Sun google 微软 东软 百度 思科 金山 等公司面试题

一、Java 基础知识
1.Java 有那些基本数据类型，String 是不是基本数据类型，他们有何区别。
2.字符串的操作：
写一个方法，实现字符串的反转，如：输入 abc，输出 cba
写一个方法，实现字符串的替换，如：输入 bbbwlirbbb，输出 bbbhhtccc。
3.数据类型之间的转换
如何将数值型字符转换为数字（Integer，Double）
如何将数字转换为字符
如何取小数点前两位，并四舍五入。
4.日期和时间
如何取得年月日，小时分秒
如何取得从 1970 年到现在的毫秒数
如何获取某个日期是当月的最后一天
如何格式化日期
5.数组和集合
6.文件和目录（I/O）操作
如何列出某个目录下的所有文件
如何列出某个目录下的所有子目录
判断一个文件或目录是否存在
如何读写文件
7.Java 多态的实现（继承、重载、覆盖）
8.编码转换，怎样实现将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串。
9.Java 中访问数据库的步骤，Statement 和 PreparedStatement 之间的区别。
10.找出下列代码可能存在的错误，并说明原因：

二、JSP&Servlet 技术
1.描述 JSP 和 Servlet 的区别、共同点、各自应用的范围
2.在 Web 开发中需要处理 HTML 标记时，应做什么样的处理，要筛选那些字符（< > & “”）
3.在 JSP 中如何读取客户端的请求，如何访问 CGI 变量，如何确定某个 Jsp 文件的真实路径。
4.描述 Cookie 和 Session 的作用，区别和各自的应用范围，Session 工作原理。
5.列出 Jsp 中包含外部文件的方式，两者有何区别。
6.说明 Jsp 中 errorPage 的作用，应用范围。
7.介绍在 Jsp 中如何使用 JavaBeans。
8.简单介绍 JSP 的标记库
9.Jsp 和 Servlet 中的请求转发分别如何实现。

三、J2EE 相关知识
1.介绍 J2EE、J2SE、J2SE 的区别。
2.J2EE 是一种技术还是一种平台，他提供了那些技术。
3.什么是 Application Server，它有什么功能和优点。
4.简单介绍连接池的优点和原理。
5.Web.xml 的作用

四、其他
1.Web 安全性的考虑(表单验证、浏览器 Basic 方式的验证，应用程序的安全性，SSL，代码考虑)
2.简单介绍您所了解的 MVC。
3.简单介绍所了解的 XML。
4.文档和编码规范
5.Java 中的分页、效率考虑。
6.简单介绍您所了解的 structs。

1.xml 在项目中的作用
2.s-EJB 与 e-EJB 的区别
3.会话面的作用
4.cmp 与 bmp 的优缺点
5.j2me 程序的必需的几个部分
6.c/s 与 b/s 的区别
7.构建一个 connect pool,然后再调用它，
8.j2ee 平台与 dotnet 平台的区别

9.ejb 的 life cycle
10.session bean 和 entity bean 的区别
11.ejb 中的 transaction 机制
12.synchronized (生产者和消费)
13.String 和 StringBuffer
14.Serializable
15.MVC （Struts 的工作流程)
16.什么是 MDA

17.tcp 与 udp 的区别
18.链表与散列表和数组的区别
19.堆和栈的区别
20.ejb 的分类及区别
21.你对现在软件业以及国内软件业的看法
22.谈谈 java 多线程
23.谈谈文件加密技术
24.软件开发生命周期
25.路由协议种类及特点

26.java 的 awt 和 swing 组件的 GUI 设计的关键
27.对于 java 流的认识

28.简单描述一下 awt 与 swing 区别。
29.简述 java 编程中事件处理模式。
30.你编写过 applet 吗？applet 的安全权限如何？试列举 java application 或者 applet 中与 servlet/jsp 通信可以采用的方式。

31.简述逻辑操作(如&,|)与条件操作(如&&,||)的区别。
32.简述 Java Server Page 和 Servlet 的联系和区别。
33.简述 synchronized 和 java.util.concurrent.locks.Lock 的异同 ？
34.EJB 规范规定 EJB 中禁止的操作有哪些？
35.java 除了 8 种基本类型外，在虚拟机里还有哪一种，有什么作用?
36.除了使用 new 关键字创建对象意外，试列举另外三种以上创建实例的方式?
37.classloader 中，JDK 的 API、Classpath 中的同 web-inf 中的 class 加载方式有什么区别？
38.列举三种以上垃圾回收算法，并比较其优缺点？
39.编写代码实现一个线程池

40.描述一下 JVM 加载 class 文件的原理机制?
41.试举例说明一个典型的垃圾回收算法？
42.请用 java 写二叉树算法，实现添加数据形成二叉树功能，并以先序的方式打印出来.
43.请写一个 java 程序实现线程连接池功能？
44.给定一个 C 语言函数，要求实现在 java 类中进行调用。

