**JAVA面试题相关基础知识**  
  
    1、面向对象的特征有哪些方面   
  
    ①抽象：  
  
    抽象是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。  
  
    ②继承：  
    继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。  
  
    ③封装：  
    封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。  
  
    ④多态性：  
    多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。  
  
    2、String是最基本的数据类型吗?  
  
    基本数据类型包括byte、int、char、long、float、double、boolean和short。  
    java.lang.String类是final类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用StringBuffer类  
  
    3、int 和 Integer 有什么区别？  
  
    Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int是java的原始数据类型，Integer是java为int提供的封装类。  
  
    4、String 和StringBuffer的区别？  
  
    JAVA平台提供了两个类：String和StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个String类提供了数值不可改变的字符串。而这个StringBuffer类提供的字符串进行修改。当你知道字符数据要改变的时候你可以使用StringBuffer。典型地，你可以使用StringBuffers来动态构造字符数据。  
  
    5、运行时异常与一般异常有何异同？  
  
    异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。  
  
    6、说出Servlet的生命周期，并说出Servlet和CGI的区别？  
  
    Servlet被服务器实例化后，容器运行其init方法，请求到达时运行其service方法，service方法自动派遣运行与请求对应的doXXX方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其destroy方法。  
    与cgi的区别在于servlet处于服务器进程中，它通过多线程方式运行其service方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而CGI对每个请求都产生新的进程，服务完成后销毁，所以效率上低于servlet。  
  
    7、说出ArrayList,Vector, LinkedList的存储性能和特性  
  
    ArrayList和Vector都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector由于使用了synchronized方法（线程安全），通常性能上较ArrayList差，而LinkedList使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。  
  
    8、EJB是基于哪些技术实现的？  
  
    EJB包括Session Bean、Entity Bean、Message Driven Bean，基于JNDI、RMI、JAT等技术实现。  
    SessionBean在J2EE应用程序中被用来完成一些服务器端的业务操作，例如访问数据库、调用其他EJB组件。EntityBean被用来代表应用系统中用到的数据。  
    对于客户机，SessionBean是一种非持久性对象，它实现某些在服务器上运行的业务逻辑。  
    对于客户机，EntityBean是一种持久性对象，它代表一个存储在持久性存储器中的实体的对象视图，或是一个由现有企业应用程序实现的实体。  
    Session Bean 还可以再细分为 Stateful Session Bean 与 Stateless Session Bean ，这两种的 Session Bean都可以将系统逻辑放在 method之中执行，不同的是 Stateful Session Bean 可以记录呼叫者的状态，因此通常来说，一个使用者会有一个相对应的 Stateful Session Bean 的实体。  
  
    9、Collection 和 Collections的区别？  
  
    Collection是集合类的上级接口，继承与他的接口主要有Set 和List.  
    Collections是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。  
  
    10、&和&&的区别？  
  
    &是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。  
  
    11、HashMap和Hashtable的区别？  
  
    HashMap是Hashtable的轻量级实现（非线程安全的实现），他们都完成了Map接口，主要区别在于HashMap允许空（null）键值（key）,由于非线程安全，效率上可能高于Hashtable。  
    HashMap允许将null作为一个entry的key或者value，而Hashtable不允许。  
    Hashtable继承自Dictionary类，而HashMap是Java1.2引进的Map interface的一个实现。  
    的不同是，Hashtable的方法是Synchronize的，而HashMap不是，在多个线程访问Hashtable时，不需要自己为它的方法实现同步，而HashMap 必须为之提供外同步。  
  
    12、final, finally, finalize的区别？  
  
    final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。  
    finally是异常处理语句结构的一部分，表示总是执行。  
    finalize是Object类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。  
  
    13、sleep() 和 wait() 有什么区别?   
  
    sleep是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep不会释放对象锁。  
    wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。  
  
    14、Overload和Override的区别？Overloaded的方法是否可以改变返回值的类型?  
  
