1.什么是Java虚拟机？为什么Java被称作是“平台无关的编程语言”？  
答：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。

2.JDK和JRE的区别是什么？  
答：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。

3.”static”关键字是什么意思？Java中是否可以覆盖(override)一个private或者是static的方法？  
答：“static”关键字表明一个成员变量或者是成员方法可以在没有所属的类的实例变量的情况下被访问。  
Java中static方法不能被覆盖，因为方法覆盖是基于运行时动态绑定的，而static方法是编译时静态绑定的。static方法跟类的任何实例都不相关，所以概念上不适用。

4.是否可以在static环境中访问非static变量？  
答：static变量在Java中是属于类的，它在所有的实例中的值是一样的。当类被Java虚拟机载入的时候，会对static变量进行初始化。如果你的代码尝试不用实例来访问非static的变量，编译器会报错，因为这些变量还没有被创建出来，还没有跟任何实例关联上。

5.Java支持的数据类型有哪些？什么是自动拆装箱？  
答：Java语言支持的8中基本数据类型是：  
byte  
short  
int  
long  
float  
double  
boolean  
char  
自动装箱是Java编译器在基本数据类型和对应的对象包装类型之间做的一个转化。比如：把int转化成Integer，double转化成double，等等。反之就是自动拆箱。

6.Java中的方法覆盖(Overriding)和方法重载(Overloading)是什么意思？  
答：Java中的方法重载发生在同一个类里面两个或者是多个方法的方法名相同但是参数不同的情况。与此相对，方法覆盖是说子类重新定义了父类的方法。方法覆盖必须有相同的方法名，参数列表和返回类型。覆盖者可能不会限制它所覆盖的方法的访问。

7.Java中，什么是构造函数？什么是构造函数重载？什么是复制构造函数？  
答：当新对象被创建的时候，构造函数会被调用。每一个类都有构造函数。在程序员没有给类提供构造函数的情况下，Java编译器会为这个类创建一个默认的构造函数。  
Java中构造函数重载和方法重载很相似。可以为一个类创建多个构造函数。每一个构造函数必须有它自己唯一的参数列表。  
Java不支持像C++中那样的复制构造函数，这个不同点是因为如果你不自己写构造函数的情况下，Java不会创建默认的复制构造函数。

8.Java支持多继承么？  
答：不支持，Java不支持多继承。每个类都只能继承一个类，但是可以实现多个接口。

9.接口和抽象类的区别是什么？  
答：Java提供和支持创建抽象类和接口。它们的实现有共同点，不同点在于：  
接口中所有的方法隐含的都是抽象的。而抽象类则可以同时包含抽象和非抽象的方法。  
类可以实现很多个接口，但是只能继承一个抽象类  
类如果要实现一个接口，它必须要实现接口声明的所有方法。但是，类可以不实现抽象类声明的所有方法，当然，在这种情况下，类也必须得声明成是抽象的。  
抽象类可以在不提供接口方法实现的情况下实现接口。  
Java接口中声明的变量默认都是final的。抽象类可以包含非final的变量。  
Java接口中的成员函数默认是public的。抽象类的成员函数可以是private，protected或者是public。  
接口是绝对抽象的，不可以被实例化。抽象类也不可以被实例化，但是，如果它包含main方法的话是可以被调用的。  
也可以参考JDK8中抽象类和接口的区别

10.什么是值传递和引用传递？  
答：对象被值传递，意味着传递了对象的一个副本。因此，就算是改变了对象副本，也不会影响源对象的值。  
对象被引用传递，意味着传递的并不是实际的对象，而是对象的引用。因此，外部对引用对象所做的改变会反映到所有的对象上。

11.进程和线程的区别是什么？  
答：进程是执行着的应用程序，而线程是进程内部的一个执行序列。一个进程可以有多个线程。线程又叫做轻量级进程。

12.创建线程有几种不同的方式？你喜欢哪一种？为什么？  
答：有三种方式可以用来创建线程：  
继承Thread类  
实现Runnable接口  
应用程序可以使用Executor框架来创建线程池  
实现Runnable接口这种方式更受欢迎，因为这不需要继承Thread类。在应用设计中已经继承了别的对象的情况下，这需要多继承（而Java不支持多继承），只能实现接口。同时，线程池也是非常高效的，很容易实现和使用。

13.概括的解释下线程的几种可用状态。  
答：线程在执行过程中，可以处于下面几种状态：  
就绪(Runnable):线程准备运行，不一定立马就能开始执行。  
运行中(Running)：进程正在执行线程的代码。  
等待中(Waiting):线程处于阻塞的状态，等待外部的处理结束。  
睡眠中(Sleeping)：线程被强制睡眠。  
I/O阻塞(Blocked on I/O)：等待I/O操作完成。  
同步阻塞(Blocked on Synchronization)：等待获取锁。  
死亡(Dead)：线程完成了执行。

14.同步方法和同步代码块的区别是什么？  
答：在Java语言中，每一个对象有一把锁。线程可以使用synchronized关键字来获取对象上的锁。synchronized关键字可应用在方法级别(粗粒度锁)或者是代码块级别(细粒度锁)。

15.在监视器(Monitor)内部，是如何做线程同步的？程序应该做哪种级别的同步？  
答：监视器和锁在Java虚拟机中是一块使用的。监视器监视一块同步代码块，确保一次只有一个线程执行同步代码块。每一个监视器都和一个对象引用相关联。线程在获取锁之前不允许执行同步代码。

16.什么是死锁(deadlock)？  
答：两个进程都在等待对方执行完毕才能继续往下执行的时候就发生了死锁。结果就是两个进程都陷入了无限的等待中。

17.如何确保N个线程可以访问N个资源同时又不导致死锁？  
答：使用多线程的时候，一种非常简单的避免死锁的方式就是：指定获取锁的顺序，并强制线程按照指定的顺序获取锁。因此，如果所有的线程都是以同样的顺序加锁和释放锁，就不会出现死锁了。

18.Java集合类框架的基本接口有哪些？  
答：Java集合类提供了一套设计良好的支持对一组对象进行操作的接口和类。Java集合类里面最基本的接口有：  
Collection：代表一组对象，每一个对象都是它的子元素。  
Set：不包含重复元素的Collection。  
List：有顺序的collection，并且可以包含重复元素。  
Map：可以把键(key)映射到值(value)的对象，键不能重复。

19.为什么集合类没有实现Cloneable和Serializable接口？  
答：集合类接口指定了一组叫做元素的对象。集合类接口的每一种具体的实现类都可以选择以它自己的方式对元素进行保存和排序。有的集合类允许重复的键，有些不允许。

20.什么是迭代器(Iterator)？  
答：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。  
克隆(cloning)或者是序列化(serialization)的语义和含义是跟具体的实现相关的。因此，应该由集合类的具体实现来决定如何被克隆或者是序列化。

21.Iterator和ListIterator的区别是什么？  
答：下面列出了他们的区别：  
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。

22.快速失败(fail-fast)和安全失败(fail-safe)的区别是什么？  
答：Iterator的安全失败是基于对底层集合做拷贝，因此，它不受源集合上修改的影响。java.util包下面的所有的集合类都是快速失败的，而java.util.concurrent包下面的所有的类都是安全失败的。快速失败的迭代器会抛出ConcurrentModificationException异常，而安全失败的迭代器永远不会抛出这样的异常。

23.Java中的HashMap的工作原理是什么？  
答：Java中的HashMap是以键值对(key-value)的形式存储元素的。HashMap需要一个hash函数，它使用hashCode()和equals()方法来向集合/从集合添加和检索元素。当调用put()方法的时候，HashMap会计算key的hash值，然后把键值对存储在集合中合适的索引上。如果key已经存在了，value会被更新成新值。HashMap的一些重要的特性是它的容量(capacity)，负载因子(load factor)和扩容极限(threshold resizing)。

24.hashCode()和equals()方法的重要性体现在什么地方？  
答：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。

25.HashMap和Hashtable有什么区别？  
答：HashMap和Hashtable都实现了Map接口，因此很多特性非常相似。但是，他们有以下不同点：  
HashMap允许键和值是null，而Hashtable不允许键或者值是null。  
Hashtable是同步的，而HashMap不是。因此，HashMap更适合于单线程环境，而Hashtable适合于多线程环境。  
HashMap提供了可供应用迭代的键的集合，因此，HashMap是快速失败的。另一方面，Hashtable提供了对键的列举(Enumeration)。  
一般认为Hashtable是一个遗留的类。

26.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？  
答：下面列出了Array和ArrayList的不同点：  
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。

27.ArrayList和LinkedList有什么区别？  
答：ArrayList和LinkedList都实现了List接口，他们有以下的不同点：  
ArrayList是基于索引的数据接口，它的底层是数组。它可以以O(1)时间复杂度对元素进行随机访问。与此对应，LinkedList是以元素列表的形式存储它的数据，每一个元素都和它的前一个和后一个元素链接在一起，在这种情况下，查找某个元素的时间复杂度是O(n)。  
相对于ArrayList，LinkedList的插入，添加，删除操作速度更快，因为当元素被添加到集合任意位置的时候，不需要像数组那样重新计算大小或者是更新索引。  
LinkedList比ArrayList更占内存，因为LinkedList为每一个节点存储了两个引用，一个指向前一个元素，一个指向下一个元素。  
也可以参考ArrayList vs. LinkedList。

28.Comparable和Comparator接口是干什么的？列出它们的区别。  
答：Java提供了只包含一个compareTo()方法的Comparable接口。这个方法可以个给两个对象排序。具体来说，它返回负数，0，正数来表明输入对象小于，等于，大于已经存在的对象。  
Java提供了包含compare()和equals()两个方法的Comparator接口。compare()方法用来给两个输入参数排序，返回负数，0，正数表明第一个参数是小于，等于，大于第二个参数。equals()方法需要一个对象作为参数，它用来决定输入参数是否和comparator相等。只有当输入参数也是一个comparator并且输入参数和当前comparator的排序结果是相同的时候，这个方法才返回true。

29.什么是Java优先级队列(Priority Queue)？  
答：PriorityQueue是一个基于优先级堆的无界队列，它的元素是按照自然顺序(natural order)排序的。在创建的时候，我们可以给它提供一个负责给元素排序的比较器。PriorityQueue不允许null值，因为他们没有自然顺序，或者说他们没有任何的相关联的比较器。最后，PriorityQueue不是线程安全的，入队和出队的时间复杂度是O(log(n))。

30.你了解大O符号(big-O notation)么？你能给出不同数据结构的例子么？  
答：大O符号描述了当数据结构里面的元素增加的时候，算法的规模或者是性能在最坏的场景下有多么好。  
大O符号也可用来描述其他的行为，比如：内存消耗。因为集合类实际上是数据结构，我们一般使用大O符号基于时间，内存和性能来选择最好的实现。大O符号可以对大量数据的性能给出一个很好的说明。

1.如何权衡是使用无序的数组还是有序的数组？  
答：有序数组最大的好处在于查找的时间复杂度是O(log n)，而无序数组是O(n)。有序数组的缺点是插入操作的时间复杂度是O(n)，因为值大的元素需要往后移动来给新元素腾位置。相反，无序数组的插入时间复杂度是常量O(1)。

2.Java集合类框架的最佳实践有哪些？  
答：根据应用的需要正确选择要使用的集合的类型对性能非常重要，比如：假如元素的大小是固定的，而且能事先知道，我们就应该用Array而不是ArrayList。  
有些集合类允许指定初始容量。因此，如果我们能估计出存储的元素的数目，我们可以设置初始容量来避免重新计算hash值或者是扩容。  
为了类型安全，可读性和健壮性的原因总是要使用泛型。同时，使用泛型还可以避免运行时的ClassCastException。  
使用JDK提供的不变类(immutable class)作为Map的键可以避免为我们自己的类实现hashCode()和equals()方法。  
[编程](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d010c0eea52ad26a&k=%B1%E0%B3%CC&k0=%B1%E0%B3%CC&kdi0=0&luki=4&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=6ad22aa5eec010d0&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4751%2Ehtml&urlid=0)的时候接口优于实现。  
底层的集合实际上是空的情况下，返回长度是0的集合或者是数组，不要返回null。

