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。抽象并不打算了解所有问题，而仅仅是选择当中的一部分，临时不用部分细节。抽象包含两个方面，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，而且同意和鼓舞类的重用，它提供了一种明白表述共性的方法。对象的一个新类能够从现有的类中派生。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）。

派生类能够从它的基类那里继承方法和实例变量，而且类能够改动或添加新的方法使之更适合特殊的须要。

3.封装：

封装是把过程和数据包围起来，对数据的訪问仅仅能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界能够被描绘成一系列全然自治、封装的对象，这些对象通过一个受保护的接口訪问其它对象。

4. 多态性：

多态性是指同意不同类的对象对同一消息作出响应。多态性包括參数化多态性和包括多态性。

多态性语言具有灵活、抽象、行为共享、代码共享的优势。非常好的攻克了应用程序函数同名问题。

5、String 是最主要的数据类型吗?

基本数据类型包含 byte、int、char、long、float、double、boolean 和 short。

java.lang.String 类是 final 类型的。因此不能够继承这个类、不能改动这个类。为了提高效率节省空间，我们应该用 StringBuffer 类

6、int 和 Integer 有什么差别

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 为每一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
int	Integer

long Long
float Float
double Double

引用类型和原始类型的行为全然不同，而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使用方法。它们包含：大小和速度问题，这样的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。

7、String 和 StringBuffer 的差别

JAVA 平台提供了两个类：String 和 StringBuffer，它们能够储存和操作字符串，即包括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改动。当你知道字符数据要改变的时候你就能够使用 StringBuffer。典型地。你能够使用 StringBuffers 来动态构造字符数据。

8、执行时异常与一般异常有何异同？

异常表示程序执行过程中可能出现的非正常状态，执行时异常表示虚拟机的通常操作中可能遇到的异常。是一种常见执行错误。java 编译器要求方法必须声明抛出可能发生的非执行时异常，可是并不要求必须声明抛出未被捕获的执行时异常。

9、说出 Servlet 的生命周期，并说出 Servlet 和 CGI 的差别。

Servlet 被 server 实例化后，容器执行其 init 方法，请求到达时执行其 service 方法。service 方法自己主动派遣执行与请求相应的 doXXX 方法（doGet，doPost）等，当 server 决定将实例销毁的时候调用其 destroy 方法。
与 cgi 的差别在于 servlet 处于 server 进程中。它通过多线程方式执行其 service 方法。一个实例能够服务于多个请求，而且事实上例一般不会销毁，而 CGI 对每一个请求都产生新的进程，服务完毕后就销毁，所以效率上低于 servlet。

10、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添加和插入元素，它们都同意直接按序号索引元素，可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 因为使用了 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，按序号索引数据须要进行前向或后向遍历，可是插入数据时仅仅须要记录本项的前后项就可以，所以插入速度较快。

11、EJB 是基于哪些技术实现的？并说出 SessionBean 和 EntityBean 的差别，StatefulBean 和 StatelessBean 的差别。

EJB 包含 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毕一些 server 端的业务操作。比如訪问数据库、调用其它 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 server 上执行的业务逻辑。

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 还能够再细分为 Stateful Session Bean 与 Stateless Session Bean 。这两种的 Session Bean 都能够将系统逻辑放在 method 之中运行，不同的是 Stateful Session Bean 能够记录呼叫者的状态，因此通常来说，一个使用者会有一个相相应的 Stateful Session Bean 的实体。Stateless Session Bean 尽管也是逻辑组件，可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 的实体来运行这个 method。换言之，非常可能数个使用者在运行某个 Stateless Session Bean 的 methods 时。会是同一个 Bean 的 Instance 在运行。从内存方面来看， Stateful Session Bean 与 Stateless Session Bean 比較， Stateful Session Bean 会消耗 J2EE Server 较多的内存。然而 Stateful Session Bean 的优势却在于他能够维持使用者的状态。

12、Collection 和 Collections 的差别。

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。

13、&和&&的差别。

&是位运算符。表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

14、HashMap 和 Hashtable 的差别。

HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完毕了 Map 接口，主要差别在于 HashMap 同意空（null）键值（key）,因为非线程安全。效率上可能高于 Hashtable。
HashMap 同意将 null 作为一个 entry 的 key 或者 value。而 Hashtable 不同意。
HashMap 把 Hashtable 的 contains 方法去掉了。改成 containsvalue 和 containsKey。由于 contains 方法 easy 让人引起误解。
Hashtable 继承自 Dictionary 类。而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。
最大的不同是。Hashtable 的方法是 Synchronize 的，而 HashMap 不是。在多个线程訪问 Hashtable 时，不须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 採用的 hash/rehash 算法都大概一样，所以性能不会有非常大的差异。

