1、面向对象的特征有哪些方面    
(1).抽象：  
抽象就

是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。

抽象包括两个方面，一是过程抽象，二是数据抽象。

[数据抽象](http://wenwen.sogou.com/s/?w=%E6%95%B0%E6%8D%AE%E6%8A%BD%E8%B1%A1&ch=w.search.intlink)就是针对对象的属性，比如建立一个鸟这样的[抽象类](http://wenwen.sogou.com/s/?w=%E6%8A%BD%E8%B1%A1%E7%B1%BB&ch=w.search.intlink)，鸟会有以下特征，两个翅膀，两支脚，有羽毛等等特性，写成类都是鸟的属性。但并没有“鸟”这种动物，在java里面就是抽象类没有实例。而[啄木鸟](http://wenwen.sogou.com/s/?w=%E5%95%84%E6%9C%A8%E9%B8%9F&ch=w.search.intlink)，麻雀等就是具体的鸟，在java里面就是[继承](http://wenwen.sogou.com/s/?w=%E7%BB%A7%E6%89%BF&ch=w.search.intlink)自抽象类的类，可以实例化的。  
  
过程抽象就是针对对象的行为特征，比如鸟会飞，会跳等等，这些方面的就会抽象为方法，即过程，写成类都是鸟的方法。但各种鸟的[飞行动作](http://wenwen.sogou.com/s/?w=%E9%A3%9E%E8%A1%8C%E5%8A%A8%E4%BD%9C&ch=w.search.intlink)是不一样的。具体到啄木鸟，麻雀等动物的话，动作是不一样的。  
  
说白了，抽象就是把事物的特点，从具体实例里面抽取出来。形成一套适合所有实例的[框架](http://wenwen.sogou.com/s/?w=%E6%A1%86%E6%9E%B6&ch=w.search.intlink)

(2).继承：  
继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的

方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派

生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。

(3).封装：  
封装是把过程和数据包围起来，对数

据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他

对象。

(4). 多态性：  
多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

2、String是最基本的数据类型吗?  
基本数据类型包括byte、int、char、long、

float、double、boolean和short。  
java.lang.String类是final类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用StringBuffer类  
3、int 和 Integer 有什么区别  
Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int是

java的原始数据类型，Integer是java为int提供的封装类。Java为每个原始类型提供了封装类。  
原始类型封装类

Boolean Boolean  
char Character  
byte Byte  
short Short  
intIn teger  
long Long  
float Float  
double Double

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。

4、String 和StringBuffer的区别  
JAVA平台提供了两个类：String和StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个String类提供了数值不可改变的字符串。而这个StringBuffer类提供的字符串进行修改。当你知道字符数据要改变的时候你就可以使用StringBuffer。典型地，你可以使用StringBuffers来动态构造字符数据。

5、运行时异常与一般异常有何异同？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java编译器要求方法必须声明

抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

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synchronized方法（线程全），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。

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。  
   1.Hashtable是Dictionary的子类，HashMap是Map接口的一个实现类；  
2.Hashtable中的方法是同步的，而HashMap中的方法在缺省情况下是非同步的。即是说，在多线程应用程序中，不用专门的操作就安全地可以使用Hashtable了；而对于HashMap，则需要额外的同步机制。但HashMap的同步问题可通过Collections的一个静态方法得到解决：  
Map Collections.synchronizedMap(Map m)  
这个方法返回一个同步的Map，这个Map封装了底层的HashMap的所有方法，使得底层的HashMap即使是在多线程的环境中也是安全的。  
3.在HashMap中，null可以作为键，这样的键只有一个；可以有一个或多个键所对应的值为null。当get()方法返回null值时，即可以表示HashMap中没有该键，也可以表示该键所对应的值为null。因此，在HashMap中不能由get()方法来判断HashMap中是否存在某个键，而应该用containsKey()方法来判断。   
  
