**基础部分：**

1. Java的序列化是什么？有哪些你知道的序列化方式？把java对象转为字节流形式，目的是复制对象，如远程可以传递字节流，复制对象。实现方式： 实现Serilizable接口，实现相应的方法。
2. New 一个HashMap，向其中添加Long，Boolean数据，主要是手机号是否可用的数据。添加一条，添加两条，一直到添加1000w条。在这个过程中，HashMap的数据结构时怎么变化的？1000w个数据添加完之后，HashMap占用了多少内存？
3. 1000w甚至1亿个手机号过滤重复，可以使用哪些方式/哪些数据类型？各有什么特点？

可用HashSet将重复号过滤掉；可以通过SQL语言distinct关键字过滤重复；存数据库，可以对手机号做唯一约束。

1. ConcurrentHashMap是如何做到高并发线程安全？
2. Java中Synchronized关键字的内存语义是什么？互斥锁，对对象上锁。
3. Java中Volatile关键字的内存语义是什么？每个线程都有自己的内存空间。当读volatile时，JVM把本地内存的变量置为无效，而对主内存的变量；写时，jvm把本地内存的变量值刷新到主内存。
4. 什么是Java中原子性操作？线程安全的操作，保证在多线程操作时，不发生更新丢失。
5. 什么是Java中的CAS操作,AtomicLong实现原理？
6. 什么是可重入锁、乐观锁、悲观锁、公平锁、非公平锁、独占锁、共享锁？
7. 讲讲独占锁 ReentrantLock 原理？谈谈读写锁 ReentrantReadWriteLock 原理？
8. ThreadLocal 作为变量的线程隔离方式，其内部是如何做的？内部为一Map，key值为线程ID，value值为实例对象。
9. CyclicBarrier内部的实现与 CountDownLatch 有何不同？
10. 有3个线程。线程A和线程B并行执行，线程C需要A和B执行完成后才能执行。可以怎么实现？可以通过全局变量实现，当A、B线程结束时，设置开关，C检测到开后，再实际执行任务。
11. 现在有两个线程同时操作一个整数I，做自增操作，如何实现I的线程安全性？加synchronized （this）{l++}
12. Java中的多个线程如何传递数据？LinkedBlockingQueue 原理
13. CountDownLatch 与线程的 Join 方法区别是什么？
14. 什么是Java指令重排序？
15. Spring是如何实现AOP的？以事物为例
16. Spring有哪些设计模式？单例、工厂、代理等模式。
17. Spring中是怎么解决循环依赖的？比如A依赖于B，B又依赖于A。
18. 对Java 接口代理模式的实现原理的理解？如何使用 Java 反射实现动态代理？ Java 接口代理模式的指定增强？
19. 谈谈对Cglib 类增强动态代理的实现？和JDK 动态代理有什么区别？
20. BeanFactory 和 FactoryBean 有什么区别？
21. BeanFactory 和 ApplicationContext 又有什么不同？
22. MySQL 有哪些存储引擎啊？都有什么区别？
23. Float、Decimal 存储金额的区别？Datetime、Timestamp 存储时间的区别？Char、Varchar、Varbinary 存储字符的区别？一般在精度，范围，占有的空间上有差别，如Char（50），固定长度50，而Varchar（50）根据具体值，存储长度会有变化。
24. 对Explain参数及重要参数的理解？
25. 索引与锁有什么关系？
26. B+tree 如何进行优化？索引遵循哪些原则？
27. Innodb事务的理解
28. MySQL表设计及规范？画出实体-关系图，表表之间主键和外键关联。
29. 索引优化？说说Sql优化的几点原则？索引有开销，尽量建立在整数类型上，不要建立在text的类型上。Where子句后面的条件的字段可建索引，尽量减少扫描表的次数；返回记录数尽量少些好。
30. redis数据结构有哪些？字符串、列表、哈希等。
31. Redis缓存穿透，缓存雪崩？
32. 如何使用Redis来实现分布式锁？
33. 讲讲tcp协议（建连过程，慢启动，滑动窗口，七层模型）？
34. webservice协议（wsdl/soap格式，与rest办议的区别）？
35. Socket交互的基本流程？
36. 深拷贝和浅拷贝
37. 描述一下 JVM 加载 Class 文件的原理机制?
38. Java 堆的结构是什么样子的？
39. 在 Java 中，对象什么时候可以被垃圾回收？置为null的
40. 简述Minor GC 和 Major GC？
41. 简述一下Java 垃圾回收机制？
42. 代码编写：快速排序；
43. 代码编写：冒泡排序；

sortt(int[] array){

for(int i=0;i<array.length;i++){

for (int j=I+1;j<aray.length;j++){

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

int temp=array[i];

array[i]=array[j];

array[j]=temp;