45.如何获得数组的长度？
46.访问修饰符“public/private/protected/缺省的修饰符”的使用
47.用关键字 final 修饰一个类或者方法时，有何意义？
48.掌握类和对象的概念，掌握面向对象编程的本质
49.静态变量和静态方法的意义，如何引用一个类的静态变量或者静态方法？
50.JAVA 语言如何进行异常处理，关键字：thorws,throw,try,catch,finally
51.Object 类(或者其子类)的 finalize()方法在什么情况下被调用？
52.一个“.java”原文件中是否可以包括多个类（不是内部类）？
53.掌握内部类和接口的概念
54.StringTokenizer 类的使用
55.数据结构，如何遍历 List 中的元素？
    如果要按照键值保存或者访问数据，使用什么数据结构？
    要掌握 Collection 相关的接口和类的使用
56.使用 StringBuffer 类与 String 类进行字符串连接时有何区别？
57.调用 Thread 类的 destroy()方法有什么后果？
58.多线程，用什么关键字修饰同步方法？stop()和 suspend()方法为何不推荐使用？
59.使用 socket 建立客户端与服务器的通信的过程
60.JAVA 语言国际化应用，Locale 类，Unicode

61.描述反射机制的作用
62.如何读写一个文件？
63.在图形界面中，一个按钮如何处理鼠标点击事件？
64.在图形界面中，一个表格，如何实现编辑单元格时弹出下拉框？
65.如何加载图片？
66.什么是模态对话框？
67.阐述 MVC 的概念
68.GUI 布局管理器的使用，FlowLayout,BorderLayout,GridBagLayout
69.如何构造一棵树？选择树的一个节点时，如何得到这个节点？
70.向编辑框中输入字符时，如何控制只输入整数？
71.描述使用 JDBC 连接数据库的过程
72.EJB 分为几类？什么是 BMP,CMP?
73.什么是 JNDI?

74.ADO 是什么?ActiveX 数据对象,是一个应用级程序接口.
75.四种 JDBC 方式?目前的版本?
76.EJB 有哪几种?区别是什么?
77.JavaBean 与 EJB 有什么区别?
78.软件开发生命周期有哪几个阶段?
79.软件开发有哪些因素?
80.软件开发中如何进行版本控制?
81.UML 中,类视图如何表示类中的继承与聚合?
82.客户端游标与服务器端游标的区别?
83.动态游标与静态游标的区别?
84.dotnet 由哪几个基本框架组成?
85.Oracle 中 SGA 是什么?
86.web servers 是什么?
87.UNIX 中 QT 是什么意思?
88.在软件开发生命周期中的哪个阶段开始测试?
89.dotnet 与 J2EE 的比较?
90.什么是 ActiveX?
91.Java 中 IDL 是什么?
92.ISO9000 和 CMM 是什么?ISO9000 和 CMM(软件能力成熟度模型)认证是国际上通用的软件质量评估方法.CMM 的五个成熟度等级。

第一，谈谈 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(面试题有的很笼统)。
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

---- **Java 面试题和答案** (http://www.bioon.net/dispbbs.asp?boardid=169&id=108010)

-- 作者：jiajia1983
-- 发布时间：2005-3-9 17:29:00

-- Java 面试题和答案
JAVA 相关基础知识
1、面向对象的特征有哪些方面
1.抽象：
抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。
2.继承：
继 承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继 承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增 加新的方法使之更适合特殊的需要。
3.封装：
封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。
4. 多态性：
多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。
2、String 是最基本的数据类型吗?
基本数据类型包括 byte、int、char、long、float、double、boolean 和 short。
java.lang.String 类是 final 类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用 StringBuffer 类
3、int 和 Integer 有什么区别
Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int 是 java 的原始数据类型，Integer 是 java 为 int 提供的封装类。Java 为每个原始类型提供了封装类。
原始类型 封装类
boolean Boolean
char Character
byte Byte
short Short
int Integer

long Long
float Float
double Double
引 用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数 据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。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 和 StringBuffer 的区别
JAVA 平台提供了两个类：String 和 StringBuffer， 它们可以储存和操作字符串，即包含多个字符的字符数据。这个 String 类提供了数值不可改变的字符串。而这个 StringBuffer 类提供的字符串进 行修改。当你知道字符数据要改变的时候你就可以使用 StringBuffer。典型地，你可以使用 StringBuffers 来动态构造字符数据。
5、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。
6、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。
7、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。
8、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。
EJB 包括 Session Bean、Entity Bean、Message Driven Bean，基于 JNDI、RMI、JAT 等技术实现。
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 还 可以再细分

为 Stateful Session Bean 与 Stateless Session Bean ，这两种 的 Session Bean 都可以将系统逻辑放在 method 之中执行，不同的是 Stateful Session Bean 可以 记录呼叫者的状态，因此通常来说，一个使用者会有一个相对应的 Stateful Session Bean 的实体。

Stateless Session Bean 虽然也是逻辑组件，但是他却不负责记录使用者状态，也就是说当使用者呼 叫 Stateless Session Bean 的时候，EJB Container 并不会找寻特定的 Stateless Session Bean 的实体来执行这个 method。换言之，很可能数个使用者在执行某个 Stateless Session Bean 的 methods 时，会是同一个 Bean 的 Instance 在执 行。从内存方面来看， Stateful Session Bean 与 Stateless Session Bean 比 较， Stateful Session Bean 会消耗 J2EE Server 较多的内存，然而 Stateful Session Bean 的优势却在于他可以维持使用者的状态。

9、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。

10、&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）。

11、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。

HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。

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。

Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。

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。

12、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。

13、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。

14、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。

15、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。

16、同步和异步有何异同，在什么情况下分别使用他们？举例说明。

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

当应用程序在对象上调用了一个需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

17、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。

18、heap 和 stack 有什么区别。
栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。
堆是栈的一个组成元素
19、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。
20、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。
21、Static Nested Class 和 Inner Class 的不同。
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。
22、JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？
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" %>
23、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。
24、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。

25、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）

26、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.

