**Java 就问 对象 类 super this static final 重载 重写 线程 集合**

**1. J2EE是什么？它包括哪些技术？**

解答：从整体上讲，J2EE是使用Java技术开发企业级应用的工业标准，它是Java技术不断适应和促进企业级应用过程中的产物。适用于企业级应用的J2EE，提供一个平台独立的、可移植的、多用户的、安全的和基于标准的企业级平台，从而简化企业应用的开发、管理和部署。J2EE是一个标准，而不是一个现成的产品。

主要包括以下这些技术：

1) Servlet Servlet是Java平台上的CGI技术。Servlet在服务器端运行，动态地生成Web页面。与传统的CGI和许多其它类似CGI的技术相比，Java Servlet具有更高的效率并更容易使用。对于Servlet，重复的请求不会导致同一程序的多次转载，它是依靠线程的方式来支持并发访问的。 2)JSP JSP(Java Server Page)是一种实现普通静态HTML和动态页面输出混合编码的技术。从这一点来看，非常类似Microsoft ASP、PHP等技术。借助形式上的内容和外观表现的分离，Web页面制作的任务可以比较方便地划分给页面设计人员和程序员，并方便地通过JSP来合成。在运行时态，JSP将会被首先转换成Servlet，并以Servlet的形态编译运行，因此它的效率和功能与Servlet相比没有差别，一样具有很高的效率。 3) EJB EJB定义了一组可重用的组件：Enterprise Beans。开发人员可以利用这些组件，像搭积木一样建立分布式应用。 4)JDBC

JDBC(Java Database Connectivity，Java数据库连接)API是一个标准SQL(Structured Query Language，结构化查询语言)数据库访问接口，它使数据库开发人员能够用标准Java API编写数据库应用程序。JDBC API主要用来连接数据库和直接调用SQL命令执行各种SQL语句。利用JDBC API可以执行一般的SQL语句、动态SQL语句及带IN和OUT参数的存储过程。Java中的JDBC相当于Microsoft平台中的ODBC(Open Database Connectivity)。

**2. 什么是Web容器?**

解答：容器就是一种服务程序，在服务器一个端口就有一个提供相应服务的程序，而这个程序就是处理从客户端发出的请求，如JAVA中的Tomcat容器，ASP的IIS或PWS都是这样的容器。

**3.运行时异常与一般异常有何异同？**

解答：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。

**1.Hibernate中：不看数据库，不看XML文件，不看查询语句，怎么样能知道表结构？**

解答：可以看与XML文件对应的域模型。

**2.存储过程和函数的区别**

解答：

从参数的返回情况来看：

如果返回多个参数值最好使用存储过程，如果只有一个返回值的话可以使用函数。

从调用情况来看：

如果在SQL语句（DML或SELECT）中调用的话一定是存储函数或存储的封装函数不可以是存储过程，但调用存储函数的时候还有好多限制以及函数的纯度等级的问题，如果是在过程化语句中调用的话，就要看你要实现什么样的功能。函数一般情况下是用来计算并返回一个计算结果而存储过程一般是用来完成特定的数据操作（比如修改、插入数据库表或执行某些DDL语句等等），所以虽然他们的语法上很相似但用户在使用他们的时候所需要完成的功能大部分情况下是不同的。

**3.试述数据库完整保护的主要任务和措施。**

解答：数据库的完整性保护也就是数据库中数据正确性的维护。数据库完整性包括三个内容：实体完整性规则，参照物完整性规则以及用户定义完整性规则。 前两个是有DBMS自动处理。

实体完整性规则是说针对于基表中的关键字中属性值不能为空值，是数据库完整性的基本要求，主关键字和元组的唯一性对应。

参照物完整性规则是不允许引用不存在的元组：即基表中的外关键字要么为空，要么关联基表中必存在元组。

用户定义完整性规则针对具体的数据环境由用户具体设置的规则，它反应了具体应用中的语义要求。 一个完整性规则一般由下面三部分组成:完整性约束条件设置,完整性约束条件的检查以及完整性约束条件的处理.后两部分在数据库中一般有相应的模块处理。另外触发器也可以做完整性的保护，但触发器大量用于主动性领域。

**1.HashMap与TreeMap的区别？**

解答：HashMap通过hashcode对其内容进行快速查找，而TreeMap中所有的元素都保持着某种固定的顺序，如果你需要得到一个有序的结果你就应该使用TreeMap（HashMap中元素的排列顺序是不固定的）。

**2.ArrayList和Vector的区别？**

解答：同步性:Vector是线程安全的，也就是说是同步的，而ArrayList是线程不安全的，不是同步的；数据增长:当需要增长时,Vector默认增长为原来一培，而ArrayList却是原来的一半。

**3.HashMap和Hashtable的区别？**

解答：HashMap是Hashtable的轻量级实现（非线程安全的实现），他们都实现了Map接口，主要区别

在于HashMap允许空（null）键值（key）,由于非线程安全，效率上高于Hashtable。HashMap允许将null作为一个entry的key或者value，而Hashtable不允许。HashMap把Hashtable的contains方法去掉了，改成containsvalue和containsKey。因为contains方法容易让人引起误解。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，Hastable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 就必须为之提供同步。

**1.事务是什么？有哪些属性，并简要说明这些属性的含义。**

解答：事务(Transaction)是访问并可能更新数据库中各种数据项的一个程序执行单元(unit)。事务通常由高级数据库操纵语言或编程语言（如SQL，C++或Java）书写的用户程序的执行所引起，并用形如begin transaction和end transaction语句（或函数调用）来界定。事务由事务开始(begin transaction)和事务结束(end transaction)之间执行的全体操作组成。

事务应该具有4个属性：原子性、一致性、隔离性、持续性。这四个属性通常称为ACID特性。

原子性（atomicity）。一个事务是一个不可分割的工作单位，事务中包括的诸操作要么都做，要么都不做。

一致性（consistency）。事务必须是使数据库从一个一致性状态变到另一个一致性状态。一致性与原子性是密切相关的。

隔离性（isolation）。一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。

持久性（durability）。持续性也称永久性（permanence），指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其有任何影响。

**2.Collection 和 Collections的区别？**

解答：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。

**3.HashMap与TreeMap的区别？**

解答：HashMap通过hashcode对其内容进行快速查找，而TreeMap中所有的元素都保持着某种固定的顺序，如果你需要得到一个有序的结果你就应该使用TreeMap（HashMap中元素的排列顺序是不固定的）。

**1.请说出 ArrayList,Vector, LinkedList 的存储性能和特性**

解答：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。

**2.简单说明什么是递归？什么情况会使用？并使用 java 实现一个简单的递归程序。**

解答：

1）递归做为一种算法在程序设计语言中广泛应用.是指函数/过程/子程序在运行过程中直接或间接调

用自身而产生的重入现象。

2）递归算法一般用于解决三类问题：

a.数据的定义是按递归定义的。(Fibonacci（斐波那契）函数)

b.问题解法按递归算法实现。(回溯)

c.数据的结构形式是按递归定义的。(树的遍历，图的搜索)

3）.这是一个排列的例子，它所做的工作是将输入的一个字符串中的所有元素进行排序并输出，例如：

你给出的参数是"abc" 则程序会输出：

abc

acb

bac

bca

cab

cba

a.算法的出口在于：ｌｏｗ＝ｈｉｇｈ也就是现在给出的排列元素只有一个时。

b.算法的逼近过程：先确定排列的第一位元素，也就是循环中ｉ所代表的元素，

然后ｌｏｗ＋１开始减少排列元素，如此下去，直到ｌｏｗ＝ｈｉｇｈ

public class Foo {

public static void main(String[] args) {

permute("abc");

}

public static void permute(String str) {

char[] strArray = str.toCharArray();

permute(strArray, 0, strArray.length - 1);

}

public static void permute(char[] list, int low, int high) {

int i;

if (low == high) {

String cout = "";

for (i = 0; i <= high; i++)

cout += list[i];

System.out.println(cout);

} else {

for (i = low; i <= high; i++) {

char temp = list[low];

list[low] = list[i];

list[i] = temp;

permute(list, low + 1, high);

temp = list[low];

list[low] = list[i];

list[i] = temp;

}

}

}

}

**3.列出自己常用的 jdk 包**

解答：JDK 常用的 package

java.lang： 这个是系统的基础类，比如 String 等都是这里面的，这个 package 是唯一一个可以不

用 import 就可以使用的 Package

java.io: 这里面是所有输入输出有关的类，比如文件操作等

java.net: 这里面是与网络有关的类，比如 URL,URLConnection 等。

java.util : 这个是系统辅助类，特别是集合类 Collection,List,Map 等。

java.sql: 这个是数据库操作的类，Connection, Statememt，ResultSet 等

**1.列出自己常用的jdk中的数据结构**

解答：线性表，链表，哈希表是常用的数据结构。

**2.List、Map、Set三个接口存储元素时各有什么特点？**

解答：

1）List是有序的Collection，使用此接口能够精确的控制每个元素插入的位置。用户能够使用索引（元素在List中的位置，类似于数组下标）来访问List中的元素，这类似于Java的数组。

2）Set是一种不包含重复的元素的Collection，即任意的两个元素e1和e2都有e1.equals(e2)=false，Set最多有一个null元素。

3）Map接口 ：请注意，Map没有继承Collection接口，Map提供key到value的映射

**3.简述基于Struts框架Web应用的工作流程**

解答：在web应用启动时就会加载初始化ActionServlet,ActionServlet从struts-config.xml文件中读取配置信息,把它们存放到各种配置对象中， 当ActionServlet接收到一个客户请求时,将执行如下流程. 1)检索和用户请求匹配的ActionMapping实例,如果不存在,就返回请求路径无效信息; 2)如果ActionForm实例不存在,就创建一个ActionForm对象,把客户提交的表单数据保存到ActionForm对象中; 3)根据配置信息决定是否需要表单验证.如果需要验证,就调用ActionForm的validate()方法; 4)如果ActionForm的validate()方法返回null或返回一个不包含ActionMessage的ActionErrors对象, 就表示表单验证成功; 5)ActionServlet根据ActionMapping所包含的映射信息决定将请求转发给哪个Action,如果相应的Action实例不存在,就先创建这个实例,然后调用Action的execute()方法; 6)Action的execute()方法返回一个ActionForward对象,ActionServlet在把客户请求转发给ActionForward对象指向的JSP组件; 7)ActionForward对象指向JSP组件生成动态网页,返回给客户;

**1.在项目中用过Spring的哪些方面？及用过哪些Ajax框架？**

解答：在项目使用过Spring IOC ，AOP，DAO，ORM，还有上下文环境。

在项目使用过Ext,Juery等Ajax框架.

**2.abstract class和interface有什么区别?**

解答：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。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public static final的。一个类可以实现多个接口，当类实现接口时，必须实现接口的所有方法。抽象类在Java语言中表示的是一种单继承的关系，对于interface 来说则不然，并不要求interface的实现者和interface定义在概念本质上是一致的，仅仅是实现了interface定义的契约而已。；抽象类中可以定义自己的成员变量，也可以包含非抽象的方法，而在接口中只能有静态的常量，所有方法必须是抽象的；实现抽象类时可以只实现其中的部分方法，而要是实现一个接口的话就必须实现这个接口中的所有抽象方法。

**3.MVC模式中M，V，C每个代表意义，并简述在Struts中MVC的表现方式。**

解答：

MVC是Model-View-Controller 的缩写，Model代表的是应用的业务逻辑（通过JavaBean，EJB组件实现），View 是应用的表示层（由JSP页面产生）Controller是通过应用的处理过程控制，（一般是一个servert）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。

在Struts框架中Controller功能由ActionServlet和ActionMapping对象构成，核心是一个Servlet类型的对象ActionServlet，它用来接收客户端的请求。ActionServlet包括一组基于配置的ActionMapping对象，每个ActionMapping对象实现了一个请求到一个具体的Model部分的Action处理器对象之间的映射。Model部分由Action和ActionForm对象构成。所有的Action处理器对象都是开发者从Struts的Action类派生的子类。Action处理器对象封装了具体的处理逻辑，调用业务逻辑模块，并且把响应提交到合适的View组件以产生响应。Struts提供的ActionForm组件对象可以通过定义属性描述客户端表单数据，开发者可以从它派生子类对象，并利用它和Struts提供的自定义标记库相结合，可以实现对客户端的表单数据的良好封装和支持，Action处理器对象可以直接对它进行读写，而不再需要和request、response对象进行数据交互。通过ActionForm组件对象实现了对View和Model之间交互的支持（View部分是通过JSP技术实现的）。Struts提供了自定义的标记库，通过这些自定义标记库可以非常容易地和系统的Model部分交互，通过使用这些自定义标记库创建的

JSP表单，可以实现对Model部分中的ActionForm的映射，完成对用户数据的封装。

**1.ArrayList和Vector的区别？**

解答：同步性:Vector是线程安全的，也就是说是同步的，而ArrayList是线程不安全的，不是同步的；数据增长:当需要增长时,Vector默认增长为原来一培，而ArrayList却是原来的一半。

**2.HashMap和Hashtable的区别？**

解答：HashMap是Hashtable的轻量级实现（非线程安全的实现），他们都实现了Map接口，主要区别

在于HashMap允许空（null）键值（key）,由于非线程安全，效率上高于Hashtable。HashMap允许将null作为一个entry的key或者value，而Hashtable不允许。HashMap把Hashtable的contains方法去掉了，改成containsvalue和containsKey。因为contains方法容易让人引起误解。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，Hastable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 就必须为之提供同步。

**3.请说出ArrayList,Vector, LinkedList的存储性能和特性**

解答：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。

**1.简要描述如何结合struts、hibernate、spring开发Web应用？**

解答：Struts可以将jsp页面的表单关联起来,就是把JSP页面的表单数据封装成javaBean,这样的话,在action中你再也不需要使用传统的request.getParameter("name");还有struts有一个控制器,你在struts编程中的控制器(XxxAction)都是继承总的ActionServlet,它能集中处理请求,然后转到相关的页面。还有struts的表单验证组件,不用你写js验证了,只需要你配置一下文件就可以了。另外struts的令牌机制可以防表单重复提交。