Map  
├Hashtable  
├HashMap  
└WeakHashMap  
Map接口  
　　请注意，Map没有继承Collection接口，Map提供key到value的映射。一个Map中不能包含相同的key，每个key只能映射一个value。Map接口提供3种集合的视图，Map的内容可以被当作一组key集合，一组value集合，或者一组key-value映射。Hashtable类  
　　Hashtable继承Map接口，实现一个key-value映射的哈希表。任何非空（non-null）的对象都可作为key或者value。  
　　添加数据使用put(key, value)，取出数据使用get(key)，这两个基本操作的时间开销为常数。  
Hashtable通过initial capacity和load factor两个参数调整性能。通常缺省的load factor0.75较好地实现了时间和空间的均衡。增大load factor可以节省空间但相应的查找时间将增大，这会影响像get和put这样的操作。  
使用Hashtable的简单示例如下，将1，2，3放到Hashtable中，他们的key分别是”one”，”two”，”three”：  
　　　　Hashtable numbers = new Hashtable();  
　　　　numbers.put(“one”, new Integer(1));  
　　　　numbers.put(“two”, new Integer(2));  
　　　　numbers.put(“three”, new Integer(3));  
　　要取出一个数，比如2，用相应的key：  
　　　　Integer n = (Integer)numbers.get(“two”);  
　　　　System.out.println(“two = ” + n);  
　　由于作为key的对象将通过计算其散列函数来确定与之对应的value的位置，因此任何作为key的对象都必须实现hashCode和equals方法。hashCode和equals方法继承自根类Object，如果你用自定义的类当作key的话，要相当小心，按照散列函数的定义，如果两个对象相同，即obj1.equals(obj2)=true，则它们的hashCode必须相同，但如果两个对象不同，则它们的hashCode不一定不同，如果两个不同对象的hashCode相同，这种现象称为冲突，冲突会导致操作哈希表的时间开销增大，所以尽量定义好的hashCode()方法，能加快哈希表的操作。  
　　如果相同的对象有不同的hashCode，对哈希表的操作会出现意想不到的结果（期待的get方法返回null），要避免这种问题，只需要牢记一条：要同时复写equals方法和hashCode方法，而不要只写其中一个。  
　　Hashtable是同步的。HashMap类  
　　HashMap和Hashtable类似，不同之处在于HashMap是非同步的，并且允许null，即null value和nullkey。，但是将HashMap视为Collection时（values()方法可返回Collection），其迭代子操作时间开销和HashMap的容量成比例。因此，如果迭代操作的性能相当重要的话，不要将HashMap的初始化容量设得过高，或者load factor过低。WeakHashMap类  
　　WeakHashMap是一种改进的HashMap，它对key实行“弱引用”，如果一个key不再被外部所引用，那么该key可以被GC回收。

12、final, finally, finalize的区别。  
　　final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。  
finally是异常处理语句结构的一部分，表示总是执行。finalize是Object类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

13、sleep() 和wait() 有什么区别?  
   sleep是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep不会释放对象锁。wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

14、Overload和Override的区别。

Overloaded的方法是否可以改变返回值的类型?  
方法的重写Overriding和重载Overloading是Java多态性的不同表现。重写Overriding

是父类与子类之间多态性的一种表现，重载Overloading是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被

重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被"屏蔽"了。如果在一个类中定义了多个同名的方法，它

们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded的方法是可以改变返回值的类型。

15、error和exception有什么区别?  
error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

16、同步和异步有何异同，在什么情况下分别使用他们？

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

数据，必须进行同步存取。  
当应用程序在对象上调用了一个需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多

情况下采用异步途径往往更有效率。

17、abstract class和interface有什么区别?  
声明方法的存在而不去实现它的类被叫做抽象类

（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建abstract 类的实例。然而可

以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法

提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。  
接口（interface）是抽象类的变体。在

接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义static final成员变

量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该

接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，

instanceof 运算符可以用来决定某对象的类是否实现了接口。

18、heap和stack有什么区别。  
栈是一种线形集合，其添加和删除元素的操作应

在同一段完成。栈按照后进先出的方式进行处理。  
堆是栈的一个组成元素

21、Static Nested Class 和 Inner

Class的不同。  
  Static Nested Class是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部

