

- 看似安全的情况下，将线程sleep一会儿就可能出问题，放大问题发生性
- 不安全买票
 - 创建线程，定义int类型票数等于10，run方法写当票数大于0时，while循环不断抢票，票数减减，打印哪个线程拿到了票。
 - 问题：
 - 有几个线程拿到了同一张票
 - 有线程抢到了第-1张票
 - 不安全原因：当票只剩最后一张的时候，多个线程发现还有票，便将数据在自己的工作内存中操作，最后一张的情况下被多次取票，导致 $1-1-1=-1$
- 不安全的银行取款
 - 余额为100元，甲取50，乙取100，二者同时取款
 - 开启线程后，有可能甲乙同时取，导致最后取出150元，余额为-50
 - 不安全的原因：甲乙同时操作数据，同时发现余额为100，满足自身取款要求，于是在自己的工作内存中操作数据， $100-50-100=-50$ ，造成数据同时被甲乙操作导致混乱
- 不安全的集合
 - 创建一个List
 - 创建一个for循环，循环一万次，每次循环用Lambda表达式创建一个线程，将线程方法体中，将自身线程名称存入集合
 - 最后打印集合的size大小，发现并没有一万个
 - 不安全原因：
 - 可能是主线程太快执行完毕，for循环中创建的线程有的还没有start，for循环后主线程需要sleep一会儿
 - 虽然线程是按照for循环以此创建的，但是线程start是根据CPU调度的，可能多个线程并发，同时add在了集合的同一个index下标，导致覆盖