3.Enumeration接口和Iterator接口的区别有哪些？  
答：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。

4.HashSet和TreeSet有什么区别？  
答：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)。  
另一方面，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)。

5.Java中垃圾回收有什么目的？什么时候进行垃圾回收？  
答：垃圾回收的目的是识别并且丢弃应用不再使用的对象来释放和重用资源。

6.System.gc()和Runtime.gc()会做什么事情？  
答：这两个方法用来提示JVM要进行垃圾回收。但是，立即开始还是延迟进行垃圾回收是取决于JVM的。

7.finalize()方法什么时候被调用？析构函数(finalization)的目的是什么？  
答：在释放对象占用的内存之前，垃圾收集器会调用对象的finalize()方法。一般建议在该方法中释放对象持有的资源。

8.如果对象的引用被置为null，垃圾收集器是否会立即释放对象占用的内存？  
答：不会，在下一个[垃圾回收](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d010c0eea52ad26a&k=%C0%AC%BB%F8%BB%D8%CA%D5&k0=%C0%AC%BB%F8%BB%D8%CA%D5&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=6ad22aa5eec010d0&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4751%2Ehtml&urlid=0)周期中，这个对象将是可被回收的。

9.Java堆的结构是什么样子的？什么是堆中的永久代(Perm Gen space)?  
答：JVM的堆是运行时数据区，所有类的实例和数组都是在堆上分配内存。它在JVM启动的时候被创建。对象所占的堆内存是由自动内存管理系统也就是垃圾收集器回收。  
堆内存是由存活和死亡的对象组成的。存活的对象是应用可以访问的，不会被垃圾回收。死亡的对象是应用不可访问尚且还没有被垃圾收集器回收掉的对象。一直到垃圾收集器把这些对象回收掉之前，他们会一直占据堆内存空间。

10.串行(serial)收集器和吞吐量(throughput)收集器的区别是什么？  
答：吞吐量收集器使用并行版本的新生代垃圾收集器，它用于中等规模和大规模数据的应用程序。而串行收集器对大多数的小应用(在现代处理器上需要大概100M左右的内存)就足够了。

11.在Java中，对象什么时候可以被垃圾回收？  
答：当对象对当前使用这个对象的应用程序变得不可触及的时候，这个对象就可以被回收了。

12.JVM的永久代中会发生垃圾回收么？  
答：[垃圾回收](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d010c0eea52ad26a&k=%C0%AC%BB%F8%BB%D8%CA%D5&k0=%C0%AC%BB%F8%BB%D8%CA%D5&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=6ad22aa5eec010d0&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4751%2Ehtml&urlid=0)不会发生在永久代，如果永久代满了或者是超过了临界值，会触发完全垃圾回收(Full GC)。如果你仔细查看垃圾收集器的输出信息，就会发现永久代也是被回收的。这就是为什么正确的永久代大小对避免Full GC是非常重要的原因。请参考下Java8：从永久代到元数据区  
(译者注：Java8中已经移除了永久代，新加了一个叫做元数据区的native内存区)

13.Java中的两种异常类型是什么？他们有什么区别？  
答：Java中有两种异常：受检查的(checked)异常和不受检查的(unchecked)异常。不受检查的异常不需要在方法或者是构造函数上声明，就算方法或者是构造函数的执行可能会抛出这样的异常，并且不受检查的异常可以传播到方法或者是构造函数的外面。相反，受检查的异常必须要用throws语句在方法或者是构造函数上声明。这里有Java异常处理的一些小建议。

14.Java中Exception和Error有什么区别？  
答：Exception和Error都是Throwable的子类。Exception用于用户程序可以捕获的异常情况。Error定义了不期望被用户程序捕获的异常。

15.throw和throws有什么区别？  
答：throw关键字用来在程序中明确的抛出异常，相反，throws语句用来表明方法不能处理的异常。每一个方法都必须要指定哪些异常不能处理，所以方法的调用者才能够确保处理可能发生的异常，多个异常是用逗号分隔的。

16.异常处理的时候，finally代码块的重要性是什么？(译者注：作者标题的序号弄错了)  
答：无论是否抛出异常，finally代码块总是会被执行。就算是没有catch语句同时又抛出异常的情况下，finally代码块仍然会被执行。最后要说的是，finally代码块主要用来释放资源，比如：I/O缓冲区，数据库连接。

17.异常处理完成以后，Exception对象会发生什么变化？  
答：Exception对象会在下一个垃圾回收过程中被回收掉。

18.finally代码块和finalize()方法有什么区别？  
答：无论是否抛出异常，finally代码块都会执行，它主要是用来释放应用占用的资源。finalize()方法是Object类的一个protected方法，它是在对象被[垃圾回收](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d010c0eea52ad26a&k=%C0%AC%BB%F8%BB%D8%CA%D5&k0=%C0%AC%BB%F8%BB%D8%CA%D5&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=6ad22aa5eec010d0&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4751%2Ehtml&urlid=0)之前由Java虚拟机来调用的。

19.什么是Applet？  
答：java applet是能够被包含在HTML页面中并且能被启用了java的客户端浏览器执行的程序。Applet主要用来创建动态交互的web应用程序。

20.解释一下Applet的生命周期  
答：applet可以经历下面的状态：  
Init：每次被载入的时候都会被初始化。  
Start：开始执行applet。  
Stop：结束执行applet。  
Destroy：卸载applet之前，做最后的清理工作。

21.当applet被载入的时候会发生什么？  
答：首先，创建applet控制类的实例，然后初始化applet，最后开始运行。

22.Applet和普通的Java应用程序有什么区别？  
答：applet是运行在启用了[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d010c0eea52ad26a&k=java&k0=java&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=6ad22aa5eec010d0&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4751%2Ehtml&urlid=0)的浏览器中，Java应用程序是可以在浏览器之外运行的独立的Java程序。但是，它们都需要有Java虚拟机。  
进一步来说，Java应用程序需要一个有特定方法签名的main函数来开始执行。Java applet不需要这样的函数来开始执行。  
最后，Java applet一般会使用很严格的安全策略，Java应用一般使用比较宽松的安全策略。

23.Java applet有哪些限制条件？  
答：主要是由于安全的原因，给applet施加了以下的限制：  
applet不能够载入类库或者定义本地方法。  
applet不能在宿主机上读写文件。  
applet不能读取特定的系统属性。  
applet不能发起网络连接，除非是跟宿主机。  
applet不能够开启宿主机上其他任何的程序。

24.什么是不受信任的applet？  
答：不受信任的applet是不能访问或是执行本地系统文件的Java applet，默认情况下，所有下载的applet都是不受信任的。

25.从网络上加载的applet和从本地文件系统加载的applet有什么区别？  
答：当applet是从网络上加载的时候，applet是由applet类加载器载入的，它受applet安全管理器的限制。  
当applet是从客户端的本地磁盘载入的时候，applet是由文件系统加载器载入的。  
从文件系统载入的applet允许在客户端读文件，写文件，加载类库，并且也允许执行其他程序，但是，却通不过字节码校验。

26.applet类加载器是什么？它会做哪些工作？  
答：当applet是从网络上加载的时候，它是由applet类加载器载入的。类加载器有自己的[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d010c0eea52ad26a&k=java&k0=java&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=6ad22aa5eec010d0&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4751%2Ehtml&urlid=0)名称空间等级结构。类加载器会保证来自文件系统的类有唯一的名称空间，来自网络资源的类有唯一的名称空间。  
当浏览器通过网络载入applet的时候，applet的类被放置于和applet的源相关联的私有的名称空间中。然后，那些被类加载器载入进来的类都是通过了验证器验证的。验证器会检查类文件格式是否遵守Java语言规范，确保不会出现堆栈溢出(stack overflow)或者下溢(underflow)，传递给字节码指令的参数是正确的。

27.applet安全管理器是什么？它会做哪些工作？  
答：applet安全管理器是给applet施加限制条件的一种机制。浏览器可以只有一个安全管理器。安全管理器在启动的时候被创建，之后不能被替换覆盖或者是扩展。

28.弹出式选择菜单(Choice)和列表(List)有什么区别  
答：Choice是以一种紧凑的形式展示的，需要下拉才能看到所有的选项。Choice中一次只能选中一个选项。List同时可以有多个元素可见，支持选中一个或者多个元素。

29.什么是布局管理器？  
答：布局管理器用来在容器中组织组件。

30.滚动条(Scrollbar)和滚动面板(JScrollPane)有什么区别？  
答：Scrollbar是一个[组件](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d010c0eea52ad26a&k=%D7%E9%BC%FE&k0=%D7%E9%BC%FE&kdi0=0&luki=2&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=6ad22aa5eec010d0&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4751%2Ehtml&urlid=0)，不是容器。而ScrollPane是容器。ScrollPane自己处理滚动事件。

1.哪些Swing的方法是线程安全的？  
答：只有3个线程安全的方法： repaint(), revalidate(), and invalidate()。

2.说出三种支持重绘(painting)的[组件](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=%D7%E9%BC%FE&k0=%D7%E9%BC%FE&kdi0=0&luki=2&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0)。  
答：Canvas, Frame, Panel,和Applet支持重绘。

3.什么是裁剪(clipping)？  
答：限制在一个给定的区域或者形状的绘图操作就做裁剪。

4.MenuItem和CheckboxMenuItem的区别是什么？  
答：CheckboxMenuItem类继承自MenuItem类，支持菜单选项可以选中或者不选中。

5.边缘布局(BorderLayout)里面的元素是如何布局的？  
答：BorderLayout里面的元素是按照容器的东西南北中进行布局的。

6.网格包布局(GridBagLayout)里面的元素是如何布局的？  
答：GridBagLayout里面的元素是按照网格进行布局的。不同大小的元素可能会占据网格的多于1行或一列。因此，行数和列数可以有不同的大小。

7.Window和Frame有什么区别？  
答：Frame类继承了Window类，它定义了一个可以有菜单栏的主应用窗口。

8.裁剪(clipping)和重绘(repainting)有什么联系？  
答：当窗口被AWT重绘线程进行重绘的时候，它会把裁剪区域设置成需要重绘的窗口的区域。

9.事件监听器接口(event-listener interface)和事件适配器(event-adapter)有什么关系？  
答：事件监听器接口定义了对特定的事件，事件处理器必须要实现的方法。事件适配器给事件监听器接口提供了默认的实现。

10.GUI组件如何来处理它自己的事件？  
答：GUI组件可以处理它自己的事件，只要它实现相对应的事件监听器接口，并且把自己作为事件监听器。

11.Java的布局管理器比传统的窗口系统有哪些优势？  
答：Java使用布局管理器以一种一致的方式在所有的窗口平台上摆放组件。因为布局管理器不会和组件的绝对大小和位置相绑定，所以他们能够适应跨窗口系统的特定平台的不同。

12.Java的Swing组件使用了哪种设计模式？  
答：Java中的Swing[组件](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=%D7%E9%BC%FE&k0=%D7%E9%BC%FE&kdi0=0&luki=2&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0)使用了MVC(视图-模型-控制器)设计模式。

13.什么是JDBC？  
答：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。

14.解释下驱动(Driver)在JDBC中的角色。  
答：JDBC驱动提供了特定厂商对JDBC API接口类的实现，驱动必须要提供[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=java&k0=java&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0).sql包下面这些类的实现：Connection, Statement, PreparedStatement,CallableStatement, ResultSet和Driver。

15.Class.forName()方法有什么作用？  
答：这个方法用来载入跟[数据库](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=%CA%FD%BE%DD%BF%E2&k0=%CA%FD%BE%DD%BF%E2&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0)建立连接的驱动。

16.PreparedStatement比Statement有什么优势？  
答：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。

17.什么时候使用CallableStatement？用来准备CallableStatement的方法是什么？  
答：CallableStatement用来执行存储过程。存储过程是由数据库存储和提供的。存储过程可以接受输入参数，也可以有返回结果。非常鼓励使用存储过程，因为它提供了安全性和模块化。准备一个CallableStatement的方法是：