15、final, finally, finalize 的差别。

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，表示总是运行。

finalize 是 Object 类的一个方法，在垃圾收集器运行的时候会调用被回收对象的此方法。能够覆盖此方法提供垃圾收集时的其它资源回收，比如关闭文件等。

16、sleep() 和 wait() 有什么差别?

sleep 是线程类（Thread）的方法，导致此线程暂停运行指定时间，给运行机会给其它线程，可是监控状态依旧保持。到时后会自己主动恢复。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，仅仅有针对此对象发出 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执行状态。

17、Overload 和 Override 的差别。Overloaded 的方法能否够改变返回值的类型?

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。假设在子类中定义某方法与其父类有同样的名称和参数，我们说该方法被重写 (Overriding)。

子类的对象使用这种方法时，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。假设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 的方法是能够改变返回值的类型。

18、error 和 exception 有什么差别?

error 表示恢复不是不可能但非常困难的情况下的一种严重问题。

比方说内存溢出。不可能指望程序能处理这种情况。

exception 表示一种设计或实现问题。也就是说。它表示假设程序执行正常。从不会发生的情况。

19、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。

假设数据将在线程间共享。比如正在写的数据以后可能被还有一个线程读到，或者正在读的数据可能已经被还有一个线程写过了，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须要花费非常长时间来运行的方法。而且不希望让程序等待方法的返回时，就应该使用异步编程，在非常多情况下採用异步途径往往更有效率。

20、abstract class 和 interface 有什么差别?

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。然而能够创建一个变量，其类型是一个抽象类，并让它指向详细子类的一个实例。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全部抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。

知道其行为的其他类能够在类中实现这些方法。

接口（interface）是抽象类的变体。

在接口中，全部方法都是抽象的。

多继承性可通过实现这种接口而获得。接口中的全部方法都是抽象的。没有一个有程序体。接口仅仅能够定义 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）全部这种接口的方法。

然后，它能够在实现了该接口的类的不论什么对象上调用接口的方法。因为有抽象类，它同意使用接口名作为引用变量的类型。

通常的动态联编将生效。

引用能够转换到接口类型或从接口类型转换，instanceof 运算符能够用来决定某对象的类是否实现了接口。

21、heap 和 stack 有什么差别。

栈是一种线形集合。其加入和删除元素的操作应在同一段完毕。

栈依照后进先出的方式进行处理。
堆是栈的一个组成元素

22、forward 和 redirect 的差别

forward 是 server 请求资源。server 直接訪问目标地址的 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 server 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 就是服务端依据逻辑,发送一个状态码,告诉浏览器又一次去请求那个地址。一般来说浏览器会用刚才请求的全部參数又一次请求。所以 session,request 參数都能够获取。

23、EJB 与 JAVA BEAN 的差别？

Java Bean 是可复用的组件，对 Java Bean 并没有严格的规范。理论上讲，不论什么一个 Java 类都能够是一个 Bean。但通常情况下。因为 Java Bean 是被容器所创建（如 Tomcat）的，所以 Java Bean 应具有一个无參的构造器，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。它是不能被跨进程訪问的。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 能够被远程訪问（跨进程、跨计算机）。但 EJB 必须被布署在诸如 Webspere、WebLogic 这种容器中，EJB 客户从不直接訪问真正的 EJB 组件。而是通过其容器訪问。EJB 容器是 EJB 组件的代理， EJB 组件由容器所创建和管理。客户通过容器来訪问真正的 EJB 组件。

24、Static Nested Class 和 Inner Class 的不同。

Static Nested Class 是被声明为静态（static）的内部类，它能够不依赖于外部类实例被

实例化。而通常的内部类须要在外部类实例化后才干实例化。

25、JSP 中动态 INCLUDE 与静态 INCLUDE 的差别？

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里的变化，适合用于包括动态页面。而且能够带参数。静态 INCLUDE 用 include 伪码实现,定不会检查所含文件的变化。适用于包括静态页面<%@ include file="included.htm" %>

26、什么时候用 assert。

assertion(断言)在软件开发中是一种经常使用的调试方式，非常多开发语言中都支持这样的机制。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；假设该值为 false，说明程序已经处于不对的状态下，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，在软件公布后，assertion 检查一般是关闭的。

27、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 easy 出现故障的地方，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 功能能够自己主动监测对象是否超过作用域从而达到自己主动回收内存的目的，Java 语言没有提供释放已分配内存的显示操作方法。

28、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，须要强制转换类型）
short s1 = 1; s1 += 1;（能够正确编译）

29、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.

30、String s = new String("xyz");创建了几个 String Object?