Spring 是一个轻量级容器,非侵入性.包含依赖注入,AOP等。它是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。

Hibernate:它可以让我们以OO的方式操作数据库，这让我们看到了hibernate的强大之处，体验到操作数据的方便。但hibernate最耀眼之处是hibernate的缓存机制，而不是以OO的方式操作数据库。Hibernate的缓存机制不外乎是一级缓存session，二级缓存sessionFactory，和第三方缓存如ehcache。也就是hibernate的最强大的地方是它的缓存，理解了这个才能真正的理解hibernate,Hibernate的命名查询/命名参数查询，就是将hql语句放在一个单独的xml文件之中，它仍然让人们以面向对象的方式去操纵数据，而不用在以OO的方式写着代码的同时，然后再转变思维，用面向关系的方式去写那些sql语句。但hibernate不仅做了这些，它的native sql查询方式，完全满足sql语句的偏爱者，它像ibatis一样，将sql语句放在配置文件之中。

**2.说明反转控制（IOC）和面向方向编程（AOP）在spring中的应用**

解答：Spring 核心容器（Core）提供Spring框架的基本功能。核心容器的主要组件是BeanFactory，它是工厂模式的实现。BeanFactory使用控制反转（Ioc）模式将应用程序的配置和依赖性规范与实际的应用代码程序分开。Spring的声明式事务基于AOP实现，却并不需要程序开发者成为AOP专家，亦可轻易使用Spring的声明式事务管理。

**3.java语言中public、private、protected三个关键字的用法,重写和重载的区别。**

解答：  
重写：发生在父子类之间，方法名相同，参数的类型、个数、顺序相同，返回值相同，访问权限不能更封闭，抛出异常不能宽泛；

重载：发生在同一个类中，方法名相同，但是参数不同（类型不同或个数不同或参数的顺序不同），返回值可以不相同。

**1.类有哪三个基本特性？各特性的优点？**

解答：类具有封装性、继承性和多态性。

封装性：类的封装性为类的成员提供公有、缺省、保护和私有等多级访问权限，目的是隐藏类中的私有变量和类中方法的实现细节。

继承性：类的继承性提供从已存在的类创建新类的机制，继承（inheritance）使一个新类自动拥有被继承类（父类）的全部可继承的成员。

多态性：类的多态性提供类中方法执行的多样性，多态性有两种表现形式：重载和覆盖。

**2.谈谈对XML的理解？说明Web应用中Web.xml文件的作用？**

解答：XML（Extensible Markup Language）即可扩展标记语言，它与HTML一样，都是SGML(Standard Generalized Markup Language,标准通用标记语言)。Xml是Internet环境中跨平台的，依赖于内容的技术，是当前处理结构化文档信息的有力工具。扩展标记语言XML是一种简单的数据存储语言，使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立，虽然XML占用的空间比二进制数据要占用更多的空间，但XML极其简单易于掌握和使用。

web.xml的作用是配置欢迎页，servlet，filter，listener等的。

**3.jsp有哪些内置对象？作用分别是什么？（至少三个）**

解答：

1）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。

2）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）。

3）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。

4）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。

5）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。

6）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。

7）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。

8）page表示从该页面产生的一个servlet实例。

9）exception 针对错误网页，未捕捉的例外

**1.简要描述如何结合struts、hibernate、spring开发Web应用？**

解答：Struts可以将jsp页面的表单关联起来,就是把JSP页面的表单数据封装成javaBean,这样的话,在action中你再也不需要使用传统的request.getParameter("name");还有struts有一个控制器,你在struts编程中的控制器(XxxAction)都是继承总的ActionServlet,它能集中处理请求,然后转到相关的页面。还有struts的表单验证组件,不用你写js验证了,只需要你配置一下文件就可以了。另外struts的令牌机制可以防表单重复提交。

Spring 是一个轻量级容器,非侵入性.包含依赖注入,AOP等。它是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。

Hibernate:它可以让我们以OO的方式操作数据库，这让我们看到了hibernate的强大之处，体验到操作数据的方便。但hibernate最耀眼之处是hibernate的缓存机制，而不是以OO的方式操作数据库。Hibernate的缓存机制不外乎是一级缓存session，二级缓存sessionFactory，和第三方缓存如ehcache。也就是hibernate的最强大的地方是它的缓存，理解了这个才能真正的理解hibernate,Hibernate的命名查询/命名参数查询，就是将hql语句放在一个单独的xml文件之中，它仍然让人们以面向对象的方式去操纵数据，而不用在以OO的方式写着代码的同时，然后再转变思维，用面向关系的方式去写那些sql语句。但hibernate不仅做了这些，它的native sql查询方式，完全满足sql语句的偏爱者，它像ibatis一样，将sql语句放在配置文件之中。

**2.说明反转控制（IOC）和面向方向编程（AOP）在spring中的应用**

解答：Spring 核心容器（Core）提供Spring框架的基本功能。核心容器的主要组件是BeanFactory，它是工厂模式的实现。BeanFactory使用控制反转（Ioc）模式将应用程序的配置和依赖性规范与实际的应用代码程序分开。Spring的声明式事务基于AOP实现，却并不需要程序开发者成为AOP专家，亦可轻易使用Spring的声明式事务管理。

**3.java语言中public、private、protected三个关键字的用法,重写和重载的区别。**

解答：

作用域 当前类 同包 子类 其它

public √ √ √ √

protected √ √ √ ×

default √ √ × ×

private √ × × ×

重写：发生在父子类之间，方法名相同，参数的类型、个数、顺序相同，返回值相同，访问权限不能更封闭，抛出异常不能宽泛；

重载：发生在同一个类中，方法名相同，但是参数不同（类型不同或个数不同或参数的顺序不同），返回值可以不相同。

**1.JSP页面之间传递参数的方法有哪些？**

解答：

1）request

2）session

3）application

4）提交表单

5）超链接

**2.forward 和redirect的区别**

解答：forward是容器中控制权的转向，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，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。

**3.Java反射机制的作用？**

解答：Java反射机制的作用是：

1）在运行时判断任意一个对象所属的类。

2）在运行时构造任意一个类的对象。

3）在运行时判断任意一个类所具有的成员变量和方法。

4）在运行时调用任意一个对象的方法

**1.你是怎么理解java的泛型的？**

解答： 在Java SE 1.5之前，没有泛型的情况的下，通过对类型Object的引用来实现参数的“任意化”，“任意化”带来的缺点是要做显式的强制类型转换，而这种转换是要求开发者对实际参数类型可以预知的情况下进行的。对于强制类型转换错误的情况，编译器可能不提示错误，在运行的时候才出现异常，这是一个安全隐患。

泛型是Java SE 1.5的新特性，泛型的本质是参数化类型，也就是说所操作的数据类型被指定为一个参数。这种参数类型可以用在类、接口和方法的创建中，分别称为泛型类、泛型接口、泛型方法。

泛型的好处是在编译的时候检查类型安全，并且所有的强制转换都是自动和隐式的，提高代码的重用率。

**2.JAVA源文件中是否可以包括多个类,有什么限制**

解答：一个java源文件中可以包含多个类，每个源文件中至多有一个public类，如果有的话，那么源文件的名字必须与之相同。如果源文件中没有public类，则源文件用什么名字都可以，但最好还是具有特定的意义，免得自己都不记得里面写的是什么了。

**3．在一个千万级的数据库查寻中，如何提高查询效率？分别说出在数据库设计、SQL语句、java等层面的解决方案。**

解答：

1）数据库设计方面：

a. 对查询进行优化，应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引。

b. 应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描，如： select id from t where num is null 可以在num上设置默认值0，确保表中num列没有null值，然后这样查询： select id from t where num=0

c. 并不是所有索引对查询都有效，SQL是根据表中数据来进行查询优化的，当索引列有大量数据重复时,查询可能不会去利用索引，如一表中有字段sex，male、female几乎各一半，那么即使在sex上建了索引也对查询效率起不了作用。

d. 索引并不是越多越好，索引固然可以提高相应的 select 的效率，但同时也降低了 insert 及 update 的效率，因为 insert 或 update 时有可能会重建索引，所以怎样建索引需要慎重考虑，视具体情况而定。一个表的索引数最好不要超过6个，若太多则应考虑一些不常使用到的列上建的索引是否有必要。

e. 应尽可能的避免更新索引数据列，因为索引数据列的顺序就是表记录的物理存储顺序，一旦该列值改变将导致整个表记录的顺序的调整，会耗费相当大的资源。若应用系统需要频繁更新索引数据列，那么需要考虑是否应将该索引建为索引。

f. 尽量使用数字型字段，若只含数值信息的字段尽量不要设计为字符型，这会降低查询和连接的性能，并会增加存储开销。这是因为引擎在处理查询和连接时会逐个比较字符串中每一个字符，而对于数字型而言只需要比较一次就够了。

g. 尽可能的使用 varchar/nvarchar 代替 char/nchar ，因为首先变长字段存储空间小，可以节省存储空间，其次对于查询来说，在一个相对较小的字段内搜索效率显然要高些。

h. 尽量使用表变量来代替临时表。如果表变量包含大量数据，请注意索引非常有限（只有主键索引）。

i. 避免频繁创建和删除临时表，以减少系统表资源的消耗。

j. 临时表并不是不可使用，适当地使用它们可以使某些例程更有效，例如，当需要重复引用大型表或常用表中的某个数据集时。但是，对于一次性事件，最好使用导出表。

k. 在新建临时表时，如果一次性插入数据量很大，那么可以使用 select into 代替 create table，避免造成大量 log ，以提高速度；如果数据量不大，为了缓和系统表的资源，应先create table，然后insert。

l. 如果使用到了临时表，在存储过程的最后务必将所有的临时表显式删除，先 truncate table ，然后 drop table ，这样可以避免系统表的较长时间锁定。

2)SQL语句方面：

a. 应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描。

b. 应尽量避免在 where 子句中使用 or 来连接条件，否则将导致引擎放弃使用索引而进行全表扫描，如： select id from t where num=10 or num=20 可以这样查询： select id from t where num=10 union all select id from t where num=20

c. in 和 not in 也要慎用，否则会导致全表扫描，如： select id from t where num in(1,2,3) 对于连续的数值，能用 between 就不要用 in 了： select id from t where num between 1 and 3

d. 下面的查询也将导致全表扫描： select id from t where name like '%abc%'

e. 如果在 where 子句中使用参数，也会导致全表扫描。因为SQL只有在运行时才会解析局部变量，但优化程序不能将访问计划的选择推迟到运行时；它必须在编译时进行选择。然而，如果在编译时建立访问计划，变量的值还是未知的，因而无法作为索引选择的输入项。如下面语句将进行全表扫描： select id from t where num=@num 可以改为强制查询使用索引： select id from t with(index(索引名)) where num=@num

f. 应尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描。如： select id from t where num/2=100 应改为: select id from t where num=100\*2

g. 应尽量避免在where子句中对字段进行函数操作，这将导致引擎放弃使用索引而进行全表扫描。如： select id from t where substring(name,1,3)='abc'--name以abc开头的id select id from t where datediff(day,createdate,'2005-11-30')=0--‘2005-11-30’生成的id 应改为: select id from t where name like 'abc%' select id from t where createdate>='2005-11-30' and createdate<'2005-12-1'

h. 不要在 where 子句中的“=”左边进行函数、算术运算或其他表达式运算，否则系统将可能无法正确使用索引。

i. 不要写一些没有意义的查询，如需要生成一个空表结构： select col1,col2 into #t from t where 1=0 这类代码不会返回任何结果集，但是会消耗系统资源的，应改成这样： create table #t(...)

j. 很多时候用 exists 代替 in 是一个好的选择： select num from a where num in(select num from b) 用下面的语句替换： select num from a where exists(select 1 from b where num=a.num)

k. 任何地方都不要使用 select \* from t ，用具体的字段列表代替“\*”，不要返回用不到的任何字段。

l. 尽量避免使用游标，因为游标的效率较差，如果游标操作的数据超过1万行，那么就应该考虑改写。

m. 尽量避免向客户端返回大数据量，若数据量过大，应该考虑相应需求是否合理。

n. 尽量避免大事务操作，提高系统并发能力。

3)java方面：

a.尽可能的少造对象。

b.合理摆正系统设计的位置。大量数据操作，和少量数据操作一定是分开的。大量的数据操作，肯定不是ORM框架搞定的。，

c.使用jDBC链接数据库操作数据

d.控制好内存，让数据流起来，而不是全部读到内存再处理，而是边读取边处理；

e.合理利用内存，有的数据要缓存

**1、请解释分布式事务管理？**

解答：分布式事务是指事务的参与者、支持事务的服务器、资源服务器以及事务管理器分别位于不同的分布式系统的不同节点之上。为了实现分布式事务，需要使用下面将介绍的两阶段提交协议。 阶段一：开始向事务涉及到的全部资源发送提交前信息。此时，事务涉及到的资源还有最后一次机会来异常结束事务。如果任意一个资源决定异常结束事务，则整个事务取消，不会进行资源的更新。否则，事务将正常执行，除非发生灾难性的失败。为了防止会发生灾难性的失败，所有资源的更新都会写入到日志中。这些日志是永久性的，因此，这些日志会幸免遇难并且在失败之后可以重新对所有资源进行更新。 阶段二：只在阶段一没有异常结束的时候才会发生。此时，所有能被定位和单独控制的资源管理器都将开始执行真正的数据更新。 在分布式事务两阶段提交协议中，有一个主事务管理器负责充当分布式事务协调器的角色。事务协调器负责整个事务并使之与网络中的其他事务管理器协同工作。 为了实现分布式事务，必须使用一种协议在分布式事务的各个参与者之间传递事务上下文信息，IIOP便是这种协议。这就要求不同开发商开发的事务参与者必须支持一种标准协议，才能实现分布式的事务。