18.数据库连接池是什么意思？  
答：像打开关闭[数据库](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=%CA%FD%BE%DD%BF%E2&k0=%CA%FD%BE%DD%BF%E2&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0)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。

19.什么是RMI？  
答：Java远程方法调用(Java RMI)是Java API对远程过程调用(RPC)提供的面向对象的等价形式，支持直接传输序列化的Java对象和分布式[垃圾回收](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=%C0%AC%BB%F8%BB%D8%CA%D5&k0=%C0%AC%BB%F8%BB%D8%CA%D5&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0)。远程方法调用可以看做是激活远程正在运行的对象上的方法的步骤。RMI对调用者是位置透明的，因为调用者感觉方法是执行在本地运行的对象上的。看下RMI的一些注意事项。

20.RMI体系结构的基本原则是什么？  
答：RMI体系结构是基于一个非常重要的行为定义和行为实现相分离的原则。RMI允许定义行为的代码和实现行为的代码相分离，并且运行在不同的JVM上。

21.RMI体系结构分哪几层？  
答：RMI体系结构分以下几层：  
存根和骨架层(Stub and Skeleton layer)：这一层对程序员是透明的，它主要负责拦截客户端发出的方法调用请求，然后把请求重定向给远程的RMI服务。  
远程引用层(Remote Reference Layer)：RMI体系结构的第二层用来解析客户端对服务端远程对象的引用。这一层解析并管理客户端对服务端远程对象的引用。连接是点到点的。  
传输层(Transport layer)：这一层负责连接参与服务的两个JVM。这一层是建立在网络上机器间的TCP/IP连接之上的。它提供了基本的连接服务，还有一些防火墙穿透策略。

22.RMI中的远程接口(Remote Interface)扮演了什么样的角色？  
答：远程接口用来标识哪些方法是可以被非本地虚拟机调用的接口。远程对象必须要直接或者是间接实现远程接口。实现了远程接口的类应该声明被实现的远程接口，给每一个远程对象定义构造函数，给所有远程接口的方法提供实现。

23.java.rmi.Naming类扮演了什么样的角色？  
答：[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=java&k0=java&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0).rmi.Naming类用来存储和获取在远程对象注册表里面的远程对象的引用。Naming类的每一个方法接收一个URL格式的String对象作为它的参数。

24.RMI的绑定(Binding)是什么意思？  
答：绑定是为了查询找远程对象而给远程对象关联或者是注册以后会用到的名称的过程。远程对象可以使用Naming类的bind()或者rebind()方法跟名称相关联。

25.Naming类的bind()和rebind()方法有什么区别？  
答：bind()方法负责把指定名称绑定给远程对象，rebind()方法负责把指定名称重新绑定到一个新的远程对象。如果那个名称已经绑定过了，先前的绑定会被替换掉。

26.让RMI程序能正确运行有哪些步骤？  
答：为了让RMI程序能正确运行必须要包含以下几个步骤：  
编译所有的源文件。  
使用rmic生成stub。  
启动rmiregistry。  
启动RMI服务器。  
运行客户端程序。

27.RMI的stub扮演了什么样的角色？  
答：远程对象的stub扮演了远程对象的代表或者代理的角色。调用者在本地stub上调用方法，它负责在远程对象上执行方法。当stub的方法被调用的时候，会经历以下几个步骤：  
初始化到包含了远程对象的JVM的连接。  
序列化参数到远程的JVM。  
等待方法调用和执行的结果。  
反序列化返回的值或者是方法没有执行成功情况下的异常。  
把值返回给调用者。

28.什么是分布式垃圾回收(DGC)？它是如何工作的？  
答：DGC叫做分布式[垃圾回收](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=d8e9ef39babeae8e&k=%C0%AC%BB%F8%BB%D8%CA%D5&k0=%C0%AC%BB%F8%BB%D8%CA%D5&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=8eaebeba39efe9d8&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4753%2Ehtml&urlid=0)。RMI使用DGC来做自动垃圾回收。因为RMI包含了跨虚拟机的远程对象的引用，垃圾回收是很困难的。DGC使用引用计数算法来给远程对象提供自动内存管理。

29.RMI中使用RMI安全管理器(RMISecurityManager)的目的是什么？  
答：RMISecurityManager使用下载好的代码提供可被RMI应用程序使用的安全管理器。如果没有设置安全管理器，RMI的类加载器就不会从远程下载任何的类。

30.解释下Marshalling和demarshalling。  
答：当应用程序希望把内存对象跨网络传递到另一台主机或者是持久化到存储的时候，就必须要把对象在内存里面的表示转化成合适的格式。这个过程就叫做Marshalling，反之就是demarshalling。

1.解释下Serialization和Deserialization。  
答:Java提供了一种叫做对象序列化的机制，他把对象表示成一连串的字节，里面包含了对象的数据，对象的类型信息，对象内部的数据的类型信息等等。因此，序列化可以看成是为了把对象存储在磁盘上或者是从磁盘上读出来并重建对象而把对象扁平化的一种方式。反序列化是把对象从扁平状态转化成活动对象的相反的步骤。

2.什么是Servlet？  
答:Servlet是用来处理客户端请求并产生动态网页内容的Java类。Servlet主要是用来处理或者是存储HTML表单提交的数据，产生动态内容，在无状态的HTTP协议下管理状态信息。

3.说一下Servlet的体系结构。  
答:所有的Servlet都必须要实现的核心的接口是javax.servlet.Servlet。每一个Servlet都必须要直接或者是间接实现这个接口，或者是继承javax.servlet.GenericServlet或者javax.servlet.http.HTTPServlet。最后，Servlet使用多线程可以并行的为多个请求服务。

4.Applet和Servlet有什么区别？  
答:Applet是运行在客户端主机的浏览器上的客户端Java程序。而Servlet是运行在web服务器上的服务端的组件。applet可以使用用户界面类，而Servlet没有用户界面，相反，Servlet是等待客户端的HTTP请求，然后为请求产生响应。

5.GenericServlet和HttpServlet有什么区别？  
答:GenericServlet是一个通用的协议无关的Servlet，它实现了Servlet和ServletConfig接口。继承自GenericServlet的Servlet应该要覆盖service()方法。最后，为了开发一个能用在网页上服务于使用HTTP协议请求的Servlet，你的Servlet必须要继承自HttpServlet。这里有Servlet的例子。

6.解释下Servlet的生命周期。  
答:对每一个客户端的请求，Servlet引擎载入Servlet，调用它的init()方法，完成Servlet的初始化。然后，Servlet对象通过为每一个请求单独调用service()方法来处理所有随后来自客户端的请求，最后，调用Servlet(译者注：这里应该是Servlet而不是server)的destroy()方法把Servlet删除掉。

7.doGet()方法和doPost()方法有什么区别？  
答:doGet：GET方法会把名值对追加在请求的URL后面。因为URL对字符数目有限制，进而限制了用在客户端请求的参数值的数目。并且请求中的参数值是可见的，因此，敏感信息不能用这种方式传递。  
doPOST：POST方法通过把请求参数值放在请求体中来克服GET方法的限制，因此，可以发送的参数的数目是没有限制的。最后，通过POST请求传递的敏感信息对外部客户端是不可见的。

8.什么是Web应用程序？  
答:Web应用程序是对Web或者是应用服务器的动态扩展。有两种类型的Web应用：面向表现的和面向服务的。面向表现的Web应用程序会产生包含了很多种标记语言和动态内容的交互的web页面作为对请求的响应。而面向服务的Web应用实现了Web服务的端点(endpoint)。一般来说，一个Web应用可以看成是一组安装在服务器URL名称空间的特定子集下面的Servlet的集合。

9.什么是服务端包含(Server Side Include)？  
答:服务端包含(SSI)是一种简单的解释型服务端脚本语言，大多数时候仅用在Web上，用servlet标签嵌入进来。SSI最常用的场景把一个或多个文件包含到Web服务器的一个Web页面中。当浏览器访问Web页面的时候，Web服务器会用对应的servlet产生的文本来替换Web页面中的servlet标签。

10.什么是Servlet链(Servlet Chaining)？  
答:Servlet链是把一个Servlet的输出发送给另一个Servlet的方法。第二个Servlet的输出可以发送给第三个Servlet，依次类推。链条上最后一个Servlet负责把响应发送给客户端。

11.如何知道是哪一个客户端的机器正在请求你的Servlet？  
答:ServletRequest类可以找出客户端机器的IP地址或者是主机名。getRemoteAddr()方法获取客户端主机的IP地址，getRemoteHost()可以获取主机名。看下这里的例子。

12.HTTP响应的结构是怎么样的？  
答:HTTP响应由三个部分组成：  
状态码(Status Code)：描述了响应的状态。可以用来检查是否成功的完成了请求。请求失败的情况下，状态码可用来找出失败的原因。如果Servlet没有返回状态码，默认会返回成功的状态码HttpServletResponse.SC\_OK。  
HTTP头部(HTTP Header)：它们包含了更多关于响应的信息。比如：头部可以指定认为响应过期的过期日期，或者是指定用来给用户安全的传输实体内容的编码格式。如何在Serlet中检索HTTP的头部看这里。  
主体(Body)：它包含了响应的内容。它可以包含HTML代码，图片，等等。主体是由传输在HTTP消息中紧跟在头部后面的数据字节组成的。

13.什么是cookie？session和cookie有什么区别？  
答:cookie是Web服务器发送给浏览器的一块信息。浏览器会在本地文件中给每一个Web服务器存储cookie。以后浏览器在给特定的Web服务器发请求的时候，同时会发送所有为该服务器存储的cookie。下面列出了session和cookie的区别：  
无论客户端浏览器做怎么样的设置，session都应该能正常工作。客户端可以选择禁用cookie，但是，session仍然是能够工作的，因为客户端无法禁用服务端的session。  
在存储的数据量方面session和cookies也是不一样的。session能够存储任意的Java对象，cookie只能存储String类型的对象。

14.浏览器和Servlet通信使用的是什么协议？  
答:浏览器和Servlet通信使用的是HTTP协议。

15.什么是HTTP隧道？  
答:HTTP隧道是一种利用HTTP或者是HTTPS把多种网络协议封装起来进行通信的技术。因此，HTTP协议扮演了一个打通用于通信的网络协议的管道的包装器的角色。把其他协议的请求掩盖成HTTP的请求就是HTTP隧道。

16.sendRedirect()和forward()方法有什么区别？  
答:sendRedirect()方法会创建一个新的请求，而forward()方法只是把请求转发到一个新的目标上。重定向(redirect)以后，之前请求作用域范围以内的对象就失效了，因为会产生一个新的请求，而转发(forwarding)以后，之前请求作用域范围以内的对象还是能访问的。一般认为sendRedirect()比forward()要慢。

17.什么是URL编码和URL解码？  
答:URL编码是负责把URL里面的空格和其他的特殊字符替换成对应的十六进制表示，反之就是解码。

18.什么是JSP页面？  
答:JSP页面是一种包含了静态数据和JSP元素两种类型的文本的文本文档。静态数据可以用任何基于文本的格式来表示，比如：HTML或者XML。JSP是一种混合了静态内容和动态产生的内容的技术。这里看下JSP的例子。

19.JSP请求是如何被处理的？  
答:浏览器首先要请求一个以.jsp扩展名结尾的页面，发起JSP请求，然后，Web服务器读取这个请求，使用JSP编译器把JSP页面转化成一个Servlet类。需要注意的是，只有当第一次请求页面或者是JSP文件发生改变的时候JSP文件才会被编译，然后服务器调用servlet类，处理浏览器的请求。一旦请求执行结束，servlet会把响应发送给客户端。这里看下如何在JSP中获取请求参数。

20.JSP有什么优点？  
答:下面列出了使用JSP的优点：  
JSP页面是被动态编译成Servlet的，因此，开发者可以很容易的更新展现代码。  
JSP页面可以被预编译。  
JSP页面可以很容易的和静态模板结合，包括：HTML或者XML，也可以很容易的和产生动态内容的代码结合起来。  
开发者可以提供让页面设计者以类XML格式来访问的自定义的JSP标签库。  
开发者可以在组件层做逻辑上的改变，而不需要编辑单独使用了应用层逻辑的页面。

