Java 基础

1、一个".java"源文件中是否可以包括多个类？有什么限制？

可以有多个类，但只能有一个 public 的类，并且 public 的类名必须与文件名相一致。

2、 Java 有没有 goto?

java 中的保留字，现在没有在 java 中使用。

3、说说&和&&的区别。

&和&&都可以用作逻辑与的运算符，当运算符两边的表达式的结果都为 true 时，整个运算结果才为 true，否则，只要有一方为 false，则结果为 false。

&&还具有短路的功能，即如果第一个表达式为 false，则不再计算第二个表达式。

&还可以用作位运算符，当&操作符两边的表达式不是 boolean 类型时，&表示按位与操作。

4、在 JAVA 中如何跳出当前的多重嵌套循环？

break 只能跳出当前的一个循环语句。

在循环体开头设置一个标志位，也就是设置一个标记，然后使用带此标号的 break 语句跳出多重循环。

设置一个 boolean 值的标记位，通过设置标志位，实现里成的代码控制外层的的循环条件。

5、 switch 语句能否作用在 byte 上，能否作用在 long 上，能否作用在 String 上?

在 switch（ expr1）中， expr1 只能是一个整数表达式或者枚举常量

byte、short、int 、char 和 string

6、 short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?

对于 short s1 = 1; s1 = s1 + 1;由于 s1+1 运算时会自动提升表达式的类型，所以结果是 int 型，再赋值给 short 类型 s1 时， 编译器将报告需要强制转换类型的错误。

对于 short s1 = 1; s1 += 1;由于 +=是 java 语言规定的运算符，java 编译器会对它进行特殊处理，因此可以正确编译。

7、 char 型变量中能不能存贮一个中文汉字?为什么？

char 型变量是用来存储 Unicode 编码的字符的，unicode 编码字符集中包含了汉字,所以，char 型变量中当然可以存储汉字啦。不过，如果某个特殊的汉字没有被包含在 unicode 编码字符集中，那么，这个 char 型变量中就不能存储这个特殊汉字。 补充说明： unicode 编码占用两个字节，所以， char 类型的变量也是占用两个字节。

8、用最有效率的方法算出 2 乘以 8 等于几?

2 << 3，

因为将一个数左移 n 位，就相当于乘以了 2 的 n 次方，那么，一个数乘以 8 只要将其左移 3 位

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

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

11、 "=="和 equals 方法究竟有什么区别?

==操作符专门用来比较两个变量的值是否相等，也就是用于比较变量所对应的内存中所存储的数值是否相同。

equals 方法是用于比较两个独立对象的内容是否相同。

字符串的比较基本上都是使用 equals 方法。

基本数据类型之间的比较，应用双等号（==）,比较的是他们的值。

复合数据类型(类)，当他们用（==）进行比较的时候，比较的是他们在内存中的存放地址，所以,除非是同一个 new 出来的对象,他们的比较后的结果为 true,否则比较后结果为 false。

12、静态变量和实例变量的区别?

在语法定义上的区别： 静态变量前要加 static 关键字，而实例变量前则不加。

在程序运行时的区别：实例变量属于某个对象的属性，必须创建了实例对象，其中的实例变量才会被分配空间，才能使用这个实例变量。静态变量不属于某个实例对象，而是属于类，所以也称为类变量，只要程序加载了类的字节码，不用创建任何实例对象，静态变量就会被分配空间，静态变量就可以被使用了。总之，实例变量必须创建对象后才可以通过这个对象来使用，静态变量则可以直接使用类名来引用。

13、是否可以从一个 static 方法内部发出对非 static 方法的调用?

