## **1.abstract class 和interface 有什么区别?**

抽象类可以有构造方法 接口不行

抽象类可以有普通成员变量 接口没有

抽象类可以有非抽象的方法 接口必须全部抽象

抽象类的访问类型都可以 接口只能是 public abstract

一个类可以实现多个接口 但只能继承一个抽象类

## **2.HashMap 和Hashtable 的区别?**

HashMap 没有排序，允许一个null键和多个null值,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，HashMap是Java1.2引进的Map接口的实现；

Hashtable 的方法是Synchronize的，而HashMap不是

## **List和set的区别**

1、List,Set都是继承自Collection接口  
2、List特点：元素有放入顺序，元素可重复 ，Set特点：元素无放入顺序，元素不可重复（注意：元素虽然无放入顺序，但是元素在set中的位置是有该元素的HashCode决定的，其位置其实是固定的）   
3、List接口有三个实现类：LinkedList，ArrayList，Vector ，Set接口有两个实现类：HashSet(底层由HashMap实现)，LinkedHashSet

## **4.ArrayList和LinkedList的底层实现 以及各自的优缺点**

ArrayList是实现了基于动态数组的数据结构，LinkedList基于双向链表的数据结构。

ArrayList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(查询)数据快，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，查询数据慢。

## **5.请描述在多线程中修饰符volatile的含义和作用**

volatile是一个类型[修饰符](http://baike.so.com/doc/2148389-2273126.html" \t "http://baike.so.com/doc/_blank)(type specifier)。它是被设计 用来修饰被不同线程访问和修改的变量。如果没有volatile，就会导致这样的结果:要么无法编写多线程程序，要么[编译器](http://baike.so.com/doc/6133192-6346352.html" \t "http://baike.so.com/doc/_blank)失去大量优化的机会。

## **请描述数据库中事物的特点**

(Atomic)原子性

(Consistent)一致性

(Insulation)隔离性

(Duration)持久性

A: atomicity(原子性) - 一个事务（transaction）中的所有操作，要么全部完成，要么全部不完成，不会结束在中间某个环节。事务在执行过程中发生错误，会被回滚（Rollback）到事务开始前的状态，就像这个事务从来没有执行过一样。

C: consistency(一致性) - 在事务开始之前和事务结束以后，数据库的完整性没有被破坏。即从一个一致性状态转换到另一个一致性状态。

I: isolation(隔离性) - 当两个或者多个事务并发访问（此处访问指查询和修改的操作）数据库的同一数据时所表现出的相互关系。通常来说，一个事务所做的修改在最终提交以前，对其他事务是不可见的。

D: durability(持久性) - 在事务完成以后，该事务对数据库所作的更改便持久地保存在数据库之中。

## **请说出sql语句中 left join ,inner join 和right join的区别**

INNER JOIN 关键字在表中存在至少一个匹配时返回行. 如果没有匹配，就不会列出这些行.

LEFT JOIN 关键字会从左表 (table\_name1) 那里返回所有的行,即使在右表 (table\_name2) 中没有匹配的行.

RIGHT JOIN 关键字会右表 (table\_name2) 那里返回所有的行,即使在左表 (table\_name1) 中没有匹配的行.

## **请描述jsp中application session request cookie 生命周期**

Request 的范围是指在一JSP 网页发出请求到另一个JSP 网页之间，随 这个属性就失效。设定Request 的范围时可利用request 对象中的setAttribute( )和getAttribute( )  
  
Session 的作用范围为一段用户持续和服务器所连接的时间(一次会话范围内)，但与服务 器断线 ，这个属性就无效。只要将数据存入session对象，数据的范围就为Session  
  
Application 的作用范围在服务器一开始执行服务，到服务器关闭为止Application 的范围最、停留的时间也最久，所以使用时要特别注意不然可能会造成服务器负载越来越重的情况。只要将数据存入application对象，数据的范围范围 (Scope) 就为Application

cookie是在客户端保存的方案.而session是在服务端保存的方案.  
如果cookie不设定时间的话就表视它的生命周期为浏览器会话的期间,只要关闭IE,cookie就消失了  
这种cookie被称为会话cookie.其一般不保存在硬盘上.而是保存在内存中.  
  
如果设置了过期时间.那么浏览器会把cookie保存到硬盘中,再次打IE时会依然有效.直到它的有效期超时;  
注:存储在硬盘中的cookie可以在不同IE间共享;

## **9.object类中的方法**

Equals ToString Finalize

## **10.请描述hibernate中一级缓存和二级缓存的区别**

1.一级缓存是针对session级别的，当这个session关闭后这个缓存就不存在了。  
 2.二级缓存是SessionFactory级别的，二级缓存我们通常使用其他的一些开源组件，比如hibernate经常使用的就是ECache，这个缓存在整个应用服务器中都会有效的。  
 区别：两者的作用范围不同。

## **11.请说出Hibernate中get()和load（）方法的区别**

1.get()采用立即加载方式,而load()采用[延迟加载](http://www.so.com/s?q=%E5%BB%B6%E8%BF%9F%E5%8A%A0%E8%BD%BD&ie=utf-8&src=wenda_link" \t "http://wenda.so.com/q/_blank);

2.get()方法执行的时候,会立即向数据库发出查询语句, 而load()方法返回的是一个代理(此代理中只有一个id属性),只有等真正使用该对象属性的时候,才会发出[sql语句](http://www.so.com/s?q=sql%E8%AF%AD%E5%8F%A5&ie=utf-8&src=wenda_link" \t "http://wenda.so.com/q/_blank)

3.如果数据库中没有对应的记录,get()方法返回的是null.而load()方法出现异常ObjectNotFoundException

## **12.请说出SpringMvc中常用的注解以及其含义**

1.@RequestMapping 可以定义不同的处理器映射规则

URL 映射路径 Method 请求方法限定

@ResponseBody 用于读取HTTP请求的内容 然后将读取的内容转换为java对象并绑定到 controller方法的参数上，加上@ResponseBody注解，就不会走视图解析器，不会返回页面，目前返回json数据

@PathVariable是获取url上数据的

@RequestParam获取请求参数的(包括post表单提交的参数)

@Controller

## **13.请用java代码写出单例模式（高频）**

懒汉式 常用模式

public class SingletonClass{

    private static SingletonClass instance=null;

    public static　synchronized　SingletonClass getInstance()

    {

        if(instance==null)

        {

               instance=new SingletonClass();

        }

        return instance;

    }

    private SingletonClass(){

    }

}

饿汉式

//对第一行static的一些解释

// java允许我们在一个类里面定义静态类。比如内部类（nested class）。

//把nested class封闭起来的类叫外部类。

//在java中，我们不能用static修饰顶级类（top level class）。

//只有内部类可以为static。

public class Singleton{

    //在自己内部定义自己的一个实例，只供内部调用

    private static final Singleton instance = new Singleton();

    private Singleton(){

        //do something

    }

    //这里提供了一个供外部访问本class的静态方法，可以直接访问

    public static Singleton getInstance(){

        return instance;

