）

j) 你知道什么是J2ME吗？和J2SE、J2EE有什么联系？（分别是JAVA三个版本的缩写：即JAVA2 MICRO EDITION(J2ME)、       JAVA2 STANDARD EDITION(J2SE)、                      JAVA2 ENTERPRISE EDITION(J2EE)）

146.说明接口与抽象类的区别

147.在Thread中wait()与sleep()有什么区别

148.列出jsp中的内置对象及其主要用途

149.Session的用途及其工作原理

150.jsp与servelt的关系

151.是否使用过struts，说明struts的结构

152.为三道UML的关系图

153.说明Exception与Error的区别

154.列出你所知道的MVC结构

155.写出你所作的struts的简单事例，不用写代码，但要说明结构

JAVA方面

156. 面向对象的特征有哪些方面

157.String是最基本的数据类型吗?

158.int 和 Integer 有什么区别

159.String 和StringBuffer的区别

160.运行时异常与一般异常有何异同？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

161. 说出一些常用的类，包,接口，请各举5个

162.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，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector由于使用了synchronized方法（线程安全），通常性能上较ArrayList差，而LinkedList使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

163.设计4个线程，其中两个线程每次对j增加1，另外两个线程对j每次减少164.写出程序。

以下程序使用内部类实现线程，对j增减的时候没有考虑顺序问题。

public class ThreadTest1{

private int j;

public static void main(String args[]){

ThreadTest1 tt=new ThreadTest1();

Inc inc=tt.new Inc();

Dec dec=tt.new Dec();

for(int i=0;i<2;i++){

Thread t=new Thread(inc);

t.start();

t=new Thread(dec);

t.start();

}

}

private synchronized void inc(){

j++;

System.out.println(Thread.currentThread().getName()+”-inc:”+j);

}

private synchronized void dec(){

j–;

System.out.println(Thread.currentThread().getName()+”-dec:”+j);

}

class Inc implements Runnable{

public void run(){

for(int i=0;i<100;i++){

inc();

}

}

}

class Dec implements Runnable{

public void run(){

for(int i=0;i<100;i++){

dec();

}

}

}

}

165.JSP的内置对象及方法。

request request表示HttpServletRequest对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取cookie, header, 和session数据的有用的方法。

response response表示HttpServletResponse对象，并提供了几个用于设置送回 浏览器的响应的方法（如cookies,头信息等）

out out 对象是javax.jsp.JspWriter的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

pageContext pageContext表示一个javax.servlet.jsp.PageContext对象。它是用于方便存取各种范围的名字空间、servlet相关的对象的API，并且包装了通用的servlet相关功能的方法。

session session表示一个请求的javax.servlet.http.HttpSession对象。Session可以存贮用户的状态信息

application applicaton 表示一个javax.servle.ServletContext对象。这有助于查找有关servlet引擎和servlet环境的信息

config config表示一个javax.servlet.ServletConfig对象。该对象用于存取servlet实例的初始化参数。

page page表示从该页面产生的一个servlet实例

166.用socket通讯写出客户端和服务器端的通讯，要求客户发送数据后能够回显相同的数据。

参见课程中socket通讯例子。

167.说出Servlet的生命周期，并说出Servlet和CGI的区别。

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

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

168.EJB是基于哪些技术实现的?并说出SessionBean和EntityBean的区别，StatefulBean和StatelessBean的区别。

169.EJB包括（SessionBean,EntityBean）说出他们的生命周期，及如何管理事务的？

170.说出数据连接池的工作机制是什么?

171.同步和异步有和异同，在什么情况下分别使用他们？举例说明。

172.应用服务器有那些？

173.你所知道的集合类都有哪些？主要方法？

174.给你一个:驱动程序A,数据源名称为B,用户名称为C,密码为D,数据库表为T，请用JDBC检索出表T的所有数据。

175.说出在JSP页面里是怎么分页的?