    方法的重写Overriding和重载Overloading是Java多态性的不同表现。重写Overriding是父类与子类之间多态性的一种表现，重载Overloading是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded的方法是可以改变返回值的类型。  
  
    15、error和exception有什么区别?  
  
    error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。  
    exception 表示一种设计或实现问题。也是说，它表示如果程序运行正常，从不会发生的情况。  
  
    16、同步和异步有何异同，在什么情况下分别使用他们？举例说明。  
  
    如果数据将在线程间共享。例如正在写的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据是共享数据，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，应该使用异步编程，在很多情况下采用异步途径往往更有效率。  
  
    17、heap和stack有什么区别？  
  
    栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素  
  
    18、forward 和redirect的区别？  
  
    forward是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。  
    redirect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。  
  
    19、Static Nested Class 和 Inner Class的不同？  
  
    Static Nested Class是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。  
  
    20、JSP中动态INCLUDE与静态INCLUDE的区别？  
  
    动态INCLUDE用jsp:include动作实现 <jsp:include page="included.jsp" flush="true" />它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数。  
静态INCLUDE用include伪码实现,定不会检查所含文件的变化，适用于包含静态页面<%@ include file="included.htm" %>   
  
    21、什么时候用assert？  
  
    assertion(断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制。在实现中，assertion是在程序中的一条语句，它对一个boolean表达式进行检查，一个正确程序必须保证这个boolean表达式的值为true；如果该值为false，说明程序已经处于不正确的状态下，系统将给出警告或退出。一般来说，assertion用于保证程序最基本、关键的正确性。assertion检查通常在开发和测试时开启。为了提高性能，在软件发布后，assertion检查通常是关闭的。  
  
    22、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。   
  
    23、short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?   
  
    short s1 = 1; s1 = s1 + 1; （s1+1运算结果是int型，需要强制转换类型）  
    short s1 = 1; s1 += 1;（可以正确编译）  
  
    24、Math.round(11.5)等於多少? Math.round(-11.5)等於多少?   
  
    Math.round(11.5)==12  
    Math.round(-11.5)==-11  
    round方法返回与参数最接近的长整数，参数加1/2后求其floor.  
  
    25、String s = new String("xyz");创建了几个String Object?   
  
    两个  
  
    26、设计4个线程，其中两个线程每次对j增加1，另外两个线程对j每次减少1。写出程序。  
  
    以下程序使用内部类实现线程，对j增减的时候没有考虑顺序问题。  
    public class ThreadTest1{  
    private int j;  
    public static void main(String args[]){  
    ThreadTest1 tt=new ThreadTest1();  
     Inc inc=tt.new Inc();  
    Dec dec=tt.new Dec();  
    for(int i=0;i<2;i++){  
    Thread t=new Thread(inc);  
    t.start();  
    t=new Thread(dec);  
    t.start();  
    }  
     }  
     private synchronized void inc(){  
     j++;  
     System.out.println(Thread.currentThread().getName()+"-inc:"+j);  
     }  
     private synchronized void dec(){  
    j--;  
    System.out.println(Thread.currentThread().getName()+"-dec:"+j);  
     }  
     class Inc implements Runnable{  
    public void run(){  
     for(int i=0;i<100;i++){  
     inc();  
    }  
    }  
    }  
    class Dec implements Runnable{  
    public void run(){  
    for(int i=0;i<100;i++){  
    dec();  
    }  
    }  
    }  
    }  
  
    27、Java有没有goto?  
  
    java中的保留字，现在没有在java中使用。  
  
    28、启动一个线程是用run()还是start()?  
  
    启动一个线程是调用start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由JVM调度并执行。这并不意味着线程会立即运行。run()方法可以产生必须退出的标志来停止一个线程。  
  
    29、应用服务器有那些？  
  
    BEA WebLogic Server，IBM WebSphere Application Server，Oracle9i Application Server，jBoss，Tomcat  
  
    30、接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类(concrete class)?  
  
     接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。  
  
    31、说出数据连接池的工作机制是什么?  
  
    J2EE服务器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，其他调用可以使用这个连接。  
  
    32、abstract的method是否可同时是static,是否可同时是native，是否可同时是synchronized?  
  
    都不能  
  
    33、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。  
  
    34、Set里的元素是不能重复的，那么用什么方法来区分重复与否呢? 是用==还是equals()? 它们有何区别?  
  