**2、请写出一段表单提交的HTML代码，表单名称为form1，提交方式为post，提交地址为submit.asp**

解答：

<form name=”form1” method=”post” action=”submit.jsp”>

<input type=”submit”>

</form>

**3、请写出一个超链接，点击链接后可以向zhangsan@d-heaven.com发送电子邮件。**

<a href=”mailto: zhangsan@d-heaven.com”>发邮件</a>

**1、请说明meta标签的作用。**

解答：

meta是用来在HTML文档中模拟HTTP协议的响应头报文。meta 标签用于网页的＜head＞与＜/head＞中，meta 标签的用处很多。meta 的属性有两种：name和http-equiv。name属性主要用于描述网页，对应于content（网页内容），以便于搜索引擎机器人查找、分类（目前几乎所有的搜索引擎都使用网上机器人自动查找meta值来给网页分类）。这其中最重要的是description（站点在搜索引擎上的描述）和keywords（分类关键词），所以应该给每页加一个meta值。比较常用的有以下几个：

name 属性

1)．＜meta name="Generator" contect=""＞用以说明生成工具（如Microsoft FrontPage 4.0）等；

2)．＜meta name="KEYWords" contect=""＞向搜索引擎说明你的网页的关键词；

3)．＜meta name="DEscription" contect=""＞告诉搜索引擎你的站点的主要内容；

4).＜meta name="Author" contect="你的姓名"＞告诉搜索引擎你的站点的制作的作者；

5).＜meta name="Robots" contect=

"all|none|index|noindex|follow|nofollow"＞

其中的属性说明如下：

设定为all：文件将被检索，且页面上的链接可以被查询；

设定为none：文件将不被检索，且页面上的链接不可以被查询；

设定为index：文件将被检索；

设定为follow：页面上的链接可以被查询；

设定为noindex：文件将不被检索，但页面上的链接可以被查询；

设定为nofollow：文件将不被检索，页面上的链接可以被查询。

6).http-equiv属性

a、＜meta http-equiv="Content-Type" contect="text/html";charset=gbk"＞

和 ＜meta http-equiv="Content-Language" contect="zh-CN"＞用以说明主页制作所使用的文字以及语言；

b、＜meta http-equiv="Refresh" contect="n;url=http://yourlink"＞定时让网页在指定的时间n内，跳转到页面http;//yourlink；

c、＜meta http-equiv="Expires" contect="Mon,12 May 2001 00:20:00 GMT"＞可以用于设定网页的到期时间，一旦过期则必须到服务器上重新调用。需要注意的是必须使用GMT时间格式；

d、＜meta http-equiv="Pragma" contect="no-cache"＞是用于设定禁止浏览器从本地机的缓存中调阅页面内容，设定后一旦离开网页就无法从Cache中再调出；

e、＜meta http-equiv="set-cookie" contect="Mon,12 May 2001 00:20:00 GMT"＞cookie设定，如果网页过期，存盘的cookie将被删除。需要注意的也是必须使用GMT时间格式；

f、＜meta http-equiv="Pics-label" contect=""＞网页等级评定，在IE的internet选项中有一项内容设置，可以防止浏览一些受限制的网站，而网站的限制级别就是通过meta属性来设置的；

g、＜meta http-equiv="windows-Target" contect="\_top"＞强制页面在当前窗口中以独立页面显示，可以防止自己的网页被别人当作一个frame页调用；

h、＜meta http-equiv="Page-Enter" contect="revealTrans(duration=10,transtion=50)"＞和＜meta http-equiv="Page-Exit"

contect="revealTrans(duration=20，transtion=6)"＞设定进入和离开页面时的特殊效果，这个功能即FrontPage中的“格式/网页过渡”，不过所加的页面不能够是一个frame页面。

**2、请写出一个隐藏控件。**

解答：<input type=”hidden” name=”userId” value=”10001”>

**3、如何将HTML页面的标题设置为“数字天堂”。**

解答：

<html>

<head><title>数字天堂</title></head>

<body>body部分</body>

</html>

**1、请写出JavaScript中常用的三种事件。**

解答： onclick，onblur，onChange

**2、请写出一段JavaScript代码，要求页面有一个按钮，点击按钮弹出确认框。程序可以判断出用户点击的是“确认”还是“取消”。**

解答：

<HTML>

<HEAD>

<TITLE>click</TITLE>

<Script >

function validateForm()

{

if(confirm("你确认提交这个表单么？"))

{

alert(“确定”);

}else{

alert(“取消”);

}

}

</Script>

</HEAD>

<BODY>

<FORM NAME="TEST" action="FirstJS.htm">

<INPUT TYPE="button" NAME="SUB" VALUE="提交"onClick="JavaScript:validateForm()">

</FORM>

</BODY>

</HTML>

**3、JavaScript如何实现计时功能。**

解答：<script> window.setTimeout( "window.location = 'http://www.csdn.net '; ", 35000); </script>

**1、JavaScript如何定义数组。**

解答： var arrTest=new Array();

**2、JavaScript能否操作cookie和session？**

解答：JavaScript可以操作cookie，但是不能操作session

**3、请写出如下正则表达式的含义；**

com︱cn︱net

^http://www.d-heaven.com/$

^http://www.sina.com.cn/new/newid=\d+

解答：com︱cn︱net表示匹配com或是cn或是net中一个

^http://www.d-heaven.com/$ 表示匹配：http://www.d-heaven.com/

^http://www.sina.com.cn/new/newid=\d+

表示匹配http://www.sina.com.cn/new/newid=其后可以出现1到多个数字

**1、请用正则表达式匹配出QQ号（假设QQ号码为5—10位）；**

解答： ^ \d{5,10}$

**2、String, StringBuffer StringBuilder的区别。**

解答：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。

**3、请写出5种常见到的runtime exception。**

解答：

NullPointerException：当操作一个空引用时会出现此错误。

NumberFormatException：数据格式转换出现问题时出现此异常。

ClassCastException：强制类型转换类型不匹配时出现此异常。

ArrayIndexOutOfBoundsException：数组下标越界，当使用一个不存在的数组下标时出现此异常。

ArithmeticException：数学运行错误时出现此异常

**1、数组有没有length()这个方法？String有没有length()这个方法？**

解答：数组没有length（）方法 它有length属性

String有length（）方法。

**2、请写出一个单例模式。**

解答：单例模式（Singleton pattern）:确保一个类只有一个实例，并提供一个全局的访问点

public class EagerSingleton {

private static final EagerSingleton m\_instance = new EagerSingleton(); /\*\* \* 私有的默认构造子 \*/

private EagerSingleton() { } /\*\* \* 静态工厂方法 \*/

public static EagerSingleton getInstance() {

return m\_instance;

}

}

**3、在java中，List是个接口，那实现List接口的类有哪些，有什么区别？**

解答： ArrayList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，

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。

**1、char类型能否存储一个中文字符？为什么**

解答：可以。一个char是两个字节，而一个中文也是两个字节。

**2、Switch选择语句能否作用在String【字符串】上，也就是能否这么写：Switch(一个字符串变量)？**

解答：不可以，只能处理int,byte,short,char，（其实是只能处理int,其它三种是可以promotion到int型）不能处理Sring 。

**3、关键字final 分别修饰一个类，一个方法，一个变量，各起什么作用**

解答：final修饰类是不能被继承

fianl修饰方法不能在子类中被覆盖

final修饰变量，称为常量，初始化以后不能改变值。

**1、启动一个线程是用run()还是start()?**

解答：start（）。

**2、是否可以继承String类**

解答：不可以。因为String类有final修饰。

**3、Java异常处理中，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?**

解答：会执行，在return前执行。

**1、EJB包含哪3种bean**

解答：session bean（会话bean）, entity bean（实体bean）, message bean（消息bean）

**2、Class.forName（String className）这个方法的作用**

解答：通过类的全名获得该类的类对象

**3、你认为在表上建立索引可以提高数据库系统的效率吗，为什么？**

解答：不一定

建立太多的索引将会影响更新和插入的速度，因为它需要同样更新每个索引文件。对于一个经常需要更新和插入的表格，就没有必要为一个很少使用的where子句单独建立索引了，对于比较小的表，排序的开销不会很大，也没有必要建立另外的索引。

**1、hibernate中的java对象有几种状态，其相互关系如何（区别和相互转换）。**

解答：在Hibernate中，对象有三种状态：临时状态、持久状态和游离状态。 临时状态：当new一个实体对象后，这个对象处于临时状态，即这个对象只是一个保存临时数据的内存区域，如果没有变量引用这个对象，则会被jre垃圾回收机制回收。这个对象所保存的数据与数据库没有任何关系，除非通过Session的save或者SaveOrUpdate把临时对象与数据库关联，并把数据插入或者更新到数据库，这个对象才转换为持久对象；

持久状态:持久化对象的实例在数据库中有对应的记录，并拥有一个持久化表示（ID）。对持久化对象进行delete操作后，数据库中对应的记录将被删除，那么持久化对象与数据库记录不再存在对应关系，持久化对象变成临时状态。

持久化对象被修改变更后，不会马上同步到数据库，直到数据库事务提交。在同步之前，持久化对象是脏的（Dirty）。

游离状态：当Session进行了Close、Clear或者evict后，持久化对象虽然拥有持久化标识符和与数据库对应记录一致的值，但是因为会话已经消失，对象不在持久化管理之内，所以处于游离状态（也

叫：脱管状态）。游离状态的对象与临时状态对象是十分相似的，只是它还含有持久化标识。

**2、对hibernate的延迟加载如何理解，在实际应用中，延迟加载与session关闭的矛盾是如何处理的？**

解答：延迟加载就是并不是在读取的时候就把数据加载进来，而是等到使用时再加载。那么Hibernate是怎么知识用户在什么时候使用数据了呢？又是如何加载数据呢？其实很简单，它使用了代理机制。返回给用户的并不是实体本身，而是实体对象的代理。代理对象在用户调用getter方法时就会去数据库加载数据。但加载数据就需要数据库连接。而当我们把会话关闭时，数据库连接就同时关闭了。这种情况就叫做未初始化的关系。

延迟加载与session关闭的矛盾一般可以这样处理：

1)、关闭延迟加载特性。

操作起来比较简单，因为hibernate的延迟加载特性是在hbm配置里面可控制的。默认lazy="true"，具体配置可以查看一下相关文档，就不详细叙述了。

但使用这个解决办法带来的隐患是十分大的。

首先，出现no session or session was closed就证明了您已经在使用外键关联表，如果去掉延迟加载的话，则表示每次查询的开销都会变得十分的大，如果关联表越多，后果也可以想象得到。所以不建议使用这个方法解决。

2)、在session关闭之前把我们想要查询的数据先获取了。

首先需要了解一下session什么时候关闭，也就是它的生命周期。通常情况下hibernate会在查询数据关闭session，而使用getHibernateTemplate().get方法查询后会延迟关闭的时间。会在事务结束后才关闭。

使用拦截器(Interceptor)或过滤器(Filter)控制session。

spring为解决hibernate这一特性提供的解决方案，可以有效的控制session生命周期。

**3、什么是AOP和OOP，IOC和DI有什么不同?**

解答：

1）面向对象编程（Object Oriented Programming，OOP，面向对象程序设计）是一种计算机编程架构。AOP是OOP的延续，是Aspect Oriented Programming的缩写，意思是面向方面编程。 将通用需求功能从不相关类之中分离出来；同时，能够使得很多类共享一个行为，一旦行为发生变化，不必修改很多类，只要修改这个行为就可以。AOP就是这种实现分散关注的编程方法，它将“关注”封装在“方面”中

2）控制反转IOC(Inversion of Control) 控制指的就是程序相关类之间的依赖关系.传统观念设计中,

通常由调用者来创建被调用者的实例, 在Spring里,创建被调用者的工作不再由调用者来完成,而是由Spring容器完成，依赖关系被反转了，称为控制反转，目的是为了获得更好的扩展性和良好的可维护性。依赖注入(Dependency injection)创建被调用者的工作由Spring容器完成，然后注入调用者，因此也称依赖注入。控制反转和依赖注入是同一个概念。

**1、Struts1中actionform和action属于MVC哪一层，为什么？**

解答：actionform和action属于MVC的Model层，Action用来处理业务逻辑，actionform保存用户表单数据以便于在不同页面间传递。而MVC中的model层就是业务逻辑层，该层用于实现具体的业务逻辑、状态维护及管理。

**2、error和exception有什么区别?**

解答：

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。

**3、Log4J是Apache组织的开源一个开源项目，通过Log4J，可以指定日志信息输出的目的地，如console、file等。Log4J采用日志级别机制，请按照输出级别由低到高的顺序写出日志输出级别。**

解答：Log4J分 为OFF、FATAL、ERROR、WARN、INFO、DEBUG、ALL或者您定义的级别。Log4j建议只使用四个级别，优先级从高到低分别是 ERROR、WARN、INFO、DEBUG。通过在这里定义的级别，您可以控制到应用程序中相应级别的日志信息的开关。比如在这里定义了INFO级别， 则应用程序中所有DEBUG级别的日志信息将不被打印出来。

**1、说出几个与spring同类型的开源框架，说出几个与hibernate同类型的开源框架,说出几个与struts同类型的开源框架**

解答：

1）与spring同类型的开源框架：JUIDE、EJB3.0、picoContainer

2）与hibernate同类型的开源框架 ：ibatis，jdo，JPA

3）几个与struts同类型的开源框架：webwork，tapestry，JSF

**2、Struts2包含哪些标签？**

解答：

A：

<s:a href=""></s:a>-----超链接，类似于html里的<a></a>

<s:action name=""></s:action>-----执行一个view里面的一个action <s:actionerror/>-----如果action的errors有值那么显示出来 <s:actionmessage/>-----如果action的message有值那么显示出来 <s:append></s:append>-----添加一个值到list，类似于list.add(); <s:autocompleter></s:autocompleter>-----自动完成<s:combobox>标签的内容，这个是ajax

B： <s:bean name=""></s:bean>-----类似于struts1.x中的，JavaBean的值

C： <s:checkbox></s:checkbox>-----复选框 <s:checkboxlist list=""></s:checkboxlist>-----多选框 <s:combobox list=""></s:combobox>-----下拉框 <s:component></s:component>-----图像符号

D： <s:date/>-----获取日期格式 <s:datetimepicker></s:datetimepicker>-----日期输入框 <s:debug></s:debug>-----显示错误信息 <s:div></s:div>-----表示一个块，类似于html的<div></div> <s:doubleselect list="" doubleName="" doubleList=""></s:doubleselect>-----双下拉框