    }

}

## **14.Spring如何处理线程并发问题（面试高频）**

ThreadLocal可以在无需同步的情况下就化解线程安全的难题

当使用ThreadLocal维护变量时，ThreadLocal为每个使用该变量的线程提供独立的变量副本，所以每一个线程都可以独立地改变自己的副本，而不会影响其它线程所对应的副本。

## **15.什么是存储过程 它有什么优点？（高频）**

存储过程（Stored Procedure）是在大型数据库系统中，一组为了完成特定功能的SQL 语句集，经编译后存储在数据库中，用户通过指定存储过程的名字并给出参数（如果该存储过程带有参数）来执行它。存储过程是数据库中的一个重要对象，任何一个设计良好的数据库应用程序都应该用到存储过程。

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，以后需要这个工作的时候，只需要调用execute就可以自动完成命令。

## **16.存储过程和存储函数的区别？**

本质上没有区别，只是函数只能返回一个变量的限制，而存储过程可以返回多个。函数是可以嵌套在sql中使用的，可以在select中调用，而存储过程不行。存储过程可以调用存储函数。

## **17.什么是触发器 什么是游标 什么是视图**

数据库触发器：是一个与表相关联的、存储的PL/SQL程序。每当一个特定的数据操作语句(Insert,update,delete)在指定的表上发出时，Oracle自动地执行触发器中定义的语句序列。

游标：在写java程序中有结果集的概念，那么在pl/sql中也会用到多条记录，这时候我们就要用到游标，游标可以存储查询返回的多条数据。游标可以理解为是PL/SQL中的结果集，我们通过游标可以提取结果集中的每行记录。

视图：视图就是封装了一条复杂查询的语句。

## **18.简述map的几种迭代方式**

方式1：根据键找值的方式(keySet())

        //a, 获取到Map集合中所有的键

        //b, 遍历键的集合，获取到每一个键

        //c, 通过键，找到对应的值

//获取到Map集合中所有的键

        Set<String> keys = map.keySet();

        //遍历键的集合，获取到每一个键

        for (String key : keys) {

            //通过键，找到对应的值

            Student s = map.get(key);

            System.out.println( key + "..." + s.getName() + "..." + s.getAge() );

        }

方式2：根据键值对对象找键和值的方式(entrySet())

        //a, 获取Map集合中所有的键值对元素,返回对应的集合

        //b, 遍历键值对元素集合，获取到每一个键值对元素对象

        //c, 通过键值对元素对象，获取对应的键，和对应的值

//获取Map集合中所有的键值对元素,返回对应的集合

        Set< Map.Entry<String, Student>> entrySet = map.entrySet();

        //遍历键值对元素集合，获取到每一个键值对元素对象

        for (Map.Entry<String, Student> entry : entrySet) {

            //通过键值对元素对象，获取对应的键，和对应的值

            //找键

            String key = entry.getKey();

            //找值

            Student s = entry.getValue();

            //打印

            System.out.println( key+"..."+s.getName()+"..."+s.getAge() );

        }  
方式3：for循环

## **19.Spring怎么进行事物管理**

①编程式事务管理：这意味你通过编程的方式管理事务，给你带来极大的灵活性，但是难维护。

②声明式事务管理：这意味着你可以将业务代码和事务管理分离，你只需用注解和XML配置来管理事务

声明式事务管理XML方式

声明式事务的注解方式

## **简述EJB和javabean 的区别**

答：Java Bean 是可复用的组件，对Java Bean 并没有严格的规范，理论上讲，任何一个Java 类都可以是一个Bean。但通常情况下，由于Java Bean 是被容器所创建（如Tomcat）的，所以Java Bean 应具有一个无参的构造器，另外，通常Java Bean 还要实现Serializable 接口用于实现Bean 的持久性。Java Bean实际上相当于微软COM 模型中的本地进程内COM 组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于DCOM，即分布式组件。它是基于Java 的远程方法调用（RMI）技术的，所以EJB 可以被远程访问（跨进程、跨计算机）。但EJB必须被布署在诸如Webspere、WebLogic 这样的容器中，EJB 客户从不直接访问真正的EJB 组件，而是通过其容器访问。EJB 容器是EJB 组件的代理，EJB 组件由容器所创建和管理。客户通过容器来访问真正的EJB 组件。

## **21.forward 和redirect 的区别?**

1.从地址栏显示来说

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，所以session,request 参数都可以获取，并且从浏览器的地址栏中可以看到跳转后的链接地址。

2.从数据共享来说

Forward转发页面和转发到的页面可以共享request里面的数据，redirect不能共享数据

3.从运用地方来说

 Forward一般用于用户登录的时候，根据角色转发到相应的模块

Redirect一般用于用户注销登录时返回的主页面和跳转到其他的网站等

## **22.get 和post 的区别？**

1）Get 是用来从服务器上获得数据，而Post 是用来向服务器上传递数据；

2）Get 将表单中数据按照variable=value 的形式，添加到action 所指向的URL 后面，并且两者使用“?”连接，而各个变量之间使用“&”连接；Post 是将表单中的数据放在form 的数据体中，按照变量和值相对应的方式，传递到action所指向URL；

3）Get 是不安全的，因为在传输过程，数据被放在请求的URL 中；Post 的所有操作对用户来说都是不可见的；

4）Get 传输的数据量小，这主要是因为受URL 长度限制；而Post 可以传输大量的数据，所以在上传文件只能使用Post；

5）Get 限制Form 表单的数据集必须为ASCII 字符，而Post 支持整个ISO10646字符集；

6）Get 是Form 的默认方法。

## **23.linux的软链式命令?**

ln -s 源文件 目标文件

## **24.来说几个linux命令?**

切换到根目录: cd /

列出文件列表:ll

创建目录和移除目录：mkdir rmdir

打包或解压：tar

Mv cp

## **25.sleep() 和wait() 有什么区别?**

sleep()方法（休眠）是线程类（Thread）的静态方法，调用此方法会让当前线程暂停执行指定的时间，将执行机会（CPU）让给其他线程，但是对象的锁依然保持，因此休眠时间结束后会自动恢复.

wait()是Object类的方法，调用对象的wait()方法导致当前线程放弃对象的锁（线程暂停执行），进入对象的等待池（wait pool），只有调用对象的notify()方法（或notifyAll()方法）时才能唤醒等待池中的线程进入等锁池（lock pool），如果线程重新获得对象的锁就可以进入就绪状态。