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

两个

28、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。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();
}
  }
   }
}
```

29、Java 有没有 goto?

java 中的保留字，现在没有在 java 中使用。

30、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。

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

SessionBean：Stateless Session Bean 的 生命周期是由容器决定的，当客户机发出请求要建立一个 Bean 的实例时，EJB 容器不一定要创建一个新的 Bean 的实例供客户机调用，而是随便找一个现有 的实例提供给客户机。当客户机第一次调用一个 Stateful Session Bean 时，容器必须立即在服务器中创建一个新的 Bean 实例， 并关联到客户机上，以后此客户机调用 Stateful Session Bean 的方法时容器会把调用分派到与此客户机相关联的 Bean 实例。

EntityBean：Entity Beans 能存活相对较长的时间，并且状态是持续的。只要数据库中的数据存在，Entity beans 就一直存活。而不是按照应用程序或者服务进程来说的。即使 EJB 容器崩溃了，Entity beans 也是存活的。 Entity Beans 生命周期能够被容器或者 Beans 自己管理。

EJB 通过以下技术管理实务：对象管理组织（OMG）的对象实务服务（OTS），Sun Microsystems 的 Transaction Service（JTS）、 Java Transaction API（JTA），开发组（X/Open）的 XA 接口。

32、应用服务器有那些?

BEA WebLogic Server，IBM WebSphere Application Server，Oracle9i Application Server，jBoss，Tomcat

33、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, RasterFormatExceptio

n, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException
34、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。
35、List, Set, Map 是否继承自 Collection 接口?
List，Set 是，Map 不是

Java 常见面试题集- -

Java 基础方面:
1、作用域 public,private,protected,以及不写时的区别
答：区别如下：
作用域 当前类 同一 package 子孙类 其他 package
public √ √ √ √
protected √ √ √ ×
friendly √ √ × ×
private √ × × ×
不写时默认为 friendly

2、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

3、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

4、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

5、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
```

6、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;}
}
C. 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

7、垃圾回收机制,如何优化程序?
希望大家补上，谢谢

8、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

9、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

10、Java 中异常处理机制，事件机制?

11、JAVA 中的多形与继承?
希望大家补上，谢谢

12、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)。

13、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

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"。
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()/SplitBy
te):(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();
}
}
```

16、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

17、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

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）：
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

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().newDocumentBui
lder();
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。

==========

一些著名的大公司面试题目往往很基础.

一、Java 基础知识
1. Java 有那些基本数据类型，String 是不是基本数据类型，他们有何区别。
  Integer literals,Floating-point literals,character literals,Boolean literal,String iteral.
  String 不是基本数据类型
2. 字符串的操作：
写一个方法，实现字符串的反转，如：输入 abc，输出 cba

```
        public static String reverse(String s){
                int length=s.length();
                StringBuffer result=new StringBuffer(length);
                for(int i=length-1;i>=0;i--)
                        result.append(s.charAt(i));
                return result.toString();
        }
```

写一个方法，实现字符串的替换，如：输入 bbbwlirbbb，输出 bbbhhtccc。

3. 数据类型之间的转换
如何将数值型字符转换为数字（Integer，Double）
如何将数字转换为字符
如何去小数点前两位，并四舍五入。
4. 日期和时间
如何取得年月日，小时分秒
Date dat=new Date();
dat.getYear();dat.getMonth();dat.getDay();dat.getHours();...
如何取得从 1970 年到现在的毫秒数
long now=dat.getTime();
如何获取某个日期是当月的最后一天
如何格式化日期
DateFormate df=DateFormate.getInstance();
df.Format(dat);
5. 数组和集合
6. 文件和目录（I/O）操作
如何列出某个目录下的所有文件
如何列出某个目录下的所有子目录
判断一个文件或目录是否存在
如何读写文件
7. Java 多态的实现（继承、重载、覆盖）
8. 编码转换，怎样实现将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串。
9. Java 中访问数据库的步骤，Statement 和 PreparedStatement 之间的区别。
10. 找出下列代码可能存在的错误，并说明原因：
二、JSP&Servlet 技术
1. 描述 JSP 和 Servlet 的区别、共同点、各自应用的范围
2. 在 Web 开发中需要处理 HTML 标记时,应做什么样的处理,要筛选那些字符(< > & ″″)
3. 在 JSP 中如何读取客户端的请求，如何访问 CGI 变量，如何确定某个 Jsp 文件的真实路径。
4. 描述 Cookie 和 Session 的作用,区别和各自的应用范围,Session 工作原理。
5. 列出 Jsp 中包含外部文件的方式，两者有何区别。
6. 说明 Jsp 中 errorPage 的作用，应用范围。
7. 介绍在 Jsp 中如何使用 JavaBeans。
8. 简单介绍 JSP 的标记库
9. Jsp 和 Servlet 中的请求转发分别如何实现。
三、J2EE 相关知识
1. 介绍 J2EE、J2SE、J2SE 的区别。
2. J2EE 是一种技术还是一种平台，他提供了那些技术。
3. 什么是 Application Server，它有什么功能和优点。
4. 简单介绍连接池的优点和原理。
5. Web.xml 的作用
四、其他

1. Web 安全性的考虑（表单验证、浏览器 Basic 方式的验证，应用程序的安全性，SSL，代码考虑）
2. 简单介绍您所了解的 MVC。
3. 简单介绍所了解的 XML。
4. 文档和编码规范
5. Java 中的分页、效率考虑。
6. 简单介绍您所了解的 structs。

找出以下程序错误。