不可以。因为非 static 方法是要与对象关联在一起的，必须创建一个对象后，才可以在该对象上进行方法调用，而 static 方法调用时不需要创建对象，可以直接调用。也就是说， 当一个 static 方法被调用时，可能还没有创建任何实例对象，如果从一个 static 方法中发出对非 static 方法的调用，那个非 static 方法是关联到哪个对象上的呢？这个逻辑无法成立，所以，一个 static 方法内部不可以发出对非 static 方法的调用。

14、 Integer 与 int 的区别

int 是 java 提供的 8 种原始数据类型之一。Java 为每个原始类型提供了封装类，Integer 是 java 为 int 提供的封装类。 int 的默认值为 0，而 Integer 的默认值为 null，即 Integer 可以区分出未赋值和值为 0 的区别， int 则无法表达出未赋值的情况

15、 Math.round(11.5)等於多少? Math.round(-11.5)等於多少?

Math 类中提供了三个与取整有关的方法： ceil、 floor、 round。

ceil 的英文意义是天花板，向上取整，Math.ceil(11.3)的结果为 12,Math.ceil(-11.3)的结果是-11；

floor 的英文意义是地板，向下取整，Math.ceil(11.6)的结果为 11,Math.ceil(-11.6)的结果是-12；

round 方法，它表示“四舍五入”，他表示将原来的数字加上 0.5 后再向下取整，所以，Math.round(11.5)的结果为 12， Math.round(-11.5)的结果为-11。

17、请说出作用域 public， private， protected，以及不写时的区别

1.什么都不写：包可见　　2.private：只能在自己的类中

3.public：任意位置都可以访问　　4.protected：子类，同包中

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

Overload 是重载，Override 是重写。
重写是父类与子类之间多态性的一种表现。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。
重载是一个类中多态性的一种表现。如果在一个类中定义了多个同名的方法，它们的参数列表各不相同（个数或类型或顺序不同），则称为方法的重载。重载的方法可以改变返回值的类型。

19、构造器 Constructor 是否可被 override?

构造器 Constructor 不能被继承，因此不能重写 Override，但可以被重载 Overload

20、接口是否可继承接口？抽象类是否可实现(implements)接口？抽象类是否可继承具体类(concrete class)?抽象类中是否可以有静态的 main 方法？

接口可以继承接口。抽象类可以实现(implements)接口，抽象类可以继承具体类。抽象类中可以有静态的 main 方法。

21、写 clone()方法时，通常都有一行代码，是什么？

clone 有缺省行为，super.clone();
因为首先要把父类中的成员复制到位，然后才是复制自己的成员。

22、面向对象的特征有哪些方面

封装，继承，多态，抽象

23、 java 中实现多态的机制是什么？

java 中实现多态的机制是依靠父类或接口的引用指向子类。从而实现了一个对象多种形态的特性。其中父类的引用是在程序运行时动态的指向具体的实例，调用该引用的方法时，不是根据引用变量的类型中定义的方法来运行，而是根据具体的实例的方法。

24、 abstract class 和 interface 有什么区别?

abstract 类不能创建实例对象。含有 abstract 方法的类必须定义为 abstract class，abstract class 类中的方法不必全是抽象的。如果的子类没有实现抽象父类中的所有抽象方法，那么子类也必须定义为 abstract 类型。
接口（ interface）可以说成是抽象类的一种特例，接口中的所有方法都必须是抽象的。 接口中的方法定义默认为 public abstract 类型，接口中的成员变量类型默认为 public static final。
下面比较一下两者的语法区别:
1.抽象类可以有构造方法，接口中不能有构造方法。
2.抽象类中可以有普通成员变量，接口中没有普通成员变量
3.抽象类中可以包含非抽象的普通方法，接口中的所有方法必须都是抽象的，不能有非抽象的普通方法。
4. 抽象类中的抽象方法的访问类型可以是 public， protected，但接口中的抽象方法只能是 public 类型的，并且默认即为 public abstract 类型。
5. 抽象类中可以包含静态方法，接口中不能包含静态方法
6. 抽象类和接口中都可以包含静态成员变量，抽象类中的静态成员变量的访问类型可以任意，但接口中定义的变量只能是 public static final 类型，并且默认即为 public static final 类型。
7. 一个类可以实现多个接口，但只能继承一个抽象类。

26、什么是内部类？ Static Nested Class 和 Inner Class 的不同。

内部类就是在一个类的内部定义的类，内部类中不能定义静态成员，内部类可以直接访问外部类中的成员变量,内部类可以定义在外部类的方法外面,也可以定义在外部类的方法体中。

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。

27、内部类可以引用它的包含类的成员吗？有没有什么限制？

完全可以。如果不是静态内部类，那没有什么限制！

28、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)?

