### 京东面试问题：

1. Log4j用到什么级别？

答：分为OFF、FATAL、ERROR、WARN、INFO、DEBUG、TRACE、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。   
 程序会打印高于或等于所设置级别的日志，设置的日志等级越高，打印出来的日志就越少。   
 如果设置级别为INFO，则优先级高于等于INFO级别（如：INFO、WARN、ERROR）的日志信息将可以被输出,小于 该级别的如DEBUG将不会被输出。   
 项目上生产环境时候建议把debug的日志级别重新调为warn或者更高，避免产生大量日志。

1. 项目发布，怎么打印异常？

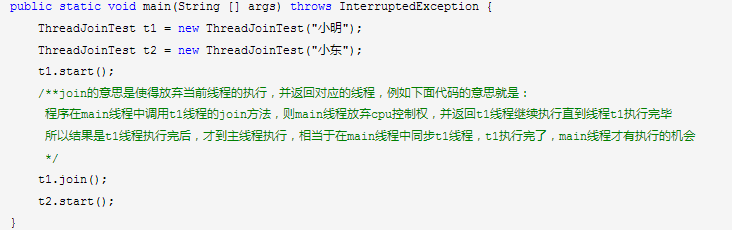
答：

1. Dubbo怎么实现负载均衡？

答：默认策略为：随机，按权重设置随机概率，当调用量大的时候分布均匀，按概率使用权重后也比较均匀。

策略二：轮循，按公约后的权重设置轮循比例，效率低的提供者设置权重低一点

1. Join的使用场景和方法？



答：在join方法中可以传递参数，表示把cpu的执行权交给该进程多少毫秒之后，继续返回执行当前线程也就是main主线程，之后的线程还是并行执行的。

1. 同步代码块的锁对象，以及wait方法的使用及锁对象？

答：同步代码块中的锁对象就是设置的锁对象，wait的使用场景就是多个线程公用一个锁对象，可以是代码块中的锁对象，也可以是公用的其他锁对象，wait方法是通过锁对象来调用的，调用后，该线程就失去当前锁，也就没有权利继续执行当前线程，这是共用该锁的其线程得到锁对象，可以执行线程，然后当锁对象调用lock.notify()时，会唤醒之前的线程，经典案例是生产者和消费者还有库存的关系，可以通过wait和notifyall来保证库存不会太多也不会不足。

和sleep的区别，sheep只是让当前线程在此等待，并没有释放线程锁，仍然占用该锁，所以共用一个锁的其他线程也还是得不到锁对象。Sleep时不占用cpu资源。

1. Dubbo的发布消费服务流程？可以实现不同领域的服务调用吗？怎么避免安全问题？

答：服务调用过程：

Step1   首先服务方启动时，向注册中心注册自己可提供的服务；

Step2   服务消费者启动时，向注册中心订阅自己所需的相关服务

Step3   注册中心在接收到订阅请求后，返回服务提供方的地址列表给服务消费方，如果有变更，注册中心将基于长连接推送变更数据给消费者。

Step4   服务消费方从提供的地址列表中基于负载均衡算法选择一台服务提供方进行调用，若调用失败，选择另一台

Step5   同时在调用的过程中，服务方和消费者都会在内存中累计调用次数和调用时间，定时将统计数据发送给监控中心。

1. 倒排索引法？

答：倒排索引法，可以解释为反向索引，正向索引是：文档--->单词，反向索引：单词--->文档；可以大大缩短查询时间。

1. Solr查询对象？
2. Hibernate的设计模式？
3. Servlet单例，在框架中Servlet的创建和使用？

答：servlet是单例，可以通过设置web.xml在服务器启动的时候，实例化servlet，也可以是在用户第一次请求的时候实例化，在服务器关闭的时候，销毁servlet实例。

