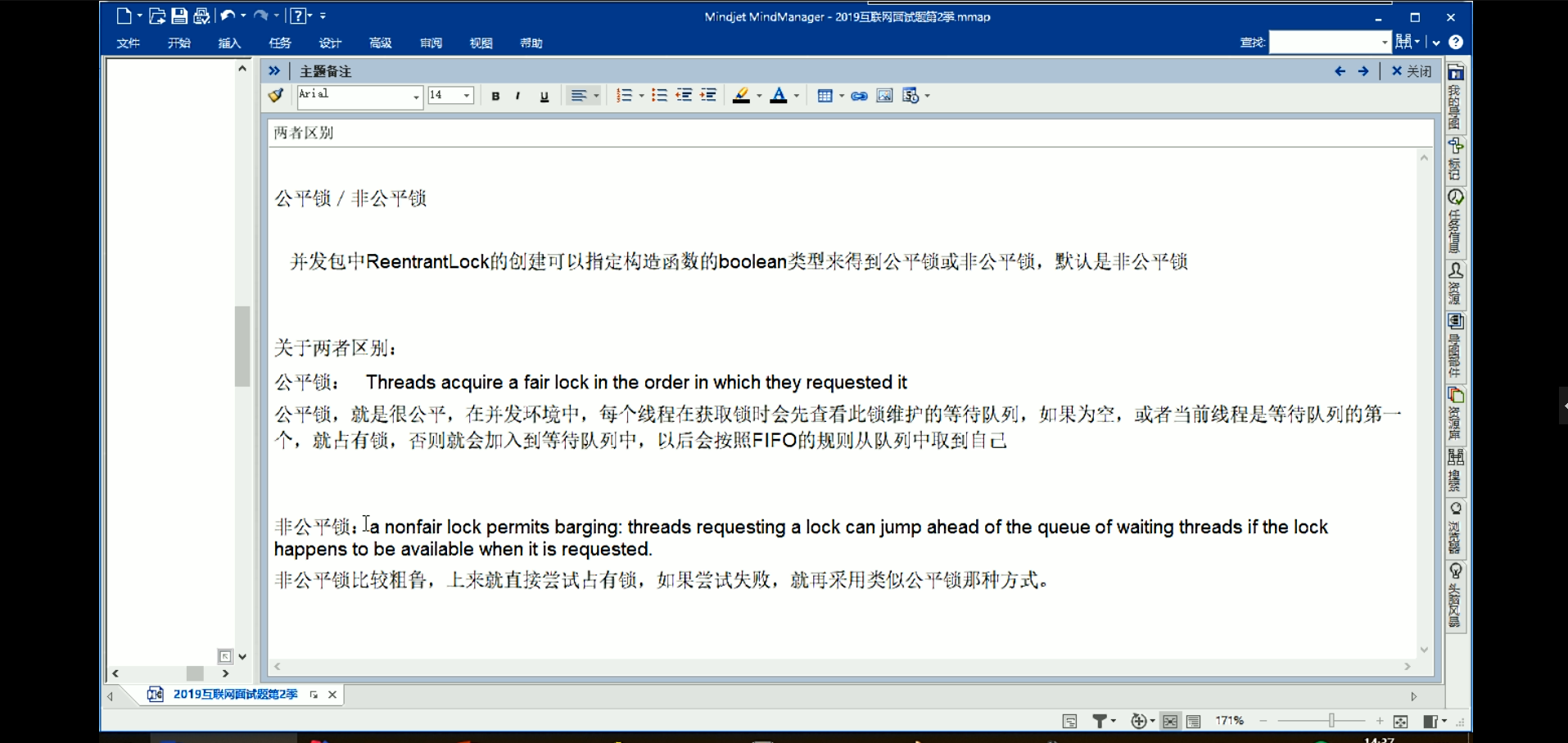
公平锁 : 是指多个线程按照申请锁的顺序来获取锁,类似于排队打饭,先来先到

非公平锁 : 是指多个线程的顺序并不是按照申请锁的顺序 , 有可能后申请的线程比先申请的线程先获取锁 . 在高并发情况下,有可能出现优先级反转或者饥饿现象

并发包下面ReentrantLock的创建可以指定构造函数的boolean类型来得到公平锁和非公平锁,默认是非公平锁



非公平锁的优点 : 吞吐量比公平锁大

Synchronyzed是一种非公平锁,可以通过参数传入true来变为公平锁

可重入锁 : 也叫递归锁

指同一个线程外部函数获取锁之后,内部递归函数仍然能获取该锁的代码

在同一个线程在外层方法获取锁的时候,在进入内层方法会自动获取锁,

也就是说,线程可以进入任何一个它已经拥有的锁所同步的代码块

Synchronyzed和ReentrantLock就是一个典型的可重入锁

可重入锁最大的作用是避免死锁

自旋锁:(spinlock)

是指尝试获取锁的线程不会立即阻塞,而是采用循环的方式去尝试获取锁这样做的好处是减少线程上下文切换的消耗,缺点是循环会消耗cpu

独占锁(写锁)-----该锁只能被一个线程所持有,reentrantLock和syncchronized都是独占锁

共享锁(读锁):--指该锁可以被多个线程所持有

对于ReentrantReadWriteLock 它的读锁是一个共享锁,写锁是一个独占锁

读锁的共享锁可以保证并发读是非常高效的 , 读写 写读 写写 的过程是互斥的

互斥锁