

悲观锁:总是假设最坏的情况, 每次去拿数据的时候都认为别人会修改, 所以每次在拿数据的时候都会上锁, 这样别人想拿这个数据就会阻塞直到它拿到锁

代表:synchronized, ReentrantLock

乐观锁:总是假设最好的情况, 每次去拿数据的时候都认为别人不会修改, 所以不会上锁, 但是在更新的时候会判断一下在此期间别人有没有去更新这个数据。

CAS这种机制我们也可以将其称之为乐观锁。代表:Atomic类下的

AtomicBoolean, AtomicInteger, AtomicLong。它们分别用于Boolean,

Integer, Long类型的原子性操作。

通常在写操作频繁时使用悲观锁, 读操作频繁时使用乐观锁