21.什么是JSP指令(Directive)？JSP中有哪些不同类型的指令？  
答:Directive是当JSP页面被编译成Servlet的时候，JSP引擎要处理的指令。Directive用来设置页面级别的指令，从外部文件插入数据，指定自定义的标签库。Directive是定义在<%@ 和 %>之间的。下面列出了不同类型的Directive：  
包含指令(Include directive)：用来包含文件和合并文件内容到当前的页面。  
页面指令(Page directive)：用来定义JSP页面中特定的属性，比如错误页面和缓冲区。  
Taglib指令： 用来声明页面中使用的自定义的标签库。

22.什么是JSP动作(JSP action)？  
答:JSP动作以XML语法的结构来控制Servlet引擎的行为。当JSP页面被请求的时候，JSP动作会被执行。它们可以被动态的插入到文件中，重用JavaBean组件，转发用户到其他的页面，或者是给Java插件产生HTML代码。下面列出了可用的动作：  
jsp:include-当JSP页面被请求的时候包含一个文件。  
jsp:useBean-找出或者是初始化Javabean。  
jsp:setProperty-设置JavaBean的属性。  
jsp:getProperty-获取JavaBean的属性。  
jsp:forward-把请求转发到新的页面。  
jsp:plugin-产生特定浏览器的代码。

23.什么是Scriptlets？  
答:JSP技术中，scriptlet是嵌入在JSP页面中的一段Java代码。scriptlet是位于标签内部的所有的东西，在标签与标签之间，用户可以添加任意有效的scriplet。

24.声明(Decalaration)在哪里？  
答:声明跟Java中的变量声明很相似，它用来声明随后要被表达式或者scriptlet使用的变量。添加的声明必须要用开始和结束标签包起来。

25.什么是表达式(Expression)？  
答:【列表很长，可以分上、中、下发布】  
JSP表达式是Web服务器把脚本语言表达式的值转化成一个String对象，插入到返回给客户端的数据流中。表达式是在<%=和%>这两个标签之间定义的。

26.隐含对象是什么意思？有哪些隐含对象？  
答:JSP隐含对象是页面中的一些Java对象，JSP容器让这些Java对象可以为开发者所使用。开发者不用明确的声明就可以直接使用他们。JSP隐含对象也叫做预定义变量。下面列出了JSP页面中的隐含对象：  
application  
page  
request  
response  
session  
exception  
out  
config  
pageContext

1、面向对象的特征有哪些方面? 【基础】  
答：面向对象的特征主要有以下几个方面：  
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](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=225f0cf50b9f4ec0&k=%C3%E6%CF%F2%B6%D4%CF%F3&k0=%C3%E6%CF%F2%B6%D4%CF%F3&kdi0=0&luki=3&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=c04e9f0bf50c5f22&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4758%2Ehtml&urlid=0)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、作用域public,private,protected,以及不写时的区别？【基础】  
答：区别如下：  
作用域当前类同包子孙类其他  
public √ √ √ √  
protected √ √ √ ×  
default √ √ × ×  
private √ × × ×  
不写时默认为default。

3、String 是最基本的数据类型吗? 【基础】  
答：不是。

4、float 型float f=3.4是否正确? 【基础】  
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 。

5、语句float f=1.3；编译能否通过？【基础】  
答:不能；应该用强制类型转换，如下所示：float f=(float)1.3； 。

6、short s1 = 1; s1 = s1 + 1;有什么错?short s1 = 1; s1 += 1;有什么错? 【基础】  
答：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,自动类型提升。

7、Java 有没有goto? 【基础】  
答：goto 是java 中的保留字，现在没有在java 中使用。

8、int 和Integer 有什么区别? 【基础】  
答：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](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=225f0cf50b9f4ec0&k=java&k0=java&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=c04e9f0bf50c5f22&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4758%2Ehtml&urlid=0) 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: boolean,char,byte,short,int,long,float,double  
封装类型：Boolean，Character，Byte，Short，Integer，Long，Float，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。

9、&和&&的区别？【基础】  
答：&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

10、简述逻辑操作(&,|,^)与条件操作(&&,||)的区别？【基础】  
答：区别主要有两点：a.条件操作只能操作布尔型的,而逻辑操作不仅可以操作布尔型,而且可以操作数值型b.逻辑操作不会产生短路。

11、heap 和stack 有什么区别？【基础】  
答：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。

12、Math.round(11.5) 等于多少? Math.round(-11.5)等于多少? 【基础】  
答：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。

13、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? 【基础】  
答：[switch](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=225f0cf50b9f4ec0&k=switch&k0=switch&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=c04e9f0bf50c5f22&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4758%2Ehtml&urlid=0)（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。

14、[编程](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=225f0cf50b9f4ec0&k=%B1%E0%B3%CC&k0=%B1%E0%B3%CC&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=c04e9f0bf50c5f22&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4758%2Ehtml&urlid=0)题: 用最有效率的方法算出2 乘以8 等於几? 【基础】  
答： 2 << 3。

15、有没有length()这个方法? String 有没有length()这个方法？【基础】  
答：数组没有length()这个方法，有length 的属性。String 有length()这个方法。

16、在JAVA 中，如何跳出当前的多重嵌套循环？【基础】  
答：在最外层循环前加label 标识,然后用break:label 方法即可跳出多重循环。

17、构造器Constructor 是否可被override? 【基础】  
答：构造器Constructor 不能被继承，因此不能重写Overriding，但可以被重载Overloading。

18、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? 【基础】  
答：不对，有相同的hash code。

19、是否可以继承String 类? 【基础】  
答：String 类是final 类，故不可以继承。

20、以下二条语句返回值为true 的有：  
A：“beijing”==“beijing”；  
B：“beijing”.equalsIgnoreCase（new String（“beijing”））；【基础】  
答：A 和B 。

21、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? 【基础】  
答：是值传递。Java [编程语言](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=225f0cf50b9f4ec0&k=%B1%E0%B3%CC%D3%EF%D1%D4&k0=%B1%E0%B3%CC%D3%EF%D1%D4&kdi0=0&luki=2&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=c04e9f0bf50c5f22&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4758%2Ehtml&urlid=0)只有值传递参数。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。

22、我们在web 应用开发过程中经常遇到输出某种编码的字符，如iso8859-1等，如何输出一个某种编码的字符串？【基础】  
答：public String translate(String str){  
String tempStr = “”;  
try{  
tempStr = new String(str.getBytes(“ISO-8859-1″), “GBK”);  
tempStr = tempStr.trim();  
}catch (Exception e){  
System.err.println(e.getMessage());  
}  
return tempStr;  
}

23、String 和StringBuffer 的区别? 【基础】  
答：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。

24、String, StringBuffer StringBuilder 的区别。【基础】  
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；线程安全；StringBuilder 是从JDK 5 开始，为StringBuffer 该类补充了一个单个线程使用的等价类；通常应该优先使用StringBuilder 类，因为它支持所有相同的操作，但由于它不执行同步，所以速度更快。

25、Overload 和Override 的区别。Overloaded 的方法是否可以改变返回值的类型? 【基础】  
答：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、定义类A 和类B 如下：【基础】  
class A {  
int a=1;  
double d=2.0;  
void show(){  
System.out.println(“Class A: a=”+a +”\td=”+d);  
}  
}  
class B extends A{  
float a=3.0f;  
String d=”Java program.”;  
void show(){  
super.show( );  
System.out.println(“Class B: a=”+a +”\td=”+d);  
}  
}  
(1) 若在应用程序的main 方法中有以下语句：  
A a=new A();  
a.show();  
则输出的结果如何？  
(2) 若在应用程序的main 方法中定义类B 的对象b：  
A b=new B();  
b.show();  
则输出的结果如何？  
答：输出结果为：  
1）Class A: a=1 d=2.0 ；  
2）Class A: a=1 d=2.0  
Class B: a=3.0 d=Java program。

27、描述一下JVM 加载class 文件的原理机制? 【基础】  
答：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。它负责在运行时查找和装入类文件的类。

28、char 型变量中能不能存贮一个中文汉字?为什么? 【基础】  
答：能够定义成为一个中文的，因为[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=225f0cf50b9f4ec0&k=java&k0=java&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=c04e9f0bf50c5f22&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4758%2Ehtml&urlid=0) 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。

29、abstract class 和interface 有什么区别? 【基础】  
答：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。新型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，所有成员变量都是publicstatic final 的。一个类可以实现多个接口，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。

30、Static Nested Class 和Inner Class 的不同？【基础】  
答：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。

1、java 中会存在内存泄漏吗，请简单描述。【基础】  
答：会；存在无用但可达的对象，这些对象不能被GC 回收，导致耗费内存资源。

2、abstract 的method 是否可同时是static,是否可同时是native，是否可同时是synchronized? 【基础】  
答：都不能。

3、静态变量和实例变量的区别？【基础】  
答：静态变量也称为类变量，归全类共有，它不依赖于某个对象，可通过类名直接访问；而实例变量必须依存于某一实例，只能通过对象才能访问到它。

4、是否可以从一个static 方法内部发出对非static 方法的调用？【基础】  
答：不可以,如果其中包含对象的method()，不能保证对象初始化。

5、写clone()方法时，通常都有一行代码，是什么？【基础】  
答：Clone 有缺省行为：super.clone()，他负责产生正确大小的空间，并逐位复制。

6、GC 是什么? 为什么要有GC? 【基础】  
答：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。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() 或Runtime.getRuntime().gc() 。

7、垃圾回收的优点和原理。并考虑2 种回收机制。【基础】  
答：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。

8、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？【基础】  
答：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。可以。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。

9、String s=new String(“xyz”);创建了几个String Object? 【基础】  
答：两个对象，一个是”xyx”,一个是指向”xyx”的引用对象s。

10、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)? 【基础】  
答：接口可以继承接口。抽象类可以实现(implements)接口，抽象类可继承实体类，但前提是实体类必须有明确的构造函数。

11、Java 的接口和C++的虚类的相同和不同处。【基础】  
答：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。

12、一个“.java”源文件中是否可以包含多个类（不是内部类）？有什么限制？【基础】  
答：可以；必须只有一个类名与文件名相同。

13、说出一些常用的类，包，接口，请各举5 个。【基础】  
答：常用的类：BufferedReader BufferedWriter FileReader FileWirter String Integer；  
常用的包：java.lang java.awt java.io java.util java.sql；  
常用的接口：Remote List Map Document NodeList

14、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)? 【基础】  
答：可以继承其他类或实现其他接口，在swing 编程中常用此方式。

15、内部类可以引用他包含类的成员吗？有没有什么限制？【基础】  
答：一个内部类对象可以访问创建它的外部类对象的内容。

16、java 中实现多态的机制是什么？【基础】  
答：方法的覆盖Overriding 和重载Overloading 是java 多态性的不同表现；覆盖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。

17、在java 中一个类被声明为final 类型，表示了什么意思？【基础】  
答：表示该类不能被继承，是顶级类。

18、下面哪些类可以被继承? 【基础】  
1）java.lang.Thread (T)  
2）java.lang.Number (T)  
3）java.lang.Double (F)  
4）java.lang.Math (F)  
5）java.lang.Void (F)  
6）java.lang.Class (F)  
7）java.lang.ClassLoader (T)  
答：1、2、7 可以被继承。

19、指出下面程序的运行结果: 【基础】  
class A{  
static{  
System.out.print(“1″);  
}  
public A(){  
System.out.print(“2″);  
}  
}  
class B extends A{  
static{  
System.out.print(“a”);  
}  
public B(){  
System.out.print(“b”);  
}  
}  
public class Hello{  
public static void main(String[] ars){  
A ab = new B(); //执行到此处,结果: 1a2b  
ab = new B(); //执行到此处,结果: 1a2b2b  
}  
}  
答：输出结果为1a2b2b；类的static 代码段,可以看作是类首次加载(虚拟机加载)执行的代码,而对于类加载,首先要执行其基类的构造,再执行其本身的构造。

20、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?【基础】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();  
}  
}  
答：输出结果为：  
FatherClass Create  
FatherClass Create  
ChildClass Create