```
Class Test{
        private String par1;
        private String par2;
        Test(){
        }

        public static void main(String[] arg){
                int a ;
                if(a){
                        System.out.println("par1="+par1);
                }else{
                        System.out.println("par2=" + par2);
                }
        }
}
```

==========

```
public class Test
{
public static int a = 5;
public static void main(String[] args)
{
Test test = new Test();
test = null;
System.out.println(test.a);
}
}
```

答案是 5，a 为静态变量；不依赖对象，即使为空。

J2EE 初学者需要理解的问题

原文出处： http://www.scjp.com.cn/news/article_show.asp?id=19617

一、J2EE 提出的背景

1、 企业级应用框架的需求

在许多企业级应用中，例如数据库连接、邮件服务、事务处理等都是一些通用企业需求模块，这些模块如果每次再开发中都由开发人员来完成的话，将会造成开发周期长和代码可*性差等问题。于是许多大公司开发了自己的通用模块服务。这些服务性的软件系列统称为中间件。

2、 为了通用必须要提出规范，不然无法达到通用

在上面的需求基础之上，许多公司都开发了自己的中间件，但其与用户的沟通都各有不同，从而导致用户无法将各个公司不同的中间件组装在一块为自己服务。从而产生瓶颈。于是提出标准的概念。其实 J2EE 就是基于 JAVA 技术的一系列标准。

注：中间件的解释 中间件处在操作系统和更高一级应用程序之间。他充当的功能是：将应用程序运行环境与操作系统隔离，从而实现应用程序开发者不必为更多系统问题忧虑，而直接关注该应用程序在解决问题上的能力 。我们后面说到的容器的概念就是中间件的一种。

二、相关名词解释

容器：充当中间件的角色

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。

WEB 容器和 EJB 容器在原理上是大体相同的，更多的区别是被隔离的外界环境。WEB 容器更多的是跟基于 HTTP 的请求打交道。而 EJB 容器不是。 它是更多的跟数据库、其它服务打交道。但他们都是把与外界的交互实现从而减轻应用程序的负担。例如 SERVLET 不用关心 HTTP 的细节，直接引用环境变 量

session,request,response 就行、EJB 不用关心数据库连接速度、各种事务控制，直接由容器来完成。

RMI/IIOP:远程方法调用/internet 对象请求中介协议，他们主要用于通过远程调用服务。例如，远程有一台计算机上运行一个程序，它提 供股票分析服务，我们可以在本地计算机上实现对其直接调用。当然这是要通过一定的规范才能在异构的系统之间进行通信。RMI 是 JAVA 特有的。

JNDI:JAVA 命名目录服务。主要提供的功能是：提供一个目录系统，让其它各地的应用程序在其上面留下自己的索引，从而满足快速查找和定位分布式应用程序的功能。

JMS:JAVA 消息服务。主要实现各个应用程序之间的通讯。包括点对点和广播。

JAVAMAIL:JAVA 邮件服务。提供邮件的存储、传输功能。他是 JAVA 编程中实现邮件功能的核心。相当 MS 中的 EXCHANGE 开发包。

JTA：JAVA 事务服务。提供各种分布式事务服务。应用程序只需调用其提供的接口即可。

JAF:JAVA 安全认证框架。提供一些安全控制方面的框架。让开发者通过各种部署和自定义实现自己的个性安全控制策略。

EAI:企业应用集成。是一种概念，从而牵涉到好多技术。J2EE 技术是一种很好的集成实现。

三、J2EE 的优越性
1、 基于 JAVA 技术，平台无关性表现突出

2、 开放的标准，许多大型公司已经实现了对该规范支持的应用服务器。如 BEA ,IBM,ORACLE 等。

3、 提供相当专业的通用软件服务。

4、 提供了一个优秀的企业级应用程序框架，对快速高质量开发打下基础

四、现状
J2EE 是由 SUN 公司开发的一套企业级应用规范。现在最高版本是 1.4。支持 J2EE 的应用服务器有 IBM WEBSPHERE APPLICATION SERVER,BEA WEBLOGIC SERVER,JBOSS,ORACLE APPLICATION SERVER,SUN ONE APPLICATION SERVER 等。

ff2004 发表评论于 2004-12-14 22:07:37 | 引用

学习 Java 的 30 个基本概念

Java 概述:

目前 Java 主要应用于中间件的开发(middleware)---处理客户机于服务器之间的通信技术,早期的实践证明,Java 不适合 pc 应用程序的开发,其发展逐渐变成在开发手持设备,互联网信息站,及车载计算机的开发.Java 于其他语言所不同的是程序运行时提供了平台的独立性,称 许可以在 windows,solaris,linux 其他操作系统上使用完全相同的代码.Java 的语法与 C++语法类似,C++/C 程序员很容易掌握, 而且 Java 是完全的彻底的面向对象的,其中提出了很好的 GC(Garbage Collector)垃圾处理机制,防止内存溢出.

Java 的白皮书为我们提出了 Java 语言的 11 个关键特性.

(1)Easy:Java 的语法比 C++的相对简单,另一个方面就是 Java 能使软件在很小的机器上运行,基础解释其和类库的支持的大小约为 40kb,增加基本的标准库和线程支持的内存需要增加 125kb.

(2)分布式:Java 带有很强大的 TCP/IP 协议族的例程库,Java 应用程序能够通过 URL 来穿过网络来访问远程对象,由于 servlet 机制的出现,使 Java 编程非常的高效,现在许多的大的 web server 都支持 servlet.

(3)OO:面向对象设计是把重点放在对象及对象的接口上的一个编程技术.其面向对象和 C++有很多不同,在与多重继承的处理及 Java 的原类模型.

(4)健壮特性:Java 采取了一个安全指针模型,能减小重写内存和数据崩溃的可能性。

(5)安全:Java 用来设计网路和分布系统,这带来了新的安全问题,Java 可以用来构建防病毒和防攻击的 System.事实证明 Java 在防毒这一方面做的比较好.

(6)中立体系结构:Java 编译其生成体系结构中立的目标文件格式可以在很多处理器上执行,编译器产生的指令字节码(Javabytecode)实现此特性,此字节码可以在任何机器上解释执行.

(7)可移植性:Java 中对基本数据结构类型的大小和算法都有严格的规定所以可移植性很好.

(8)多线程:Java 处理多线程的过程很简单,Java 把多线程实现交给底下操作系统或线程程序完成.所以多线程是 Java 作为服务器端开发语言的流行原因之一

(9)Applet 和 servlet:能够在网页上执行的程序叫 Applet,需要支持 Java 的浏览器很多,而 applet 支持动态的网页,这是很多其他语言所不能做到的.

基本概念:

1.OOP 中唯一关系的是对象的接口是什么,就像计算机的销售商她不管电源内部结构是怎样的,他只关系能否给你提供电就行了,也就是只要知道 can or not 而不是 how and why.所有的程序是由一定的属性和行为对象组成的,不同的对象的访问通过函数调用来完成,对象间所有的交流都是通过方法调用,通过对封装对象数据,很大限度上提高复用率.

2.OOP 中最重要的思想是类,类是模板是蓝图,从类中构造一个对象,即创建了这个类的一个实例(instance)

3.封装:就是把数据和行为结合起在一个包中)并对对象使用者隐藏数据的实现过程,一个对象中的数据叫他的实例字段(instance field)

4.通过扩展一个类来获得一个新类叫继承(inheritance),而所有的类都是由 Object 根超类扩展而得,根超类下文会做介绍.

5.对象的 3 个主要特性

behavior---说明这个对象能做什么.

state---当对象施加方法时对象的反映.

identity---与其他相似行为对象的区分标志.

每个对象有唯一的 indentity 而这 3 者之间相互影响.

6.类之间的关系:

use-a :依赖关系

has-a :聚合关系

is-a :继承关系--例:A 类继承了 B 类,此时 A 类不仅有了 B 类的方法,还有其自己的方法.(个性存在于共性中)

7.构造对象使用构造器:构造器的提出,构造器是一种特殊的方法,构造对象并对其初始化.

例:Data 类的构造器叫 Data

new Data()---构造一个新对象,且初始化当前时间.

Data happyday=new

Data()---把一个对象赋值给一个变量 happyday,从而使该对象能够多次使用,此处要声明的使变量与对象变量二者是不同的.new 返回的值是一个引用.

构造器特点:构造器可以有 0 个,一个或多个参数

构造器和类有相同的名字

一个类可以有多个构造器

构造器没有返回值

构造器总是和 new 运算符一起使用.

8.重载:当多个方法具有相同的名字而含有不同的参数时,便发生重载.编译器必须挑选出调用哪个方法.

9.包(package)Java 允许把一个或多个类收集在一起成为一组,称作包,以便于组织任务,标准 Java 库分为许多包.java.lang java.util java,net 等,包是分层次的所有的 java 包都在 java 和 javax 包层次内.

10.继承思想:允许在已经存在的类的基础上构建新的类,当你继承一个已经存在的类时,那么你就复用了这个类的方法和字段,同时你可以在新类中添加新的方法和字段.

11.扩展类:扩展类充分体现了 is-a 的继承关系. 形式为:class (子类) extends (基类).

12.多态:在 java 中,对象变量是多态的.而 java 中不支持多重继承.

13.动态绑定:调用对象方法的机制.

(1)编译器检查对象声明的类型和方法名.

(2)编译器检查方法调用的参数类型.