## **26.你们用过的线程安全的类?**

Vector

StringBuffer

Hashtable

## **多线程实现以及线程同步方法?**

多线程实现方式主要有2种：

继承Thread类、实现Runnable接口

线程同步方法：

wait()：使一个线程处于等待（阻塞）状态，并且释放所持有的对象的锁；   
 sleep()：使一个正在运行的线程处于睡眠状态，是一个静态方法;  
 notify()：唤醒一个处于等待状态的线程，当然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，而且与优先级无关；   
 notityAll()：唤醒所有处于等待状态的线程，该方法并不是将对象的锁给所有线程，而是让它们竞争，只有获得锁的线程才能进入就绪状态；

## **28.简述struts2的工作流程**

客户端浏览器发出HTTP请求。

根据web.xml配置，该请求被FilterDispatcher接收。

根据struts.xml配置，找到需要调用的Action类和方法， 并通过IoC方式，将值注入给Aciton。

4、Action调用业务逻辑组件处理业务逻辑，这一步包含表单验证。

Action执行完毕，根据struts.xml中的配置找到对应的返回结果result，并跳转到相应页面。

返回HTTP响应到客户端浏览器

## **29.JSP 和Servlet 有哪些相同点和不同点，他们之间的联系是什么？**

答：JSP是Servlet技术的扩展，本质上是Servlet的简易方式，更强调应用的外表表达。JSP编译后是"类servlet"。Servlet和JSP最主要的不同点在于，Servlet的应用逻辑是在Java文件中，并且完全从表示层中的HTML里分离开来。而JSP的情况是Java和HTML可以组合成一个扩展名为.jsp的文件。JSP侧重于视图，Servlet主要用于控制逻辑。

## **SpringMVC流程.**

1.用户发送请求到前端控制器

2.前端控制器收到请求调用处理器映射器

3.处理器映射器根据请求URL找到具体的处理器，生出处理器对象以及处理器拦截器，返回给 前端控制器

4.前端控制器通过处理器适配器调用处理器

5.执行处理器Controller

6.处理器执行完成后返回ModelAndView

7.处理器适配器将执行结果ModelAndView返回给前端控制器

8.前端控制器将ModelAndView传给视图解析器

9.视图解析器解析后返回具体view

10.前端控制器对view进行渲染视图(即将模型数据填充到视图中)

11.前端控制器响应用户

## **springMvc和struts2的区别?**

1.springMVC入口是一个servlet即前端控制器，而struts2的入口是一个filter过滤器

2.springMVC是基于方法开发(一个URL对应一个方法)，请求参数传递到方法的形参，可以设计为单例或多例(建议单例)，struts2是基于类开发，传递参数是通过类的属性，只能设计为多例。

3.struts2采用值栈存储请求和响应的数据，通过OGNL存取数据，springMVC通过参数解析器是将request请求内容解析，并给方法形参赋值，将数据和视图封装成ModelAndView对象，最后又将ModelAndView中的模型数据通过request域传输到页面。JSP视图解析器默认使用jstl

## **你是怎么理解spring?**

Spring 框架是一个热门的流行框架，由 7 个定义良好的模块组成。

组成 Spring 框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。每个模块的功能如下：

1、核心容器：核心容器提供 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程序代码分开。

2、Spring 上下文：Spring 上下文是一个配置文件，向 Spring 框架提供上下文信息。Spring 上下文包括企业服务，例如 JNDI、EJB、电子邮件、国际化、校验和调度功能。

3、Spring AOP：通过配置管理特性，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持 AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。通过使用 Spring AOP，不用依赖 EJB 组件，就可以将声明性事务管理集成到应用程序中。

4、Spring DAO：JDBC DAO 抽象层提供了有意义的异常层次结构，可用该结构来管理异常处理和不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写 的异常代码数量（例如打开和关闭连接）。Spring DAO 的面向 JDBC 的异常遵从通用的 DAO 异常层次结构。

5、 Spring ORM：Spring 框架插入了若干个 ORM 框架，从而提供了 ORM 的对象关系工具，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。

6、Spring Web 模块：Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。所以，Spring 框架支持与 Jakarta Struts 的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。

7、Spring MVC 框架：MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。通过策略接口，MVC 框架变成为高度可配置的，MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI。

IOC ,DI和 AOP

控制反转的基本概念是：调用者不再创建被调用者的实例，由spring容器创建，所以称为控制反转

依赖注入的基本概念是：[依赖注入](https://www.baidu.com/s?wd=%E4%BE%9D%E8%B5%96%E6%B3%A8%E5%85%A5&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkrHIWPAn4PWcYnW-BPvms0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1f1PW6srHfd" \t "https://zhidao.baidu.com/question/_blank)是[spring框架](https://www.baidu.com/s?wd=spring%E6%A1%86%E6%9E%B6&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkrHIWPAn4PWcYnW-BPvms0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1f1PW6srHfd" \t "https://zhidao.baidu.com/question/_blank)中的解耦的一种策略,主要有set方式（提供set和get方法)和constractor(构造方法)方式，它使得类与类之间以配置文件的形式组织在一起，而不是硬编码的方式，

面向切面变成概念是：面向切面编程，主要意图是将日志记录，安全控制，事物处理，异常处理等代码从业务逻辑代码中分出来，我们希望可以将他们独立到非逻辑业务的方法中，进而改变这些行为的时候不影响业务逻辑的代码。

## **33.讲一下jdbc流程?**

1）注册驱动；

2）建立连接；

3）创建Statement；

4）执行sql 语句；

5）处理结果集`

6）关闭连接。

## **34.数据库连接池的工作机制?**

1.J2EE服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其标记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量由配置参数决定。

2.当使用的池连接调用完成后，池驱动程序将此连接表记为空闲，其他调用就可以使用这个连接。

## **hibernate和mybatis区别?**

1.Mybatis不完全是一个ORM框架，因为它需要程序员自己编写sql语句，而hibernate完全是一个ORM框架。

2.Mybatis学习门槛低，简单易学，而hibernate学习门槛高，不易精通。

3.Mybatis程序员直接编写原生态sql，可以严格控制sql执行性能，进行更为细致的sql优化，灵活度高，非常适合对关系型数据模型要求不高的软件开发，例如互联网软件，企业运营类软件，而hibernate不需要编写sql语句，灵活性不高，