匿名类本身就是通过继承类或者接口来实现的。但是不能再显式的 extends 或者 implements 了。

30、 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

31、 String s = "Hello";s = s + " world!";这两行代码执行后，原始的 String 对象中的内容到底变了没有?

没有。因为 String 被设计成不可变类，所以它的所有对象都是不可变对象。在这段代码中，s 原先指向一个 String 对象，内容是 "Hello"，然后我们对 s 进行了+操作， s 所指向的那个对象并没有发生改变。这时， s 不指向原来那个对象了，而指向了另一个 String 对象，内容为"Hello world!",原来那个对象还存在于内存之中,只是 s 这个引用变量不再指向它了。

33、 String s = new String("xyz");创建了几个 String Object?二者之间有什么区别?

如果 String 常理池中，已经创建"xyz"，则不会继续创建，此时只创建了一个对象 new String("xyz");

如果 String 常理池中，没有创建"xyz"，则会创建两个对象，一个对象的值是"xyz"，一个对象 new String("xyz")。

34、 String 和 StringBuffer 的区别

String 和 StringBuffer，它们可以储存和操作字符串。

这个 String 类提供了数值不可改变的字符串。而这个 StringBuffer 类提供的字符串可以修改。当你知道字符数据要改变的时候你就可以使用 StringBuffer。

String 实现了 equals 方法，,而 StringBuffer 没有实现 equals 方法。

StringBuffer 类的 append 方法追加字符比 String 使用 + 操作符添加字符到一个已经存在的字符串后面有效率得多

35、如何把一段逗号分割的字符串转换成一个数组?

```
String msg="a,b,c,d,e,f";
String [] arr = msg.split(",");
```

36、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。

38、 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?

finally 语句放在 try catch 语句后，不管异常是否被处理，finally 语句总是执行，不管是 return 还是 break 都执行，除了 System.exit（0）。前提是这些中断的都是在 try catch 中。

40、 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。 JVM 不保证此方法总被调用

41、运行时异常与一般异常有何异同?

运行时异常表示虚拟机的正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。

42、 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。

44、请写出你最常见到的 5 个 runtime exception。

NullPointerException 、 ArrayIndexOutOfBoundsException 、 ClassCastException 、 ArithmeticException、IllegalArgumentException

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

throws 捕获并向外抛出异常

throw 抛出异常（手动）

try catch 是内部捕获异常并做自定义处理

finally 是无论是否有异常都会被处理的语句，除非在 finally 前存在被执行的 System.exit(int i)时除外

46、java 中有几种方法可以实现一个线程？用什么关键字修饰同步方法? stop()和 suspend()方法为何不推荐使用?

有两种实现方法， 分别是继承 Thread 类与实现 Runnable 接口

用 synchronized 关键字修饰同步方法

反对使用 stop()，是因为它不安全。

suspend()方法容易发生死锁。

47、 sleep()和 wait()有什么区别?

sleep 就是正在执行的线程主动让出 cpu，cpu 去执行其他线程，在 sleep 指定的时间过后，cpu 才会回到这个线程上继续往下执行，如果当前线程进入了同步锁， sleep 方法并不会释放锁，即使当前线程使用 sleep 方法让出了 cpu，但其他被同步锁挡住了的线程也无法得到执行。

wait 是指在一个已经进入了同步锁的线程内，让自己暂时让出同步锁，以便其他正在等待此锁的线程可以得到同步锁并运行，只有其他线程调用了 notify 方法（ notify 并不释放锁，只是告诉调用过 wait 方法的线程可以去参与获得锁的竞争了，但不是马上得到锁，因为锁还在别人手里，别人还没释放。如果 notify 方法后面的代码还有很多，需要这些代码执行完后才会释放锁，可以在 notfiy 方法后增加一个等待和一些代码，看看效果），调用 wait 方法的线程就会解除 wait 状态和程序可以再次得到锁后继续向下运行。

48、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。

50、多线程有几种实现方法?同步有几种实现方法?