21、内部类的实现方式? 【基础】  
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();  
}  
}  
输出结果为:  
InterClass Create  
OuterClass Create

22、关于内部类： 【基础】  
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();  
}  
答：答案为C、E；说明如下：  
1）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；  
2）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；故D 错；  
3）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。

23、数据类型之间的转换:  
1)如何将数值型字符转换为数字？  
2)如何将数字转换为字符？  
3)如何取小数点前两位并四舍五入? 【基础】  
答：1)调用数值类型相应包装类中的方法parse\*\*\*(String)或valueOf(String)即可返回相应基本类型或包装类型数值；  
2)将数字与空字符串相加即可获得其所对应的字符串;另外对于基本类型数字还可调用String 类中的valueOf(…)方法返回相应字符串,而对于包装类型数字则可调用其toString()方法获得相应字符串；  
3)可用该数字构造一java.math.BigDecimal 对象,再利用其round()方法进行四舍五入到保留小数点后两位,再将其转换为字符串截取最后两位。

24、字符串操作：如何实现字符串的反转及替换？【基础】  
答：可用字符串构造一StringBuffer 对象,然后调用StringBuffer 中的reverse方法即可实现字符串的反转,调用replace 方法即可实现字符串的替换。

25、编码转换：怎样将GB2312 编码的字符串转换为ISO-8859-1 编码的字符串？【基础】  
答：示例代码如下:  
String s1 = “你好”;  
String s2 = new String(s1.getBytes(“GB2312″), “ISO-8859-1″);

26、写一个函数，要求输入一个字符串和一个字符长度，对该字符串进行分隔。【基础】  
答：函数代码如下：  
public String[] split(String str, int chars){  
int n = (str.length()+ chars – 1)/chars;  
String ret[] = new String[n];  
for(int i=0; i<n; i++){  
if(i < n-1){  
ret[i] = str.substring(i\*chars , (i+1)\*chars);  
}else{  
ret[i] = str.substring(i\*chars);  
}  
}  
return ret;  
}

27、写一个函数，2 个参数，1 个字符串，1 个字节数，返回截取的字符串，要求字符串中的中文不能出现乱码：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”。【基础】  
答：代码如下：  
public String subString(String str, int subBytes) {  
int bytes = 0; // 用来存储字符串的总字节数  
for (int i = 0; i < str.length(); i++) {  
if (bytes == subBytes) {  
return str.substring(0, i);  
}  
char c = str.charAt(i);  
if (c < 256) {  
bytes += 1; // 英文字符的字节数看作1  
} else {  
bytes += 2; // 中文字符的字节数看作2  
if(bytes – subBytes == 1){  
return str.substring(0, i);  
}  
}  
}  
return str;  
}

28、日期和时间：  
1)如何取得年月日、小时分秒？  
2)如何取得从1970 年到现在的毫秒数？  
3)如何取得某个日期是当月的最后一天？  
4)如何格式化日期？【基础】  
答：1)创建java.util.Calendar 实例(Calendar.getInstance()),调用其get()  
方法传入不同的参数即可获得参数所对应的值,如：  
第12 页共59 页  
calendar.get(Calendar.YEAR);//获得年  
2)以下方法均可获得该毫秒数:  
Calendar.getInstance().getTimeInMillis();  
System.currentTimeMillis();  
3)示例代码如下:  
Calendar time = Calendar.getInstance();  
time.set(Calendar.DAY\_OF\_MONTH,  
time.getActualMaximum(Calendar.DAY\_OF\_MONTH));  
4)利用java.text.DataFormat 类中的format()方法可将日期格式化。

29、Java 编程,打印昨天的当前时刻。【基础】  
答：public class YesterdayCurrent{  
public static void main(String[] args){  
Calendar cal = Calendar.getInstance();  
cal.add(Calendar.DATE, -1);  
System.out.println(cal.getTime());  
}  
}

30、java 和javasciprt 的区别。【基础】  
答：JavaScript 与Java 是两个公司开发的不同的两个产品。Java 是SUN 公司推出的新一代面向对象的程序设计语言，特别适合于Internet 应用程序开发；而JavaScript 是Netscape 公司的产品，其目的是为了扩展Netscape Navigator功能,而开发的一种可以嵌入Web 页面中的基于对象和事件驱动的解释性语言,它的前身是Live Script；而Java 的前身是Oak 语言。下面对两种语言间的异同作如下比较：  
1）基于对象和面向对象：  
Java 是一种真正的面向对象的语言，即使是开发简单的程序，必须设计对象；JavaScript 是种脚本语言，它可以用来制作与网络无关的，与用户交互作用的复杂软件。它是一种基于对象（Object Based）和事件驱动（Event Driver）的编程语言。因而它本身提供了非常丰富的内部对象供设计人员使用；  
2）解释和编译：  
Java 的源代码在执行之前，必须经过编译；JavaScript 是一种解释性编程语言，其源代码不需经过编译，由浏览器解释执行；  
3）强类型变量和类型弱变量：  
Java 采用强类型变量检查，即所有变量在编译之前必须作声明；JavaScript 中变量声明，采用其弱类型。即变量在使用前不需作声明，而是解释器在运行时检查其数据类型；  
4）代码格式不一样。

1、什么时候用assert？【中等难度】  
答：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。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。在实现中，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。断言用于调试目的：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。  
要在系统类中启用断言，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。异常部分：（共8 题：基础8 道）

2、Java 中的异常处理机制的简单原理和应用？【基础】  
答：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。

3、error 和exception 有什么区别? 【基础】  
答：error 表示系统级的错误和程序不必处理的异常，是恢复不是不可能但很困难的情况下的一种严重问题；比如内存溢出，不可能指望程序能处理这样的情况；exception 表示需要捕捉或者需要程序进行处理的异常，是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。

4、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? 【基础】  
答：会执行，在return 前执行。

5、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？【基础】  
答：Java 通过面向对象的方法进行异常处理，把各种不同的异常进行分类，并提供了良好的接口。在Java 中，每个异常都是一个对象，它是Throwable 类或其它子类的实例。当一个方法出现异常后便抛出一个异常对象，该对象中包含有异常信息，调用这个对象的方法可以捕获到这个异常并进行处理。Java 的异常处理是通过5 个关键词来实现的：try、catch、throw、throws 和finally。一般情况下是用try 来执行一段程序，如果出现异常，系统会抛出（throws）一个异常，这时候你可以通过它的类型来捕捉（catch）它，或最后（finally）由缺省处理器来处理；  
try 用来指定一块预防所有“异常”的程序；  
catch 子句紧跟在try 块后面，用来指定你想要捕捉的“异常”的类型；  
throw 语句用来明确地抛出一个“异常”；  
throws 用来标明一个成员函数可能抛出的各种“异常”；  
Finally 为确保一段代码不管发生什么“异常”都被执行一段代码；  
可以在一个成员函数调用的外面写一个try 语句，在这个成员函数内部写另一个try 语句保护其他代码。每当遇到一个try 语句，“异常”的框架就放到堆栈上面，直到所有的try 语句都完成。如果下一级的try 语句没有对某种“异常”进行处理，堆栈就会展开，直到遇到有处理这种“异常”的try 语句。

6、运行时异常与一般异常有何异同？【基础】  
答：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。

7、给我一个你最常见到的runtime exception？【基础】  
答：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

8、final, finally, finalize 的区别? 【基础】  
答：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。

9、类Example A 继承Exception，类ExampleB 继承Example A；【基础】  
有如下代码片断：  
try{  
throw new ExampleB(“b”)；  
}catch（ExampleA e）{  
System.out.printfln（“ExampleA”）；  
}catch（Exception e）{  
System.out.printfln（“Exception”）；  
}  
输出的内容应该是：  
A：ExampleA B：Exception C：b D：无  
答：输出为A。  
集合部分：（共11 题：基础11 道）

10、介绍JAVA 中的Collection FrameWork(及如何写自己的数据结构)【基础】  
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。

11、List,Set,Map 是否继承自Collection 接口？【基础】  
答：List,Set 是；Map 不是。

12、你所知道的集合类都有哪些？主要方法？【基础】  
答：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。

13、说出ArrayList,Vector, LinkedList 的存储性能和特性？【基础】  
答：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。

14、Collection 和Collections 的区别？【基础】  
答：Collection 是java.util 下的接口，它是各种集合的父接口，继承于它的接口主要有Set 和List；Collections 是个java.util 下的类，是针对集合的帮助类，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。

15、HashMap 和Hashtable 的区别? 【基础】  
答：二者都实现了Map 接口，是将惟一键映射到特定的值上；主要区别在于：  
1)HashMap 没有排序，允许一个null 键和多个null 值,而Hashtable 不允许；  
2)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；  
3)Hashtable 继承自Dictionary 类，HashMap 是Java1.2 引进的Map 接口的实现；  
4)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 算法大致一样，所以性能不会有很大的差异。

16、Arraylist 与Vector 区别？【基础】  
答：就ArrayList 与Vector 主要从二方面来说：  
1）同步性：Vector 是线程安全的（同步），而ArrayList 是线程序不安全的；  
2）数据增长：当需要增长时,Vector 默认增长一倍，而ArrayList 却是一半。

17、List、Map、Set 三个接口，存取元素时，各有什么特点？【基础】  
答：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。

18、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? 【基础】  
答：Set 里的元素是不能重复的，用equals ()方法来区分重复与否。覆盖equals()方法用来判断对象的内容是否相同，而”==”判断地址是否相等,用来决定引用值是否指向同一对象。

19、用程序给出随便大小的10 个数，序号为1-10，按从小到大顺序输出，并输出相应的序号。【基础】  
答：代码如下：  
package test;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Iterator;  
import java.util.List;  
import java.util.Random;  
public class RandomSort {  
public static void printRandomBySort() {  
Random random = new Random(); // 创建随机数生成器  
List list = new ArrayList();  
// 生成10 个随机数，并放在集合list 中  
for (int i = 0; i < 10; i++) {  
list.add(random.nextInt(1000));  
}  
Collections.sort(list); // 对集合中的元素进行排序  
Iterator it = list.iterator();  
int count = 0;  
while (it.hasNext()) { // 顺序输出排序后集合中的元素  
System.out.println(++count + “: ” + it.next());  
}  
}  
public static void main(String[] args) {  
printRandomBySort();  
}  
}

20、用JAVA 实现一种排序，JAVA 类实现序列化的方法？ 在COLLECTION 框架中，实现比较要实现什么样的接口？【基础】  
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(){  
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。

线程部分：（共10 题：基础7 道，中等难度3 道）  
21、sleep() 和wait() 有什么区别? 【基础】  
答：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。

22、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? 【基础】  
答：其它线程只能访问该对象的其它非同步方法，同步方法则不能进入。

23、请说出你所知道的线程同步的方法。【基础】  
答：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；  
notityAll():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。

24、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? 【基础】  
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。

25、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。【基础】  
答：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。

26、启动一个线程是用run()还是start()?【基础】  
答：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。

27、线程的基本概念、线程的基本状态以及状态之间的关系？【基础】  
答：线程指在程序执行过程中，能够执行程序代码的一个执行单位，每个程序至少都有一个线程，也就是程序本身；Java 中的线程有四种状态分别是：运行、就绪、挂起、结束。

28、简述synchronized 和java.util.concurrent.locks.Lock 的异同？【中等难度】  
答：主要相同点：Lock 能完成synchronized 所实现的所有功能；主要不同点：Lock 有比synchronized 更精确的线程语义和更好的性能。synchronized 会自动释放锁，而Lock 一定要求程序员手工释放，并且必须在finally 从句中释放。

29、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？【中等难度】  
答：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。它会解除由线程获取的所有锁定，而且如果对象处于一种不连贯状态，那么其他线程能在那种状态下检查和修改它们。结果很难检查出真正的问题所在；  
suspend()方法容易发生死锁。调用suspend()的时候，目标线程会停下来，但却仍然持有在这之前获得的锁定。此时，其他任何线程都不能访问锁定的资源，除非被“挂起”的线程恢复运行。对任何线程来说，如果它们想恢复目标线程，同时又试图使用任何一个锁定的资源，就会造成死锁。故不应该使用suspend()，  
而应在自己的Thread 类中置入一个标志，指出线程应该活动还是挂起。若标志指出线程应该挂起，便用wait()命其进入等待状态。若标志指出线程应当恢复，则用一个notify()重新启动线程。