E： <s:if test=""></s:if> <s:elseif test=""></s:elseif> <s:else></s:else>-----这3个标签一起使用，表示条件判断

F： <s:fielderror></s:fielderror>-----显示文件错误信息 <s:file></s:file>-----文件上传 <s:form action=""></s:form>-----获取相应form的值

G： <s:generator separator="" val=""></s:generator>----和<s:iterator>标签一起使用

H： <s:head/>-----在<head></head>里使用，表示头文件结束

<s:hidden></s:hidden>-----隐藏值

I： <s:i18n name=""></s:i18n>-----加载资源包到值堆栈 <s:include value=""></s:include>-----包含一个输出，servlet或jsp页面 <s:inputtransferselect list=""></s:inputtransferselect>-----获取form的一个输入 <s:iterator></s:iterator>-----用于遍历集合

L： <s:label></s:label>-----只读的标签

M： <s:merge></s:merge>-----合并遍历集合出来的值

O： <s:optgroup></s:optgroup>-----获取标签组 <s:optiontransferselect doubleList="" list="" doubleName=""></s:optiontransferselect>-----左右选择框

P： <s:param></s:param>-----为其他标签提供参数 <s:password></s:password>-----密码输入框 <s:property/>-----得到'value'的属性 <s:push value=""></s:push>-----value的值push到栈中,从而使property标签的能够获取value的属性

R：

<s:radio list=""></s:radio>-----单选按钮 <s:reset></s:reset>-----重置按钮 S： <s:select list=""></s:select>-----单选框 <s:set name=""></s:set>-----赋予变量一个特定范围内的值 <s:sort comparator=""></s:sort>-----通过属性给list分类 <s:submit></s:submit>-----提交按钮 <s:subset></s:subset>-----为遍历集合输出子集

T：

<s:tabbedPanel id=""></s:tabbedPanel>-----表格框 <s:table></s:table>-----表格 <s:text name=""></s:text>-----I18n文本信息 <s:textarea></s:textarea>-----文本域输入框 <s:textfield></s:textfield>-----文本输入框 <s:token></s:token>-----拦截器 <s:tree></s:tree>-----树 <s:treenode label=""></s:treenode>-----树的结构

U： <s:updownselect list=""></s:updownselect>-----多选择框 <s:url></s:url>-----创建url

**3、struts2中，OGNL访问值栈的时候查找的顺序是什么？请排序：模型对象、临时对象、固定名称的对象、Action对象**

解答：struts2的值栈排列顺序为：1）.临时对象；2）.模型对象；3）.Action对象；4）.固定名称对象（如#application, #session, #request等对象）。

**1、struts2中，Action通过什么方式获得用户从页面输入的数据，又是通过什么方式把其自身的数据传给视图的？**

解答：

1）可以直接通过与表单元素相同名称的数据成员（需要存在符合命名规范set和get方法）获取页面表单数据。

2）会把处理好的数据成员放入值栈中，到页面可以使用struts2标签取值就可以了。

**2、常用的设计模式有哪些？说明工厂模式。**

解答：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。

**3、什么是数据库的参照完整性？**

解答：数据库的参照完整性是指表与表之间的一种对应关系，通常情况下可以通过设置两表之间的主键、外键关系，或者编写两表的触发器来实现。 有对应参照完整性的两张表格，在对他们进行数据插入、更新、删除的过程中，系统都会将被修改表格与另一张对应表格进行对照，从而阻止一些不正确的数据的操作。

**1、如何优化数据库，如何提高数据库的性能?**

解答：

1） 硬件调整性能 最有可能影响性能的是磁盘和网络吞吐量,解决办法扩大虚拟内存，并保证有足够可以扩充的空间；把数据库服务器上的不必要服务关闭掉；把数据库服务器和主域服务器分开；把SQL数据库服务器的吞吐量调为最大；在具有一个以上处理器的机器上运行SQL。

2）调整数据库

若对该表的查询频率比较高，则建立索引；建立索引时，想尽对该表的所有查询搜索操作， 按照where选择条件建立索引，尽量为整型键建立为有且只有一个簇集索引，数据在物理上按顺序在数据页上，缩短查找范围，为在查询经常使用的全部列建立非簇集索引，能最大地覆盖查询；但是索引不可太多，执行UPDATE DELETE INSERT语句需要用于维护这些索引的开销量急剧增加；避免在索引中有太多的索引键；避免使用大型数据类型的列为索引；保证每个索引键值有少数行。

3）使用存储过程

应用程序的实现过程中，能够采用存储过程实现的对数据库的操作尽量通过存储过程来实现，因为存储过程是存放在数据库服务器上的一次性被设计、编码、测试，并被再次使用，需要执行该任务的应用可以简单地执行存储过程，并且只返回结果集或者数值，这样不仅可以使程序模块化，同时提高响应速度，减少网络流量，并且通过输入参数接受输入，使得在应用中完成逻辑的一致性实现。

4）应用程序结构和算法

建立查询条件索引仅仅是提高速度的前提条件，响应速度的提高还依赖于对索引的使用。因为人们在

使用SQL时往往会陷入一个误区，即太关注于所得的结果是否正确，特别是对数据量不是特别大的数据库操作时，是否建立索引和使用索引的好坏对程序的响应速度并不大，因此程序员在书写程序时就忽略了不同的实现方法之间可能存在的性能差异，这种性能差异在数据量特别大时或者大型的或是复杂的数据库环境中（如联机事务处理OLTP或决策支持系统DSS）中表现得尤为明显。在工作实践中发现，不良的SQL往往来自于不恰当的索引设计、不充份的连接条件和不可优化的where子句。在对它们进行适当的优化后，其运行速度有了明显地提高！

**2、JS中的三种弹出式消息提醒(警告窗口、确认窗口、信息输入窗口)的命令是什么？**

解答：alert confirm prompt

**3、描述JSP和Servlet的区别、共同点、各自应用的范围**

解答：JSP在本质上就是SERVLET,但是两者的创建方式不一样.Servlet完全是JAVA程序代码构成，擅长于流程控制和事务处理，通过Servlet来生成动态网页很不直观.JSP由HTML代码和JSP标签构成，可以方便地编写动态网页.因此在实际应用中采用Servlet来控制业务流程，而采用JSP来生成动态网页.

**1、在java语言中int 和 Integer 有什么区别**

解答：int是基本数据类型，Integer是int的包装类，属于引用类型

85.计算下列结果：

25&3 =

25|3=

解答：25的二进制是11001 ， 3的二进制是00011， 按位与的结果是：00001 ，按位或的结果是：11010，因此转成10进制分别为：1 和27

**2、如何获得<div id=”div1”>This is first layer</div>中的值？**

解答：

<script>

var div1=Document.getElementById(“div1”);

alert(div1.innerHTML)

</script>

**3、JDK1.5中支持的 for循环的语法**

解答：

for(type element : array) { System.out.println(element)

} type 集合(不仅仅指Collection，也包含普通的数组)中元素的数据类型 element 遍历到的元素 array 集合对象本身(当然不只是Collection)

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

解答：

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。

**2、如何格式化日期**

解答：

Date now=new Date();

SimpleDateFormat sdf=new SimpleDateFormat(“yyyy-MM-dd hh:mm:ss”);

String formatNow=sdf.format(now);

变量formatNow就是格式化好的日期。

**3、将字符“12345”转换成long型**

解答： String s="12345";

long num=Long.valueOf(s).longValue();

**1、struts中如何实现国际化，涉及哪些文件？**

解答：“国际化”是指一个应用程序在运行时能够根据客户端请求所来自的国家/地区、语言的不同而显示不同的用户界面。Struts框架通过使用<bean:message>标记，以及使用java.util数据包中定义的Locale和ResourceBundle类来支持国际化。java.text.MessageFormat类定义的技术可以支持消息的格式。利用此功能，开发人员不需了解这些类的细节就可进行国际化和设置消息的格式。会涉及到资源文件，不需了解这些类的细节就可进行国际化和设置消息的格式。会涉及到资源文件，struts-config.xml配置文件,web.xml配置文件。

**2、例举在诊断Oracle性能问题时，常用的工具、方法**

解答： 1）简单一点的可以用toad及dbartisan这样的工具。 2）纯做性能监测，比较出色的有spolight和emc的I3，这两个软件都比较贵。 3）一些网管系统这方面也不错，如hp的openview。不过定制起来不太容易，且很贵。

4）不用花钱又好用的就是Oracle的statpack了。 5）再有就是自己写脚本了，大多数有经验的DBA都喜欢这样的方式。优点是实用灵活。缺点是如果想出一个性能变化的曲线图等图表，且做的美观就需要些delphi,c++builder或是pb的开发功底了。

**3、Oracle启动中，startup nomount、 startup mount 有什么差别？**

解答： startup nomount：启动实例，读取参数文件，分配内存空间，启动后台进程，打开跟踪文件和报警文件。startup mount：装载数据库，打开控制文件。nomount方式下还没有读取控制文件，该选项用于在数据库的控制文件全部损坏，需要重新创建数据库控制文件或创建一个新的数据库时使用。mount 选项下并没有打开数据文件，该选项可以用来修改数据库的运行模式或进行数据库恢复。

**1、Oracle启动中，spfile.ora、init<SID>.ora、spfile<SID>.ora 这三个文件正确的先后顺序是什么？**

解答：启动数据库，使用startup命令，Oralce将会按照以下顺序在缺省目录中搜索参数文件：spfile<SID>.ora , spfile.ora ,init<SID>.ora

**2、说明Oracle数据库逻辑备份和物理备份的方式。**

解答：Oracle备份包括逻辑备份和物理备份。

1）．逻辑备份

数据库的逻辑备份包含读一个数据库记录集和将记录集写入文件。

a.输出（Export）输出可以是整个数据库、指定用户或指定表。

b.输入（Import）输入将输出建立的二进制转储文件读入并执行其命令。

2)．物理备份

物理备份包含拷贝构成数据库的文件而不管其逻辑内容。

Oracle支持两种不同类型的物理文件备份：脱机备份（offline backup）和联机备份（online backup）。

a.脱机备份: 脱机备份用在当数据库已正常关闭，数据库处于"offline"时，要备份下列文件：

所有数据文件

所有控制文件

所有联机日志

init.ora（可选的）

b 联机备份:联机备份可用来备份任何运作在ARCHIVELOG方式下的数据库。在这种方式下，联机日志被归档，在数据库内部建立一个所有作业的完整记录。联机备份过程具备强有力的功能。第一，提供了完全的时间点（point-in-time）恢复。第二，在文件系统备份时允许数据库保持打开状态。

**3、有2个类Cat及WhiteCat，代码如下：**

public class Cat {

protected static String color = "random";

public Cat() {

}

public void showCatColor() {

System.out.println("Cat:" + color);

}

public static void showColor() {

System.out.println("Cat:" + color);

}

}

public class WhiteCat extends Cat {

protected static String color = "white";

public WhiteCat() {

super();

}

public void showCatColor() {

System.out.println("WhiteCat:" + color);

}

public static void showColor() {

System.out.println("WhiteCat:" + color);

}

}

请分析下面各段程序的运行结果

A．WhiteCat whiteCat = new WhiteCat(); Cat cat = whiteCat; cat.showColor(); cat.showCatColor();

B．Cat cat = new Cat(); WhiteCat whiteCat = (WhiteCat) cat;

cat.showColor(); cat.showCatColor();

C．Cat cat = new WhiteCat(); WhiteCat whiteCat = (WhiteCat) cat; cat.showColor(); cat.showCatColor();

解答：A段执行的结果是：

Cat:random

WhiteCat:white

B段执行的结果是：

会抛出java.lang.ClassCastException异常

C段执行的结果是：

Cat:random

WhiteCat:white

**1、说说下面语句是否有错误，或可能出现的缺陷，并指出错误，或缺陷在哪里？**

public class MyFile implements Runnable{

public void run(){

while (true){

try{

FileReader fr=new FileReader(new File("a.txt")) ;

String line=fr.readLine();

System.out.println(line);

}catch(IOException err) {

}

Sleep(1000); }

}

解答： 1.fr.readLine()没有这个方法

2.Sleep(1000)需要用Thread.sleep(1000);

**2、判断下列语句是否正确，如果有错误，请指出错误所在？**

List<Short> a = new ArrayList<Short>();

a.add(5);

解答：错误,默认封装int类型。

**3、判断下列语句是否正确，如果有错误，请指出错误所在？**

void foo(final int []arg){

if (arg.length > 1)

arg[0] = 5;

}

解答：正确

**1、判断下列语句是否正确，如果有错误，请指出错误所在？**

void foo(final int []arg){

if (arg.length > 1)

arg[0] = 5;

}

解答：正确

**2、判断下列语句是否正确，如果有错误，请指出错误所在？**

interface A{

int add(final A a);

}

class B implements A{

long add(final A a){

return this.hashCode() + a.hashCode();

}

}

解答：返回值不是long类型

**3、指出下面程序的运行结果：**

class A{

static{

System.out.print("a");

}

public A (){

System.out.print("x");

}

}

class B extends A{

static{

System.out.print("b");

}

public B (){

System.out.print("y");

}

}

public class Test{

public static void main(String[] args){

A ab = new B ();

ab = new B ();

}

}

解答：abxyxy

**1、下列代码的输出结果是什么？**

public class MyFor {

public static void main (String argv[]){

int i; int j;

outer:for(i=1;i<3;i++)

inner:for(j=1;j<3;j++){

if (j==2) continue outer;

System.out .println("Value for i="+i+" Value for j=" +j);

}

}

}

解答：Value for i=1 Value for j=1

Value for i=2 Value for j=1

**2、查看下面的代码，写出可以使程序正常执行的修改方法**

1.public class MyClass {

2.static String s1;

3. String s2;

4. public static void main(String args[]) {

5. String s3;

6. System.out.println("s1 =" + s1);

7. System.out.println("s2 =" + s2);

8. System.out.println("s3 =" + s3);

9. }

10.}

解答：删除第8行或者将第6行改为String s3 = "";

**3、为了显示myStr = 23 这样的结果，写出在控制台输入的命令**

public class MyClass {

public static void main(String args[]) {

String s1 = args[0];

String s2 = args[1];

String myStr = args[2];

System.out.printin("myStr =" + s2 + myStr);

}

}

解答：java MyClass 1 2 3 4

**1、简述一下java基本数据类型及所占位数**

java基本数据类型：4类8种

逻辑型：boolean false/true 1byte

文本型：char 2byte

整数型：byte(1byte)，short(2byte)，int(4byte)，long(8byte)

浮点数型：float(4byte)，double(8byte)

**2、说出5个的启动时异常:**

RunTimeException

|---NullPointerException

|——ArrayIndexOutOfBoundsException

|——ArithmeticException

|——ClassCastException

|——NumberFormatException

**3. HashMap和HashTable的区别：**

1.HashMap允许空键值对，HashTable不允许

2.HashMap不是线程安全的，HashTable是

3.HashMap直接实现Map接口，HashTable继承Dictionary类

**1、ArrayList，Vector，LinkedList存储性能和特性**

它们都实现List接口

ArrayList和Vector都是基于数组实现的

LinkedList基于双向循环链表（查找效率低，添加删除容易）

ArrayList不是线程安全的而Vector是线程安全的，所以速度上ArrayList高于Vector

**2、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。

**3、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。

**1、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。

**2、 Overload和Override的区别。Overload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,方法名，参数列表，

返回值类型都得与父类的方法一致.

