1. 为什么要用锁

避免并发导致的脏读、数据不一致问题

1. 锁的实现

2.1 volatile ：实现方式有总线锁和缓存锁，缓存锁使用的是缓存一致性协议，总线锁是避免另外一个高速缓存的数据写入内存。

2.2 synchronized：使用方式有同步方法和同步代码块，同步代码块包含通过锁定静态对象和锁定变量两种方式。同步代码块实现机制是使用moniterenter和moniterexit，当执行临界区代码时，线程必须获取monitor锁，获取锁成功后，成为该监视则对象的拥有者。同步方法使用acc\_synchronized标志来实现的，jdk8中没有使用此。锁标志存放在对象头中的mark word中。Synchronized锁优化，引入了偏向锁、轻量锁。

2.3 cas操作，compareAndSwap，涉及到我预期值和原来值，当将预期值赋值给变量时需要判断当前值是否同原来值相等，如果相等，就修改，否则则不断循环操作。

2.4 java中锁的实现依赖队列同步器。

2.5 concurrentHashMap的实现原理：与hashtable的区别，分段锁，主要有segment和hashentry

2.6 juc lock接口<https://segmentfault.com/a/1190000015562196>

<https://zhuanlan.zhihu.com/p/66123372>

1. 线程池

参数：corePoolSize,池初始线程数目为0,除非使用的是预创建线程池方法，当线程数量达到corePoolSize时候，任务就会到达等待队列中。

maximumPoolSize:线程池最大线程数。当等待队列也饱和了，此时还有任务来，就需要采取饱和策略，增加线程数量，直到线程数量达到maximumPoolSize值，当线程数量已经达到maximumPoolSize值时，就拒绝任务。

Keepalivetime: 线程空闲时间，大于这个时间后，会被销毁

排队队列有哪几种（ArrayBlockingQueue,LinkedBlockingQueue,SychronizedQueue）；队列满了有哪几种处理方式（抛出异常，直接抛弃，添加到队列并抛弃队列中的第一个任务）；Executors.newFixedThreadPool(),Executors.cachedFixedThreadPool();

<https://www.cnblogs.com/kuoAT/p/6714762.html>

设计模式：常见的单例模式(一共有6种，能够手写出来一种)，io中的设计模式（适配器模式和装饰器模式）

Java8：hashmap的变化，jvm内存管理，元空间替换了永久代，lambda表达式，函数式接口，接口可以实现default方法，注解的使用场景，java.time包

<https://www.cnblogs.com/xingzc/p/6002873.html>

数据库：mysql和mongodb的区别；

事务：原子性、一致性、隔离性、持久性（acid）

原子性：事务作为一个整体，要么成功，要么失败

一致性：保证数据库状态从一个一致性状态转换为另一个一致性状态

隔离性：多个事务执行时，事务间不应相互影响

持久性：一个事务一旦提交，对数据库的修改应该永久保存

事务的并发问题有哪几种：丢失更新、脏读、不可重复读和幻读

数据库的锁有哪几种：

独占锁、排他锁和更新锁

事务的隔离级别：读未提交、读已提交、可重复读和序列化

<https://juejin.im/post/5c9b1b7df265da60e21c0b57>

数据库的索引有什么作用，底层数据结构是什么

索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息，使用B+树的原因：查找速度快、效率更高，在查找的过程中，每次都能抛弃一部分节点，减少遍历个数。

聚簇索引和非聚簇索引的区别？

MyISAM和InnoDB的区别有哪些？MyISAM表不支持事务、不支持行级锁、不支持外键。InnoDB表支持事务、支持行级锁、支持外键。

<https://blog.csdn.net/u013252072/article/details/52912385>

<https://www.cnblogs.com/Cheney222/articles/5876382.html>