对于关系模型要求高的软件，用它开发可以节省很多代码，提高开发效率。

4.mybatis本身提供的缓存机制不佳，而hibernate有更好的一级缓存和二级缓存机制，可以使用第三方缓存，如ECache

## **spring的运行流程?**

1.用户发送请求到前端控制器

2.前端控制器收到请求调用处理器

3.处理器执行完成后返回处理结果给前端控制器

4.前端控制器将处理结果转发给jsp

5.Jsp页面返回html页面给前端控制器、

6.前端控制器响应给用户

## **37.Collection 和Collections 的区别？**

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。

## **38.JSP 和Servlet 中的请求转发分别如何实现？**

JSP中的请求转发可利用forward动作实现<jsp:forward />；

Serlvet转发的方式getServletContext().getRequestDispatcher(path).forward(req,res)。

## **39.说出ArrayList,Vector, LinkedList 的存储性能和特性？**

ArrayList和Vector都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢

Vector由于使用synchronized(同步代码块)方法（线程安全），通常性能上较ArrayList差

LinkedList使用双向链表实现存储，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。

## **40.Arraylist 与Vector 区别？**

答：就ArrayList 与Vector 主要从二方面来说：

1）同步性：Vector 是线程安全的（同步），而ArrayList 是线程序不安全的；

2）数据增长：当需要增长时,Vector 默认增长一倍，而ArrayList 却是一半。

## **41.wait,join,sleep,notify区别?**

wait()：使一个线程处于等待（阻塞）状态，并且释放所持有的对象的锁；   
- sleep()：使一个正在运行的线程处于睡眠状态，是一个静态方法   
- notify()：唤醒一个处于等待状态的线程，当然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，而且与优先级无关；   
- notityAll()：唤醒所有处于等待状态的线程，该方法并不是将对象的锁给所有线程，而是让它们竞争，只有获得锁的线程才能进入就绪状态；

## **42.构造器Constructor 是否可被override?**

答：构造器Constructor 不能被继承，因此不能重写Overriding，但可以被重载Overloading。

## **43.A继承Base,A引用C.哪种耦合度更高?**

继承的耦合度更高，继承属于强耦合

## **44、XML 文档定义有几种形式？它们之间有何本质区别？解析(读取)XML 文档有哪几种方式？**

1）两种形式：dtd 以及schema；

2）本质区别：schema 本身是xml 的，可以被XML 解析器解析(这也是从DTD上发展schema 的根本目的)；

3）解析方式：有DOM,SAX,STAX 等：

DOM:处理大型文件时其性能下降的非常厉害。这个问题是由DOM 的树结构所造成的，这种结构占用的内存较多，而且DOM 必须在解析文件之前把整个文档装入内存,适合对XML 的随机访问；SAX:不同于DOM,SAX 是事件驱动型的XML 解析方式。它顺序读取XML 文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理XML 文件，适合对XML 的顺序访问；

STAX:Streaming API for XML (StAX)。

## **45.JSP 中动态INCLUDE 与静态INCLUDE 的区别？**

动态INCLUDE 用jsp:include 它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数；

静态INCLUDE 用include 伪码实现,它不会检查所含文件的变化，适用于包含静态页面

动态包含,包含其他页面的运行的结果.最终翻译成多个Servlet解释执行的.

静态包含相对于代码的copy,最终被翻译成一个Servlet解释执行的.

## **46.hashmap,treemap底层结构?**

HashMap:底层是哈希表数据结构。线程不同步。适用于在Map中插入、删除和定位元素。

TreeMap:底层是二叉树(红黑树)数据结构，线程不同步，可用于给Map集合中的键进行排序。适用于按自然顺序或自定义顺序遍历键(key)。

## **47.加密算法用到了什么?**

MD5 、 [SHA](https://www.baidu.com/s?wd=SHA&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dhrHnzPymsujf4P17BPyRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbYPWn1PW04Pj01nHfLPj6vr0" \t "http://zhidao.baidu.com/_blank) 、 [HMAC](https://www.baidu.com/s?wd=HMAC&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dhrHnzPymsujf4P17BPyRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbYPWn1PW04Pj01nHfLPj6vr0" \t "http://zhidao.baidu.com/_blank)

## **异常分类?**

系统中异常包括两类：预期异常和运行时异常RuntimeException,前者通过捕获异常从而获取异常信息，后者主要通过规范代码开发，测试通过手段减少运行时异常的发生。

## **Spring创建对象方式?**

一、通过构造函数创建对象。

二、通过静态方法创建对象

三、通过工厂方法创建对象。

## **50.线程的生命周期**

线程的生命周期：

新建：创建线程对象

就绪：有执行资格 没有执行权

运行：有执行资格 有执行权

阻塞：由于一些操作让线程处于该状态 没有执行资格 没有执行权

而另一些操作却可以把它给激活 激活后处于就绪状态

死亡：线程对象变成垃圾 等待被回收

## **51.java 中常用的设计模式？说明工厂模式？**

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。

## **52.Overload 和Override 的区别。Overloade的方法是否可以改变返回值的类型?**

答：方法的重载和重写都是实现多态的方式，区别在于前者实现的是编译时的多态性，而后者实现的是运行时的多态性。重载发生在一个类中，同名的方法如果有不 同的参数列表（参数类型不同、参数个数不同或者二者都不同）则视为重载；重写发生在子类与父类之间，重写要求子类被重写方法与父类被重写方法有相同的返回 型，比父类被重写方法更好访问，不能比父类被重写方法声明更多的异常（里氏代换原则）。重载对返回类型没有特殊的要求,可以改变返回值类型。

## **53.char 型变量中能不能存贮一个中文汉字?为什么?**

答：能够定义成为一个中文的，因为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。

## **54.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，是恢复很困难的一种严重问题；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

## **55.final, finally, finalize 的区别?**

final：修饰符（关键字）有三种用法：如果一个类被声明为final，它就不能被继承，将变量声明为final，可以保证它们在使用中不被改变，被声明为final的变量必须在声明时给定初值，而在以后的引用中只能读取不可修改。被声明为final的方法也同样只能使用，不能在子类中被重写。   
 finally：通常放在try…catch…的后面构造总是执行代码块，这就意味着程序无论正常执行还是发生异常，这里的代码只要JVM不关闭都能执行。   
 finalize：Object类中定义的方法，Java中允许使用finalize()方法在垃圾收集器将对象从内存中清除出去之前做必要的清理工作。

## **56.Bean Factory 与 ApplicationContext 的区别?**

Bean Factory 只提供基础的 DI(Dependes inject) 支持

Application contexts 提供了处理 text messages 的功能,读取 file resources 的功能, 发布事件给已经注册为 listener 的 bean.