(3)静态绑定:若方法类型为 priavte static final 编译器会准确知道该调用哪个方法.

(4)当程序运行并且使用动态绑定来调用一个方法时,那么虚拟机必须调用 x 所指向的对象的实际类型相匹配的方法版本.

(5)动态绑定:是很重要的特性,它能使程序变得可扩展而不需要重编译已存代码.

14.final 类:为防止他人从你的类上派生新类,此类是不可扩展的.

15.动态调用比静态调用花费的时间要长,

16.抽象类:规定一个或多个抽象方法的类本身必须定义为 abstract 例: public abstract string getDescripition

17.Java 中的每一个类都是从 Object 类扩展而来的.

18.object 类中的 equal 和 toString 方法.equal 用于测试一个对象是否同另一个对象相等.toString 返回一个代表该对象的字符串,几乎每一个类都会重载该方法,以便返回当前状态的正确表示.(toString 方法是一个很重要的方法)

19.通用编程:任何类类型的所有值都可以同 object 类性的变量来代替.

20.数组列表:ArrayList 动态数组列表,是一个类库,定义在 java.uitl 包中,可自动调节数组的大小.

21.class 类 object 类中的 getclass 方法返回 ckass 类型的一个实例,程序启动时包含在 main 方法的类会被加载,虚拟机要加载他需要的所有类,每一个加载的类都要加载它需要的类.

22.class 类为编写可动态操纵 java 代码的程序提供了强大的功能反射,这项功能为 JavaBeans 特别有用,使用反射 Java 能 支持 VB 程序员习惯使用的工具.能够分析类能力的程序叫反射器,Java 中提供此功能的包叫 Java.lang.reflect 反射机制十分强大.

1.在运行时分析类的能力.

2.在运行时探察类的对象.

3.实现通用数组操纵代码.

4.提供方法对象.

而此机制主要针对是工具者而不是应用及程序.

反射机制中的最重要的部分是允许你检查类的结构.用到的 API 有:

java.lang.reflect.Field 返回字段.

java.reflect.Method 返回方法.

java.lang.reflect.Constructor 返回参数.

方法指针:java 没有方法指针,把一个方法的地址传给另一个方法,可以在后面调用它,而接口是更好的解决方案.

23.接口(Interface)说明类该做什么而不指定如何去做,一个类可以实现一个或多个 interface.

24.接口不是一个类,而是对符合接口要求的类的一套规范.若实现一个接口需要 2 个步骤:

1.声明类需要实现的指定接口.

2.提供接口中的所有方法的定义.

声明一个类实现一个接口需要使用 implements 关键字 class actionB implements Comparable 其 actionb 需要提供 CompareTo 方法,接口不是类,不能用 new 实例化一个接口.

25.一个类只有一个超类,但一个类能实现多个接口.Java 中的一个重要接口 Cloneable

26.接口和回调.编程一个常用的模式是回调模式,在这种模式中你可以指定当一个特定时间发生时回调对象上的方法.例:ActionListener 接口监听.

类似的 API 有:java.swing.JOptionPane

java.swing.Timer

java.awt.Tookit

27.对象 clone:clone 方法是 object 一个保护方法,这意味着你的代码不能简单的调用它.

28.内部类:一个内部类的定义是定义在另一个内部的类

原因是:1.一个内部类的对象能够访问创建它的对象的实现,包括私有数据

2.对于同一个包中的其他类来说,内部类能够隐藏起来.

3.匿名内部类可以很方便的定义回调.

4.使用内部类可以非常方便的编写事件驱动程序.

29.代理类(proxy):1.指定接口要求所有代码 2.object 类定义的所有的方法 (toString equals)

30.数据类型:Java 是强调类型的语言,每个变量都必须先申明它都类型,java 中总共有 8 个基本类型.4 种是整型,2 种是浮点型,一种是字符型,被用于 Unicode 编码中的字符,布尔型.

JAVA 面试题集（7）- -

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。

JAVA 华为面试题

JAVA 方面

1 面向对象的特征有哪些方面

2 String 是最基本的数据类型吗?

3 int 和 Integer 有什么区别

4 String 和 StringBuffer 的区别

5 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。

6 说出一些常用的类，包,接口，请各举 5 个

7 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。

8 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。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();
}
}
}
}
```