30、设计4 个线程，其中两个线程每次对j 增加1，另两个线程对j 每次减少1;写出程序。【中等难度】  
答：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 TestThread {  
private int j;  
public TestThread(int j) {this.j = j;}  
private synchronized void inc(){  
j++;  
System.out.println(j + “–Inc–” +  
Thread.currentThread().getName());  
}  
private synchronized void dec(){  
j–;  
System.out.println(j + “–Dec–” +  
Thread.currentThread().getName());  
}  
public void run() {  
(new Dec()).start();  
new Thread(new Inc()).start();  
(new Dec()).start();  
new Thread(new Inc()).start();  
}  
class Dec extends Thread {  
public void run() {  
for(int i=0; i<100; i++){  
dec();  
}  
}  
}  
class Inc implements Runnable {  
public void run() {  
for(int i=0; i<100; i++){  
inc();  
}  
}  
}  
public static void main(String[] args) {  
(new TestThread(5)).run();  
}  
}

IO 流及Socket 部分：（共5 题：基础5 道）

1、什么是java 序列化，如何实现java 序列化？【基础】  
答：序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题；序列化的实现：将需要被序列化的类实现Serializable 接口，该接口没有需实现的方法，implements Serializable 只是为了标注该对象是可被序列化的，然后使用一个输出流(如FileOutputStream)来构造一个ObjectOutputStream(对象流)对象，接着，使用ObjectOutputStream 对象的writeObject(Object obj)方法就可以将参数为obj 的对象写出(即保存其状态)，要恢复的话则用输入流。

2、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？【基础】  
答：字节流，字符流。字节流继承于InputStream、OutputStream，字符流继承于Reader、Writer。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。

3、文件和目录（IO）操作：  
1)如何列出某个目录下的所有文件？  
2)如何列出某个目录下的所有子目录？  
3)如何判断一个文件或目录是否存在？  
4)如何读写文件？【基础】  
答：1)示例代码如下:  
File file = new File(“e:\\总结”);  
File[] files = file.listFiles();  
for(int i=0; i<files.length; i++){  
if(files[i].isFile()) System.out.println(files[i]);  
}  
2)示例代码如下:  
File file = new File(“e:\\总结”);  
File[] files = file.listFiles();  
for(int i=0; i<files.length; i++){  
if(files[i].isDirectory()) System.out.println(files[i]);  
}  
3)创建File 对象,调用其exsit()方法即可返回是否存在,如:  
第22 页共59 页  
System.out.println(new File(“d:\\t.txt”).exists());  
4)示例代码如下:  
//读文件:  
FileInputStream fin = new FileInputStream(“e:\\tt.txt”);  
byte[] bs = new byte[100];  
while(true){  
int len = fin.read(bs);  
if(len <= 0) break;  
System.out.print(new String(bs,0,len));  
}  
fin.close();  
//写文件:  
FileWriter fw = new FileWriter(“e:\\test.txt”);  
fw.write(“hello world!” + System.getProperty(“line.separator”));  
fw.write(“你好！北京！”);  
fw.close();

4、写一个方法,输入一个文件名和一个字符串,统计这个字符串在这个文件中出现的次数。【基础】  
答：代码如下：  
public int countWords(String file, String find) throws Exception{  
int count = 0;  
Reader in = new FileReader(file);  
int c;  
while ((c = in.read()) != -1) {  
while (c == find.charAt(0)) {  
for (int i = 1; i < find.length(); i++) {  
c = in.read();  
if (c != find.charAt(i)) break;  
if (i == find.length() – 1) count++;  
}  
}  
}  
return count;  
}

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？【基础】  
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();  
}  
}

OOA/D 与UML 部分：（共6 题：基础2 道，中等难度4 道）  
6、UML 是什么？常用的几种图？【基础】  
答：UML 是标准建模语言；常用图包括：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。

7、编程题: 写一个Singleton 出来。【基础】  
答：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。举例：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。  
第一种形式:  
public class Singleton {  
private Singleton(){}  
private static Singleton instance = new Singleton();  
public static Singleton getInstance(){  
return instance;  
}  
}  
第二种形式:  
public class Singleton {  
private static Singleton instance = null;  
public static synchronized Singleton getInstance(){  
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。

8、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。【中等难度】  
答：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。

9、Java 中常用的设计模式？说明工厂模式？【中等难度】  
答：Java 中的23 种设计模式：Factory（ 工厂模式），Builder（ 建造模式）， Factory Method（工厂方法模式），Prototype（原始模型模式），Singleton（单例模式），Facade（门面模式），Adapter（适配器模式）， Bridge（桥梁模式）， Composite（合成模式），Decorator（装饰模式）， Flyweight（享元模式）， Proxy（代理模式），Command（命令模式）， Interpreter（解释器模式）， Visitor（访问者模式），Iterator（迭代子模式）， Mediator（调停者模式）， Memento（备忘录模式），Observer（观察者模式），State（状态模式），Strategy（策略模式），Template Method（模板方法模式）， Chain Of Responsibleity（责任链模式）。  
工厂模式：工厂模式是一种经常被使用到的模式，根据工厂模式实现的类可以根据提供的数据生成一组类中某一个类的实例，通常这一组类有一个公共的抽象父类并且实现了相同的方法，但是这些方法针对不同的数据进行了不同的操作。首先需要定义一个基类，该类的子类通过不同的方法实现了基类中的方法。然后需要定义一个工厂类，工厂类可以根据条件生成不同的子类实例。当得到子类的实例后，开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子类的实例。

10、开发中都用到了那些设计模式?用在什么场合? 【中等难度】  
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；及简单工厂模式等。

11、你对软件开发中迭代的含义的理解；【中等难度】  
答：软件开发中，各个开发阶段不是顺序执行的，应该是并行执行,也就是迭代的意思。这样对于开发中的需求变化，及人员变动都能得到更好的适应。

XML 部分：（共4 题：基础1 道，中等难度1 道，较难2 道）  
12、XML 文档定义有几种形式？它们之间有何本质区别？解析XML 文档有哪几种方式？【基础】  
答：1）两种形式：dtd 以及schema；  
2）本质区别：schema 本身是xml 的，可以被XML 解析器解析(这也是从DTD  
上发展schema 的根本目的)；  
3）解析方式：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:不同于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)。

13、你在项目中用到了xml 技术的哪些方面?如何实现的? 【中等难度】  
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。

14、用jdom 解析xml 文件时如何解决中文问题?如何解析? 【较难】  
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());  
}  
}  
}

15、编程用JAVA 解析XML 的方式。【较难】  
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>

四、数据库及SQL 部分：（共4 题：基础3 道，中等难度1 道）  
16、有3 个表（15 分钟）：【基础】  
Student 学生表(学号，姓名，性别，年龄，组织部门)  
Course 课程表(编号，课程名称)  
Sc 选课表(学号，课程编号，成绩)  
表结构如下：  
1）写一个SQL 语句，查询选修了’计算机原理’的学生学号和姓名（3 分钟）  
2）写一个SQL 语句，查询’周星驰’同学选修了的课程名字（3 分钟）  
3）写一个SQL 语句，查询选修了5 门课程的学生学号和姓名（9 分钟）  
答：1）SQL 语句如下：  
select stu.sno, stu.sname from Student stu  
where (select count(\*) from sc where sno=stu.sno and cno =  
(select cno from Course where cname=’计算机原理’)) != 0;  
2）SQL 语句如下：  
select cname from Course  
where cno in ( select cno from sc where sno =  
(select sno from Student where sname=’周星驰’));  
3）SQL 语句如下：  
select stu.sno, stu.sname from student stu  
where (select count(\*) from sc where sno=stu.sno) = 5;

17、有三张表,学生表S,课程C,学生课程表SC,学生可以选修多门课程,一门课程可以被多个学生选修,通过SC 表关联。【基础】  
1）写出建表语句；  
2）写出SQL 语句,查询选修了所有选修课程的学生；  
3）写出SQL 语句,查询选修了至少5 门以上的课程的学生。  
答：1）建表语句如下（mysql 数据库）：  
create table s(id integer primary key, name varchar(20));  
create table c(id integer primary key, name varchar(20));  
create table sc(  
sid integer references s(id),  
cid integer references c(id),  
primary key(sid,cid)  
);  
2）SQL 语句如下：  
select stu.id, stu.name from s stu  
where (select count(\*) from sc where sid=stu.id)  
= (select count(\*) from c);  
3）SQL 语句如下：  
select stu.id, stu.name from s stu  
where (select count(\*) from sc where sid=stu.id)>=5;

18、数据库表(Test)结构如下：【基础】  
ID NAME AGE MANAGER(所属主管人ID)  
106 A 30 104  
109 B 19 104  
104 C 20 111  
107 D 35 109  
112 E 25 120  
119 F 45 NULL  
要求:列出所有年龄比所属主管年龄大的人的ID 和名字?  
答：SQL 语句如下：  
select employee.name from test employee  
where employee.age > (select manager.age from test manager  
where manager.id=employee.manager);

19、有如下两张表：【中等难度】  
表city： 表state：  
CityNo CityName StateNo  
BJ 北京（Null）  
SH 上海（Null）  
GZ 广州GD  
DL 大连LN  
欲得  
到如下结果： City  
No City Name State No State Name BJ  
北京（Null） （Null） DL  
大连LN 辽宁GZ  
广州GD 广东SH  
上海（Null） （Null） 写相  
应的SQL 语句。  
答：SQL 语句为：  
SELECT C.CITYNO, C.CITYNAME, C.STATENO, S.STATENAME  
FROM CITY C, STATE S  
WHERE C.STATENO=S.STATENO(+)  
ORDER BY(C.CITYNO);

五、JDBC 及Hibernate:（共12 题：基础10 道，中等难度2 道）  
20、数据库，比如100 用户同时来访，要采取什么技术解决？【基础】  
答：可采用连接池。

21、什么是ORM？【基础】  
答：对象关系映射（Object—Relational Mapping，简称ORM）是一种为了解决面向对象与面向关系数据库存在的互不匹配的现象的技术；简单的说，ORM 是通过使用描述对象和数据库之间映射的元数据，将java 程序中的对象自动持久化到关系数据库中；本质上就是将数据从一种形式转换到另外一种形式。

22、Hibernate 有哪5 个核心接口？【基础】  
答：Configuration 接口：配置Hibernate，根据其启动hibernate，创建  
SessionFactory 对象；  
SessionFactory 接口：初始化Hibernate，充当数据存储源的代理，创建session 对象，sessionFactory 是线程安全的，意味着它的同一个实例可以被应用的多个线程共享，是重量级、二级缓存；  
Session 接口：负责保存、更新、删除、加载和查询对象，是线程不安全的，避免多个线程共享同一个session，是轻量级、一级缓存；  
Transaction 接口：管理事务；  
Query 和Criteria 接口：执行数据库的查询。

23、关于hibernate: 【基础】  
1）在hibernate 中，在配置文件呈标题一对多，多对多的标签是什么；  
2）Hibernate 的二级缓存是什么；  
3）Hibernate 是如何处理事务的；  
答：1）一对多的标签为<one-to-many> ；多对多的标签为<many-to-many>；  
2）sessionFactory 的缓存为hibernate 的二级缓存；  
3）Hibernate 的事务实际上是底层的JDBC Transaction 的封装或者是JTA  
Transaction 的封装；默认情况下使用JDBCTransaction。  
State  
No  
State  
Name  
GD 广东  
LN 辽宁  
SD 山东  
NMG 内蒙古