类实例化后才能实例化。  
22、JSP中动态INCLUDE与静态INCLUDE的区别？  
动态INCLUDE用jsp:include动作实现 <jsp:include

page="included.jsp" flush="true" />它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数。  
静态INCLUDE用

include伪码实现,定不会检查所含文件的变化，适用于包含静态页面<%@ include file="included.htm" %>  
23、什么时候用assert  
  assertion(断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制。在实现中，assertion就是在程序中的一条语句，它对

一个boolean表达式进行检查，一个正确程序必须保证这个boolean表达式的值为true；如果该值为false，说明程序已经处于不正确的状态下，系统将给出

警告或退出。一般来说，assertion用于保证程序最基本、关键的正确性。assertion检查通常在开发和测试时开启。为了提高性能，在软件发布后，

assertion检查通常是关闭的。

24、GC是什么? 为什么要有GC?  
　　GC是垃圾收集的意思（Gabage Collection）,内存处理是编程人员容

易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java提供的GC功能可以自动监测对象是否超过作用域从而达到自动回收内存的

目的，Java语言没有提供释放已分配内存的显示操作方法。  
25、short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 +=

1;有什么错?  
  short s1 = 1; s1 = s1 + 1; （s1+1运算结果是int型，需要强制转换类型）  
short s1 = 1; s1 += 1;（

可以正确编译）  
26、Math.round(11.5)等於多少? Math.round(-11.5)等於多少?  
  Math.round(11.5)==12

Math.round(-11.5)==-11  
round方法返回与参数最接近的长整数，参数加1/2后求其floor.  
27、String s = new

String("xyz");创建了几个String Object?  
  两个  
28、设计4个线程，其中两个线程每次对j增加1，另外两个线程对j每次减少1。写

出程序。  
以下程序使用内部类实现线程，对j增减的时候没有考虑顺序问题。  
public class ThreadTest1{  
private int

j;  
public static void main(String args[]){  
ThreadTest1 tt=new ThreadTest1();  
Inc

inc=tt.new Inc();  
Dec dec=tt.new Dec();  
for(int i=0;i<2;i++){  
Thread t=new Thread

(inc);  
t.start();  
t=new Thread(dec);  
t.start();  
}  
}  
private synchronized void

inc(){  
j++;  
System.out.println(Thread.currentThread().getName()+"-inc:"+j);  
}  
private

synchronized void dec(){  
j--;  
System.out.println(Thread.currentThread().getName()+"-

dec:"+j);  
}  
class Inc implements Runnable{  
public void run(){  
for(int i=0;i<100;i++)

{  
inc();  
}  
}  
}  
class Dec implements Runnable{  
public void run(){  
for(int

i=0;i<100;i++){  
dec();  
}  
}  
}  
}  
29、Java有没有goto?  
java中的保留字，现在没有在java中使用。

30、启动一个线程是用run()还是start()?  
启动一个线程是调用start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由

JVM调度并执行。这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

31、EJB包括

（SessionBean,EntityBean）说出他们的生命周期，及如何管理事务的？  
SessionBean：Stateless Session Bean 的生命周期是由容

器决定的，当客户机发出请求要建立一个Bean的实例时，EJB容器不一定要创建一个新的Bean的实例供客户机调用，而是随便找一个现有的实例提供给客户机。当

客户机第一次调用一个Stateful Session Bean 时，容器必须立即在服务器中创建一个新的Bean实例，并关联到客户机上，以后此客户机调用

Stateful Session Bean 的方法时容器会把调用分派到与此客户机相关联的Bean实例。  
EntityBean：Entity Beans能存活相对较长的

时间，并且状态是持续的。只要数据库中的数据存在，Entity beans就一直存活。而不是按照应用程序或者服务进程来说的。即使EJB容器崩溃了，Entity

beans也是存活的。Entity Beans生命周期能够被容器或者 Beans自己管理。  
EJB通过以下技术管理实务：对象管理组织（OMG）的对象实务服务

（OTS），Sun Microsystems的Transaction Service（JTS）、Java Transaction API（JTA），开发组（X/Open）的XA接口。  
32、