两个

31、EJB 包含（SessionBean,EntityBean）说出他们的生命周期。及怎样管理事务的？

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，

容器必须马上在 server 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。而且状态是持续的。仅仅要数据库中的数据存在，Entity beans 就一直存活。

而不是依照顾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 生命周期可以被容器或者 Beans 自己管理。
EJB 通过下面技术管理实务：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 server 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)?
接口能够继承接口。抽象类能够实现(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白的构造函数。

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 server 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。client 程序须要连接时。池驱动程序会返回一个未使用的池连接并将其表记为忙。

假设当前没有空暇连接，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毕后，池驱动程序将此连接表记为空暇。其它调用就能够使用这个连接。

37、abstract 的 method 是否可同一时候是 static,是否可同一时候是 native，是否可同一时候是 synchronized?

都不能

38、数组有没有 length()这种方法?

String 有没有 length()这种方法？

数组没有 length()这种方法，有 length 的属性。

String 有有 length()这种方法。

39、Set 里的元素是不能反复的，那么用什么方法来区分反复与否呢?

是用==还是 equals()? 它们有何差别?

Set 里的元素是不能反复的。那么用 iterator()方法来区分反复与否。

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，但能够被重载 Overloading。

41、能否够继承 String 类?

String 类是 final 类故不能够继承。

42、swtich 能否作用在 byte 上，能否作用在 long 上，能否作用在 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 会不会被运

行，什么时候被运行，在 return 前还是后?
会运行，在 return 前运行。

44、编程题：用最有效率的方法算出 2 乘以 8 等於几?
2 << 3

45、两个对象值同样(x.equals(y) == true)，但却可有不同的 hash code，这句话对不正确?
不正确，有同样的 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。那么这里究竟是值传递还是引用传递?
是值传递。Java 编程语言仅仅有值传递参数。

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。对象的内容能够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，其他线程是否可进入此对象的其他方法?

不能，一个对象的一个 synchronized 方法仅仅能由一个线程訪问。

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 仅仅有一个实例存在。

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,继而调用当中的方法。

```
public class Singleton {
private Singleton(){}
        //在自己内部定义自己一个实例，是不是非常奇怪？
        //注意这是 private 仅仅供内部调用
        private static Singleton instance = new Singleton();
        //这里提供了一个供外部訪问本 class 的静态方法，能够直接訪问
        public static Singleton getInstance() {
            return instance;
        }
    }
```

另外一种形式:

```
public class Singleton {
    private static Singleton instance = null;
    public static synchronized Singleton getInstance() {
    //这种方法比上面有所改进，不用每次都进行生成对象，仅仅是第一次
    //使用时生成实例，提高了效率！
    if (instance==null)
        instance＝new Singleton();
return instance;       }
}
```

其它形式:

定义一个类，它的构造函数为 private 的，全部方法为 static 的。

一般觉得第一种形式要更加安全些

49、Java 的接口和 C++的虚类的同样和不同处。

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。由于接口中没有不论什么实现代码。当一个类实现了接口以后，该类要实现接口里面全部的方法和属性。而且接口里面的属性在默认状态以下都是 public static,全部方法默认情况下是 public.一个类能够实现多个接口。

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。违反语义规则包含 2 种情况。

一种是 JAVA 类库内置的语义检查。比如数组下标越界,会引发 IndexOutOfBoundsException;訪问 null 的对象时会引发 NullPointerException。

还有一种情况就是 JAVA 同意程序猿扩展这样的语义检查，程序猿能够创建自己的异常。并自由选择在何时用 throwkeyword 引发异常。

全部的异常都是 java.lang.Thowable 的子类。

51、垃圾回收的长处和原理。

并考虑 2 种回收机制。

Java 语言中一个显着的特点就是引入了垃圾回收机制。使 c++程序猿最头疼的内存管理的问题迎刃而解，它使得 Java 程序猿在编敲代码的时候不再须要考虑内存管理。因为有个垃圾回收机制。 Java 中的对象不再有"作用域"的概念，仅仅有对象的引用才有"作用域"。垃圾回收能够有效的防止内存泄露，有效的使用能够使用的内存。

垃圾回收器一般是作为一个单独的低级别的线程执行，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，程序猿不能实时的调用垃圾回收器对某个对象或全部对象进行垃圾回收。

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。

52、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。而且释放所持有的对象的 lock。
sleep():使一个正在执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，并且不是按优先级。
Allnotity():唤醒全部处入等待状态的线程，注意并非给全部唤醒线程一个对象的锁，而是让它们竞争。

53、你所知道的集合类都有哪些？主要方法？
最经常使用的集合类是 List 和 Map。 List 的详细实现包含 ArrayList 和 Vector，它们是可变大小的列表，比较适合构建、存储和操作不论什么类型对象的元素列表。 List 适用于按数值索引訪问元素的情形。
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"），当中每一个键映射到一个值。

54、描写叙述一下 JVM 载入 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 执行时系统组件。它负责在执行时查找和装入类文件的类。

55、char 型变量中能不能存贮一个中文汉字?为什么?