页面需要保存以下参数：

总行数：根据sql语句得到总行数

每页显示行数：设定值

当前页数：请求参数

页面根据当前页数和每页行数计算出当前页第一行行数，定位结果集到此行，对结果集取出每页显示行数的行即可。

|  |
| --- |
| 176.谈谈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的此方法，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。  　   177.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)?  可以继承其他类或完成其他接口，在swing编程中常用此方式。 　　178.Static Nested Class 和 Inner Class的不同，说得越多越好(面试题有的很笼统)。  Static Nested Class是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。 　　179.&和&&的区别。  & 是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）. 　    180.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允许空（null）键值（key）,由于非线程安全，效率上可能高于Hashtable.  　　181.Collection 和 Collections的区别。  　　Collection是集合类的上级接口，继承与他的接口主要有Set 和List.  Collections是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。  　　182.什么时候用assert。  1.4新增关键字（语法），用于测试boolean表达式状态，可用于调试程序。  使用方法 assert <boolean表达式>，表示如果表达式为真（true）,则下面的语句执行，否则抛出AssertionError。  另外的使用方式assert < boolean表达式>:<other表达式>，表示如果表达式为真，后面的表达式忽略，否则后面表达式的值用于AssertionError的构建参数。  注意编译时要增加-source 1.4 参数，否则报错。]运行时要增加 –ea参数，否则assert行被忽略  　　183.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。  　　184.String s = new String(“xyz”);创建了几个String Object?  两个 　　185.Math.round(11.5)等於多少? Math.round(-11.5)等於多少?  Math.round(11.5)==12  Math.round(-11.5)==-11  round方法返回与参数最接近的长整数，参数加1/2后求其floor. 　　186.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） 　　187.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时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep不会释放对象锁。  wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。  　　188.Java有没有goto?  没有 　　很十三的问题，如果哪个面试的问到这个问题，我劝你还是别进这家公司。（完）（job.sohu.com） |
|  |

189.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。

190.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。

191.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。

192.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

193.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。

194.List, Set, Map是否继承自Collection接口?  
List，Set是

Map不是

195.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。

196.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

197.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。

198.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。

199.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。

200.是否可以继承String类?  
String类是final类故不可以继承。

201.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。

202.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。

203.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

204.两个对象值相同(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。

205.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。

206.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。

207.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

JAVA培训资料

**208.Java语言**

   1、面向对象的三个基本特征

   2、方法重载和方法重写的概念和区别

   3、接口和内部类、抽象类的特性

   4、文件读写的基本类

   \*\*5、串行化的注意事项以及如何实现串行化

   6、线程的基本概念、线程的基本状态以及状态之间的关系

   7、线程的同步、如何实现线程的同步

   8、几种常用的数据结构及内部实现原理。

   9、Socket通信(TCP、UDP区别及Java实现方式)

  \*\*10、Java的事件委托机制和垃圾回收机制

  11、JDBC调用数据库的基本步骤

  \*\*12、解析XML文件的几种方式和区别

  13、Java四种基本权限的定义

  14、Java的国际化

**209、JSP**

   1、至少要能说出7个隐含对象以及他们的区别

  \*\* 2、forward 和redirect的区别

   3、JSP的常用指令

**210、servlet**

   1、什么情况下调用doGet()和doPost()？

   2、servlet的init()方法和service()方法的区别

   3、servlet的生命周期

   4、如何现实servlet的单线程模式

   5、servlet的配置

   6、四种会话跟踪技术

**211、EJB**

   \*\*1、EJB容器提供的服务

主要提供声明周期管理、代码产生、持续性管理、安全、事务管理、锁和并发行管理等服务。

   2、EJB的角色和三个对象

EJB角色主要包括Bean开发者 应用组装者 部署者 系统管理员 EJB容器提供者 EJB服务器提供者

三个对象是Remote（Local）接口、Home（LocalHome）接口，Bean类

   2、EJB的几种类型

会话（Session）Bean ，实体（Entity）Bean 消息驱动的（Message Driven）Bean

会话Bean又可分为有状态（Stateful）和无状态（Stateless）两种

实体Bean可分为Bean管理的持续性（BMP）和容器管理的持续性（CMP）两种

   3、bean 实例的生命周期

对于Stateless Session Bean、Entity Bean、Message Driven Bean一般存在缓冲池管理，而对于Entity Bean和Statefull Session Bean存在Cache管理，通常包含创建实例，设置上下文、创建EJB Object（create）、业务方法调用、remove等过程，对于存在缓冲池管理的Bean，在create之后实例并不从内存清除，而是采用缓冲池调度机制不断重用实例，而对于存在Cache管理的Bean则通过激活和去激活机制保持Bean的状态并限制内存中实例数量。