    Set里的元素是不能重复的，那么用iterator()方法来区分重复与否。equals()是判读两个Set是否相等。  
  
    equals()和==方法决定引用值是否指向同一对象equals()在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值。  
  
    35、构造器Constructor是否可被override?  
  
    构造器Constructor不能被继承，因此不能重写Overriding，但可以被重载Overloading。  
  
    36、是否可以继承String类?  
  
    String类是final类故不可以继承。  
  
    37、swtich是否能作用在byte上，是否能作用在long上，是否能作用在String上?  
  
    switch（expr1）中，expr1是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、 short、 char 或者 byte。long,string 都不能作用于swtich。  
  
    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，在return前还是后?  
  
    会执行，在return前执行。  
  
    39、编程题: 写一个Singleton出来。  
  
    Singleton模式主要作用是保证在Java应用程序中，一个类Class只有一个实例存在。  
    一般Singleton模式通常有几种种形式:  
    种形式: 定义一个类，它的构造函数为private的，它有一个static的private的该类变量，在类初始化时实例话，通过一个public的getInstance方法获取对它的引用,继而调用其中的方法。  
public class Singleton {  
private Singleton(){}  
　　 //在自己内部定义自己一个实例，是不是很奇怪？  
　　 //注意这是private 只供内部调用  
　　 private static Singleton instance = new Singleton();  
　　 //这里提供了一个供外部访问本class的静态方法，可以直接访问　　  
　　 public static Singleton getInstance() {  
　　　　 return instance; 　　  
　　 }   
      }   
    第二种形式:   
    public class Singleton {   
　　private static Singleton instance = null;  
　　public static synchronized Singleton getInstance() {  
　　//这个方法比上面有所改进，不用每次都进行生成对象，只是次　　　 　  
　　//使用时生成实例，提高了效率！  
　　if (instance==null)  
　　　　instance＝new Singleton();  
    return instance; 　　}   
     }   
  
    40、Java的接口和C++的虚类的相同和不同处  
  
    由于Java不支持多继承，而有可能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制不能满足要求。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。  
  
    41、Java中的异常处理机制的简单原理和应用  
  
    当JAVA程序违反了JAVA的语义规则时，JAVA虚拟机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。  
  
    42、请说出你所知道的线程同步的方法  
  
    wait():使一个线程处于等待状态，并且释放所持有的对象的lock。  
    sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉InterruptedException异常。  
    notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由JVM确定唤醒哪个线程，而且不是按优先级。  
    Allnotity():唤醒所有处入等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。  
  
     43、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。  
  
    44、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。它负责在运行时查找和装入类文件的类。  
  
    45、char型变量中能不能存贮一个中文汉字?为什么?   
  
    能够定义成为一个中文的，因为java中以unicode编码，一个char占2个字节，所以放一个中文是没问题的  
  
    46、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?   
  
    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  
  
    47、JSP的内置对象及方法  
  
    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javax.servlet.jsp.PageContext对象。它是用于方便存取各种范围的名字空间、servlet相关的对象的API，并且包装了通用的servlet相关功能的方法。   
    session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   
    applicaton 表示一个javax.servle.ServletContext对象。这有助于查找有关servlet引擎和servlet环境的信息   
    config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  
  
    48、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，也是程序本身。  
    Java中的线程有四种状态分别是：运行、绪、挂起、结束。   
  
    49、servlet的生命周期  
  
    web容器加载servlet，生命周期开始。通过调用servlet的init()方法进行servlet的初始化。通过调用service()方法实现，根据请求的不同调用不同的do\*\*\*()方法。结束服务，web容器调用servlet的destroy()方法。  
  
    50、如何现实servlet的单线程模式  
  
    <%@ page isThreadSafe=”false”%>  
  
    51、页面间对象传递的方法  
  
    request，session，application，cookie等  
  
    52、JSP和Servlet有哪些相同点和不同点，他们之间的联系是什么？   
  
    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。  
  
    52、J2EE是技术还是平台还是框架？  
  
    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。  
  
    53、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？  
  
    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;  
    }  
  
    54、EJB的几种类型  
  
    会话（Session）Bean ，实体（Entity）Bean 消息驱动的（Message Driven）Bean  
    会话Bean又可分为有状态（Stateful）和无状态（Stateless）两种  
    实体Bean可分为Bean管理的持续性（BMP）和容器管理的持续性（CMP）两种  
  
    55、如何给weblogic指定大小的内存?   
  