可以定义成为一个中文的，由于 java 中以 unicode 编码，一个 char 占 16 个字节，所以放一个中文是没问题的

56、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，各自是继承 Thread 类与实现 Runnable 接口
同步的实现方面有两种，各自是 synchronized,wait 与 notify

57、JSP 的内置对象及方法。
request 表示 HttpServletRequest 对象。它包括了有关浏览器请求的信息。而且提供了几个用于获取 cookie, header, 和 session 数据的实用的方法。

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，而且包装了通用的 servlet 相关功能的方法。

session 表示一个请求的 javax.servlet.http.HttpSession 对象。Session 能够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

58、线程的基本概念、线程的基本状态以及状态之间的关系

线程指在程序运行过程中，可以运行程序代码的一个运行单位，每一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各自是：执行、就绪、挂起、结束。

59、JSP 的经常使用指令

<%@page language="java" contenType="text/html;charset=gb2312"
session="true" buffer="64kb" autoFlush="true" isThreadSafe="true" info="text"
errorPage="error.jsp" isErrorPage="true" isELIgnored="true"
pageEncoding="gb2312" import="java.sql.*"%>

isErrorPage(能否使用 Exception 对象)，isELIgnored(是否忽略表达式)

<%@include file="filename"%>

<%@taglib prefix="c"uri="http://....."%>

60、什么情况下调用 doGet()和 doPost()？

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()。

61、servlet 的生命周期

web 容器载入 servlet，生命周期開始。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，依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。

62、怎样现实 servlet 的单线程模式

<%@ page isThreadSafe="false"%>

63、页面间对象传递的方法

request，session。application，cookie 等

64、JSP 和 Servlet 有哪些同样点和不同点，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 最基本的不同点在于，Servlet 的应用逻辑是在 Java 文件里，而且全然从表示层中的 HTML 里分离开来。而 JSP 的情况是 Java 和 HTML 能够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。

65、四种会话跟踪技术
会话作用域 ServletsJSP 页面描写叙述
page 否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（能够带有不论什么的 include 指令，可是没有 include 动作）表示。这既包含 servlet 又包含被编译成 servlet 的 JSP 页面
request 是是代表与 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（因为 forward 指令和 include 动作的关系）
session 是是代表与用于某个 Web 客户机的一个用户体验相关的对象和属性。一个 Web 会话能够也常常会跨越多个客户机请求
application 是是代表与整个 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。包含多个页面、请求和会话的一个全局作用域

66、Request 对象的主要方法：
setAttribute(String name,Object)：设置名字为 name 的 request 的参数值
getAttribute(String name)：返回由 name 指定的属性值
getAttributeNames()：返回 request 对象全部属性的名字集合，结果是一个枚举的实例
getCookies()：返回 client 的全部 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)：返回指定名字的 request Header 的全部值，结果是一个枚举的实例
getHeaderNames()：返回所以 request Header 的名字。结果是一个枚举的实例
getInputStream()：返回请求的输入流。用于获得请求中的数据
getMethod()：获得 client 向 server 端传送数据的方法
getParameter(String name)：获得 client 传送给 server 端的有 name 指定的參数值
getParameterNames()：获得 client 传送给 server 端的全部參数的名字，结果是一个枚举的实例
getParameterValues(String name)：获得有 name 指定的參数的全部值
getProtocol()：获取 client 向 server 端传送数据所根据的协议名称
getQueryString()：获得查询字符串
getRequestURI()：获取发出请求字符串的 client 地址
getRemoteAddr()：获取 client 的 IP 地址
getRemoteHost()：获取 client 的名字
getSession([Boolean create])：返回和请求相关 Session
getServerName()：获取 server 的名字
getServletPath()：获取 client 所请求的脚本文件的路径
getServerPort()：获取 server 的 port 号

removeAttribute(String name)：删除请求中的一个属性

67、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，包含 JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