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

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(中断异常)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。

51、启动一个线程是用 run()还是 start()?.

启动一个线程是调用 start()方法，使线程就绪状态，以后可以被调度为运行状态，一个线程必须关联一些具体的执行代码， run()方法是该线程所关联的执行代码。

52、当一个线程进入一个对象的一个 synchronized 方法后，其它线程是否可进入此对象的其它方法?

分几种情况：

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

53、 线程的基本概念、线程的基本状态以及状态之间的关系

线程：是进程中的一个执行控制单元，执行路径

一个进程中至少有一个线程在负责控制程序的执行

一个进程中如果只有一个执行路径，这个程序称为单线程

一个进程中有多个执行路径时，这个程序成为多线程

状态：就绪，运行， synchronize 阻塞， wait 和 sleep 挂起，结束。 wait 必须在 synchronized 内部调用。

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

57、介绍 Collection 框架的结构

集合框架：

Collection：List 列表，Set 集

Map：Hashtable，HashMap，TreeMap

59、 ArrayList 和 Vector 的区别

答:

这两个类都实现了 List 接口（ List 接口继承了 Collection 接口），他们都是有序集合，即存储在这两个集合中的元素的位置都是有顺序的，相当于一种动态的数组，我们以后可以按位置索引号取出某个元素，并且其中的数据是允许重复的。

Vector 是线程安全的，也就是说是它的方法之间是线程同步的，而 ArrayList 是线程序不安全的，它的方法之间是线程不同步的。如果只有一个线程会访问到集合，那最好是使用 ArrayList，因为它不考虑线程安全，效率会高些；如果有多个线程会访问到集合，那最好是使用 Vector，因为不需要我们自己再去考虑和编写线程安全的代码。（记住 Vector 与 Hashtable 是旧的，是 java 一诞生就提供了的，它们是线程安全的，ArrayList 与 HashMap 是 java2 时才提供的，它们是线程不安全的。）

ArrayList 与 Vector 都有一个初始的容量大小，当存储进它们里面的元素的个数超过了容量时，就需要增加 ArrayList 与 Vector 的存储空间，每次要增加存储空间时，不是只增加一个存储单元，而是增加多个存储单元，即 Vector 增长原来的一倍， ArrayList 增加原来的 0.5 倍。

60、 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。

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。

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

61、 List 和 Map 区别?

一个是存储单列数据的集合，另一个是存储键和值这样的双列数据的集合， List 中存储的数据是有顺序，并且允许重复；Map 中存储的数据是没有顺序的，其键是不能重复的，它的值是可以有重复的。

62、 List,Set,Map 是否继承自 Collection 接口?

List， Set 是， Map 不是

63、 List、 Map、 Set 三个接口，存取元素时，各有什么特点?

List 与 Set 具有相似性,它们都是单列元素的集合,它们有一个共同的父接口,叫 Collection。

Set 里面不允许有重复的元素

Set 取元素时，没法说取第几个，只能以 Iterator 接口取得所有的元素，再逐一遍历各个元素。

List 表示有先后顺序的集合，可以重复

List 除了可以以 Iterator 接口取得所有的元素，再逐一遍历各个元素之外，还可以调用 get(index i)来明确说明取第几个。

Map 与 List 和 Set 不同，它是双列的集合，其中有 put 方法，定义如下： put(obj key,obj value)，每次存储时，要存储一对 key/value，不能存储重复的 key

取则可以根据 key 获得相应的 value

另外,也可以获得所有的 key 的结合,还可以获得所有的 value 的结合,还可以获得 key 和 value 组合成的 Map.Entry 对象的集合。

List 以特定次序来持有元素，可有重复元素。 Set 无法拥有重复元素,内部排序。 Map 保存 key-value 值， value 可多值。

64、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。

LinkedList 也是线程不安全的， LinkedList 提供了一些方法,使得 LinkedList 可以被当作堆栈和队列来使用。

66、 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。

67、 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，元素重复与否是使用 equals()方法进行判断的。

68、你所知道的集合类都有哪些？主要方法?