重载Overloading是一个类中多态性的一种表现。重载的方法是可以改变返回值的类型。

**3、用一句话总结一下冒泡排序。**

依次比较相邻的两个数，将小数放在前面，大数放在后面.

**1、实现线程安全的两种方式**

1）synchronized方法：通过在方法声明中加入synchronized关键字来声明

synchronized方法。

2）synchronized块：通过synchronized关键字来声明synchronized块。

**2、说一下 “==”和equals()方法在字串变量操作中的不同?**

”==”比较的是两个字符串对象的地址，equals()是比较的两个字符串的具体值。

**3、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。

**1、&与&&的区别?**

&位运算符:非短路运算符，它会把所有条件执行完毕之后，才会返回结果

&&逻辑运算符（and）：短路运算符，遇到不符合条件，立即终止程序的执行

**2、 error和exception区别**

error:表示恢复不是不可能的一种严重的问题,比如：内存溢出，不指望程序处理

exception程序运行时的异常，如果程序设计合理从不会出现的情况

**3、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；

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。

**1、网络编程中设计并发服务器，使用多进程与多线程，请问有什么区别?**

1)进程：子进程是父进程的复制品子进程获得父进程数据空间堆和栈的复制品

2)线程：相对与进程而言，线程是一个更加接近与执行体的概念，它可以与同进程的其他线程共享数据，但拥有自己的栈空间，拥有独立的执行序列

两者都可以提高程序的并发度，提高程序运行效率和响应时间

线程和进程在使用上各有优缺点：线程执行开销小，但不利于资源管理和保护;而进程正相反同时，线程适合于在SMP机器上运行，而进程则可以跨机器迁移

**2、什么是反射？**

在运行过程中:

1.对于任意一个类,可以知道这个类的属性和方法.

2.对于任意一个对象,可以调用这个对象的任意方法.

对于这种动态获取的信息,以及动态调用对象方法的功能称为反射机制.

java反射机制提供的功能:

1.运行时判断任意对象的所属类;

2.运行时构造任意类的对象;

3.运行时判断和调用对象的成员变量和方法;

4.生成动态代理.

**3、什么是回调函数？**

某个程序S(Student.main) 调用 服务程序A(Arrays) 中的某个方法(sort), 服务程序A的sort方法在某个时候反过来调用S的某个方法(compareTo),这种情况下,compareTo叫做S的回调方法。

例如：public class Student implements Comparable{

private int id;

private String name;

private int age;

private int score;

//构造器

//getter / setter方法

//回调方法

public int compareTo(Object obj){

return

this.id - ((Student)obj).id;

}

}

Student s1 = new Student(1,"a",18,89);

Student s2 = new Student(2,"x",22,94);

Student s3 = new Student(3,"w",19,78);

Student [] arrs = {s1,s2,s3};

Arrays.sort(arrs);

**1、遍历文件夹下所有.java的文件?**

public void listFiles(String path){

File dir = new File(path);

Files files [] = dir.listFiles(new FileFilter(){

public boolean accept(File f){

return f.getName().endWith(".java");

}

});

for(File file : files){

System.out.println(file.getName());

}

}

**2、线程的基本概念、线程的本状态以及状态之间的关系**

新建 (Born) : 新建的线程处于新建状态

就绪 (Ready) : 在创建线程后，它将处于就绪状态，等待 start() 方法被调用

运行 (Running) : 线程在开始执行时进入运行状态

睡眠 (Sleeping) : 线程的执行可通过使用 sleep() 方法来暂时中止。在睡眠后，线程将进入就绪状态

等待 (Waiting) : 如果调用了 wait() 方法，线程将处于等待状态。用于在两个或多个线程并发运行时。

挂起 (Suspended) : 在临时停止或中断线程的执行时，线程就处于挂起状态。

恢复 (Resume) : 在挂起的线程被恢复执行时，可以说它已被恢复。

阻塞 (Blocked) – 在线程等待一个事件时（例如输入/输出操作），就称其处于阻塞状态。 死亡 (Dead) – 在 run() 方法已完成执行或其 stop() 方法被调用之后，线程就处于死亡状态。

串行化的注意事项以及如何实现串行化答：如果有循环引用是不可以串行化的。对象输出流的WriteObject方法和 对象输入流的ReadObect 方法

**3、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。

**1、什么是触发器和存储过程？**

触发器是存储在数据库中的块，这些块一旦被构造后，就可以多次执行，当触发它的事件发生时调用该触发器。触发事件是指对表中数据的操作，如插入、删除和修改。

存储过程是将常用的或很复杂的工作，预先用SQL语句写好并用一个指定的名称存储起来, 那么以后要叫数据库提供与已定义好的存储过程的功能相同的服务时,只需调用execute,即可自动完成命令。我的理解就是一堆sql的集合，可以建立非常复杂的查询，编译运行，所以运行一次后，以后再运行速度比单独执行SQL快很多

**2、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。

**3、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 doGet()或者doPost()方法。

结束服务，web 容器调用 servlet 的 destroy()方法。

**1、jsp与servlet的区别及联系**

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。

**2、 数据库连接池的工作机制：**

J2EE服务器启动的时候，会创建一定数量的池连接，并维持不少于此数量的池连接。

程序需要时，池驱动程序会返回一个未使用的池连接并将其标记为忙。

如果当前没有空闲连接，池驱动会新建一批，数量由配置参数决定。

当调用池连接完成后，池驱动将此连接标记为空闲，其他调用就可以使用这个连接。

**3、jsp有哪些内置对象?作用分别是什么?**

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

**1、forward 和redirect的区别**

答：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。

**2、Jsp的四种范围**

page 否是代表与一个页面相关的对象和属性。作用域在当前页.

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。

session 只要访问的浏览器不关闭，作用域就一直存在。

application 只要访问的服务器不关闭，作用域就一直存在。

**3、什么是B/S结构,C/S结构？**

C/S是Client/Server的缩写。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，客户机上只要安装一个浏览器(Browser)，如Netscape Navigator或Internet Explorer，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端实现。浏览器通过Web Server 同数据库进行数据交互。

**1.说说jdbc连接数据库的步骤**

1.注册驱动

2.获得连接

3.执行sql语句

4.获得结果集，进行结果集的处理

5.关闭结果集

6.关闭连接，释放资源

**2. statement和preparedstatement的区别**

1.statement是preparedstatemnet的父类

2.statement是直接发送sql语句到数据库，事先没有进行预编译。

prepatedstatement会将sql进行预编译，当sql语句要重复执行时，数据库会调用以前编译好的sql，所以preparedstatement在性能方面会更好

3.preparedstatement在执行sql时，对传入的参数进行强制类型转换，以保证数据格式与底层数据库格式一致。

**3. 解析一个XML文档有哪些方式？**

解析有：dom和sax两种

dom：把整个XML文档放入内存，适合XML随机访问,占用内存资源大

sax：事件驱动型的XML解析方式，顺序读取，不用一次装载整个文件，

遇到标签会触发一个事件，适和对XML的顺序访问，占用内存资源稍小

**1、XML文档定义有几种形式？**

a: 两种形式dtd schema

b: 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)，

c:有DOM,SAX,STAX 等

**2、薪水排序后薪水排名在第3－5的员工**

1)select \* from(select ename,sal,rownum rn from

(select ename,sal from emp\_44 where sal is not null order by sal desc) where rownum<6)where rn>2;

2)select \* from(select ename,sal,rownum rn from

(select ename,sal from emp\_44 where sal is not null order by sal desc))where rn between 3 and 5;

**3、删除一张表中所有数据的方式？**

1.truncate table命令将快速删除数据表中的所有记录

2.delete产生rollback，如果删除在数据量的表速度会很慢，同时会占用很多的rollback,segments.truncate是DLL操作，不产生 rollback，速度会快一些。

**1、 用一条sql语句取出所有姓名有重复的学员姓名和重复的记录数.**

select name, count(\*) from student group by name

having count(\*) > 1 order by count(\*) desc;

**2、 去除oracle数据库 表中重复数据应有如下两种方法：**

方法一：创建新表删除旧表法

1）create table student1 as select distinct id, name, score from student;

2）drop table student;

3）rename student1 to student;

方法二：使用rowid(地址)伪列

删除伪列地址除了最大地址值以外的记录

delete from temp where rowid not in (select max(rowid) from temp group by id);

删除伪列地址除了最大地址值以外的记录

delete from temp where rowid not in (select min(rowid) from temp group by id);

**3、 应用服务器有哪些：**

weblogic ，jboss，tomcat

**1、Hibernate优于JDBC的地方**

1）对jdbc访问数据库进行了封装，简化了数据访问层的重复代码

2）Hibernate操作数据库是面向对象的操作

**2、 hibernate工作原理**

1）读取并解析配置文件

2）读取并解析映射信息，创建SessionFactory

3）打开Sesssion

4）创建事务Transation

5）持久化操作

6）提交事务

7）关闭Session

8）关闭SesstionFactory

**为什么要用：**

1. 对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。

2. Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作

3. hibernate的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系。

**3、J2EE共有23中设计模式，说出其中几种常用的：**

工厂模式（Factory）单例模式（Singleton）

适配器模式（Adapter）代理模式（Proxy）

模板方法模式（Template Method）

**1、Hibernate对象状态有哪几种，并简单介绍一下。**

1）临时状态：内存对象，并没有保存在数据库

2）持久化状态 已经保存在数据库并纳入了session缓存中

3）游离状题啊 已经保存在数据库中，但没有纳入session缓存中

**2、spring 的优点？**

1）降低了组件之间的耦合性 ，实现了软件各层之间的解耦

2）可以使用容易提供的众多服务，如事务管理，消息服务等

3）容器提供单例模式支持

4）容器提供了AOP技术，利用它很容易实现如权限拦截，运行期监控等功能

5）容器提供了众多的辅助类，能加快应用的开发

6）spring对于主流的应用框架提供了集成支持，如hibernate，JPA，Struts等

7）spring属于低侵入式设计，代码的污染极低

8）独立于各种应用服务器

9）spring的DI机制降低了业务对象替换的复杂性

10）Spring的高度开放性，并不强制应用完全依赖于Spring，开发者可以自由选择spring的部分或全部

**3、介绍一下Struts的工作原理？**

1）读取配置。加载struts控制器.

2）用户请求 用户提交表单或调用URL向WEB应用程序服务器提交一个请求，请求的数据用HTTP协议上传给WEB服务器。

3）通过struts控制器进行处理，

4）经过一系列的拦截器处理

5）进行业务逻辑的处理

6）响应用户JSP将结果展现给用户。

**为什么要用：**

1）JSP、Servlet、JavaBean技术的出现给我们构建强大的企业应用系统提供了可能。但用这些技术构建的系统非常的繁乱。

2）基于Struts开发的应用:

不用再考虑公共问题

专心在业务实现上

结构统一，易于学习、维护

新手也可写出好程序

**1、什么是DI机制？**

依赖注入（Dependecy Injection）和控制反转（Inversion of Control）是同一个概念，具体的讲：当某个角色 需要另外一个角色协助的时候，在传统的程序设计过程中，通常由调用者来创建被调用者的实例。但在spring中 创建被调用者的工作不再由调用者来完成，因此称为控制反转。创建被调用者的工作由spring来完成，然后注入调用者

因此也称为依赖注入。

spring以动态灵活的方式来管理对象 ， 注入的两种方式，设置注入和构造注入。 设置注入的优点：直观，自然 构造注入的优点：可以在构造器中决定依赖关系的顺序。

**2、什么是AOP？**

面向切面编程（AOP）完善spring的依赖注入（DI），面向切面编程在spring中主要表现为两个方面：

1）面向切面编程提供声明式事务管理

2）spring支持用户自定义的切面 面向切面编程（aop）是对面向对象编程（oop）的补充， 面向对象编程将程序分解成各个层次的对象，面向切面编程将程序运行过程分解成各个切面。 AOP从程序运行角度考虑程序的结构，提取业务处理过程的切面，oop是静态的抽象，aop是动态的抽象， 是对应用执行过程中的步骤进行抽象，从而获得步骤之间的逻辑划分。

aop框架具有的两个特征：

1）各个步骤之间的良好隔离性

2）源代码无关性

**3、什么是Hibernate延迟加载?**

延迟加载机制是为了避免一些无谓的性能开销而提出来的，所谓延迟加载就是当在真正需要数据的时候，才真正执行数据加载操作。在Hibernate中提供了对实体对象的延迟加载以及对集合的延迟加载，另外在Hibernate3中还提供了对属性的延迟加载。

**1、什么是Hibernate延迟加载?**

延迟加载机制是为了避免一些无谓的性能开销而提出来的，所谓延迟加载就是当在真正需要数据的时候，才真正执行数据加载操作。在Hibernate中提供了对实体对象的延迟加载以及对集合的延迟加载，另外在Hibernate3中还提供了对属性的延迟加载。

