第一部分 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< p>

{
```

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 objecs 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 objecs?

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*树算法，实现添加数据形成二*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 objecs，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？

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)。

第二部分 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 *= 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< ) ;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<>

{

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< ) ;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<>
{
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()/SplitByte):(SplitStr.length()/Split
    Byte+1);
    System.out.println("Will Split into "+loopCount);
    for (int i=1;i<=loopCount ;i++ )
    {
    if (i==loopCount){
    System.out.println(SplitStr.substring((i-1)*SplitByte,SplitStr.length()));
    } else {
```

```
System.out.println(SplitStr.substring((i-1)*SplitByte,(i*SplitByte)));
}
}
}
public static void main(String[] args)
{
SplitString ss = new SplitString("test 中 dd 文 dsaf 中男大3443n 中国43中国人

0ewldfls=103",4);
ss.SplitIt();
}
}
```

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

第三部分 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 objecs 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
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

4、两种跳转方式分别是什么?有什么区别?

答：有两种，分别为：

前者页面不会转向 include 所指的页面，只是显示该页的结果，主页面还是原来的页面。执行完后还会回来，相当于函数调用。并且可以带参数.后者完全转向新页面，不会再回来。相当于 go to 语句。

第四部分 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 {
}
}
```

第五部分 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 objecs 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

```
//输出内容

//输出翻页连接

合计:/第一页<>
```

href="List.jsp?page=">上一页

[]

下一页最后页

第六部分 Xml 方面

1、xml 有哪些解析技术?区别是什么?

答:有 DOM,SAX,STAX 等

DOM:处理大型文件时其性能下降的非常厉害。这个问题是由DOM的树结构所造成的，这种结构占用的内存较多，而且DOM必须在解析文件之前把整个文档装入内存,适合对XML的随机访问SAX:不现于DOM,SAX是事件驱动型的XML解析方式。它顺序读取XML文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理XML文件，适合对XML的顺序访问

STAX:Streaming API for XML (StAX)

2、你在项目中用到了 xml 技术的哪些方面?如何实现的?

答:用到了数据存贮，信息配置两方面。在做数据交换平台时，将不能数据源的数据组装成 XML 文件，然后将 XML 文件压缩打包加密后通过网络传送给接收者，接收解密与解压缩后再同 XML 文件中还原相关信息进行处理。在做软件配置时，利用 XML 可以很方便的进行，软件的各种配置参数都存贮在 XML 文件中。

3、用 jdom 解析 xml 文件时如何解决中文问题?如何解析?

答:看如下代码,用编码方式加以解决

```
package test;

import java.io.*;

public class DOMTest

{
```

```
private String inFile = "c:\people.xml";
private String outFile = "c:\people.xml";
public static void main(String args[])
{
new DOMTest();
}
public DOMTest()
{
try
{
javax.xml.parsers.DocumentBuilder builder =
javax.xml.parsers.DocumentBuilderFactory.newInstance().newDocumentBuilder();
org.w3c.dom.Document doc = builder.newDocument();
org.w3c.dom.Element root = doc.createElement("老师");
org.w3c.dom.Element wang = doc.createElement("王");
org.w3c.dom.Element liu = doc.createElement("刘");
wang.appendChild(doc.createTextNode("我是王老师"));
root.appendChild(wang);
doc.appendChild(root);
javax.xml.transform.Transformer transformer =
javax.xml.transform.TransformerFactory.newInstance().newTransformer();
transformer.setOutputProperty(javax.xml.transform.OutputKeys.ENCODING, "gb2312");
transformer.setOutputProperty(javax.xml.transform.OutputKeys.INDENT, "yes");
```

transformer.transform(new javax.xml.transform.dom.DOMSource(doc),

new

javax.xml.transform.stream.StreamResult(outFile));

}

catch (Exception e)

{

System.out.println (e.getMessage());

}

}

}

4、编程用 JAVA 解析 XML 的方式.

答:用 SAX 方式解析 XML，XML 文件如下：

王小明
信息学院

6258113

男,1975年生,博士，2005年调入河南大学

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:

第七部分 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.EJBobjecs;

import java.rmi.RemoteException;

public interface Add extends EJBobjecs

{

//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

}

第八部分 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 objecs 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,

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 objecs Request Broker Architecture)，由对象管理组织 (objecs 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。