## **57.String, StringBuffer StringBuilder 的区别。**

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。

## **58.悲观锁和乐观锁有何区别？**

悲观锁:数据开始更改时,就锁住数据,直到修改完成后才释放

访问数据库时间长,效率低

乐观锁:数据更改完毕后才锁住数据,修改完成后释放

访问数据库时间短,效率高

## **59.Git和svn各自的优点**

1.Git的优点：

Git分布式优于svn集中式

Git分布式数据仓库速度快，效率更高和多副本

Git可离线工作

GIT任意两个开发者之间很容易解决冲突

缺点：

学习周期长，

不符合常规思维

2.Svn的优点：

Svn的理念符合常规思维

Svn支持空目录

Svn有更好的windows平台支持

Svn支持二进制文件 更容易处理大文件

## **60.为什么要使用Ajax技术 ，有什么好处？**

使用Ajax的最大优点，就是能在不更新整个页面的前提下维护数据，减少网页重新载入的次数。这使得Web应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变过的信息,提高客户的体验性。

## **61.如何解决ajax跨域问题**

什么是跨域问题？

Js使用AJAX请求进行跨域请求，无法返回数据（json，xml等）

为什么会有跨域问题？

浏览器基于安全考虑，不允许Ajax请求跨域调用数据（json，xml等）

什么是跨域？

跨域是跨域名或端口号进行调用

Jsonp核心原理：js使用ajax无法跨域调用数据（json，xml等），但是跨域跨域调用js数据，我们跨域把json伪装成js，再进行调用

使用接收到的请求参数callback作为方法名对json数据进行包裹

## **62.Hibernate的优化**

优先使用缓存技术。检索策略：一般情况下都用默认的

1: 是设定JDBC的Statement读取数据的时候每次从数据库中取出的记录条数，一般设置为30、50。如果继续增大，超出100，性能提升不明显，反而会消耗内存。Oracle数据库的JDBC驱动默认的Fetch

Fetch Size设的越大，读数据库的次数越少，速度越快；FetchSize越小，读数据库的次数越多，速度越慢。

2、如果是超大的系统，建议生成htm文件。加快页面提升速度。

3、对代码进行重构，减少对数据库的操作，尽量避免在数据库查询时使用in操作，以及避免递归查询操作，代码质量、系统设计的合理性决定系统性能的高低。

4、 对大数据量查询时，慎用list()或者iterator()返回查询结果，

5、在一对多、多对一的关系中，使用延迟加载机制，会使不少的对象在使用时方会初始化，这样可使得节省内存空间以及减少数据库的负荷，而且若PO中的集合没有被使用时，就可减少互数据库的交互从而减少处理时间。

6、对于大数据量新增、修改、删除操作或者是对大数据量的查询，与数据库的交互次数是决定处理时间的最重要因素，减少交互的次数是提升效率的最好途径，所以在开发过程中，请将show\_sql设置为true，深入了解Hibernate的处理过程，尝试不同的方式，可以使得效率提升。尽可能对每个页面的显示，对数据库的操作减少到100----150条以内。越少越好。

## **63.spring依赖注入的几种方式和三大核心思想是什么？**

使用构造器注入

使用setter方法注入

使用Filed注入（用于注解方式）

三大核心思想：AOP IOC DI

## **struts2中 # % $的作用**

#号的使用

获取context的数据

<s:property value=”#request.name”/>

用于构建一个map集合，使用struts的UI标签的时候

<s:iterator value=”#{‘aaa’:’111’,’bbb’:’222’}” var=”entry”/>

%号的使用

强制解析OGNL表达式<s:textfield name=”name” value=”%{#request.name}”/>

强制不解析OGNL表达式<s:property value=”%{#request.name}”/>

$号的使用

在配置文件中使用OGNL表达式

在struts的配置文件中使用.xml文件或者是属性文件。

## **65.&和&&的区别？**

答：&运算符有两种用法：(1)按位与；(2)逻辑与。

&&运算符是短路与运算。

逻辑与跟短路与的差别是非常巨大的，虽然二者都 要求运算符左右两端的布尔值都是true整个表达式的值才是true。&&之所以称为短路运算是因为，如果&&左边的表 达式的值是false，右边的表达式会被直接短路掉，不会进行运算。很多时候我们可能都需要用&&而不是&，例如在验证用户登录 时判定用户名不是null而且不是空字符串，应当写为：username != null &&!username.equals("")，二者的顺序不能交换，更不能用&运算符，因为第一个条件如果不成立，根本不能进 行字符串的equals比较，否则会产生NullPointerException异常。注意：逻辑或运算符（|）和短路或运算符（||）的差别也是如 此。

## **66.cookie与session的区别与联系**

1.Cookie是一种客户端的技术，将数据保存到客户端浏览器的内存中,Session是一种服务器端技术，将数据保存到服务器端

2.Cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗，考虑到安全应该用session

3.Session会在一定时间内保存在服务器上，当访问增多，会比较占用服务器的性能，考虑到减轻服务器性能方面，应当使用cookie

所以建议：将登陆信息等重要信息存放在session，其他信息如需要保留，可以放在cookie中

## **67.TCP与UDP的区别**

TCP协议是面向连接的通信协议，即在传输数据前先在发送端和接收端建立逻辑连接，然后再传输数据。每次连接的创建都需要经过“三次握手”。

第一次握手，客户端向服务器端发出连接请求，等待服务器确认

第二次握手，服务器端向客户端回送一个响应，通知客户端收到了连接请求

第三次握手，客户端再次向服务器端发送确认信息，确认连接

UDP是面向非连接的协议，它不与对方建立连接，而是直接就把数据包发送过去！

UDP适用于一次只传送少量数据、对可靠性要求不高的应用环境。比如，我们经常使用“ping”命令来测试两台主机之间TCP/IP通信是否正常，其实“ping”命令的原理就是向对方主机发送UDP数据包，然后对方主机确认收到数据包，如果数据包是否到达的消息及时反馈回来，那么网络就是通的。UDP协议是面向非连接的协议，没有建立连接的过程。正因为UDP协议没有连接的过程，所以它的通信效果高；但也正因为如此，它的可靠性不如TCP协议高。QQ就使用UDP发消息，因此有时会出现收不到

## **68.jQuery优缺点 和工作中用到哪些？**

优点：

实现脚本与页面的分离

最少的代码做最多的事情

性能支持比较好

节约学习成本

缺点：

不能向后兼容

多个插件冲突

对动画和特效的支持差

工作中jQuery基本选择器和属性选择器用的比较多，

## **69.JSP 和Servlet 有哪些相同点和不同点，他们之间的联系是什么？**

JSP 是Servlet 技术的扩展，本质上是Servlet 的简易方式，更强调应用的外表表达。JSP 编译后是"类servlet"。Servlet 和JSP 最主要的不同点在于，Servlet 的应用逻辑是在Java 文件中，并且完全从表示层中的HTML 里分离开来。而JSP 的情况是Java 和HTML 可以组合成一个扩展名为.jsp 的文件。JSP 侧重于视图，Servlet 主要用于控制逻辑。