**2、Hibernate中类之间的关联关系有几种?(如：一对多、多对多的关系)**

many-to-one、one-to-many、many-to-many、 one-to-one

**3、说下Hibernate的缓存机制**

一、hibernate一级缓存

（1）hibernate支持两个级别的缓存，默认只支持一级缓存；

（2）每个Session内部自带一个一级缓存；

（3）某个Session被关闭时，其对应的一级缓存自动清除；

二、hibernate二级缓存

(1) 二级缓存独立于session，默认不开启；

**1、Hibernate的查询方式**

本地SQL查询、Criteria、Hql

**2、为什么要用spring?**

Spring是一个轻量级的IOC和AOP框架。

IOC（控制反转）意味着将你设计好的类交给系统去控制，而不是在你的类内部控制。这称为控制反转

AOP（面向切面），它将那些影响多个类的行为封装到可重用的模块中，面向对象是把问题从同类事物中抽象出来，面向切面是把问题从不同类问题中抽象出来。

**3、struts框架提供了哪些辅助功能**

1）自动收集数据（成员变量）

2）支持类型转换

3）国际化

4）异常处理

5）标签

**1、struts1和struts2的区别**

1 struts1 控制器servlet struts2控制器filter

2 struts 收集数据时 ActionForm Struts2 成员变量

3 struts1 execute(Resquest,Response)

struts2 execute()

4 struts1 与ServletAPI耦合性强

struts2 与ServletAPI耦合性低

5 struts1 没有拦截器 struts2有拦截器(可以使代码各司其职)

6 struts1只能使用jstl标签，而struts2提供了OGNL表达式

**2、简述一下java基本数据类型及所占位数,**

java基本数据类型：4类8种

逻辑型：boolean false/true 1byte

文本型：char 2byte

整数型：byte(1byte)，short(2byte)，int(4byte)，long(8byte)

浮点数型：float(4byte)，double(8byte)

**3、说出5个的启动时异常:**

RunTimeException

|---NullPointerException

|——ArrayIndexOutOfBoundsException

|——ArithmeticException

|——ClassCastException

|——NumberFormatException

**1、HashMap和HashTable的区别：**

1）HashMap允许空键值对，HashTable不允许

2）HashMap不是线程安全的，HashTable是

3）HashMap直接实现Map接口，HashTable继承Dictionary类

**2、ArrayList，Vector，LinkedList存储性能和特性**

它们都实现List接口

ArrayList和Vector都是基于数组实现的

LinkedList基于双向循环链表（查找效率低，添加删除容易）

ArrayList不是线程安全的而Vector是线程安全的，所以速度上ArrayList高于Vector

**3、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。

**1、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。

**2、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。

**3、Overload和Override的区别。Overload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,方法名，参数列表，

返回值类型都得与父类的方法一致.

重载Overloading是一个类中多态性的一种表现。重载的方法是可以改变返回值的类型。

**1、用一句话总结一下冒泡排序。**

依次比较相邻的两个数，将小数放在前面，大数放在后面.

**2、实现线程安全的两种方式**

1）synchronized方法：通过在方法声明中加入synchronized关键字来声明

synchronized方法。

2）synchronized块：通过synchronized关键字来声明synchronized块。

**3、说一下 “==”和equals()方法在字串变量操作中的不同?**

”==”比较的是两个字符串对象的地址，equals()是比较的两个字符串的具体值。

**1、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。

**2、&与&&的区别?**

&位运算符:非短路运算符，它会把所有条件执行完毕之后，才会返回结果

&&逻辑运算符（and）：短路运算符，遇到不符合条件，立即终止程序的执行

**3、error和exception区别**

error:表示恢复不是不可能的一种严重的问题,比如：内存溢出，不指望程序处理

exception程序运行时的异常，如果程序设计合理从不会出现的情况

**1、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；

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。

**2、网络编程中设计并发服务器，使用多进程与多线程，请问有什么区别?**

1)进程：子进程是父进程的复制品子进程获得父进程数据空间堆和栈的复制品

2)线程：相对与进程而言，线程是一个更加接近与执行体的概念，它可以与同进程的其他线程共享数据，但拥有自己的栈空间，拥有独立的执行序列

两者都可以提高程序的并发度，提高程序运行效率和响应时间

线程和进程在使用上各有优缺点：线程执行开销小，但不利于资源管理和保护;而进程正相反同时，线程适合于在SMP机器上运行，而进程则可以跨机器迁移

**3、什么是反射？**

在运行过程中:

1.对于任意一个类,可以知道这个类的属性和方法.

2.对于任意一个对象,可以调用这个对象的任意方法.

对于这种动态获取的信息,以及动态调用对象方法的功能称为反射机制.

**1、java反射机制提供的功能:**

1）运行时判断任意对象的所属类;

2）运行时构造任意类的对象;

3）运行时判断和调用对象的成员变量和方法;

4）生成动态代理.

**2、什么是回调函数？**

某个程序S(Student.main) 调用 服务程序A(Arrays) 中的某个方法(sort), 服务程序A的sort方法在某个时候反过来调用S的某个方法(compareTo),这种情况下,compareTo叫做S的回调方法。

例如：public class Student implements Comparable{

private int id;

private String name;

private int age;

private int score;

//构造器

//getter / setter方法

//回调方法

public int compareTo(Object obj){

return

this.id - ((Student)obj).id;

}

}

Student s1 = new Student(1,"a",18,89);

Student s2 = new Student(2,"x",22,94);

Student s3 = new Student(3,"w",19,78);

Student [] arrs = {s1,s2,s3};

Arrays.sort(arrs);

**3、遍历文件夹下所有.java的文件?**

public void listFiles(String path){

File dir = new File(path);

Files files [] = dir.listFiles(new FileFilter(){

public boolean accept(File f){

return f.getName().endWith(".java");

}

});

for(File file : files){

System.out.println(file.getName());

}

}

**1、线程的基本概念、线程的本状态以及状态之间的关系**

新建 (Born) : 新建的线程处于新建状态

就绪 (Ready) : 在创建线程后，它将处于就绪状态，等待 start() 方法被调用

运行 (Running) : 线程在开始执行时进入运行状态

睡眠 (Sleeping) : 线程的执行可通过使用 sleep() 方法来暂时中止。在睡眠后，线程将进入就绪状态

等待 (Waiting) : 如果调用了 wait() 方法，线程将处于等待状态。用于在两个或多个线程并发运行时。

挂起 (Suspended) : 在临时停止或中断线程的执行时，线程就处于挂起状态。

恢复 (Resume) : 在挂起的线程被恢复执行时，可以说它已被恢复。

阻塞 (Blocked) – 在线程等待一个事件时（例如输入/输出操作），就称其处于阻塞状态。 死亡 (Dead) – 在 run() 方法已完成执行或其 stop() 方法被调用之后，线程就处于死亡状态。 串行化的注意事项以及如何实现串行化答：如果有循环引用是不可以串行化的。对象输出流的WriteObject方法和 对象输入流的ReadObect 方法

**2、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。

**3、什么是触发器和存储过程？**

触发器是存储在数据库中的块，这些块一旦被构造后，就可以多次执行，当触发它的事件发生时调用该触发器。触发事件是指对表中数据的操作，如插入、删除和修改。

存储过程是将常用的或很复杂的工作，预先用SQL语句写好并用一个指定的名称存储起来, 那么以后要叫数据库提供与已定义好的存储过程的功能相同的服务时,只需调用execute,即可自动完成命令。我的理解就是一堆sql的集合，可以建立非常复杂的查询，编译运行，所以运行一次后，以后再运行速度比单独执行SQL快很多

**1.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。

**2.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 doGet()或者doPost()方法。

结束服务，web 容器调用 servlet 的 destroy()方法。

**3. jsp与servlet的区别及联系**

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。

**1、数据库连接池的工作机制**：

J2EE服务器启动的时候，会创建一定数量的池连接，并维持不少于此数量的池连接。

程序需要时，池驱动程序会返回一个未使用的池连接并将其标记为忙。

如果当前没有空闲连接，池驱动会新建一批，数量由配置参数决定。

当调用池连接完成后，池驱动将此连接标记为空闲，其他调用就可以使用这个连接。

**2、jsp有哪些内置对象?作用分别是什么?**

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

**3、forward 和redirect的区别**

答：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。

**1、Jsp的四种范围**

page 否是代表与一个页面相关的对象和属性。作用域在当前页.

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。

session 只要访问的浏览器不关闭，作用域就一直存在。

application 只要访问的服务器不关闭，作用域就一直存在。

**2、什么是B/S结构,C/S结构？**

C/S是Client/Server的缩写。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，客户机上只要安装一个浏览器(Browser)，如Netscape Navigator或Internet Explorer，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端实现。浏览器通过Web Server 同数据库进行数据交互。

**3、说说jdbc连接数据库的步骤**

1）注册驱动

2）获得连接

3）执行sql语句

4）获得结果集，进行结果集的处理

5）关闭结果集

6）关闭连接，释放资源

**1、statement和preparedstatement的区别**

1）statement是preparedstatemnet的父类

2）statement是直接发送sql语句到数据库，事先没有进行预编译。

prepatedstatement会将sql进行预编译，当sql语句要重复执行时，数据库会调用以前编译好的sql，所以preparedstatement在性能方面会更好

3）preparedstatement在执行sql时，对传入的参数进行强制类型转换，以保证数据格式与底层数据库格式一致。

**2、解析一个XML文档有哪些方式？**

解析有：dom和sax两种

dom：把整个XML文档放入内存，适合XML随机访问,占用内存资源大

sax：事件驱动型的XML解析方式，顺序读取，不用一次装载整个文件，

遇到标签会触发一个事件，适和对XML的顺序访问，占用内存资源稍小

**3、XML文档定义有几种形式？**

a: 两种形式dtd schema

b: 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)，

c:有DOM,SAX,STAX 等

**1.薪水排序后薪水排名在第3－5的员工**

1)select \* from(select ename,sal,rownum rn from

(select ename,sal from emp\_44 where sal is not null order by sal desc) where rownum<6)where rn>2;

2)select \* from(select ename,sal,rownum rn from

(select ename,sal from emp\_44 where sal is not null order by sal desc))where rn between 3 and 5;

**2. 删除一张表中所有数据的方式？**

1.truncate table命令将快速删除数据表中的所有记录

2.delete产生rollback，如果删除在数据量的表速度会很慢，同时会占用很多的rollback,segments.truncate是DLL操作，不产生 rollback，速度会快一些。

**3. 用一条sql语句取出所有姓名有重复的学员姓名和重复的记录数.**

select name, count(\*) from student group by name

having count(\*) > 1 order by count(\*) desc;

**1.什么是面向对象？面向对象有哪些特征？**

以事物为驱动的编程思想。

我理解的面向对象是必须有一个具体的事物，

这个事物可以是一个具体的人，一个具体的物，

这个事物有自己的特征（属性），有自己的行为（方法）,

那么对这个事物特征的描述，和对行为的操作称为面向对象的。

继承：

实体和实体之间的一种关系。

例如：父子类之间 对于父类而言，他的一些东西（属性和方法）是可以共享的，

对于子类而言，它可以共享到父类的一些东西（属性和方法），那么

我们把父类与子类的这种关系称为继承。

封装：

实体的有些东西是自身特有的，不需要共享。

例如：

小明的爸爸有自己不想公开的东西，那么对于小明的爸爸而言，

这些东西应该设置为私有的 。这就体现出java 的封装。

多态：行为和引用。

一个类中 （行为）。

例如: 打 ， 打人，打车，打麻将 ，也叫方法重载

父子类之间 （引用）

儿子可以引用父亲的行为,也叫方法的重写。

**2.说一下什么是javaBean 规范：**

1.要有包--类必须写在包中

2.属性都是私有--private

3.要有无参构造方法

4.写有效的setXXX getXXX 方法

5.实现序列化接口

**3.简述一下java 基本数据类型及所占位数,**

java 基本数据类型：4 类8 种

逻辑型：boolean false/true 1byte 8bit

字符型：char 2byte 16bit 运算时按照int 类型运算

整数型：

byte(1byte 8bit) 运算时按照int 类型运算

short(2byte 16bit) 运算时按照int 类型运算

**1、说出9 个的启动时异常:**

RunTimeException

|---NullPointerException

|——ArrayIndexOutOfBoundsException

|——ArithmeticException

|——ClassCastException

|——NumberFormatException

|——SystemException

|——ArrayStoreException

|——EmptyStackException

|——BufferOverflowException

注：异常Exception 分为两大类：

编译时异常：编译期间要检查的

运行时异常：编译期间不检查的

**2、说出5 个常用的java-api 包**

java.lang

java.util

java.sql

java.text

java.io

**3、说出String 类中常用的5 个方法**

charAt() 返回指定索引处的char 值

concat() 将指定字符串连接到此字符串的结尾

length()返回此字符串的长度

split()根据给定正则表达式的匹配拆分此字符串

trim()返回字符串的副本，忽略前导空白和尾部空白.

**1、HashMap 和HashTable 的区别：**

1.HashMap 允许空键值对，HashTable 不允许

2.HashMap 不是线程安全的，HashTable 是

3.HashMap 直接实现Map 接口，HashTable 直接继承Dictionary 类

**2、ArrayList ，Vector ，LinkedList 存储性能和特性**

它们都实现List 接口

ArrayList 和Vector 都是基于数组实现的

LinkedList 基于双向循环链表（查找效率低，添加删除容易）

ArrayList 不是线程安全的而Vector 是线程安全的，所以速度上ArrayList 高于Vector

**3、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。

**1、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。

**2、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。

**3、Overload 和Override 的区别。Overload 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,方法名，参数列表，

返回值类型都得与父类的方法一致.

重载Overloading 是一个类中多态性的一种表现。重载的方法是可以改变返回值的类型。

**1、用一句话总结一下冒泡排序。**

依次比较相邻的两个数，将小数放在前面，大数放在后面.