24、Hibernate 的应用（Hibernate 的结构）？【基础】  
答：//首先获得SessionFactory 的对象  
SessionFactory sessionFactory = new Configuration().configure().  
buildSessionFactory();  
//然后获得session 的对象  
Session session = sessionFactory.openSession();  
//其次获得Transaction 的对象  
Transaction tx = session.beginTransaction();  
//执行相关的数据库操作:增,删,改,查  
session.save(user); //增加, user 是User 类的对象  
session.delete(user); //删除  
session.update(user); //更新  
Query query = session.createQuery(“from User”); //查询  
List list = query.list();  
//提交事务  
tx.commit();  
//如果有异常,我们还要作事务的回滚,恢复到操作之前  
tx.rollback();  
//最后还要关闭session,释放资源  
session.close();

25、什么是重量级？什么是轻量级？【基础】  
答：轻量级是指它的创建和销毁不需要消耗太多的资源，意味着可以在程序中经常创建和销毁session 的对象；重量级意味不能随意的创建和销毁它的实例，会占用很多的资源。

26、数据库的连接字符串？【基础】  
答：MS SQL Server  
//第二种连接方式  
Class.forName(“com.microsoft.jdbc.sqlserver.SQLServerDriver”).  
newInstance();  
conn = DriverManager.getConnection(“jdbc:Microsoft:sqlserver  
://localhost:1433;DatabaseName=pubs”,”sa”,””);  
//Oracle  
Class.forName(“oracle.jdbc.driver.OracleDriver”).newInstance();  
conn = DriverManager.getConnection(“jdbc:oracle:thin:  
@localhost:1521:sid”, uid, pwd);  
//Mysql  
Class.forName(“org.git.mm.mysql.Driver”).newInstance();  
conn = DriverManager.getConnection(“jdbc:mysql  
://localhost:3306/pubs”,”root”,””);  
处理中文的问题:  
jdbc:mysql://localhost:3306/pubs?useUnicode=true  
&characterEncoding=GB2312

27、事务处理？【基础】  
答：Connection 类中提供了3 个事务处理方法：  
setAutoCommit(Boolean autoCommit):设置是否自动提交事务，默认为自动提交事务，即为true，通过设置false 禁止自动提交事务；  
commit():提交事务；  
rollback():回滚事务。

28、Java 中访问数据库的步骤？Statement 和PreparedStatement 之间的区别？【基础】  
答：Java 中访问数据库的步骤如下：  
1）注册驱动；  
2）建立连接；  
3）创建Statement；  
4）执行sql 语句；  
5）处理结果集（若sql 语句为查询语句）；  
6）关闭连接。  
PreparedStatement 被创建时即指定了SQL 语句，通常用于执行多次结构相同的SQL 语句。

29、用你熟悉的语言写一个连接ORACLE 数据库的程序,能够完成修改和查询工作。【基础】  
答：JDBC 示例程序如下：  
public void testJdbc(){  
Connection con = null;  
PreparedStatement ps = null;  
ResultSet rs = null;  
try{  
//step1：注册驱动；  
Class.forName(“oracle.jdbc.driver.OracleDriver”);  
//step 2：获取数据库连接；  
con=DriverManager.getConnection(  
“jdbc:oracle:thin:@192.168.0.39:1521:TARENADB”,  
“sd0605″,”sd0605″);  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*查询\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
//step 3：创建Statement；  
String sql = “SELECT id, fname, lname, age, FROM  
Person\_Tbl”;  
ps = con.prepareStatement(sql);  
//step 4 ：执行查询语句，获取结果集；  
rs = ps.executeQuery();  
//step 5：处理结果集—输出结果集中保存的查询结果；  
while (rs.next()){  
System.out.print(“id = ” + rs.getLong(“id”));  
System.out.print(” , fname = ” +rs.getString(“fname”));  
System.out.print(” , lname = ” +rs.getString(“lname”));  
System.out.print(” , age = ” + rs.getInt(“age”));  
}  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JDBC 修改\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
sql = “UPDATE Person\_Tbl SET age=23 WHERE id = ?”;  
ps = con.prepareStatement(sql);  
ps.setLong(1, 88);  
int rows = ps.executeUpdate();  
System.out.println(rows + ” rows affected.”);  
} catch (Exception e){  
e.printStackTrace();  
} finally{  
try{  
con.close(); //关闭数据库连接，以释放资源。  
} catch (Exception e1) {  
}  
}  
}

30、JDBC，Hibernate 分页怎样实现？【中等难度】  
答：方法分别为：  
1) Hibernate 的分页：  
Query query = session.createQuery(“from Student”);  
query.setFirstResult(firstResult);//设置每页开始的记录号  
query.setMaxResults(resultNumber);//设置每页显示的记录数  
Collection students = query.list();  
2) JDBC 的分页：根据不同的数据库采用不同的sql 分页语句  
例如: Oracle 中的sql 语句为: “SELECT \* FROM (SELECT a.\*, rownum r FROM  
TB\_STUDENT) WHERE r between 2 and 10″ 查询从记录号2 到记录号10 之间的所有记录

Web 部分：（共题：基础40 道，基础37 道，中等难度3 道）  
1、说出Servlet 的生命周期，并说出Servlet 和CGI 的区别? 【基础】  
答：Web 容器加载Servlet 并将其实例化后，Servlet 生命周期开始，容器运行其init 方法进行Servlet 的初始化，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](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)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。

2、Servlet 的基本架构。【基础】  
答：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 {  
}  
}

3、forward 和redirect 的区别? 【基础】  
答：forward 是容器中控制权的转向，是[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)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，并且从浏览器的地址栏中可以看到跳转后的链接地址。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、JSP 中动态INCLUDE 与静态INCLUDE 的区别？【基础】  
答：动态INCLUDE 用[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=jsp&k0=jsp&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0):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,它不会检查所含文件的变化，适用于包含静态页面<%@ include file=”included.htm” %>

5、说出数据连接池的工作机制是什么? 【基础】  
答：J2EE [服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其表记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接表记为空闲，其他调用就可以使用这个连接。

6、JSP 的内置对象及方法？【基础】  
答：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 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 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 表示一个[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=java&k0=java&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)x.servlet.[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=jsp&k0=jsp&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0).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 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 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 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 表示从该页面产生的一个servlet 实例。

7、JSP 的常用指令？【基础】  
答：<%@page language=”java” contenType=”text/html;charset=gb2312”  
session=”true” buffer=”64kb” autoFlush=”true” isThreadSafe=”true”  
info=”text” errorPage=”error.[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=jsp&k0=jsp&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)” isErrorPage=”true” isELIgnored=”  
true” pageEncoding=”gb2312” import=”[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=java&k0=java&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0).sql.\*”%>  
isErrorPage：是否能使用Exception 对象；isELIgnored：是否忽略EL 表达式；  
<%@include file=”filename”%>  
<%@taglib prefix=”c”uri=”http://……”%>

8、jsp 有哪些动作?作用分别是什么? 【基础】  
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。

9、[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=jsp&k0=jsp&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0) 有哪些内置对象?作用分别是什么? 【基础】  
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。

10、get 和post 的区别？【基础】  
答：Form 中的get 和post 方法，在数据传输过程中分别对应了HTTP 协议中的  
GET 和POST 方法。二者主要区别如下：  
1）Get 是用来从[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)上获得数据，而Post 是用来向服务器上传递数据；  
2）Get 将表单中数据按照variable=value 的形式，添加到action 所指向的URL 后面，并且两者使用“?”连接，而各个变量之间使用“&”连接；Post 是将表单中的数据放在form 的数据体中，按照变量和值相对应的方式，传递到action所指向URL；  
3）Get 是不安全的，因为在传输过程，数据被放在请求的URL 中；Post 的所有操作对用户来说都是不可见的；  
4）Get 传输的数据量小，这主要是因为受URL 长度限制；而Post 可以传输大量的数据，所以在上传文件只能使用Post；  
5）Get 限制Form 表单的数据集必须为ASCII 字符，而Post 支持整个ISO10646字符集；  
6）Get 是Form 的默认方法。

11、什么情况下调用doGet()和doPost()？【基础】  
答：Jsp 页面中的form 标签里的method 属性为get 时调用doGet()，为post时调用doPost()。

12、如何从form 表单中得取checkbox 的值；【基础】  
答：可在页面把checkbox 的name 属性取同一个，value 属性取每个条目的id,后台用getParamter(“name”)能取到checkbox 的一组值。

13、页面中有一个命名为bank No 的下拉列表，写脚本获取当前选项的索引值。【基础】  
答：用java 或javaScript 的处理方式分别如下：  
Java：  
request.getParameter(“bank No”);  
javaScript：  
var selectItems = document.getElementsByName(“bank No”);  
selectItems[0].value;

14、[java](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=java&k0=java&kdi0=0&luki=6&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)script 常用的方面；【基础】  
答：常用于数据输入校验和页面特殊效果等。

15、常用的web 容器和开发工具；【基础】  
答：最常用的容器包括：tomcat、weblogic；  
[开发工具](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%BF%AA%B7%A2%B9%A4%BE%DF&k0=%BF%AA%B7%A2%B9%A4%BE%DF&kdi0=0&luki=3&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)有：eclipse,jbuilder。

16、JSP 和Servlet 有哪些相同点和不同点，他们之间的联系是什么？【基础】  
答：JSP 是Servlet 技术的扩展，本质上是Servlet 的简易方式，更强调应用的外表表达。JSP 编译后是”类servlet”。Servlet 和JSP 最主要的不同点在于，Servlet 的应用逻辑是在Java 文件中，并且完全从表示层中的HTML 里分离开来。而JSP 的情况是Java 和HTML 可以组合成一个扩展名为.jsp 的文件。JSP 侧重于视图，Servlet 主要用于控制逻辑。

17、[jsp](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=jsp&k0=jsp&kdi0=0&luki=5&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0) 的四种范围？【基础】  
答：a.page 是代表与一个页面相关的对象和属性。一个页面由一个编译好的Java servlet 类（可以带有任何的include 指令，但是没有include 动作）表示。这既包括servlet 又包括被编译成servlet 的JSP 页面  
b.request 是代表与Web 客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面，涉及多个Web 组件（由于forward 指令和include 动作的关系）  
c.session 是代表与用于某个Web 客户机的一个用户体验相关的对象和属性。一个Web 会话可以也经常会跨越多个客户机请求  
d.application 是代表与整个Web 应用程序相关的对象和属性。这实质上是跨越整个Web 应用程序，包括多个页面、请求和会话的一个全局作用域。

18、Request 对象的主要方法? 【基础】  
答：setAttribute(String name,Object)：设置名字为name 的属性值  
getAttribute(String name)：返回由name 指定的属性值  
getAttributeNames()：返回request 对象所有属性的名字集合(枚举)  
getCookies()：返回客户端的所有Cookie 对象，结果是一个Cookie 数组  
getCharacterEncoding()：返回请求中的字符编码方式  
getContentLength()：返回请求的Body 的长度  
getHeader(String name)：获得HTTP 协议定义的文件头信息  
getHeaders(String name)：返回指定名的request Header 的所有值(枚举)  
getHeaderNames()：返回所有request Header 的名字(枚举)  
getInputStream()：返回请求的输入流，用于获得请求中的数据  
getMethod()：获得客户端向[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)端传送数据的方法  
getParameter(String name)：获得客户端请求中传送的name 指定的参数值  
getParameterNames()：获得客户端传送给服务器端的所有参数的名字(枚举)  
getParameterValues(String name)：获得有name 指定的参数的所有值  
getProtocol()：获取客户端向服务器端传送数据所依据的协议名称  
getQueryString()：获得查询字符串  
getRequestURI()：获取发出请求字符串的客户端地址  
getRemoteAddr()：获取客户端的IP 地址  
getRemoteHost()：获取客户端的名字  
getSession([Boolean create])：返回和请求相关Session  
getServerName()：获取服务器的名字  
getServletPath()：获取客户端所请求的脚本文件的路径  
getServerPort()：获取[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)的端口号  
removeAttribute(String name)：删除请求中的一个属性

19、如何实现servlet 的单线程模式？【基础】  
答：<%@page isThreadSafe=”false”%>

20、页面间对象传递的方法。【基础】  
答：request,session,application,cookie 等。

21、详细描述MVC。【基础】  
答：基于Java 的Web 应用系统采用MVC 架构模式，即model（模型）、view（视图）、control（控制）分离设计；这是目前WEB 应用服务系统的主流设计方向。  
Model：即处理业务逻辑的模块，每一种处理一个模块；  
View：负责页面显示，显示MODEL 处理结果给用户，主要实现数据到页面转换过程；  
Control：负责每个请求的分发，把FORM 数据传递给MODEL 处理，把处理结果的数据传递给VIEW 显示。