## **70.error 和exception 有什么区别?**

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。

## **71.请说明SQLsever中delete from tablea & truncate table tablea的区别**

1.Delete from table会删除表数据和表结构，truncate table 摧毁表数据和结构，并重新创建一张空表

2.Delete事物提交后才生效，truncate操作立即生效，不能回滚

## **72.在测试生命周期中，测试过程分为几个阶段**

[测试](https://www.baidu.com/s?wd=%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y1nvfYmW0smWTzPhD1uj9b0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWRdnHcLn104" \t "http://zhidao.baidu.com/_blank)过程一般有4个步骤：[单元测试](https://www.baidu.com/s?wd=%E5%8D%95%E5%85%83%E6%B5%8B%E8%AF%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y1nvfYmW0smWTzPhD1uj9b0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWRdnHcLn104" \t "http://zhidao.baidu.com/_blank)、[集成测试](https://www.baidu.com/s?wd=%E9%9B%86%E6%88%90%E6%B5%8B%E8%AF%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y1nvfYmW0smWTzPhD1uj9b0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWRdnHcLn104" \t "http://zhidao.baidu.com/_blank)、[确认测试](https://www.baidu.com/s?wd=%E7%A1%AE%E8%AE%A4%E6%B5%8B%E8%AF%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y1nvfYmW0smWTzPhD1uj9b0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWRdnHcLn104" \t "http://zhidao.baidu.com/_blank)、[系统测试](https://www.baidu.com/s?wd=%E7%B3%BB%E7%BB%9F%E6%B5%8B%E8%AF%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y1nvfYmW0smWTzPhD1uj9b0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWRdnHcLn104" \t "http://zhidao.baidu.com/_blank)。

## **73.int 和Integer 有什么区别?**

答：Java 提供两种不同的类型：引用类型和原始类型（或内置类型）；

int 是java 的原始数据类型，Integer 是java 为int 提供的封装类。

Int的默认值为0，Integer的默认值为null

## **74有没有length()这个方法? String 有没有length()这个方法？**

答：数组没有length()方法，有length 的属性。String 有length()方法。JavaScript中，获得字符串的长度是通过length属性得到的，这一点容易和Java混淆。

## **75.构造器Constructor 是否可被override?**

答：构造器Constructor 不能被继承，因此不能重写Overriding，但可以被重载Overloading。

## **76.String 和StringBuffer 的区别?**

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

## **77.描述一下JVM 加载class 文件的原理机制?**

答：JVM 中类的装载是由ClassLoader 和它的子类来实现的,Java ClassLoader是一个重要的Java 运行时系统组件。它负责在运行时查找和装入类文件的类。

## **78.char 型变量中能不能存贮一个中文汉字?为什么?**

答：能够定义成为一个中文的，因为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。

## **79.String s=new String(“xyz”);创建了几个String Object?**

答：两个对象，一个是静态区的"xyz"，一个是用new创建在堆上的对象。

## **80.数据类型之间的转换:**

1)如何将数值型字符转换为数字？

2)如何将数字转换为字符？

3)如何取小数点前两位并四舍五入? 【基础】

答：1)调用数值类型相应包装类中的方法parse\*\*\*(String)或valueOf(String)即可返回相应基本类型或包装类型数值；

2)将数字与空字符串相加即可获得其所对应的字符串;另外对于基本类型数字还可调用String 类中的valueOf(…)方法返回相应字符串,而对于包装类型数字则可调用其toString()方法获得相应字符串；

3)可用该数字构造一java.math.BigDecimal 对象,再利用其round()方法进行四舍五入到保留小数点后两位,再将其转换为字符串截取最后两位。

## **81.你在项目中用到了xml 技术的哪些方面?如何实现的?**

答:用到了数据存贮，信息配置两方面。在做数据交换平台时，将不能数据源的数据组装成XML 文件，然后将XML 文件压缩打包加密后通过网络传送给接收者，接收解密与解压缩后再同XML 文件中还原相关信息进行处理。在做软件配置时，利用XML 可以很方便的进行，软件的各种配置参数都存贮在XML 文件中。

## **82.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。

## **83.什么情况下调用doGet()和doPost()？**

答：Jsp页面中的form标签里的method属性为get时调用doGet()，为post时调用doPost()。

## **84.如何从form 表单中得取checkbox 的值**

答：可在页面把checkbox 的name 属性取同一个，value 属性取每个条目的id,后台用getParamter(“name”)能取到checkbox 的一组值。

## **？85.SQL优化,数据库优化以及据库表设计?**

一、mysql如何优化：

1、开启mysql数据库的缓存（内存）

2、分表，分库

3、建立索引（提高查询效率）

4、采用集群

5、主主互备，主从复制，读写分离

二、sql语句如何优化

1、建立索引：create index 索引名 on id名；

2、尽可能的使用 NOT NULL

3、使用 ENUM 而不是 VARCHAR

4、避免 SELECT \*

5、当知道只有一行数据的时候使用 LIMIT 1

6、当有一批处理的插入或更新时，我会用批量插入或批量更新，不会一条条记录的去更新!

7.慎用distinct关键字

8.不要在in关键字中使用子查询

9.不要在where条件中使用函数

10.使用LECCO SQL Expert人工智能自动SQL优化

## **86.对MVC的理解**

答：基于Java 的Web 应用系统采用MVC 架构模式，即model（模型）、view（视图）、control（控制）分离设计；这是目前WEB 应用服务系统的主流设计方向。

Model：即处理业务逻辑的模块，每一种处理一个模块；

View：负责页面显示，显示MODEL 处理结果给用户，主要实现数据到页面转换过程；

Control：负责每个请求的分发，把FORM 数据传递给MODEL 处理，把处理结果的数据传递给VIEW 显示。

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。

## **清空表结构**

truncate table

## **88.Solr和lucene区别**

Lucene只是有个工具包

Solr是服务器，底层是lucene，可以直接运行

## **89..dubbo的配置**

服务端

<!-- dubbo服务启动名称 -->

<dubbo:application name="taotao-manager-service" />

<!-- 使用multicast注册中心暴露服务地址 -->

<dubbo:registry protocol="zookeeper" address="192.168.37.161:2181" />

<!-- 用dubbo协议在20880端口暴露服务 -->

<dubbo:protocol name="dubbo" port="20880" />

<!-- 声明需要暴露的服务接口 -->

<dubbo:service interface="com.taotao.manager.service.TestService" ref="testServiceImpl" />

客户端

<!-- 配置dubbo服务启动名称 -->

<dubbo:application name="taotao-manager-web" />

<!-- 使用zookeeper注册中心 -->

<dubbo:registry protocol="zookeeper" address="192.168.37.161:2181" />

<!-- 声明要调用的服务-->

<dubbo:reference interface="com.taotao.manager.service.TestService" id="testService" timeout="1000000" />

## **90.ajax和jquery的写法**

 $.post(URL,{data},function(dt){});

## **91.Hibernate的几种查询方式？**

Sql HQL QBC QBE EJBQL

## **reaultType和resultMap区别**

resultType:

1.指定输出的结果类型，将sql查询结果的一行记录数据映射为resultType指定类型的对象，如果有多条数据，则分别进行映射，并把对象放到list中

2.可以将指定查询结果映射为pojo，但需要pojo的属性名和sql查询的列名一致才可映射成功

resultMap：

1.如果sql查询字段和pojo的属性名不一致，可以用resultMap将字段名和属性名做一个对应关系

2.可以将查询结果映射为复杂的pojo，比如在查询结果映射对象中包括pojo和list实现一对一查询和一堆多查询

## **请手写出一段冒泡排序的代码？**

public class BubbleSort

{

    public void sort(int[] a)

    {

        int temp = 0;

        for (int i = a.length - 1; i > 0; --i)

        {

            for (int j = 0; j < i; ++j)

            {

                if (a[j + 1] < a[j])

                {

                    temp = a[j];

                    a[j] = a[j + 1];

                    a[j + 1] = temp;

                }

            }

        }

    }

}

## **java线程池?**

使用线程池的好处:   
  
1.减少在创建和销毁线程上所花的时间以及系统资源的开销   
2.如不使用线程池，有可能造成系统创建大量线程而导致消耗完系统内存

## **95.Soa和普通架构区别**

soa(Serice-oriented architecture,[面向服务架构](https://www.baidu.com/s?wd=%E9%9D%A2%E5%90%91%E6%9C%8D%E5%8A%A1%E6%9E%B6%E6%9E%84&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkuW-bn1NhmHwBPh7hPWR30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWc3PW01rjms" \t "https://zhidao.baidu.com/question/_blank)）是在计算环境下设计、开发、应用、管理分散的逻辑（服务）单元的一种规范。SOA的一个中心思想就是使得企业应用摆脱面向技术的解决方案的束缚，轻松应对企业商业服务变化、发展的需要。

## **96.什么是依赖注入**

[依赖注入](https://www.baidu.com/s?wd=%E4%BE%9D%E8%B5%96%E6%B3%A8%E5%85%A5&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkrHIWPAn4PWcYnW-BPvms0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1f1PW6srHfd" \t "https://zhidao.baidu.com/question/_blank)是[spring框架](https://www.baidu.com/s?wd=spring%E6%A1%86%E6%9E%B6&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkrHIWPAn4PWcYnW-BPvms0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1f1PW6srHfd" \t "https://zhidao.baidu.com/question/_blank)中的解耦的一种策略,称为DI或IOC([控制反转](https://www.baidu.com/s?wd=%E6%8E%A7%E5%88%B6%E5%8F%8D%E8%BD%AC&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkrHIWPAn4PWcYnW-BPvms0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1f1PW6srHfd" \t "https://zhidao.baidu.com/question/_blank)),主要有set方式（提供set和get方法)和constractor(构造方法)方式，它使得类与类之间以配置文件的形式组织在一起，而不是硬编码的方式，

## **97.描述Spring中的编程式事物处理及声明式事物处理**

编程式事物就是自己写Try catch语句调用commit/rollback等函数来实现那段业务数据改变的事务性

声明式业务是通过框架和事物定义配置给自己的业务代码添加事务性，比如Spring里面的AOP

前者需要自己写大量重复的事物控制代码 后者通过设置可以一次性给所有的业务方法添加上事物特性

## **98.描述JSP MVC模式下重复提交问题产生的原因和解决办法**

生成一个令牌保存在用户session中，在form中加一个hidden域，显示该令牌的值，form提交后重新生成一个新的令牌，将用户提交的令牌和session中的令牌比较，如相同则是重复提交,有效地防止了重复提交的发生。

## **99.数据库的硬优化和软优化?**

硬优化：硬件方面就可以优化硬盘IO，[内存分配](https://www.baidu.com/s?wd=%E5%86%85%E5%AD%98%E5%88%86%E9%85%8D&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4nyN-uhckrjTvuAfdPWb40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHmsPHm3rHbLPWRsnHb1PHbsPs" \t "https://zhidao.baidu.com/question/_blank)，就是安装时候调整的一系列[操作系统](https://www.baidu.com/s?wd=%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4nyN-uhckrjTvuAfdPWb40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHmsPHm3rHbLPWRsnHb1PHbsPs" \t "https://zhidao.baidu.com/question/_blank)级的内核参数，

软优化：数据库架构上的优化，如逻辑、数据结构等等，最后就是代码上的优化。

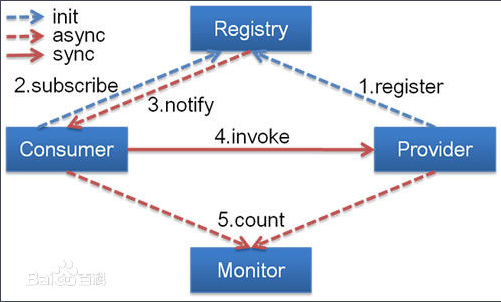
## **100.对dubbo内部结构了解不?**

主要核心部件

Remoting:网络通信框架，实现了sync-over-async和request-response消息机制

RPC: 一个[远程过程调用](http://baike.baidu.com/view/431455.htm" \t "http://baike.baidu.com/item/Dubbo/_blank)的抽象，支持[负载均衡](http://baike.baidu.com/view/51184.htm" \t "http://baike.baidu.com/item/Dubbo/_blank)、[容灾](http://baike.baidu.com/view/1088749.htm" \t "http://baike.baidu.com/item/Dubbo/_blank)和[集群](http://baike.baidu.com/view/302477.htm" \t "http://baike.baidu.com/item/Dubbo/_blank)功能

Registry: 服务目录框架用于服务的注册和服务事件发布和订阅



## 101.介绍redis

什么是redis

Redis是一个开源的使用C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value形式的Nosql数据库

为什么要添加缓存？(端口号6379)

网站性能优化第一定律：优先使用缓存优化性能

随着访问用户越来越多，并发越来越大，直接从Mysql数据库查询性能低，可以使用把数据进行缓存，从缓存读取速度更快

什么样的数据需要添加缓存？

访问频率高，更新频率低

可以使用redis增加缓存，因为redis的读写速度，访问速度快，服务层把需要缓存的数据放在redis中，先从redis中查询数据，如果redis没有数据吗，再去mysql查询

Redis把数据存放到内存中，内存的读写速度比硬盘的读写速度快

Redis是非关系型数据库，即Nosql数据库，存储的数据模型是key-value

数据类型：

String(字符串) Hash(哈希类型) List(链表) Set(集合) zSet(有序集合)

Redis是单线程，但是Redis的读写速度太快了

**搭建Redis集群**

为了保证可以进行投票(容错机制)，需要至少3个主节点

每个主节点都需要至少一个从节点，所以需要至少3个从节点(高可用)