68、我们在 web 应用开发过程中常常遇到输出某种编码的字符，如 iso8859-1 等，怎样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;
}
```

69、简述逻辑操作(&,|,^)与条件操作(&&,||)的差别。

差别主要答两点：

a.条件操作仅仅能操作布尔型的,而逻辑操作不仅能够操作布尔型,并且能够操作数值型

b.逻辑操作不会产生短路

70、XML 文档定义有几种形式？它们之间有何本质差别？解析 XML 文档有哪几种方式？

a: 两种形式 dtd　schema，b: 本质差别:schema 本身是 xml 的。能够被 XML 解析器解析(这也是从 DTD 上发展 schema 的根本目的)。c:有 DOM,SAX,STAX 等

DOM:处理大型文件时其性能下降的很厉害。这个问题是由 DOM 的树结构所造成的。这样的结构占用的内存较多。并且 DOM 必须在解析文件之前把整个文档装入内存,适合对 XML 的随机訪问

SAX:不现于 DOM,SAX 是事件驱动型的 XML 解析方式。它顺序读取 XML 文件，不须要一次所有装载整个文件。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 的顺序訪问

STAX:Streaming API for XML (StAX)

71、简述 synchronized 和 java.util.concurrent.locks.Lock 的异同？

主要同样点：Lock 能完毕 synchronized 所实现的全部功能

主要不同点：Lock 有比 synchronized 更精确的线程语义和更好的性能。synchronized 会自己主动释放锁。而 Lock 一定要求程序猿手工释放，而且必须在 finally 从句中释放。

72、EJB 的角色和三个对象

一个完整的基于 EJB 的分布式计算结构由六个角色组成。这六个角色能够由不同的开发商提供。每一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。

这六个角色各自是 EJB 组件开发人员（Enterprise Bean Provider）、应用组合者（Application Assembler）、部署者（Deployer）、EJB server 提供者（EJB Server Provider）、EJB 容器提供者（EJB Container Provider）、系统管理员（System Administrator）
三个对象是 Remote（Local）接口、Home（LocalHome）接口，Bean 类

73、EJB 容器提供的服务
主要提供声明周期管理、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。

74、EJB 规范规定 EJB 中禁止的操作有哪些？
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 server 功能。
4.不能对静态属生存取，
5.不能使用 IO 操作直接存取文件系统。
6.不能载入本地库.，
7.不能将 this 作为变量和返回，
8.不能循环调用。

75、remote 接口和 home 接口主要作用
remote 接口定义了业务方法。用于 EJBclient 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

76、bean 实例的生命周期
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，通常包括创建实例。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。

77、EJB 的激活机制
以 Stateful Session Bean 为例：其 Cache 大小决定了内存中能够同一时候存在的 Bean 实例的数量。依据 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 client 调用某个 EJB 实例业务方法时。假设相应 EJB Object 发现自己没有绑定相应的 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。状态变迁前会调用相应的 ejbActive 和 ejbPassivate 方法。

78、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

79、客服端调用 EJB 对象的几个基本步骤
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。

80、怎样给 weblogic 指定大小的内存?

在启动 Weblogic 的脚本中（位于所在 Domian 相应 server 文件夹下的 startServerName）。添加 set MEM_ARGS=-Xms32m -Xmx200m。能够调整最小内存为 32M，最大 200M
81、怎样设定的 weblogic 的热启动模式(开发模式)与产品公布模式?
能够在管理控制台中改动相应 server 的启动模式为开发或产品模式之中的一个。或者改动服务的启动文件或者 commenv 文件，添加 set PRODUCTION_MODE=true。

82、怎样启动时不需输入 username 与 password?
改动服务启动文件。添加 WLS_USER 和 WLS_PW 项。也能够在 boot.properties 文件里添加加密过的 username 和 password.

83、在 weblogic 管理制台中对一个应用域(或者说是一个站点,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里?
保存在此 Domain 的 config.xml 文件里，它是 server 的核心配置文件。

84、说说 weblogic 中一个 Domain 的缺省文件夹结构?比方要将一个简单的 helloWorld.jsp 放入何文件夹下,然的在浏览器上就可打入 http://主机:port 号//helloword.jsp 就能够看到执行结果了? 又比方这当中用到了一个自己写的 javaBean 该怎样办?

Domain 文件夹 server 文件夹 applications。将应用文件夹放在此文件夹下将能够作为应用訪问。假设是 Web 应用，应用文件夹须要满足 Web 应用文件夹要求，jsp 文件能够直接放在应用文件夹中，Javabean 须要放在应用文件夹的 WEB-INF 文件夹的 classes 文件夹中，设置 server 的缺省应用将能够实如今浏览器上无需输入应用名。

85、在 weblogic 中公布 ejb 需涉及到哪些配置文件
不同类型的 EJB 涉及的配置文件不同，都涉及到的配置文件包含
ejb-jar.xml,weblogic-ejb-jar.xmlCMP 实体 Bean 一般还须要
weblogic-cmp-rdbms-jar.xml