**2、写出选择，冒泡，插入排序的代码**

选择排序：

for (int i=0;i<ary.length-1;i++){

for (int j=i+1;j<ary.length;j++){

if (ary[i]>ary[j]){

int t = ary[i];

ary[i]=ary[j];

ary[j]=t;

}

}

}

冒泡排序：

for (int i=0;i<ary.length-1;i++){

for (int j=0;j<ary.length-i-1;j++){

if (ary[j]>ary[j+1]){

int t = ary[j];

ary[j]=ary[j+1];

ary[j+1]=t;

}

}

}

插入排序：

for (int i = 1; i < ary.length; i++) {

int temp = ary[i];

int j;

for (j = i - 1; j >= 0 && temp < ary[j]; j--) {

ary[j + 1] = ary[j];

}

ary[j + 1] = temp;

}

**3、写出二分查找的代码：**

注：二分法从数组查询元素，必须保证数组内部元素是有顺序的。

意思就是先对数组进行排序。

public class BinarySearch {

static int idx=1;

public static void find(int leftIndex,int rightIndex,int val,int[] arr){

int midIndex=(rightIndex+leftIndex)/2;

int midVal=arr[midIndex];//找到中间的数

if(rightIndex>=leftIndex){

//如果要找的数比midVal大

if(midVal>val){

idx++;

//在arr左边数中找

find (leftIndex,midIndex-1,val,arr);

}else if(midVal<val){

//在arr 的右边去查找

idx++;

find (midIndex+1,rightIndex,val,arr);

}else if(midVal==val){

System.out.println("找到下标"+midIndex+"共查找了"+idx+"次");

}

}else{

System.out.println("没有该数!");

}

}

public static void main(String[] args) {

int[] arr=new int[]{100,300,500,800,1000,2000,3000};

find (0,arr.length-1,1000,arr);

}

}

**1、实现多线程的两种方式**

1)继承Thread 类

2)实现Runnable 接口

优先选择实现Runnable 接口 因为比较灵活。

**2、简述如下几个概念。**

程序：指令+数据的byte 序列。PC 上面的可运行文件

进程：正在运行的程序，是程序动态的执行过程。

线程：在进程的内部，程序并发执行的过程。

并发: 进程是并发运行的, 操作系统(OS)将时间划分为很多时间片段, 尽可能

均匀分配给正在运行的程序, 微观上进程走走停停, 宏观上都在运行

这种都运行的现象叫: 并发。

**3、说一下 “==”和equals()方法在字串变量操作中的不同?**

”==”比较的是两个字符串对象的地址，equals()是比较的两个字符串的具体值。

**1、sleep()和wait()有什么区别？**

sleep 是线程类（Thread ）的方法，导致此线程暂停，然后执行给定时间，让出

cpu 给其他线程，但不会释放对象锁，时间到了自动恢复.

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。

**2、&与&& 的区别?**

&位运算符,非短路逻辑运算符，它会把所有条件执行完毕之后，才会返回结果

&&逻辑运算符（and ）：短路运算符，遇到不符合条件，立即终止程序的执行

**3、error 和exception 区别**

error:表示恢复不是不可能的一种严重的问题, 比如：内存溢出，不指望程序处理

exception 程序运行时的异常，如果程序设计合理从不会出现的情况

**1、请说出你所知道的线程同步的方法。**

join():合并当前线程，相当于方法调用

yield():让出cpu

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。

suspended() –挂起，该方法已过时: 在临时停止或中断线程的执行时，线程就处于挂起状态。

resume()—恢复，该方法已过时 : 在挂起的线程被恢复执行时，可以说它已被恢复。

stop()--暴力结束当前正在执行的线程，该方法已过时

**2、IO流**

字节流：

FileInputStream 和FileOutputStream 对文件基本的输入输出操作

BufferedInputStream 和BufferedOutputStream 加缓冲区的操作

DataInputStream 和DataOutputStream 对基本操作的扩展,

可以一次读一种基本数据类型的数据.

字符流:

InputStreamReader 和OutputStreamWriter 基本字符流 可以处理字符编码

BuffereReader 和PrintWriter 可以一次读取一行数据

**3、什么是反射？**

在运行过程中:

1.对于任意一个类,可以知道这个类的属性和方法.

2.对于任意一个对象,可以调用这个对象的任意方法.

对于这种动态获取的信息, 以及动态调用对象方法的功能称为反射机制.

java 反射机制提供的功能:

1.运行时判断任意对象的所属类;

2.运行时构造任意类的对象;

3.运行时判断和调用对象的成员变量和方法;

4.生成动态代理.

**1、什么是回调函数？**

某个程序S(Student.main) 调用 服务程序A(Arrays) 中的某个方法(sort), 服务程

序 A 的 sort 方法在某个时候反过来调用 S 的某个方法(compareTo),这种情况

下,compareTo 叫做S 的回调方法。

例如：public class Student implements Comparable{

private int id;

private String name;

private int age;

private int score;

//构造器

//getter / setter 方法

// 回调方法

public int compareTo(Object obj){

return

this.id - ((Student)obj).id;

}

}

Student s1 = new Student(1,"a",18,89);

Student s2 = new Student(2,"x",22,94);

Student s3 = new Student(3,"w",19,78);

Student [] arrs = {s1,s2,s3};

Arrays.sort(arrs);

**2、遍历文件夹下所有.java 的文件?**

public void listFiles(String path){

File dir = new File(path);

Files files [] = dir.listFiles(new FileFilter(){

public boolean accept(File f){

return f.getName().endWith(".java");

}

});

for(File file : files){

System.out.println(file.getName());

}

}

**3、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。

**1.薪水排序后薪水排名在第3--5 的员工**

1)select \* from(select ename,sal,rownum rn from

(select ename,sal from emp\_44 where sal is not null order by sal desc) where rownum<6)where

rn>2;

2)select \* from(select ename,sal,rownum rn from

(select ename,sal from emp\_44 where sal is not null order by sal desc))where rn between 3 and

5;

**2.删除一张表中所有数据的方式？**

1.truncate table 命令将快速删除数据表中的所有记录

2.delete 产生 rollback ，如果删除在数据量的表速度会很慢，同时会占用很多的

rollback,segments.truncate 是DLL 操作，不产生 rollback，速度会快一些。

**3.用一条sql 语句取出所有姓名有重复的学员姓名和重复的记录数.**

select name, count(\*) from student group by name

having count(\*) > 1 order by count(\*) desc;

**1、去除oracle 数据库 表中重复数据应有如下两种方法：**

方法一：创建新表删除旧表法

1）create table student1 as select distinct id, name, score from student;

2 ）drop table student;

3 ）rename student1 to student;

方法二：使用rowid(地址)伪列

删除伪列地址除了最大地址值以外的记录

delete from temp where rowid not in (select max(rowid) from temp group by id);

删除伪列地址除了最大地址值以外的记录

delete from temp where rowid not in (select min(rowid) from temp group by id);

**2、按工资进行排名，排名从1 开始，工资相同排名相同（如果两人并列第一则没有第二名，从第三名继续排）**

方法一：

select empno,ename,sal, (select count(\*)+1 from iv\_emp where sal>e.sal) rank

from iv\_emp e

order by rank asc;

方法二：select e.empno,e.ename,e.salary,rank() over(order by

salary desc) mingci from emp\_ding e;

**3、按工资进行排名，排名从1 开始，工资相同排名相同。**

select e.empno,e.ename,e.salary,dense\_rank() over(order by

salary desc) mingci from emp\_ding e;

**1、查询每个部门工资最高的前2 名员工**

方法一:

select \*

from( select empno,ename,salary, deptno,(select count(\*)+1 from emp\_ding where

salary>e.salary and deptno = e.deptno ) rank

from emp\_ding e) ee

where ee.rank<=2

order by deptno,rank asc;

方法二：

select \* from ( select e.empno,e.deptno,e.ename,e.salary, dense\_rank()

over (partition by deptno order by salary desc) mingci from emp\_ding e ) x where

x.mingci<=2;

PL/SQL 部分：

解释如下概念

1.触发器(trigger)：是存储在数据库中的块，这些块一旦被构造后，就可以多次执行，

当触发它的事件发生时调用该触发器。触发事件是指对表中数据的操作，如插入(inserting)、

删除(deleting)和修改(updating) 。需要注意的是：触发器没有参数，没有返回值，不能够显

示调用。

create or replace trigger user\_log –user\_log 该触发器的名字

before delete or update or insert

on t\_user

for each row

begin

if deleting then

insert into t\_user\_log values

(myseq.nextval,'用户'||:old.username||'被删除了'); --:old 特殊的变量表示操作的原数据

elsif inserting then

insert into t\_user\_log values

(myseq.nextval,'用户'||:new.username||'被添加了');--new,特殊的变量表示操做的新数据

elsif updating then

insert into t\_user\_log values

(myseq.nextval,'用户'||:old.username||'被修改了'||:new.username);

else

null;

end if;

end;

2.存储过程(procedure)：是将常用的或很复杂的工作，预先用SQL 语句写好并用一个指定

的名称存储起来, 那么以后要叫数据库提供与已定义好的存储过程的功能相同的服务时, 只

需调用execute,即可自动运行。我的理解就是一堆sql 的集合，可以建立非常复杂的查询，

编译运行运行一次后，以后再运行速度比单独执行SQL 快很多.

1、具有删除功能的存储过程

create or replace procedure

delete\_stu(v\_stuid number)

is

begin

delete from t\_student where

sid=v\_stuid;

end;

调用存储过程：

exec delete\_stu(1001);

3.游标(cursor) ：游标(cursor)，是一种控制结构，可以帮助我们处理多条记录。游标不是一

种数据类型。

对于游标的使用：

1、声明游标 cursor cursor\_name is select....

2、打开游标 open cursor\_name

3、从游标中抓取数据 fetch cursor\_name into v\_row\_data

4 、关闭游标 close cursor\_name

静态游标的使用：

declare

v\_data t\_student%rowtype;

cursor mycursor

is

select \* from t\_student;

begin

open mycursor;

fetch mycursor into v\_data;

dbms\_output.put\_line(v\_data.stuname);

close mycursor;

end;

动态游标的使用：

declare

type cur\_type is ref cursor;

v\_data t\_student%rowtype;

v\_mycursor cur\_type;

begin

open v\_mycursor

for

select \* from t\_student;

fetch v\_mycursor into v\_data;

while v\_mycursor%found loop

dbms\_output.put\_line(v\_data.stuname);

fetch v\_mycursor into v\_data;

end loop;

close v\_mycursor;

end;

4.索引（index ）：相当于书签，能够提高数据的访问效率

主键列自动创建索引

其他列必须手动创建索引。

5.视图(view):虚拟的存在于数据库中。

用到视图的时候，系统会自动搜索数据库中的表，数据都是从表中取到的，

并不是从视图中直接拿到的。

6.函数(function):必须有返回值，一般用于查询。

create or replace function emp\_income

(v\_empno emp\_lc.empno%type)

return number

is

v\_income number(10,2);

begin

select sal\*12+nvl(comm,0)\*12

into v\_income

from emp\_lc where empno=v\_empno;

return v\_income;

end;

7.序列(sequence):oracle 数据库没有提供主键自动增长，为了方便操作，

oracle 数据库提供了序列，它的作用就是能够间接的实现主键自增长。

**1.说说jdbc 连接数据库的步骤**

1.注册驱动Class.forName(“”);

2.获得连接Connection conn=DriverManager.getConnection(“”);

3.创建声明

4.执行sql 语句,获得结果集，进行结果集的处理

5.关闭结果集

6.关闭声明

7.关闭连接，释放资源

**2. statement 和preparedstatement 的区别**

1.statement 是preparedstatemnet 的父类

2.statement 是直接发送sql 语句到数据库，事先没有进行预编译,这样会导致

恶意注入sql 语句的问题出现.prepatedstatement 会将sql 进行预编译，当sql 语句要重

复执行时，数据库会调用以前编译好的sql，所以preparedstatement 在性能方面会更好

3.preparedstatement 在执行sql 时，对传入的参数进行强制类型转换，以保证数据格式

与底层数据库格式一致。

4.PreparedStatement 相对于Statemnet 要安全一些，可以有效防止sql 注入.

5.PreparedStatement 能够执行批处理，而Statement 不行.

另：CallableStatement 是用来调用存储过程的

**3.数据库数据如何提高查询速度？**

1. jdbc setFetchSize （）；

2. 数据库：索引

3. 把大表可以拆分成多个小表

4. dba 对表进行分区

**1、解析一个XML 文档有哪些方式？**

解析有：dom 和sax 两种

dom：把整个XML 文档放入内存，适合XML 随机访问, 占用内存资源大

sax：事件驱动型的XML 解析方式，顺序读取，不用一次装载整个件，遇到 标签会触发一个事件，适和对XML 的顺序访问，占用内存资源稍小

**2、XML文档定义有几种形式？**

a: 两种形式dtd schema

b: 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)，

c:有DOM,SAX,STAX 等

**3、Java 连接数据库有几种模式(方式) ？**

1.直接连接JDBC （瘦连接）

2.通过客户端（胖连接）

3.使用桥连接ODBC(桥接)

**1、分页？**

1）缓存分页（假分页）

特点: 第一次取全部数据时比较慢.(可能非常慢) 以后每次都从缓存中取数据, 比较快只访问数据库一次. 内存压力大.如果需求是一般只查看前几页,浪费内存资源.

2）数据库分页（查询分页）

特点:每次只向数据库查询并返回一页的数据频繁的数据库访问.每次查询的时间都差不多.没有缓存压力.

两者比较：

基于缓存的方式,适合小表,全部查看

基于查询的方式,适合大表,只查询部分数据

**2、一个完整的网页由结构、表现、行为组成。**

结构指HTML、XHTML、XML 等

表现指CSS 样式

行为指脚本(JS)

JavaScript 是Netscape(网景)推出客户端运行的解释性脚本语言。

注：Javascript 是弱类型语言，变量的声明只能用var

Html 基本结构：<html> <head><title></title></head><body></body></html>

设置隐藏文本框：<input type="hidden" name="id">

设置只读文本框：<input type="text" readonly="readonly" name="username">

Undefined:未定义

1.声明但没有赋值

2.没有声明的变量

获得对象：document.getElementById(“id”);

获得对象的值：document.getElementById(“id”).value

**3、创建数组：**

1）var 变量名称 = new Array()

2 ）var 变量名称 = new Array(长度)

3 ）var 变量名称 = new Array(值,....)

