ReentrantLock的加锁解锁语义跟进入和退出同步代码块相同。

ReentrantLock和同步代码块都具有可重入特性。

使用ReentrantLock的理由：

1. 灵活的加锁解锁，使用灵活可能带来更好的性能
2. 使用tryLock实现锁轮训和定时锁，有限避免死锁
3. synchronized无法响应中断，ReentrantLock可用在可中断场景
4. 读写锁
5. 是否公平

使用synchronized的理由：

1. 避免忘记解锁
2. synchronized仍有优化空间，在新版本中synchronized性能已经略优于ReentrantLock。