一共需要6台redis服务器，可以使用6个redis实例

## **单点登录 购物车 商品添加（高频）**

**后台管理系统：**

**商品添加流程：**

商品添加功能主要有2个地方稍微复杂一点：

一个是选择商品目录，我们要把商品目录按树形结构展示在页面上，这时我们用到的是easyUI-tree。使用parentId作为查询条件进行查询，第一次请求是没有id参数，需要设置默认的parentId为0，查询一级目录，然后点击某个一级目录，会传递此父节点的id作为parentId对其子节点进行查询并显示在页面上，所以这样我们就能把商品目录全部展示出来。

第二个是图片上传功能，用到的服务器是FastDFS，它是一个分布式文件系统，存储空间可扩展，访问效率高，充分考虑了负载均衡，横向扩展等机制，并注重高可用，高性能等指标。

**单点登录系统：**

**为什么要使用单点登录流程？**

由于此项目使用了dubbo分布式架构，搭建可tomcat服务器集群，使得用户每次访问新的服务器，都需要登录，所以用户信息不放在session中，可以放在redis中，由redis实现session的功能，使得相互信任的系统之间的访问不用继续登录

**单点登录流程：**

1.用户登录商城系统首页，输入相关信息

2.商城系统跨域调用单点登录sso系统

3.单点登录系统调用单点登录系统服务，查询Mysql数据库查询用户信息

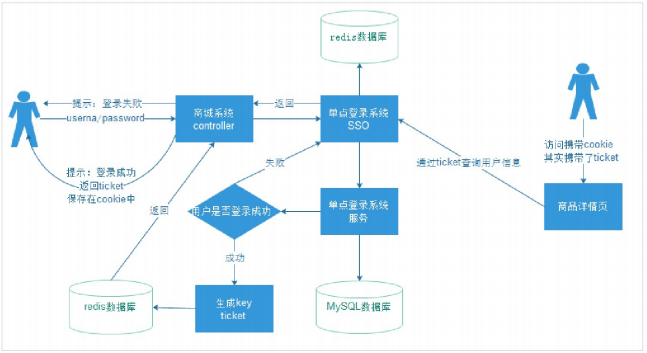
4.根据携带用户输入信息和查询到的数据判断用户是否登录成功

5.登陆成功的话会同时生成key\_ticket,将用户信息缓存到redis数据库

6.返回到商城系统首页，提示登录成功，并将ticket保存在cookie中。

7.当用户需要访问此商城别的系统时，访问会携带cookie中的ticket从redis数据库中查询用户信息而可以直接登录

需要开发2个接口，一个是检查用户数据是否可用，另一个是通过用户传递过来的ticket查询redis数据库中用户的信息



## **103.RESTful的了解**

RESTful web service是一种遵守REST风格的服务

RESTful风格的一大特色就是使用URI来标识资源

URL(统一资源定位符),URN(统一资源名称)是URI(统一资源标识符)的子类

GET 获取一个资源 POST创建一个新的资源

PUT 修改一个资源的状态 DELETE删除一个资源

http响应码

200 操作成功

201 对象创建成功

204 操作已执行成功，但是没有返回数据

303 重定向

404 资源服务找不到

500 系统内部错误

## **104.dubbo服务开发流程，运行流程？zookeeper注册中心的作用？端口是多少？**

使用dubbo进行远程调用实现服务交互，只需要在配置中指定使用的协议即可

Dubbo的客户端和服务端有三种连接方式，分别是广播，直连，和使用zookeeper注册中心

Zookeeper注册中心负责服务地址的注册与查找，相当于目录服务，服务提供者在启动时与注册中心交互，消费者不断的发起请求获取服务信息，注册中心不转发请求，压力较小、

端口号为2181

## **105.jvm执行流程?**

## **106.rpc?**

## **107.ActiveMQ使用场景?**

Mysql数据库和Solr索引库都进行商品数据的保存，会存在数据同步的问题，后台增加商品数据的时候，索引库并没有同步更新，这时候就需要用到MQ

后台商品进行商品修改，通知MQ，其他系统监听MQ是否有消息，如果没有消息，持续监听，如果有消息，处理相关业务逻辑

存在的问题：

1. MQ要是挂了，所有相关服务都挂了
2. MQ性能瓶颈的问题，在使用时，尽量减少消息内容大小

MQ消息的传递有两种类型：

1. queue点对点的，即一个生产者和一个消费者一一对应
2. topic发布、订阅模式，即一个生产者产生消息并进行发送后，可以由多个消费者进行接收

## **108.操作共享数据的安全性?**

## **109.没登录的时候添加购物车了 登陆以后怎么返回原来的页面**

## **110.WebService shiro？**

WebService就是实现两个系统之间远程调用的技术

不过一般不直接使用webService， 而是使用其框架apache CXF

服务端提供接口，客户端使用接口

权限控制一般要用到5张表，function role user user\_role role\_function

Apache shiro是一个强大而灵活的开源安全框架

**四大核心功能：**

认证

授权

会话管理

加密

**步骤**：

第一步：引入shiro框架相关的jar

第二步：在web.xml中配置spring框架提供的用于整合shiro框架的过滤器

第三步：在spring配置文件中配置shiro框架的过滤工厂bean

第四步：配置安全管理器

第五步：修改UserAction中的login方法，使用shiro提供的方式进行认证操作

第六步：自定义realm，并注入给安全管理器

**Shiro框架提供的权限控制方式**

URL拦截权限控制

方法注解权限控制

页面标签权限控制

代码级别权限控制

## **FreeMarker（商品详情页）的介绍**

商品详情页是消费者了解商品的主要途径，访问的频率非常高，所以需要对其进行优化，提高访问的速度。

优化方案：1.使用redis添加缓存，如果使用缓存的方式，需要注意数据同步的问题

2.使用页面静态化，把动态页面转化为静态页面html，

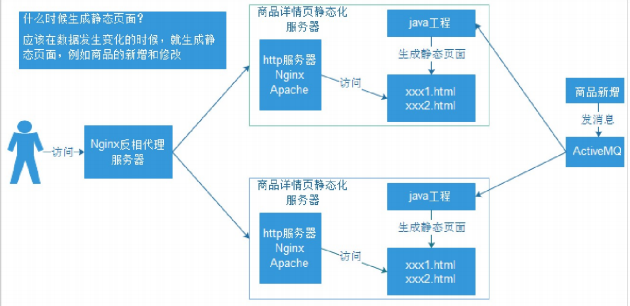
好处：a访问静态页面不需要经过程序处理，可以提高速度

B.稳定性高

C 从安全角度讲，静态页面不容易遭到黑客攻击

D 静态页面相对于动态页面更容易被搜索引擎收入(SEO)

静态访问的流程



Freemarker是一个模板引擎，可以基于模板生成文本(html)

企业一般用freemarker做生成静态页面功能

过程： 添加依赖

配置文件（Spring与freemarker整合，Spring与MQ整合，web.xml）

接收消息并消费消息(静态化java实现)

获取消息，消息不能为空

根据消息，生成静态页面

编写商品模板