22、MVC 的各个部分都有那些技术来实现?如何实现? 【基础】  
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。

23、应用[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)有那些？【基础】  
答：BEA WebLogic Server, IBM WebSphere Application Server, Oracle9i,Application Server, JBoss, Tomcat。

24、Servlet 执行时一般实现哪几个方法？【基础】  
答：public void init(ServletConfig config)  
public ServletConfig getServletConfig()  
public String getServletInfo()  
public void service(ServletRequest request,ServletResponse response)  
public void destroy()

25、struts 的入口类？【基础】  
答：是ActionServlet，所有的[struts](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=struts&k0=struts&kdi0=0&luki=4&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0) 请求都经由该类转发处理。

26、STRUTS 的应用(如STRUTS 架构)？【基础】  
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：  
1)包含一个controller servlet，能将用户的请求发送到相应的Action 对象；  
2)JSP 自由tag 库，并且在controller servlet 中提供关联支持，帮助开发人员创建交互式表单应用；  
3)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。

27、几种会话跟踪技术？【基础】  
答：cookie、URL 重写、设置表单隐藏域。

28、BS 与CS 的联系与区别？【基础】  
答：C/S 是Client/Server 的缩写，是客户机与服务器结构的应用程序，服务器通常采用高性能的PC、工作站或小型机，并采用大型数据库系统，如Oracle、Sybase、Informix 或SQL Server。客户端需要安装专用的客户端软件。B/Ｓ是Brower/Server 的缩写，是浏览器和服务器结构的应用程序，即Web 应用程序，客户机上只要安装一个浏览器（Browser），如Netscape Navigator 或InternetExplorer，服务器安装Oracle、Sybase、Informix 或SQL Server 等数据库。在这种结构下，用户界面完全通过WWW 浏览器实现，一部分事务逻辑在前端实现，但是主要事务逻辑在[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)端实现。浏览器通过Ｗeb Server 同数据库进行数据交互。  
C/S 与B/S 区别：  
1)硬件环境不同:  
C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=128&fv=0&jk=f3b4b1ed986ba9ce&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=1&n=10&p=baidu&q=baidusiteerror_cpr&rb=0&rs=1&seller_id=1&sid=cea96b98edb1b4f3&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1905736&u=http%3A%2F%2Fwww%2Ealigoto%2Ecom%2Fjava%2Fjava%2Dbase%2F4844%2Ehtml&urlid=0)提供连接和数据交换服务；B/S 建立在广域网之上的, 不必是专门的网络硬件环境,例与电话上网, 租用设备. 信息自己管理. 有比C/S 更强的适应范围, 一般只要有操作系统和浏览器就行；  
2)对安全要求不同：  
C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用C/S 结构适宜. 可以通过B/S 发布部分可公开信息；B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户；  
3)对程序架构不同：  
C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑；  
B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上.  
比C/S 有更高的要求B/S 结构的程序架构是发展的趋势, 从MS 的.Net 系列的BizTalk 2000 Exchange 2000 等, 全面支持网络的构件搭建的系统. SUN 和IBM 推的JavaBean 构件技术等,使B/S 更加成熟；  
4)软件重用不同：  
C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在B/S 要求下的构件的重用性好；  
B/S 对的多重结构,要求构件相对独立的功能. 能够相对较好的重用.就入买来的餐桌可以再利用,而不是做在墙上的石头桌子；  
5)系统维护不同：  
C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统；  
B/S 构件组成,方面构件个别的更换,实现系统的无缝升级. 系统维护开销减到最小.用户从网上自己下载安装就可以实现升级；  
6)处理问题不同：  
C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统；  
B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是C/S 无法作到的. 与操作系统平台关系最小；  
7)用户接口不同：  
C/S 多是建立的Window 平台上,表现方法有限,对程序员普遍要求较高；  
B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低,减低开发成本；  
8)信息流不同：  
C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低；  
B/S 信息流向可变化, B-B B-C B-G 等信息、流向的变化, 更像交易中心。

29、过滤器有哪些作用？【基础】  
答：可以验证客户是否来自可信的网络，可以对客户提交的数据进行重新编码，可以从系统里获得配置的信息，可以过滤掉客户的某些不应该出现的词汇，可以验证用户是否登录，可以验证客户的浏览器是否支持当前的应用，可以记录系统的日志等等。

30、过滤器的用法？（对客户端的请求统一编码和对客户端进行认证）【基础】  
答：首先要实现（implements）Filter 接口，同时覆盖Filter 接口的三个方法：  
init(FilterConfig config) //用于获得FilterConfig 对象；  
doFilter(ServletRequest request, ServletResponse response,  
FilterChain chain) //进行过滤处理一些业务；  
destroy() //销毁Filter。

1、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。【中等难度】  
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%>页  
<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>

2、简述HttpSession 的作用、使用方法，可用代码说明。（3 分钟）【基础】  
答：HttpSession 中可以跟踪并储存用户信息，把值设置到属性中，有2 个方法：  
setAttribute(),getAttrribute()；  
例如：在一个方法中用session.setAttribute(“student”,student);在session 中设置一个属性名为student,值为一个名为student 的对象。而后可在同一session 范围内用getAttribute(“student”)取出该属性，得到student对象。

3、介绍在JSP 中如何使用JavaBeans？【基础】  
答：在JSP 中使用JavaBean 常用的动作有：  
1）<jsp:useBean />：用来创建和查找bean 对象；  
2）<jsp:setProperty />：用来设置bean 的属性，即调用其setXxx()方法；  
3）<jsp:getProperty />：用来获得bean 的属性，即调用其getXxx()方法。

4、JSP 和Servlet 中的请求转发分别如何实现？【基础】  
答：JSP 中的请求转发可利用forward 动作实现：<jsp:forward />；  
Serlvet 中实现请求转发的方式为：  
getServletContext().getRequestDispatcher(path).forward(req,res)。

5、Web.Xml 的作用？【基础】  
答：用于配置web 应用的信息；如listener、filter 及servlet 的配置信息等。

6、写出熟悉的JSTL 标签。【基础】  
答：<c:if>、<c:choose>、<c: when>、<c: otherwise>、<c:forEach>、<c:set>。

7、说出struts 中的标签。【基础】  
答：<bean:message /> <html:errors />  
<bean:include /> <html:messages />  
<bean:define /> <html:html>  
<bean:write /> <html:form>  
<bean:resource /> <html:link>  
<bean:cokkie /> <html:text>  
<bean:heder /> <logic:present />  
<bean:parameter /> <logic:equal />  
<bean:size />

8、JSP 标签的作用？如何定义？【中等难度】  
答：作用：分离jsp 页面的内容和逻辑；  
业务逻辑开发者可以创建自定义标签；  
封装业务逻辑；  
可重用并且易维护；  
易于手工修改、易于工具维护；  
提供简洁的语法；  
定义:  
写标签处理器；  
写tld 文件；  
讲标签处理器和tld 文件放到同一个包里面；  
把jsp 页面和标签库配置部署在一起。

9、写一个自定义标签；【中等难度】  
答：代码如下：  
import javax.servlet.jsp.tagext.\*;  
import javax.servlet.jsp.\*;  
import java.io.\*;  
public class TimeTag extends SimpleTagSupport{  
private boolean isServer = true;  
public void setServer(boolean isServer){  
this.isServer = isServer;  
}  
public void doTag() throws JspException, IOException{  
JspWriter out = getJspContext().getOut();  
if(isServer) {  
out.println(new java.util.Date());  
}else{  
out.println(“<script language=\”javascript\”>”);  
out.println(“document.write(new Date());”);  
out.println(“</script>”);  
}  
}  
}

10、javascript 的优缺点和内置对象；【中等难度】  
答：1）优点：简单易用，与Java 有类似的语法，可以使用任何文本编辑工具编写，只需要浏览器就可执行程序，并且事先不用编译，逐行执行，无需进行严格的变量声明，而且内置大量现成对象，编写少量程序可以完成目标；  
2）缺点：不适合开发大型应用程序；  
3）Javascript 有11 种内置对象：  
Array、String、Date、Math、Boolean、Number、Function、Global、Error、RegExp、Object。

EJB 及Spring 部分（共18 题：基础4 道，中等难度13 道，较难1 道）

11、EJB 与JAVA BEAN 的区别？【基础】  
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。

12、EJB 的几种类型？【基础】  
答：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。

13、remote 接口和home 接口主要作用？【基础】  
答：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。

14、客服端口调用EJB 对象的几个基本步骤？【基础】  
答：设置JNDI 服务工厂以及JNDI 服务地址系统属性，查找Home 接口，从Home接口调用Create 方法创建Remote 接口，通过Remote 接口调用其业务方法。

15、EJB 的角色和三个对象？【中等难度】  
答：一个完整的基于EJB 的分布式计算结构由六个角色组成，这六个角色可以由不同的开发商提供，每个角色所作的工作必须遵循Sun 公司提供的EJB 规范，以保证彼此之间的兼容性。这六个角色分别是EJB 组件开发者（Enterprise Bean Provider） 、应用组合者（Application Assembler）、部署者（Deployer） 、EJB 服务器提供者（EJB Server Provider）、EJB 容器提供者（EJB ContainerProvider）、系统管理员（System Administrator），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；三个对象是Remote（Local）接口、Home（LocalHome）接口，Bean 类。

16、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。【中等难度】  
答：EJB 包括Session Bean、Entity Bean、Message Driven Bean，基于JNDI、RMI、JTA 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。

17、bean 实例的生命周期？【中等难度】  
答：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。

18、EJB 的激活机制？【中等难度】  
答：以Stateful 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。

19、EJB 包括（SessionBean,EntityBean）说出他们的生命周期，及如何管理事务的？【中等难度】  
答：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：EntityBeans 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 通过以下技术管理事务：对象管理组织（OMG）的对象实务服务（OTS），Sun Microsystems 的Transaction Service（JTS）、JavaTransaction API（JTA），开发组（X/Open）的XA 接口。

20、EJB 的事务是如何实现的？何时进行回滚；【中等难度】  
答：是通过使用容器或Bean 自身管理事务的；当产生一个系统异常时容器就自动回滚事务。

21、EJB 容器提供的服务？【中等难度】  
答：主要提供生命周期管理、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。

22、EJB 需直接实现它的业务接口或Home 接口吗？请简述理由。【中等难度】  
答：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。

23、请对以下在J2EE 中常用的名词进行解释(或简单描述) 【中等难度】  
答：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 直接跟容器中的环境变量接口交互，不必关注其它系统问题。主要由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；  
JNDI：（Java Naming & Directory Interface）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 Message Service）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；  
JTA：（ Java Transaction API）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 Action FrameWork）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；  
RMI/IIOP:（Remote Method Invocation /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。

24、J2EE 是什么？【中等难度】  
答：J2EE 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。

25、J2EE 是技术还是平台还是框架？【中等难度】  
答：J2EE 本身是一个标准，一个为企业分布式应用的开发提供的标准平台；J2EE 也是一个框架，包括JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

26、请写出spring 中I0C 的三种实现机制。【中等难度】  
答：三种机制为：通过setter 方法注入、通过构造方法注入和接口注入。

27、写出你熟悉的开源框架以及各自的作用。【中等难度】  
答：框架：hibernate、spring、struts；  
Hibernate 主要用于数据持久化；  
Spring 的控制反转能起到解耦合的作用；  
Struts 主要用于流程控制。

28、EJB 规范规定EJB 中禁止的操作有哪些？【较难】  
答：1)不能操作线程和线程API(线程API 指非线程对象的方法，如notify,wait等)；  
2)不能操作awt；  
3)不能实现服务器功能；  
4)不能对静态属性存取；  
5)不能使用IO 操作直接存取文件系统；  
6)不能加载本地库；  
7)不能将this 作为变量和返回；  
8)不能循环调用。