9. 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

10.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。

11 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。

12.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。

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

14. 说出数据连接池的工作机制是什么？

15 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。

JAVA 面试题集（8）- -

16 应用服务器有那些？

17 你所知道的集合类都有哪些？主要方法？

18 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。

19. 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。

数据库方面：

1. 存储过程和函数的区别

存储过程是用户定义的一系列 sql 语句的集合，涉及特定表或其它对象的任务，用户可以调用存储过程，而函数通常是数据库已定义的方法，它接收参数并返回某种类型的值并且不涉及特定用户表。

2. 事务是什么？

事务是作为一个逻辑单元执行的一系列操作，一个逻辑工作单元必须有四个属性，称为 ACID（原子性、一致性、隔离性和持久性）属性，只有这样才能成为一个事务：

原子性

事务必须是原子工作单元；对于其数据修改，要么全都执行，要么全都不执行。

一致性

事务在完成时，必须使所有的数据都保持一致状态。在相关数据库中，所有规则都必须应用于事务的修改，以保持所有数据的完整性。事务结束时，所有的内部数据结构（如 B 树索引或双向链表）都必须是正确的。

隔离性

由并发事务所作的修改必须与任何其它并发事务所作的修改隔离。事务查看数据时数据所处的状态，要么是另一并发事务修改它之前的状态，要么 是另一事务修改它之后的状态，事务不会查看中间状态的数据。这称为可串行性，因为它能够重新装载起始数据，并且重播一系列事务，以使数据结束时的状态与原 始事务执行的状态相同。

持久性

事务完成之后，它对于系统的影响是永久性的。该修改即使出现系统故障也将一直保持。

3. 游标的作用？如何知道游标已经到了最后？

游标用于定位结果集的行，通过判断全局变量@@FETCH_STATUS 可以判断是否到了最后，通常此变量不等于 0 表示出错或到了最后。

4. 触发器分为事前触发和事后触发，这两种触发有和区别。语句级触发和行级触发有何区别。

事前触发器运行于触发事件发生之前，而事后触发器运行于触发事件发生之后。通常事前触发器可以获取事件之前和新的字段值。

语句级触发器可以在语句执行前或后执行，而行级触发在触发器所影响的每一行触发一次。

中远面试题

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

二、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

三、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

四、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

五、数据库
1、存储过程的编写

2、基本的 SQL 语句

六、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

JAVA 面试题集- -

基础知识：
1.C++或 Java 中的异常处理机制的简单原理和应用。
当 JAVA 程序违反了 JAVA 的语义规则时，JAVA 虚拟机就会将发生的错误表示为一个异常。违反语义规则包括 2 种情况。一种是 JAVA 类库内置的语义检查。例如数组下标越界,会引发 IndexOutOfBoundsException;访问 null 的对象时会引发 NullPointerException。另一种情况就是 JAVA 允 许程序员扩展这种语义检查，程序员可以创建自己的异常，并自由选择在何时用 throw 关键字引发异常。所有的异常都是 java.lang.Thowable 的子类。
2. Java 的接口和 C++的虚类的相同和不同处。
由于 Java 不支持多继承，而有 可能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制就不能满足要求。与继承相比，接口有更高的灵活性，因为接口中没有任何实现 代码。当一个类实现了接口以后，该类要实现接口里面所有的方法和属性，并且接口里面的属性在默认状态下面都是 public static,所有方法默认情况下是 public.一个类可以实现多个接口。
3. 垃圾回收的优点和原理。并考虑 2 种回收机制。
Java 语言中一个显著的特点就是引入了垃圾回收机制，使 c++程序员最头疼的内存管理的问题迎刃而解，它使得 Java 程序员在编写程序的时候不再需要考虑内存管 理。由于有个垃圾回收机制，Java 中的对象不再有"作用域"的概念，只有对象的引用才有"作用域"。垃圾回收可以有效的防止内存泄露，有效的使用可以使 用的内存。垃圾回收器通常是作为一个单独的低级别的线程运行，不可预知的情况下对内存堆中已经死亡的或者长时间没有使用的对象进行清楚和回收，程序员不能 实时的调用垃圾回收器对某个对象或所有对象进行垃圾回收。回收机制有分代复制垃圾回收和标记垃圾回收，增量垃圾回收。
4. 请说出你所知道的线程同步的方法。
wait():使一个线程处于等待状态，并且释放所持有的对象的 lock。
sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 InterruptedException 异常。
notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级。
Allnotity():唤醒所有处入等待状态的线程,注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。
5. 请讲一讲析构函数和虚函数的用法和作用。
6. Error 与 Exception 有什么区别？
Error 表示系统级的错误和程序不必处理的异常，
Exception 表示需要捕捉或者需要程序进行处理的异常。
7. 在 java 中一个类被声明为 final 类型，表示了什么意思？
表示该类不能被继承，是顶级类。

8. 描述一下你最常用的编程风格。
9. heap 和 stack 有什么区别。
栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。
堆是栈的一个组成元素
10. 如果系统要使用超大整数（超过 long 长度范围），请你设计一个数据结构来存储这种超大型数字以及设计一种算法来实现超大整数加法运算）。

```
public class BigInt()
{
int[] ArrOne = new ArrOne[1000];
String intString="";
public int[] Arr(String s)
{
intString = s;
for(int i=0;i<ArrOne.leght;i++)
{
```

11. 如果要设计一个图形系统，请你设计基本的图形元件
(Point,Line,Rectangle,Triangle)的简单实现
12，谈谈 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。
13，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。

14，Static Nested Class 和 Inner Class 的不同，说得越多越好(面试题有的很笼统)。

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。
15，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。
16，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。
17，什么时候用 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。
18，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()
19，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。

20，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;

21，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。

22，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。

23，Java 有没有 goto?

Goto—java 中的保留字，现在没有在 java 中使用。

24，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。

25，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。

26，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。

27，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

28，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。

29，List, Set, Map 是否继承自 Collection 接口?

List，Set 是

Map 不是

30，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。

31，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

32，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。

33，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。

34，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。

35，是否可以继承 String 类?

String 类是 final 类故不可以继承。

36，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。

37,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。

38，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

39，两个对象值相同(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。

40，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。

41，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。

42，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

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。

43.描述一下 JVM 加载 class 文件的原理机制?