}

}

}

}

**进阶部分：**

ConcurrentLinkedQueue 内部是如何使用 CAS 非阻塞算法来保证多线程下入队出队操作的线程安全？入队过程主要就是定位出尾节点，然后使用CAS算法将入队节点设置成尾节点的next节点，如不成功则重试。CAS原子操作由硬件保证。

1. CyclicBarrier内部的实现与 CountDownLatch 有何不同？
2. Semaphore 的内部实现是怎样的？
3. 并发组件CopyOnWriteArrayList 是如何通过写时拷贝实现并发安全的 List？
4. 同步与异步？阻塞与非阻塞？同步： 程序线性执行，提交完某请求，会等待结果，方能继续发请求。异步： 程序提交完某请求后，不必等待结果，可以继续发请求 。阻塞：调用结果返回之前，当前程序会被挂起。非阻塞：调用结果不能马上返回，当前线程也不会被挂起。
5. ThreadLocalRandom 是如何利用 ThreadLocal 的原理来解决 Random 的局限性？在多线程条件下，为每个线程维护一个Seed
6. 使用过MQ吗？在什么场景下使用的？解决了什么问题？遇到了什么问题？用过，对接的公司向MQ发数据，我获取数据。没记得有啥问题呢。
7. Kafka的topic和partition如何设计？Kafka时如何保证数据不丢失的呢？topic是逻辑概念，具体的存储是partition，Kafaka有主从机制，数据有副本，副本数最好大于3个，以便保证高可用
8. DB和MQ是两种介质。有哪些方式可以保证DB和MQ数据都写入成功的？你们是怎么做的？
9. 什么是最终一致性？最终一致性实现方式？读取数据时，多个请求能得到一致的结果。根据实际情况吧，比如单库，可以通过事务。消息队列，可以通过基于事务性消息队列的最终一致性。
10. Redis的分布式锁是怎么使用的？有什么优缺点？
11. 说一说RPC调用的整个过程，发生了哪些事情，各个模块都做了哪些事情？客户端发起调用，Client Stub打包参数，调用网络服务，服务端收到请求，解包，服务端返回结果，服务端stub打包结果，服务端通过网络返回结果。
12. MySQL的Innodb在什么情况下会发生死锁？你遇到的死锁是什么场景？是怎么解决死锁的？
13. 说说数据库事务特点及潜在问题？保证一个操作要么执行完，要么能回退到初始状态，不能只是执行了一半。事务如只提交了一半，就坏了，这是潜在问题。
14. 什么是MySQL隔离级别？Innodb如何解决幻读？
15. 讲讲Innodb行锁？可以锁定行，而不必锁定整个表，有共享锁和排它锁
16. 死锁及监控是什么？什么情况下会出现死锁？举例SQL。
17. 自增长与锁 ，锁的算法，锁问题，锁升级是什么？
18. Redis持久化的几种方式，优缺点是什么，怎么实现的？RDB持久化，指定的时间间隔内将内存中的数据集快照写入磁盘，优点：方便，容灾性好。AOF持久化是以日志的形式记录服务器所处理的每一个写、删除操作，好处：安全性高。
19. Redis的缓存失效策略？
20. Redis的数据淘汰策略？
21. redis队列应用场景？
22. 分布式缓存和一致性哈希？
23. 说说Netty线程模型，什么是零拷贝？
24. Dubbo的底层实现原理和机制？
25. 描述一个服务从发布到被消费的详细过程？
26. Dubbo的服务请求失败怎么处理？怎么熔断？
27. 对分布式事务的理解？
28. 分布式集群下如何做到唯一序列号？
29. 如何保证消息的一致性?
30. TCP如何保证数据的可靠传输的？通过分块，编号和校验等机制。
31. 代码编写：两个有序数组合并；
32. 代码编写：二叉树的前/中/后序遍历

TreeNode {

Int data;

TreeNode left;

TreeNode right;

}

//前序

BrowserTree(TreeNode tn){

If (tn!=null)

System.out.println();

BrowserTree(tn.left);

BrowserTree(tn.right);

}

中序和后序类似上面，只是改一下语句顺序即可。

1. 代码编写：使用栈实现一个队列；
2. 代码编写：打印倒三角

void printInverseTriangel(){

int num=51;

for (int i=0;i<num/2+1;i++){

for (int j=0;j<I;j++){

System.out.print(“ “);

}

for (int k=num-2\*I;k>0;k--){

System.out.print(“\*”);

}

System.out.println();

}

}

这是我们的开发工程师测评试题，请各位筛选已通过测评的候选人简历，测评要求：1、基础部分任意抽取15题，建议80%以上的正确率；  
2、进阶部分抽取15题，至少7道正确。  
3、基础、进阶两部分，每一部分至少一道编程题