4 ）var 变量名称 = [值,...]

属性

length --> 返回数组成员的数目

方法

push() --> 向数组的未尾添加一个或多个成员,返回为新数组的长度

unshift() --> 向数组的开头添加一个或多个成员,返回为新数组的长度

shift() -> 删除数组的第一个成员，并且返回该成员。

pop() --> 删除数组的最后一个成员，并且返回该成员。

splice() --> 插入/删除数组成员，并且用新成员取代原有成员。

slice(start[,end]) --> 返回数组的一部分

join() --> 使用分隔符将数据成员连接在一起

**1、什么是ajax ？asynchronous javascript and xml:异步的javascript 和xml.**

为了解决传统的 web 应用当中“请求-处理-等待-响应”的弊端而创建的技术，其实质是：使用javascript 调用浏览器内置的一个对象（XmlHttpRequest)异步向服务器发送请求，服务器返回xml 或者text 给XmlHttpRequest ，然后，javascript 使用服务器返回的数据更新页面。在整个过程当中，页面没有任何的刷新。

**2.使用ajax 发送异步请求：**

首先获得浏览器内置对象

function getXhr(){

var xhr=null;

/\* (typeof XMLHttpRequest) !='undefined' \*/

if(window.XMLHttpRequest){

xhr=new XMLHttpRequest();

}else{

xhr=new ActiveXObject('Microsoft.XMLHttp');

}

return xhr;}

/\* Ajax 发送get 请求\*/

function sendAjaxGet(){

var username=document.getElementById("t1").value;

var password=document.getElementById("t2").value;

var s1=document.getElementById("s1");

var xhr=getXHR();

var uri="test.do?username="+username+"&password="+password;

xhr.open("get",encodeURI(uri),true);

xhr.onreadystatechange=function(){

if(xhr.readyState==4){

if(xhr.status==200){ var txt=xhr.responseText; s1.innerHTML=txt;

}else{ s1.innerHTML="系统异常，稍后再试!";

}

}else{ s1.innerHTML="正在验证....";

}

}

xhr.send(null);

}

/\*ajax 发送post 请求 \*/

function sendAjaxPost(){

var username=document.getElementById("t1").value;

var password=document.getElementById("t2").value;

var s1=document.getElementById("s1");

var xhr=getXHR();

var uri="test.do";

xhr.open("post",encodeURI(uri),true);

xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

xhr.onreadystatechange=function(){

if(xhr.readyState==4){

if(xhr.status==200){ var txt=xhr.responseText;

s1.innerHTML=txt;

}else{ s1.innerHTML="系统异常，稍后再试!"; }

}else{ s1.innerHTML="正在验证....";

}

}

xhr.send("username="+username+"&password="+password);

}

**3、什么是JQuery?**

Jquery 是一个js 框架，对底层js 代码进行封装，方便用户操作。

并且jquery 提供了对ajax 的支持，使用jquery 也可以发送异步请求，

不需要屏蔽浏览器差异，代码书写简单。

**1、MVC 的各个部分都有那些技术来实现?如何实现?**

M-Model 模型

模型的职责是负责业务逻辑。包含两部分：业务数据和业务处理逻辑。

例如：实体类、DAO、Service 等都属于模型层

V-View 试图

试图的职责是负责显示界面和用户交互 （收集用户信息）。

属于视图的类是不包含业务逻辑和控制逻辑的JSP(如果 JSP 页面中有<% %>就不能

算是视图层的类，或者JSP 中更有转发和重定向的控制逻辑也是不可以的)。

C-Controller 控制器

控制器是模型层M 和视图层V 之间的桥梁，用于控制流程。

比如我们之前项目中写的Action 或Servlet

**2、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 doGet()或者doPost()方法。

结束服务，web 容器调用 servlet 的 destroy()方法。

**3、jsp与servlet的区别及联系**

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。

**1、数据库连接池的工作机制：**

J2EE 服务器启动的时候，会创建一定数量的池连接，并维持不少于此数量的池连接。

程序需要时，池驱动程序会返回一个未使用的池连接并将其标记为忙。

如果当前没有空闲连接，池驱动会新建一批，数量由配置参数决定。

当调用池连接完成后，池驱动将此连接标记为空闲，其他调用就可以使用这个连接。

**2、jsp 有哪些内置对象?作用分别是什么?**

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）：

page jsp 网页本身

pageContext 网页的属性是在这里管理

request 客户端请求，此请求会包含来自GET/POST 请求的参数

session 与请求有关的会话期，只要浏览器不关闭，保存在session 里面的值都存在.

application servlet 正在执行的内容

config servlet 的构架部件

response 网页传回用户端的回应

out 用来传送回应的输出

exception 针对错误网页，未捕捉的例外

**3、forward(转发) 和redirect(重定向)的区别**

答：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 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。

**1、Jsp 的四种会话范围**

page 是代表与一个页面相关的对象和属性。作用域在当前页.

request 是代表与Web客户机发出的一个请求相关的对象和属性。

session 只要访问的浏览器不关闭，作用域就一直存在。

application 只要访问的服务器不关闭，作用域就一直存在。

**2、什么是B/S 结构,C/S 结构？**

C/S 是Client/Server 的缩写。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，客户机上只要安装一个浏览器(Browser)，如 Netscape

Navigator 或Internet Explorer，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端实现。浏览器通过Web Server 同数据库进行数据交互。

**3、编码格式转换问题：**

String str=new String(" 中国".getBytes("ISO-8859-1"),"GBK").trim();

**1、URL 和URI 的区别？**

URL：统一资源定位符, 指的是Internet 文件在网上的地址, 用在客户程序和服务器上，

定位客户端连接服务器所需要的信息，它不仅定位了这个信息资源，而且定义了如何找到这

个资源.

URI：统一资源标识符，Web 上可用的每种资源 ：HTML 文档、图像、视频片段、程序

等„是由一个通过通用资源标志符(Universal Resource Identifier, 简称"URI")进行

定位,不局限于客户端服务器.

URI 一般由三部分组成:

1.访问资源的命名机制。

2.存放资源的主机名。

3.资源自身的名称，由路径表示。

URL 是URI 的一个子集。

**2、介绍一下Struts的工作原理？**

1）读取配置,加载struts 控制器.

2）用户请求 用户提交表单或调用URL 向WEB 应用程序服务器提交一个请求，请求的数据

用HTTP 协议上传给WEB 服务器。

3）通过struts 控制器进行处理，

4）经过一系列的拦截器处理

5）进行业务逻辑的处理

6）响应用户JSP 将结果展现给用户。

**3、为什么要用Struts框架：**

1） JSP、Servlet、JavaBean技术的出现给我们构建强大的企业应用系统提供了可能。但用

这些技术构建的系统非常的繁乱。

2）基于Struts开发的应用:

不用再考虑公共问题

专心在业务实现上

结构统一，易于学习、维护

新手也可写出好程序

**1、struts 框架提供了哪些辅助功能**

1） 自动收集数据（成员变量）

2） 支持类型转换

3） 国际化

4） 异常处理

5） 标签

**2、 struts1 和struts2 的区别**

1） struts1 控制器servlet struts2 控制器filter

2） struts 收集数据时 ActionForm Struts2 成员变量

3） struts1 execute(Resquest,Response)

struts2 execute()

4） struts1 与ServletAPI 耦合性强

struts2 与ServletAPI 耦合性低

5） struts1 没有拦截器 struts2 有拦截器(可以使代码各司其职)

6） struts1 只能使用jstl 标签，而struts2 提供了OGNL 表达式

**3、 hibernate工作原理**

1）读取并解析配置文件

2）读取并解析映射信息，创建SessionFactory

3）打开Sesssion

4）创建事务Transation

5）持久化操作

6）提交事务

7）关闭Session

8）关闭SesstionFactory

**1、为什么要用Hibernate框架：**

1）对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。

2）Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度

的简化DAO层的编码工作

3）hibernate 的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持

各种关系数据库，从一对一到多对多的各种复杂关系。

4）Hibernate 是面向对象进行操作的

**2、Hibernate 对象状态有哪几种，并简单介绍一下。**

1.临时状态：内存对象，并没有保存在数据库

2.持久化状态 已经保存在数据库并纳入了session 缓存中

3.游离状态 已经保存在数据库中，但没有纳入session 缓存中

**3、什么是Hibernate延迟加载?**

延迟加载机制是为了避免一些无谓的性能开销而提出来的，所谓延迟加载就是当在真正需要

数据的时候，才真正执行数据加载操作。在Hibernate 中提供了对实体对象的延迟加载以及

对集合的延迟加载，另外在Hibernate3 中还提供了对属性的延迟加载。

**1、下面是Spring依赖注入方式的是（AB）**

A.set方法注入

B.构造方法的注入

C.get方法的注入

D.接口的注入

**2、下面关于在Spring中配置Bean的id属性的说法正确的是（BD）**

A.id属性是必须，没有id属性就会报错

B.id属性不是必须的，可以没有

C.id属性的值可以重复

D.id属性的值不可以重复

**3、为什么要使用Spring框架，它有什么优点？**

(1) 轻量级的框架

(2) 非侵入性的

(3) 可以整合其它的框架，比如Struts， Hibernate等

(4) 可以提供事务管理

**1、数据库题**

1）建一个员工信息表employee, 表中 id （员工代码)、sex (员工性别)、name(姓名）、departmentid (部门代码) 、address (地址)，birthdate (生日)、postcode (邮编) 、salary〔薪水)、workdate (入职日期) 、remark （备注信息），其中 postcode 、 remark 可以为空，薪水需为numher 类型.，生日、入职日期为 date 类型。以员工代码作为主键。

2）插入两条记录，id分别为0023, 1023，其余信息自己编造。

3）修改id为 0023 的员工的入职日期为2007-12-31

1).create table employee(

id number(10) primary key,

sex varchar2(4) not null,

name varchar2(20) not null,

departmentid number(10) not null,

address varchar2(20) not null,

birthdate date not null,

postcode varchar2(20),

salary number(20) not null,

workdate date not null,

remark varchar2(20)

)

2).insert into employee values(0023,'男','zhangsan',1001,'北京',to\_date('199 0-12-03','yyyy-mm-dd'),null,5000,to\_date('2013-11-06','yyyy-mm-dd'),null);

insert into employee values(1023,'女','lisi',1002,'上海',to\_date('1989-0 7-04','yyyy-mm-dd'),null,8000,to\_date('2012-10-23','yyyy-mm-dd'),null);

3).update employee set workdate=to\_date('2007-12-31','yyyy-mm-dd') where id=0023;

**2、简述一下servlet的生命周期**

Servlet 生命周期就是指创建 Servlet 实例后响应客户请求直至销毁的全过程。

Serlvet 生命周期的三个方法:init()-->service()-->destroy()

Servlet 生命周期的各个阶段:

1.实例化:Servlet 容器创建Servlet 类的实例对象；

2.初始化:Servlet 容器调用 Servlet 的 init()方法；

3.服务:如果请求 Servlet,则容器调用 service()方法；

4.销毁:销毁实例之前调用 destroy()方法。

**3、什么是垃圾回收？什么时候触发垃圾回收？如何降低垃圾回收的触发频率？它能保证程序有足够的可用内存吗？**

垃圾回收(GC)是 Java 语言的一个重要特性，作用是释放不再被使用的内存。垃圾回收由系统 进行管理。在系统认为需要的时候自动启动一个线程进行处理。 尽量减少垃圾内存，也就是新建对象的数量，可以降低垃圾回收的频率。 垃圾回收机制无法保证有足够的内存。

**1、什么是 java 序列化，如何实现 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。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、请写出5 种常见到的runtime exception异常**

NullPointerException：当操作一个空引用时会出现此错误。

NumberFormatException：数据格式转换出现问题时出现此异常。

ClassCastException：强制类型转换类型不匹配时出现此异常。

ArrayIndexOutOfBoundsException：数组下标越界，当使用一个不存在的数组下标时出现此异常。

ArithmeticException：数学运行错误时出现此异常

**3、error和exception 有什么区别？**

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。

**1、事务是什么？有哪些属性，并简要说明这些属性的含义。**

事务(Transaction)是访问并可能更新数据库中各种数据项的一个程序执行单元(unit)。事务 通常由高级数据库操纵语言或编程语言（如 SQL，C++或 Java）书写的用户程序的执行所引起，并用形如 begin transaction 和 end transaction 语句（或函数调用）来界定。事务由事务开始(begin transaction)和事务结束(end transaction)之间执行的全体操作组成。

事务应该具有 4个属性：原子性、一致性、隔离性、持续性。这四个属性通常称为 ACID 特性。 原子性（atomicity）。一个事务是一个不可分割的工作单位，事务中包括的诸操作要么都做，要么都不做。一致性（consistency）。事务必须是使数据库从一个一致性状态变到另一个一致性状态。一致性与原子性是密切相关的。隔离性（isolation）。一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。 持久性（durability）。持续性也称永久性（permanence），指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其有任何影响。

**2、Java 数据库编程包含哪些类？Java数据库编程的基本过程是什么？**

用到的类：Connection、ResultSet、Statement、PreparedStatement

Java 中访问数据库的步骤如下：

1）加载驱动；

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。

**3、Statement,PreparedStatement,CallableStatment的区别。**

区别有以下几点：

1） Statement 是PreparedStatement 和CallableStatement 的父类；

2）Statement 是直接发送 Sql 语句到数据库，事先没有进行预编译。PreparedStatement 会将sql 进行预编译，当sql 语句要重复执行时，数据库会调用以前预编译好的sql 语句，所以PreparedStatement 在性能方面会更好；

3）PreparedStatement 在执行sql 时，对传入的参数可以进行强制的类型转换。以保证数据格式与底 层的数据库格式一致。

4）CallableStatement 适用与存储过程的查询表达语句

**1、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。

**2、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：继承Thread类或者实现Runnable接口。 实现同步也有两种方法：一种是同步方法，另一种是同步代码块。 同步方法是在方法返回类型前面加上synchronized关键字 同步代码块是synchronized (这里写需要同步的对象){...}

**3、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；

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。