44.试举例说明一个典型的垃圾回收算法？

45.请用 java 写二叉树算法，实现添加数据形成二叉树功能，并以先序的方式打印出来.

46.请写一个 java 程序实现线程连接池功能？

47.给定一个 C 语言函数，要求实现在 java 类中进行调用。

48、编一段代码，实现在控制台输入一组数字后，排序后在控制台输出；

49、列出某文件夹下的所有文件；

50、调用系统命令实现删除文件的操作；

51、实现从文件中一次读出一个字符的操作；

52、列出一些控制流程的方法；

53、多线程有哪些状态？

54、编写了一个服务器端的程序实现在客户端输入字符然后在控制台上显示，直到输入"END"为止，让你写出客户端的程序；

55、作用域 public,private,protected,以及不写时的区别

答：区别如下：

作用域 当前类 同一 package 子孙类 其他 package

public √ √ √ √

protected √ √ √ ×

friendly √ √ × ×

private √ × × ×

不写时默认为 friendly

56、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
57、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
58、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
59、垃圾回收机制,如何优化程序?
希望大家补上，谢谢
60、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
61、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
62、Java 中异常处理机制，事件机制?
11、JAVA 中的多形与继承?
希望大家补上，谢谢
63、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)。

编程题：
1．现在输入 n 个数字，以逗号，分开；
然后可选择升或者降序排序；
按提交键就在另一页面显示
　按什么 排序，结果为，　，
提供 reset
答案（1）　public static String[] splitStringByComma(String source){
　　if(source==null||source.trim().equals(""))
　　return null;
　　StringTokenizer commaToker =　new
StringTokenizer(source,",");
　　String[] result = new
String[commaToker.countTokens()];
　　int i=0;
　　while(commaToker.hasMoreTokens()){
　　result[i] =
commaToker.nextToken();
　　i++;
　　}
　　return result;
　}
循环遍历 String 数组
Integer.parseInt(String s)变成 int 类型
组成 int 数组
Arrays.sort(int[] a),
a 数组升序
降序可以从尾部开始输出

2．金额转换，阿拉伯数字的金额转换成中国传统的形式如：
（￥1011）－>（一千零一拾一元整）输出。
3、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
```

4、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;}
}
C. 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

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();
}
};
```

6、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
7、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()/SplitBy
te):(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();
}
```

}
8、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
9、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
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）：
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
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().newDocumentBui
lder();
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。
JAVA 华为面试题
JAVA 方面
1 面向对象的特征有哪些方面
2 String 是最基本的数据类型吗?
3 int 和 Integer 有什么区别
4 String 和 StringBuffer 的区别
5 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。
6 说出一些常用的类，包,接口，请各举 5 个
7 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。
8 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。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.currentT
hread().getName()+"-inc:"+j);
                    }
                    private synchronized void dec(){
                                      j--;
                                    System.out.println(Thread.currentT
hread().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();
                                                        }
                                      }
                    }
}
```

9. 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
10.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。
11 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。
12.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。

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

14. 说出数据连接池的工作机制是什么？

15 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。

16 应用服务器有那些？

17 你所知道的集合类都有哪些？主要方法？

18 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。

19．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。

数据库方面：

1. 存储过程和函数的区别
存储过程是用户定义的一系列 sql 语句的集合，涉及特定表或其它对象的任务，用户可以调用存储过程，而函数通常是数据库已定义的方法，它接收参数并返回某种类型的值并且不涉及特定用户表。
2. 事务是什么？
事务是作为一个逻辑单元执行的一系列操作，一个逻辑工作单元必须有四个属性，称为 ACID（原子性、一致性、隔离性和持久性）属性，只有这样才能成为一个事务：
原子性
事务必须是原子工作单元；对于其数据修改，要么全都执行，要么全都不执行。
一致性
事务在完成时，必须使所有的数据都保持一致状态。在相关数据库中，所有规则都必须应用于事务的修改，以保持所有数据的完整性。事务结束时，所有的内部数据结构（如 B 树索引或双向链表）都必须是正确的。
隔离性
由 并发事务所作的修改必须与任何其它并发事务所作的修改隔离。事务查看数据时数据所处的状态，要么是另一并发事务修改它之前的状态，要么是另一事务修改它之 后的状态，事务不会查看中间状态的数据。这称为可串行性，因为它能够重新装载起始数据，并且重播一系列事务，以使数据结束时的状态与原始事务执行的状态相 同。
持久性
事务完成之后，它对于系统的影响是永久性的。该修改即使出现系统故障也将一直保持。

3. 游标的作用？如何知道游标已经到了最后？
游标用于定位结果集的行，通过判断全局变量@@FETCH_STATUS 可以判断是否到了最后，通常此变量不等于 0 表示出错或到了最后。
4. 触发器分为事前触发和事后触发，这两种触发有和区别。

语句级触发和行级触发有何区别。
事前触发器运行于触发事件发生之前，而事后触发器运行于触发事件发生之后。通常事前触发器可以获取事件之前和新的字段值。
语句级触发器可以在语句执行前或后执行，而行级触发在触发器所影响的每一行触发一次。

中远面试题

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

二、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

三、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

四、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

五、数据库

1、存储过程的编写

2、基本的 SQL 语句

六、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