最常用的集合类是 List 和 Map。 List 的具体实现包括 ArrayList 和 Vector，它们是可变大小的列表，比较适合构建、存储和操作任何类型对象的元素列表。 List 适用于按数值索引访问元素的情形。

Map 提供了一个更通用的元素存储方法。Map 集合类用于存储元素对(称作"键"和"值"),其中每个键映射到一个值。

它们都有增删改查的方法，但这些方法的具体名称，我记得不是很清楚，对于 set，大概的方法是 add,remove, contains；对于 map，大概的方法就是 put,remove， contains 等
List 类会有 get(int index)这样的方法，因为它可以按顺序取元素，而 set 类中没有 get(int index)这样的方法。
List 和 set 都可以迭代出所有元素,迭代时先要得到一个 iterator 对象,所以，set 和 list 类都有一个 iterator 方法，用于返回那个 iterator 对象。
map 可以返回三个集合，一个是返回所有的 key 的集合，另外一个返回的是所有 value 的集合，再一个返回的 key 和 value 组合成的 EntrySet 对象的集合， map 也有 get 方法，参数是 key，返回值是 key 对应的 value。

69、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?
对。
如果对象要保存在 HashSet 或 HashMap 中，它们的 equals 相等，那么，它们的 hashcode 值就必须相等。
如果不是要保存在 HashSet 或 HashMap，则与 hashcode 没有什么关系了，这时候 hashcode 不等是可以的

71、说出一些常用的类，包，接口，请各举 5 个
常用的类： BufferedReader、BufferedWriter 、FileReader、 FileWirter 、String、 Integer、 java.util.Date、ArrayList、HashMap
常用的包： java.lang 、java.io、 java.util、java.sql、org.apache.strtuts.action、org.hibernate
常用的接口： List、Map、HttpServletRequest、HttpServletResponse、Transaction(Hibernate)、Session(Hibernate)、HttpSession

72、java 中有几种类型的流? JDK 为每种类型的流提供了一些抽象类以供继承，请说出他们分别是哪些类?
字节流，字符流。
字节流继承 InputStream、OutputStream，字符流继承于 InputStreamReader、OutputStreamWriter。在 java.io 包中还有许多其他的流，主要是为了提高性能和使用方便。

73、字节流与字符流的区别
字符流的底层就是字节流。
字符流主要是读取文本文件内容的，可以一个字符一个字符的读取，也可以一行一行的读取文本文件内容。而字节流读取单位为 byte。byte 作为计算机存储最基本单位，可以用字节流来读取很多其他格式的文件，比如图片视频等等。基于 B/S 和 C/S 的文件传输都可以采用字节流的形式。

74、什么是 java 序列化，如何实现 java 序列化? 或者请解释 Serializable 接口的作用。
java 序列化是将一个 java 对象变成字节流的形式传出去或存在硬盘或者从一个字节流中恢复成一个 java 对象
要被传输的对象必须实现 serializable 接口，只是为了标注该对象是可被序列化的。
1 生成你要保存的对象 s1，s2.。。。
2.将对象存入文件。
生成一个对象序列化的对象。

ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("C:\\stu.ser"));
这个对象来保存每一个要保存的对象，最后要存一个空对象。
oos.writeObject(s1);
oos.writeObject(null);
关闭这个序列化对象

3.读取文件中的对象。

生成一个对象反序列化的对象。
ObjectInputStream ois = new ObjectInputStream(new FileInputStream("c:\\stu.ser"));
可以通过循环来用这个对象去读取每一个已保存的对象，读到 null 时终止循环。
ois.readObject();
将读取到的对象强转为他原来的类型
关闭这个反序列化的对象

75、描述一下 JVM 加载 class 文件的原理机制?