86、怎样在 weblogic 中进行 ssl 配置与 client 的认证配置或说说 j2ee(标准)进行 ssl 的配置
缺省安装中使用 DemoIdentity.jks 和 DemoTrust.jks KeyStore 实现 SSL，须要配置 server 使用 Enable SSL，配置其 port，在产品模式下须要从 CA 获取私有密钥和数字证书，创建 identity 和 trust keystore，装载获得的密钥和数字证书。能够配置此 SSL 连接是单向还是双向的。

87、怎样查看在 weblogic 中已经公布的 EJB?
能够使用管理控制台。在它的 Deployment 中能够查看全部已公布的 EJB

88、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)和同意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执行，为不同的操作系统开发。

89、说说你所熟悉或听说过的 j2ee 中的几种经常使用模式?及对设计模式的一些看法
Session Facade Pattern：使用 SessionBean 訪问 EntityBean
Message Facade Pattern：实现异步调用
EJB Command Pattern：使用 Command JavaBeans 代替 SessionBean，实现轻量级訪问
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 类实现同样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。

90、说说在 weblogic 中开发消息 Bean 时的 persistent 与 non-persisten 的区别
persistent 方式的 MDB 能够保证消息传递的可靠性,也就是假设 EJB 容器出现故障而 JMSserver 依旧会将消息在此 MDB 可用的时候发送过来，而 non－persistent 方式的消息将被丢弃。

91、Servlet 运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()

92、j2ee 经常使用的设计模式？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（訪问者模式），
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：工厂模式是一种常常被使用到的模式。依据工厂模式实现的类能够依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而且实现了同样的方法，可是

这些方法针对不同的数据进行了不同的操作。首先须要定义一个基类。该类的子类通过不同的方法实现了基类中的方法。然后须要定义一个工厂类。工厂类能够依据条件生成不同的子类实例。当得到子类的实例后,开发者能够调用基类中的方法而不必考虑究竟返回的是哪一个子类的实例。

93、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 接口不须要直接实现。他们的实现代码是由 server 产生的。程序执行中相应实现类会作为相应接口类型的实例被使用。

94、排序都有哪几种方法？请列举。用 JAVA 实现一个高速排序。

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、高速排序），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）

高速排序的伪代码。
/ /使用高速排序方法对 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
把余下的元素切割为两段 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
递归地使用高速排序方法对 left 进行排序
递归地使用高速排序方法对 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

95、请对下面在 J2EE 中经常使用的名词进行解释(或简单描写叙述)
web 容器：给处于当中的应用程序组件（JSP，SERVLET）提供一个环境。使 JSP,SERVLET 直接更容器中的环境变量接口交互,不必关注其他系统问题。主要有 WEBserver 来实现。比如：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 WEBserver 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。他提供给执行在当中的组件 EJB 各种管理功能。

仅仅要满足 J2EE 规范的 EJB 放入该容器。立即就会被容器进行高效率的管理。

而且能够通过现成的接口来获得系统级别的服务。比如邮件服务、事务管理。
JNDI：（Java Naming & Directory Interface）JAVA 命名文件夹服务。主要提供的功能是：提供一个文件夹系统。让其他各地的应用程序在其上面留下自己的索引，从而满足高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。包含点对点和广播。
JTA：（Java Transaction API）JAVA 事务服务。

提供各种分布式事务服务。应用程序仅仅需调用其提供的接口就可以。
JAF：（Java Action FrameWork）JAVA 安全认证框架。提供一些安全控制方面的框架。让开发人员通过各种部署和自己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。比如，远程有一台计算机上执行一个程序，它提供股票分析服务，我们能够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干在异构的系统之间进行通信。

RMI 是 JAVA 特有的。

96、JAVA 语言怎样进行异常处理，keyword：throws,throw,try,catch,finally 分别代表什么意义？在 try 块中能够抛出异常吗？
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。每一个异常都是一个对象，它是 Throwable 类或其他子类的实例。

当一个方法出现异常后便抛出一个异常对象。该对象中包括有异常信息，调用这个对象的方法能够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。

普通情况下是用 try 来运行一段程序。假设出现异常。系统会抛出（throws）一个异常，这时候你能够通过它的类型来捕捉（catch）它，或最后（finally）由缺省处理器来处理。

用 try 来指定一块预防全部"异常"的程序。紧跟在 try 程序后面，应包括一个 catch 子句来指定你想要捕捉的"异常"的类型。

throw 语句用来明白地抛出一个"异常"。
throws 用来标明一个成员函数可能抛出的各种"异常"。