    在启动Weblogic的脚本中（位于所在Domian对应服务器目录下的startServerName），增加set MEM\_ARGS=-Xms32m -Xmx200m，可以调整最小内存为32M，200M  
  
    56、如何启动时不需输入用户名与密码?  
  
    修改服务启动文件，增加 WLS\_USER和WLS\_PW项。也可以在boot.properties文件中增加加密过的用户名和密码.  
  
    57、在weblogic管理制台中对一个应用域(或者说是一个网站,Domain)进行jms及ejb或连接池等相关信息进行配置后,实际保存在什么文件中?  
  
    保存在此Domain的config.xml文件中，它是服务器的核心配置文件。  
  
    58、如何查看在weblogic中已经发布的EJB?  
  
    可以使用管理控制台，在它的Deployment中可以查看所有已发布的EJB  
  
    59、CORBA是什么?用途是什么?   
  
    CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture)，由对象管理组织 (Object Management Group，缩写为 OMG)标准化。它的组成是接口定义语言(IDL), 语言绑定(binding:也译为联编)和允许应用程序间互操作的协议。 其目的为：用不同的程序设计语言书写在不同的进程中运行，为不同的操作系统开发。  
  
    60、说说你所熟悉或听说过的j2ee中的几种常用模式?及对设计模式的一些看法？  
  
    Session Facade Pattern：使用SessionBean访问EntityBean  
    Message Facade Pattern：实现异步调用  
    EJB Command Pattern：使用Command JavaBeans取代SessionBean，实现轻量级访问  
    Data Transfer Object Factory：通过DTO Factory简化EntityBean数据提供特性  
    Generic Attribute Access：通过AttibuteAccess接口简化EntityBean数据提供特性  
    Business Interface：通过远程（本地）接口和Bean类实现相同接口规范业务逻辑一致性  
    EJB架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂，项目队伍越庞大则越能体现良好设计的重要性。  
  
    61、Servlet执行时一般实现哪几个方法？  
  
    public void init(ServletConfig config)  
    public ServletConfig getServletConfig()  
    public String getServletInfo()  
    public void service(ServletRequest request,ServletResponse response)  
    public void destroy()  
  
    62、j2ee常用的设计模式？说明工厂模式  
  
    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。  
  
    63、EJB需直接实现它的业务接口或Home接口吗，请简述理由。  
  
    远程接口和Home接口不需要直接实现，他们的实现代码是由服务器产生的，程序运行中对应实现类会作为对应接口类型的实例被使用。  
  
    64、排序都有哪几种方法？请列举。用JAVA实现一个快速排序。  
  
    排序的方法有：插入排序（直接插入排序、希尔排序），交换排序（冒泡排序、快速排序），选择排序（直接选择排序、堆排序），归并排序，分配排序（箱排序、基数排序）  
    快速排序的伪代码。  
    / /使用快速排序方法对a[ 0 :n- 1 ]排序  
    从a[ 0 :n- 1 ]中选择一个元素作为m i d d l e，该元素为支点  
    把余下的元素分割为两段left 和r i g h t，使得l e f t中的元素都小于等于支点，而right 中的元素都大于等于支点  
    递归地使用快速排序方法对left 进行排序  
    递归地使用快速排序方法对right 进行排序  
    所得结果为l e f t + m i d d l e + r i g h t  
  
    65、一个“.java”源文件中是否可以包括多个类（不是内部类）？有什么限制？  
  
    可以，必须只有一个类名与文件名相同。  
  
    66、MVC的各个部分都有那些技术来实现?如何实现?   
  