   4、激活机制

以Statefull Session Bean 为例：其Cache大小决定了内存中可以同时存在的Bean实例的数量，根据MRU或NRU算法，实例在激活和去激活状态之间迁移，激活机制是当客户端调用某个EJB实例业务方法时，如果对应EJB Object发现自己没有绑定对应的Bean实例则从其去激活Bean存储中（通过序列化机制存储实例）回复（激活）此实例。状态变迁前会调用对应的ejbActive和ejbPassivate方法。

   5、remote接口和home接口主要作用

remote接口定义了业务方法，用于EJB客户端调用业务方法

home接口是EJB工厂用于创建和移除查找EJB实例

   6、客服端调用EJB对象的几个基本步骤

一、 设置JNDI服务工厂以及JNDI服务地址系统属性

二、 查找Home接口

三、 从Home接口调用Create方法创建Remote接口

四、 通过Remote接口调用其业务方法

**212、数据库**

   1、存储过程的编写

   2、基本的SQL语句

**213、weblogic**

1、 如何给weblogic指定大小的内存?

在启动Weblogic的脚本中（位于所在Domian对应服务器目录下的startServerName），增加set MEM\_ARGS=-Xms32m -Xmx200m，可以调整最小内存为32M，最大200M

2、 如何设定的weblogic的热启动模式(开发模式)与产品发布模式?

可以在管理控制台中修改对应服务器的启动模式为开发或产品模式之一。或者修改服务的启动文件或者commenv文件，增加set PRODUCTION\_MODE=true。

3、 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.

4、 在weblogic管理制台中对一个应用域(或者说是一个网站,Domain)进行jms及ejb或连接池等相关信息进行配置后,实际保存在什么文件中?

保存在此Domain的config.xml文件中，它是服务器的核心配置文件。

5、 说说weblogic中一个Domain的缺省目录结构?比如要将一个简单的helloWorld.jsp放入何目录下,然的在浏览器上就可打入http://主机:端口号//helloword.jsp就可以看到运行结果了? 又比如这其中用到了一个自己写的javaBean该如何办?

Domain目录\服务器目录\applications，将应用目录放在此目录下将可以作为应用访问，如果是Web应用，应用目录需要满足Web应用目录要求，jsp文件可以直接放在应用目录中，Javabean需要放在应用目录的WEB-INF目录的classes目录中，设置服务器的缺省应用将可以实现在浏览器上无需输入应用名。

6、 如何查看在weblogic中已经发布的EJB?

可以使用管理控制台，在它的Deployment中可以查看所有已发布的EJB

7、 如何在weblogic中进行ssl配置与客户端的认证配置或说说j2ee(标准)进行ssl的配置

缺省安装中使用DemoIdentity.jks和DemoTrust.jks  KeyStore实现SSL，需要配置服务器使用Enable SSL，配置其端口，在产品模式下需要从CA获取私有密钥和数字证书，创建identity和trust keystore，装载获得的密钥和数字证书。可以配置此SSL连接是单向还是双向的。

   8、在weblogic中发布ejb需涉及到哪些配置文件

不同类型的EJB涉及的配置文件不同，都涉及到的配置文件包括ejb-jar.xml,weblogic-ejb-jar.xmlCMP实体Bean一般还需要weblogic-cmp-rdbms-jar.xml

   9、EJB需直接实现它的业务接口或Home接口吗,请简述理由.

远程接口和Home接口不需要直接实现，他们的实现代码是由服务器产生的，程序运行中对应实现类会作为对应接口类型的实例被使用。

  10、说说在weblogic中开发消息Bean时的persistent与non-persisten的差别

persistent方式的MDB可以保证消息传递的可靠性,也就是如果EJB容器出现问题而JMS服务器依然会将消息在此MDB可用的时候发送过来，而non－persistent方式的消息将被丢弃。

  11、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

215.Weblogic应用和部署？它和TomCat有什么不同？怎样把在TomCat中的Web应用程序部署到WebLogic中？

216.叙述一下MVC中的业务逻辑层的开发过程？

217.为什么要把数据库的连接打开/关闭放在中间控制层Servlet中？

218.你开发的javaBean怎么样View层交互？以及你是怎么样传递信息的？是使用什么传递信息？传递的是什么东西？

219.对MVC模式的理解与你在应用中的主要职责

220.怎样把你的项目移到 weblogic服务器下

221.你的项目MVC实现说明

222.怎样使用数据库连接Bean 怎样使用业务逻辑bean 他们之间关系

223.项目为什么没有使用struts

224.weblogic服务器的使用方法

225.配制文件是怎么同步的？