JVM 中类的装载是由 ClassLoader 和它的子类来实现的，Java ClassLoader 是一个重要的 Java 运行时系统组件。它是负责在运行时查找和装入类文件的类。

76、heap 和 stack 有什么区别

1.heap 是堆，stack 是栈。

2.stack 的空间由操作系统自动分配和释放，heap 的空间是手动申请和释放的，heap 常用 new 关键字来分配。

3.stack 空间有限，heap 的空间是很大的自由区。在 Java 中，若只是声明一个对象，则先在栈内存中为其分配地址空间，若再 new 一下，实例化它，则在堆内存中为其分配地址。

4.举例：数据类型 变量名；这样定义的东西在栈区。如：Object a =null; 只在栈内存中分配空间。new 数据类型()，或者 malloc(长度); 这样定义的东西就在堆区。如：Object b =new Object(); 则在堆内存中分配空间

77、 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。

78、垃圾回收的优点和原理。并考虑 2 种回收机制。

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，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。

79、垃圾回收器的基本原理是什么？垃圾回收器可以马上回收内存吗？有什么办法主动通知虚拟机进行垃圾回收?

对于 GC 来说，当程序员创建对象时， GC 就开始监控这个对象的地址、大小以及使用情

况。
通常， GC 采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是"可达的"，哪些对象是"不可达的"。当 GC 确定一些对象为"不可达"时， GC 就有责任回收这些内存空间。
程序员可以手动执行 System.gc()，通知 GC 运行，但是 Java 语言规范并不保证 GC 一定会执行。

80.什么是 Java 虚拟机？为什么 Java 被称作是“平台无关的编程语言”？
Java 虚拟机是一个可以执行 Java 字节码的虚拟机进程。Java 源文件被编译成能被 Java 虚拟机执行的字节码文件。
Java 被设计成允许应用程序可以运行在任意的平台，而不需要程序员为每一个平台单独重写或者是重新编译。 Java 虚拟机让这个变为可能，因为它知道底层硬件平台的指令长度和其他特性。

81.JDK 和 JRE 的区别是什么？
Java 运行时环境(JRE)是将要执行 Java 程序的 Java 虚拟机。它同时也包含了执行 applet 需要的浏览器插件。 Java 开发工具包(JDK)是完整的 Java 软件开发包，包含了 JRE，编译器和其他的工具(比如：JavaDoc，Java 调试器)，可以让开发者开发、编译、执行 Java 应用程序。

82.什么是迭代器(Iterator)？
Iterator 接口提供了很多对集合元素进行迭代的方法。每一个集合类都包含了可以返回迭代器实例的迭代方法。迭代器可以在迭代的过程中删除底层集合的元素。

83.Iterator 和 ListIterator 的区别是什么？
Iterator 可用来遍历 Set 和 List 集合，但是 ListIterator 只能用来遍历 List。
Iterator 对集合只能是前向遍历，ListIterator 既可以前向也可以后向。
ListIterator 实现了 Iterator 接口，并包含其他的功能，比如：增加元素，替换元素，获取前一个和后一个元素的索引，等等。

84.hashCode()和 equals()方法的重要性体现在什么地方？
Java 中的 HashMap 使用 hashCode()和 equals()方法来确定键值对的索引，当根据键获取值的时候也会用到这两个方法。如果没有正确的实现这两个方法，两个不同的键可能会有相同的 hash 值，因此，可能会被集合认为是相等的。而且，这两个方法也用来发现重复元素。所以这两个方法的实现对 HashMap 的精确性和正确性是至关重要的。