1. Solr过滤流程和方法？
2. **WebService的soap以及三要素？**
3. Jsp的九大内置对象？



1. IO流打印到控制台？
2. 用户登录后，把什么用户信息存放到了redis中？
3. Memcache和redis的区别？

答：redis

1. Shiro怎么实现权限控制以及认证?关键的单词
2. 公司的编码要求？

答：为提高控制程序的阅读性与可理解性，现制定相关代码程序代码注释编写的编写规范。一般情况下，源程序有效注释量必须在20％以上，注释的原则是有助于对程序的阅读理解，在该加的地方都加了，注释不宜太多也不能太少，注释语言必须准确、易懂、简洁。

1基本注释（必须加）

(a) 类（接口）的注释

(b) 构造函数的注释

(c) 方法的注释

(d) 全局变量的注释

(e) 字段/属性的注释

2特殊必加注释（必须加）

(a) 典型算法必须有注释。

(b) 在代码不明晰处必须有注释。

(c) 在代码修改处加上修改标识的注释。

(d) 在循环和逻辑分支组成的代码中加注释。

(e) 为他人提供的接口必须加详细注释。

### 中大网校面试问题（基础）：

1. Java中常用的用原始数据类型和引用类型：

原始数据类型：byte、short、int、long、boolean、char、float、double

引用数据类型：String、类、接口和数组。

它们的不同点：原始数据类型在创建的时候，是在栈内存上划分一块区域，将数值直接存储在栈上，而引用数据类型是在划分的栈内存上存放一个堆地址，指向的是在堆中创建的对象。

1. String和StringBuilder，StringBuffer的区别？

答：String是字符串常量，而另外俩个是字符串变量，也就是说在拼接字符串的时候，String是通过创建一个新的字符串来实现拼接的，造成了资源的浪费，冗余的那个对象由JVM的GC回收，而StringBuilder和StringBUffer都是在远对象的基础上进行修改。StringBuilder是线程非安全的，但是速度是最快的，StringBuffer是线程安全的，因为内部添加了同步方法，所以速度有所下降。

1. Hashcode的意义和计算方法，equals为true，hashcode相同吗？Hashcode相同，equals为true吗？

答：Hashcode存在的意义主要是用于查询的快捷性，为了减少调用equals的次数，如果hashcode相同，equals不一定为true，如果hashcode不相同，equals一定是false，如果equals为true，那么hashcode一定相同，另外如果我们重写了equals方法，也一定要重写equals的方法，保证他俩逻辑的一致性，这样才能起到真正的作用。

1. 作用域public，private，protected以及不写的区别？

答：public是该应用中所有的类都能使用该类，protected是同包下以及子类可以调用；默认是同包下的类可以调用；private是只有本类内部才可以调用。

1. Double float处理小数时会有精度损失，如何避免？

答：double和float的底层实现都是二进制的，但是有限位的十进制，在转换为二进制的时候，小数位会出现循环不止的情况，这时进行计算的结果也就有了误差，避免的方法可以使用大整型BigDecimal，它的底层是字符串原理。

1. ArrayLIst和LinkedList的区别？

答：Arraylist是基于数组实现的，所以查询块，支持高效的随机元素访问；Linkedlist是基于链表实现的，所以增删块，不支持高效的随机元素访问。在插入元素的时候如果都是在末尾进行操作，那么俩者的效率相差不大，我们尽量使用ArrayLIst，如果是频繁在集合的中间操作数据，那么可以使用LinkedList。

1. HashMap的工作原理？

答：HashMap底层实现是hash数组和链表，数组可以理解为是一个Entity[]，Entity对象包括key、value和next，是一个链式数据结构对象。put方法：实际上是通过计算得到key的hashcode，将hashcode%数组.length获取到该entity存放的数组的地址，如果多个值的hashcode%数组.length是相同的，则保存到数组的同一位置，后来的对象在数组上，之前的存在next属性中。Get方法：先是通过对key的hashcode进行同等的计算获取到在hash数组的位置，然后通过equals方法，获取相同的key，进而获取到value。另外，随着存储量越来越多，如果存储量大于设定的存储值的75%的时候，会自动扩容为原来的两倍。负载因子：默认值为0.75。