Finally 为确保一段代码无论发生什么"异常"都被运行一段代码。
能够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还有一个 try 语句保护其它代码。每当遇到一个 try 语句，"异常"的框架就放到堆栈上面，直到全部的 try 语句都完毕。假设下一级的 try 语句没有对某种"异常"进行处理。堆栈就会展开，直到遇到有处理这样的"异常"的 try 语句。

97、一个".java"源文件里能否够包含多个类（不是内部类）？有什么限制？
能够。必须仅仅有一个类名与文件名称同样。

98、MVC 的各个部分都有那些技术来实现?怎样实现?
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（通常是一个 Servlet）。通过这样的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。这些组件能够进行交互和重用。

99、java 中有几种方法能够实现一个线程？用什么 keyword 修饰同步方法? stop()和 suspend()方法为何不推荐使用？
有两种实现方法，各自是继承 Thread 类与实现 Runnable 接口
用 synchronizedkeyword 修饰同步方法
反对使用 stop()，是由于它不安全。它会解除由线程获取的全部锁定，并且假设对象处于一种不连贯状态，那么其它线程能在那种状态下检查和改动它们。结果非常难检查出真正的问题所在。

suspend()方法 easy 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，其它不论什么线程都不能訪问锁定的资源，除非被"挂起"的线程恢复执行。对不论什么线程来说，假设它们想恢复目标线程，同一时候又试图使用不论什么一个锁定的资源，就会造成死锁。所以不应该使用 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()又一次启动线程。

100、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。请说出他们各自是哪些类？
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 包中还有更多的流，主要是为了提高性能和使用方便。

101、java 中会存在内存泄漏吗，请简单描写叙述。

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。

102、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。

103、垃圾回收器的基本原理是什么？垃圾回收器能够立即回收内存吗？有什么办法主动通知虚拟机进行垃圾回收？
对于 GC 来说，当程序猿创建对象时。GC 就開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)中的全部对象。

通过这样的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。能够。程序猿能够手动运行 System.gc()，通知 GC 运行，可是 Java 语言规范并不保证 GC 一定会运行。

104、静态变量和实例变量的差别？
static i = 10; //常量
class A a; a.i =10;//可变

105、什么是 java 序列化。怎样实现 java 序列化？
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。能够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：将须要被序列化的类实现 Serializable 接口。该接口没有须要实现的方法。implements Serializable 仅仅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)方法就能够将参数为 obj 的对象写出(即保存其状态)，要恢复的话则用输入流。

106、能否够从一个 static 方法内部发出对非 static 方法的调用？
不能够,假设当中包括对象的 method()；不能保证对象初始化.

107、写 clone()方法时，通常都有一行代码。是什么？
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。

108、在 JAVA 中，怎样跳出当前的多重嵌套循环？
用 break; return 方法。

109、List、Map、Set 三个接口，存取元素时，各有什么特点？
List 以特定次序来持有元素，可有反复元素。Set 无法拥有反复元素,内部排序。

Map 保存 key-value 值。value 可多值。

110、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).在这种一个应用系统中，可依照功能划分为不同的组件。这些组件又可在不同计算机上，

而且处于对应的层次(tier)中。

所属层次包含客户层(clietn tier)组件,web 层和组件,Business 层和组件,企业信息系统(EIS)层。

111、UML 方面

标准建模语言 UML。用例图,静态图(包含类图、对象图和包图),行为图,交互图(顺序图,合作图),实现图。

112、说出一些经常使用的类，包，接口，请各举 5 个
经常使用的类：
BufferedReader BufferedWriter FileReader FileWirter String Integer
经常使用的包：java.lang java.awt java.io java.util java.sql
经常使用的接口：Remote List Map Document NodeList

113、开发中都用到了那些设计模式?

用在什么场合?

每一个模式都描写叙述了一个在我们的环境中不断出现的问题,然后描写叙述了该问题的解决方式的核心。通过这样的方式,你能够无数次地使用那些已有的解决方式,无需在反复同样的工作。

主要用到了 MVC 的设计模式。用来开发 JSP/Servlet 或者 J2EE 的相关应用。简单工厂模式等。

114、jsp 有哪些动作?作用各自是什么?

JSP 共同拥有下面 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：依据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记。

115、Anonymous Inner Class (匿名内部类) 能否够 extends(继承)其他类，能否够 implements(实现)interface(接口)?

能够继承其它类或完毕其它接口。在 swing 编程中经常使用此方式。

116、应用 server 与 WEB SERVER 的差别?
应用 server：Weblogic、Tomcat、Jboss
WEB SERVER：IIS、 Apache