85.数组(Array)和列表(ArrayList)有什么区别？什么时候应该使用 Array 而不是 ArrayList？
Array 可以包含基本类型和对象类型，ArrayList 只能包含对象类型。
Array 大小是固定的，ArrayList 的大小是动态变化的。
ArrayList 提供了更多的方法和特性，比如：addAll()，removeAll()，iterator()等等。
对于基本类型数据，集合使用自动装箱来减少编码工作量。但是，当处理固定大小的基本数据类型的时候，这种方式相对比较慢。

86.Enumeration 接口和 Iterator 接口的区别有哪些？

Enumeration 速度是 Iterator 的 2 倍，同时占用更少的内存。但是，Iterator 远远比 Enumeration 安全，因为其他线程不能够修改正在被 iterator 遍历的集合里面的对象。 同时，Iterator 允许调用者删除底层集合里面的元素，这对 Enumeration 来说是不可能的。

87.HashSet 和 TreeSet 有什么区别?

HashSet 是由一个 hash 表来实现的，因此，它的元素是无序的。add()， remove()， contains()方法的时间复杂度是 O(1)。

TreeSet 是由一个树形的结构来实现的，它里面的元素是有序的。因此，add()，remove()，contains()方法的时间复杂度是 O(logn)。

88.什么是 JDBC?

JDBC 是允许用户在不同数据库之间做选择的一个抽象层。 JDBC 允许开发者用 JAVA 写数据库应用程序，而不需要关心底层特定数据库的细节。

89.Class.forName()方法有什么作用?

这个方法用来载入跟数据库建立连接的驱动。

90.PreparedStatement 比 Statement 有什么优势?

PreparedStatements 是预编译的，因此，性能会更好。同时，不同的查询参数值，PreparedStatement 可以重用。

PreparedStatement 可以防止 sql 注入。

91.数据库连接池是什么意思?

像打开关闭数据库连接这种和数据库的交互可能是很费时的,尤其是当客户端数量增加的时候，会消耗大量的资源，成本是非常高的。可以在应用服务器启动的时候建立很多个数据库连接并维护在一个池中。连接请求由池中的连接提供。在连接使用完毕以后，把连接归还到池中，以用于满足将来更多的请求。

92.大数据量下的分页解决方法。

最好的办法是利用 sql 语句进行分页,这样每次查询出的结果集中就只包含某页的数据内容。再 sql 语句无法实现分页的情况下，可以考虑对大的结果集通过游标定位方式来获取某页的数据。

sql 语句分页，不同的数据库下的分页方案各不一样，下面是主流的三种数据库的分页 sql:

sql server:

```
String sql =
"select top " + pageSize + " * from students where id not in" +

  "(select top " + pageSize * (pageNumber-1) + " id from students order by id)" +

  "order by id";
```

mysql:

```
String sql =
```

"select * from students order by id limit " + pageSize*(pageNumber-1) + "," + pageSize;

oracle:

```
String sql =
"select * from " +
  (select *,rownum rid from (select * from students order by postime desc) where rid<=" +
pagesize*pagenumber + ") as t" +
        "where t>" + pageSize*(pageNumber-1);
```

92.写一个 jdbc 连接数据库的程序

```
public class DBHelper {
      public static Connection getConnection(){
            Connection con = null;
            try {
                  Class.forName("oracle.jdbc.driver.OracleDriver");
                  con = DriverManager.getConnection(
                        "jdbc:oracle:thin:@localhost:1521:orcl", "user", "pwd");
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return con;
      }
      public static void closeConnection(Connection con){
            try {
                  con.close();
            } catch (Exception e) {
                  e.printStackTrace();
            }
      }
}
```

```
String sql = "select * from    users";
PreparedStatement pstmt = con.prepareStatement(sql);
ResultSet rs = pstmt.executeQuery();
或
count = pstmt.executeUpdate();
```

93．写一个单例模式

特点：1.私有的构造器　　2.私有的静态的当前类对象　　3.公有的静态方法

```
public class Singleton {

      private static Singleton sin =null;
```

```
    private    Singleton() {
    }

    public static Singleton  getInstance() {
        if(sin==null)  {
            stu = new Singleton  ();
        }
        return sin;
    }
}
```