1. HashSet和TreeSet的区别？TreeSet的排序方法？

答：hashSet是保证数据的唯一，但是无序；TreeSet也可以保证数据的唯一，但是它还是有序的，通过存入元素实现Comparable接口，这种排序手段为自然排序，是升序；另外一种排序手段为定制排序，例如可以指定降序，在创建treeSet时，并联加入匿名内部类Comparator。

1. 同步方法和同步代码块的区别？

答：同步方法，锁对象是本身this，而同步代码块可以指定锁对象，另外同步方法的同步范围大，所以性能也相对差。

1. 如何保证n个线程访问n个资源的同时又不导致死锁？

答：死锁造成的原因就是多个线程想要获取对方占用的资源，导致谁也获取不到，造成死锁。解决方案是指定所有的线程获取锁的顺序保存一致，让所有的进程按照相同的顺序获取或者释放锁资源，就避免了死锁。

1. 数据库表设计，简单的分类，详情表结构是俩张表，分类表、详情表，如果要做一个电影的网站，一部电影可能既是动作类又是喜剧类，如何设计表？
2. 行政区结构，国家、省、市、县这样的数据，如果要存放在数据库中，如何设计表结构？

答：对于省市县村这样的数据库设计，有两种设计可选择：  
 (1)对省、市、县、村各建立一张表，每一张表中有个字段关联其所属上级；  
 比如：省数据表tb\_Province(id,province\_name,province\_desc,provoince\_zidcode）  
       市数据表tb\_City(id,city\_name,city\_code,province\_id) .......  
 (2)把省市县村的数据都放在一张表里，在这张表里增加一个字段标识每一条记录的上级关系  
 比如:tb\_mytable(id,name,type,parient\_id),其中type//标识是省/市/县/村，parent\_id标识上下级关系；

1. 一个数据量比较大的表中有创时间CreatTime字段，如果我要查2016年的数据有两种
2. where createtime >=’2016-1-1’ and createtime <’2017-1-1’
3. Where year(createtime)=2016

这两找那个方式哪一种更好？为什么？

1. 解析xml的手段？

答：　XML的解析方式分为四种：1、DOM解析；2、SAX解析；3、JDOM解析；4、DOM4J解析。其中前两种属于基础方法，是官方提供的平台无关的解析方式；后两种属于扩展方法，它们是在基础的方法上扩展出来的，只适用于java平台。

1. 数据库索引？以及什么情况下索引失效？

答：如果一张表很大，字段很多，可以在设计的时候选择添加查询频率高的字段到索引，它可以加快查询速度，索引的类型分为B-tree索引和hash索引，B-tree是数据库默认的索引方法，对添加的字段值进行了排序，方便我们查询。Hash索引是无序的，在固定key查询的情况下，效率很高，但是无法处>、<等查询，灵活性不强，所以数据库默认是B-tree索引。索引中有指定字段的值还有该行数据在磁盘上的地址，可以通过该地址获取到对应行数据的其他字段。

我们在查询数据库的时候，数据库会先判断查询的字段是否添加了索引，然后再判断该查询语句是否适合使用索引。

创建索引的语句：CREATE INDEX name\_indexON Employee (Employee\_Name, Employee\_Age)；

以下查询条件，不使用索引：

1. 查询条件中带or的，即使查询条件中某些字段有索引，也不走索引，要想索引有效，得所有查询字段都添加索引。
2. Like查询是以%开头。
3. 如果列的数据属性是字符串类型，那么在查询的时候一定要把查询关键字用’’引起来，否则不走索引。不要以字符格式声明数字，要以数字格式声明字符值。（日期同样）
4. 对字段进行判空查询时，例如is null等，不走索引，所以我们设计表的时候尽量不要有空的字段。
5. 判断某个字段索引列不等于某个值时，’!=’、>、<操作符不走索引。
6. 避免在索引列使用函数。

Sql优化手动：

1. 避免使用select \* from，要写出查询的字段。
2. 在查询语句中，能用between就避免使用in。
3. 避免对添加索引的字段进行运算，这样会不走索引。