117、BS 与 CS 的联系与差别。

C/S 是 Client/Server 的缩写。server 通常採用高性能的 PC、工作站或小型机，并採用大型数据库系统。如 Oracle、Sybase、Informix 或 SQL Server。client 须要安装专用的 client 软件。

B/S是Brower/Server的缩写，客户机上仅仅要安装一个浏览器（Browser），如Netscape Navigator或Internet Explorer，server安装Oracle、Sybase、Informix或 SQL Server等数据库。在这样的结构下。用户界面全然通过WWW浏览器实现，一部分事务逻辑在前端实现，可是主要事务逻辑在server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 差别：

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server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，一般仅仅要有操作系统和浏览器即可

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非常强．一般高度机密的信息系统採用C/S 结构适宜．能够通过B/S公布部分可公开信息.

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 程序能够更加注重流程，能够对权限多层次校验，对系统执行速度能够较少考虑.

B/S 对安全以及訪问速度的多重的考虑，建立在须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 程序能够不可避免的总体性考虑，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．可以相对较好的重用.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 程序因为总体性，必须总体考察，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能够实现升级.

6．处理问题不同

C/S 程序能够处理用户面固定，而且在同样区域，安全要求高需求，与操作系统相关．应该都是同样的系统

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,对程序猿普遍要求较高

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．而且大部分难度减低,减低开发成本.

8．信息流不同

C/S 程序通常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。

118、LINUX 下线程。GDI 类的解释。

LINUX 实现的就是基于核心轻量级进程的"一对一"线程模型，一个线程实体相应一个核心轻量级进程。而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库。

119、STRUTS 的应用(如 STRUTS 架构)

Struts 是採用 Java Servlet/JavaServer Pages 技术。开发 Web 应用程序的开放源代码的 framework。採用 Struts 能开发出基于 MVC(Model-View-Controller)设计模式的应用构架。 Struts 有例如以下的主要功能：一.包括一个 controller servlet，能将用户的请求发送到对应的 Action 对象。

二.JSP 自由 tag 库，而且在 controller servlet 中提供关联支持，帮助开发员创建交互式表单应用。三.提供了一系列有用对象：XML 处理、通过 Java reflection APIs 自己主动处理 JavaBeans 属性、国际化的提示和消息。

120、Jdo 是什么?

JDO 是 Java 对象持久化的新的规范。为 java data object 的简称,也是一个用于存取某种数据仓库中的对象的标准化 API。

JDO 提供了透明的对象存储。因此对开发者来说。存储数据对象全然不须要额外的代码（如 JDBC API 的使用）。这些繁琐的例行工作已经转移到 JDO 产品提供商身上，使开发者解脱出来。从而集中时间和精力在业务逻辑上。另外，JDO 非常灵活。由于它能够在不论什么数据底层上执行。

JDBC 仅仅是面向关系数据库（RDBMS）JDO 更通用，提供到不论什么数据底层的存储功能，比方关系数据库、文件、XML 以及对象数据库（ODBMS）等等，使得应用可移植性更强。

121、内部类能够引用他包括类的成员吗？有没有什么限制？

一个内部类对象能够訪问创建它的外部类对象的内容

122、WEB SERVICE 名词解释。JSWDL 开发包的介绍。

JAXP、JAXM 的解释。

SOAP、UDDI,WSDL 解释。

Web ServiceWeb Service 是基于网络的、分布式的模块化组件。它运行特定的任务，遵守详细的技术规范。这些规范使得 Web Service 能与其它兼容的组件进行互操作。

JAXP(Java API for XML Parsing) 定义了在 Java 中使用 DOM, SAX, XSLT 的通用的接口。

这样在你的程序中你仅仅要使用这些通用的接口。当你须要改变详细的实现时候也不须要改动代码。

JAXM(Java API for XML Messaging) 是为 SOAP 通信提供訪问方法和传输机制的 API。

WSDL 是一种 XML 格式，用于将网络服务描写叙述为一组端点。这些端点对包括面向文档信息或面向过程信息的消息进行操作。这样的格式首先对操作和消息进行抽象描写叙述，然后将其绑定到详细的网络协议和消息格式上以定义端点。相关的详细端点即组合成为抽象端点（服务）。

SOAP 即简单对象訪问协议(Simple Object Access Protocol)，它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准；UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范，同一时候也包括一组使企业能将自身提供的 Web Service 注册，以使别的企业可以发现的訪问协议的实现标准。

123、设计 4 个线程，当中两个线程每次对 j 添加 1，另外两个线程对 j 每次降低 1。

写出程序。

下面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();
}
 }
  }
}
```

124、Java 有没有 goto?

java 中的保留字，如今没有在 java 中使用。

125、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执行状态，这意味着它能够由 JVM 调度并执行。这并不意味着线程就会马上执行。run()方法能够产生必须退出的标志来停止一个线程。