应用服务器有那些？  
BEA WebLogic Server，IBM WebSphere Application Server，Oracle9i Application Server，

jBoss，Tomcat  
33、给我一个你最常见到的runtime exception。  
ArithmeticException, ArrayStoreException,

BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException,

ClassCastException, CMMException, ConcurrentModificationException, DOMException,

EmptyStackException, IllegalArgumentException, IllegalMonitorStateException,

IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException,

MissingResourceException, NegativeArraySizeException, NoSuchElementException,

NullPointerException, ProfileDataException, ProviderException, RasterFormatException,

SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException,

UnsupportedOperationException  
34、接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类

(concrete class)?  
接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

35、List, Set, Map是否继承自Collection接口?  
  List，Set是，Map不是  
36、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。  
37、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  
38、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。  
39、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。  
40、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。  
41、是否可以继承String

类?  
String类是final类故不可以继承。  
42、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。  
43、try {}里有一个return语句，那么紧跟在这个try后的finally {}里的code会不会被执行，什

么时候被执行，在return前还是后?  
会执行，在return前执行。  
44、编程题: 用最有效率的方法算出2乘以8等於几?  
2 << 3  
45

、两个对象值相同(x.equals(y) == true)，但却可有不同的hash code，这句话对不对?  
不对，有相同的hash code。  
46、当一个对象

被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回变化后的结果，那么这里到底是值传递还是引用传递?  
是值传递。Java 编程语言只

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

synchronized方法只能由一个线程访问。  
48、编程题: 写一个Singleton出来。  
Singleton模式主要作用是保证在Java应用程序中，一

个类Class只有一个实例存在。  
一般Singleton模式通常有几种种形式:  
第一种形式: 定义一个类，它的构造函数为private的，它有一个

static的private的该类变量，在类初始化时实例话，通过一个public的getInstance方法获取对它的引用,继而调用其中的方法。  
public

class Singleton {  
private Singleton(){}  
　　   //在自己内部定义自己一个实例，是不是很奇怪？  
　　   //注意这是

private 只供内部调用  
　　   private static Singleton instance = new Singleton();  
　　   //这里提供了一个

供外部访问本class的静态方法，可以直接访问　　  
　　   public static Singleton getInstance() {  
　　　　   return

instance; 　　  
　　   }  
  }  
  第二种形式:  
public class Singleton {  
　　private static

Singleton instance = null;  
　　public static synchronized Singleton getInstance() {  
　　//这个方法

比上面有所改进，不用每次都进行生成对象，只是第一次　　　 　  
　　//使用时生成实例，提高了效率！  
　　if (instance==null)

instance＝new Singleton();  
return instance; 　　}  
}  
其他形式:  
定义一个类，它的构造函数为private的，

所有方法为static的。  
一般认为第一种形式要更加安全些  
49、Java的接口和C++的虚类的相同和不同处。  
由于Java不支持多继承，而有可

能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制就不能满足要求。与继承相比，接口有更高的灵活性，因为接口中没有任何实现代码。

当一个类实现了接口以后，该类要实现接口里面所有的方法和属性，并且接口里面的属性在默认状态下面都是public static,所有方法默认情况下是

public.一个类可以实现多个接口。  
50、Java中的异常处理机制的简单原理和应用。  
当JAVA程序违反了JAVA的语义规则时，JAVA虚拟机就会将

发生的错误表示为一个异常。违反语义规则包括2种情况。一种是JAVA类库内置的语义检查。例如数组下标越界,会引发IndexOutOfBoundsException;

访问null的对象时会引发NullPointerException。另一种情况就是JAVA允许程序员扩展这种语义检查，程序员可以创建自己的异常，并自由选择在何时

用throw关键字引发异常。所有的异常都是java.lang.Thowable的子类。