    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。  
  
    67、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。字节流继承于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。  
  
    68、java中会存在内存泄漏吗，请简单描述。  
  
    会，如：int i,i2; return (i-i2); //when i为足够大的正数,i2为足够大的负数。结果会造成溢位，导致错误。  
  
    69、java中实现多态的机制是什么？  
  
    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。  
  
    70、静态变量和实例变量的区别？  
  
    static i = 10; //常量  
    class A a; a.i =10;//可变  
  
    71、什么是java序列化，如何实现java序列化？  
  
    序列化是一种用来处理对象流的机制，所谓对象流也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，该接口没有需要实现的方法，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。  
  
    72、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,如果其中包含对象的method()；不能保证对象初始化.  
  
    73、J2EE是什么？  
  
    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。  
  
    74、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  
    常用的包：java.lang java.awt java.io java.util java.sql  
    常用的接口：Remote List Map Document NodeList   
  
    75、jsp有哪些动作?作用分别是什么?   
  
    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。  
   
    **JAVA代码查错**  
  
    1.  
    abstract class Name {  
    private String name;  
    public abstract boolean isStupidName(String name) {}  
    }  
    答案: 错。abstract method必须以分号结尾，且不带花括号。  
    2.  
    public class Something {  
    void doSomething () {  
    private String s = "";  
    int l = s.length();  
    }  
    }  
    答案: 错。局部变量前不能放置任何访问修饰符 (private，public，和protected)。final可以用来修饰局部变量  
    (final如同abstract和strictfp，都是非访问修饰符，strictfp只能修饰class和method而非variable)。  
    3.  
    abstract class Something {  
    private abstract String doSomething ();  
    }  
    答案: 错。abstract的methods不能以private修饰。abstract的methods是让子类implement(实现)具体细节的，怎么可以用private把abstract  
    method封锁起来呢? (同理，abstract method前不能加final)。  
    4.  
    public class Something {  
    public int addOne(final int x) {  
    return ++x;  
    }  
    }  
    答案: 错。int x被修饰成final，意味着x不能在addOne method中被修改。  
    5.  
    public class Something {  
    public static void main(String[] args) {  
    Other o = new Other();  
    new Something().addOne(o);  
    }  
    public void addOne(final Other o) {  
    o.i++;  
    }  
    }  
    class Other {  
    public int i;  
    }  
    答案: 正确。在addOne method中，参数o被修饰成final。如果在addOne method里我们修改了o的reference  
    (比如: o = new Other();)，那么如同上例这题也是错的。但这里修改的是o的member vairable  
    (成员变量)，而o的reference并没有改变。  
    6.  
    class Something {  
    int i;  
    public void doSomething() {  
    System.out.println("i = " + i);  
    }  
    }   
    答案: 正确。输出的是"i = 0"。int i属於instant variable (实例变量，或叫成员变量)。instant variable有default value。int的default value是0。  
    7.  
    class Something {  
    final int i;  
    public void doSomething() {  
    System.out.println("i = " + i);  
    }  
    }  
    答案: 错。final int i是个final的instant variable (实例变量，或叫成员变量)。final的instant variable没有default value，必须在constructor (构造器)结束之前被赋予一个明确的值。可以修改为"final int i = 0;"。  
    8.  
    public class Something {  
    public static void main(String[] args) {  
    Something s = new Something();  
    System.out.println("s.doSomething() returns " + doSomething());  
    }  
    public String doSomething() {  
    return "Do something ...";  
    }  
   }  
    答案: 错。看上去在main里call doSomething没有什么问题，毕竟两个methods都在同一个class里。但仔细看，main是static的。static method不能直接call non-static methods。可改成"System.out.println("s.doSomething() returns " + s.doSomething());"。同理，static method不能访问non-static instant variable。  
    9.  
   此处，Something类的文件名叫OtherThing.java  
    class Something {  
    private static void main(String[] something\_to\_do) {   
    System.out.println("Do something ...");  
    }  
    }  
    答案: 正确。从来没有人说过Java的Class名字必须和其文件名相同。但public class的名字必须和文件名相同。  
    10．  
    interface A{  
     int x = 0;  
    }  
    class B{  
    int x =1;  
    }  
    class C extends B implements A {  
    public void pX(){  
    System.out.println(x);  
    }  
    public static void main(String[] args) {  
    new C().pX();  
    }  
    }