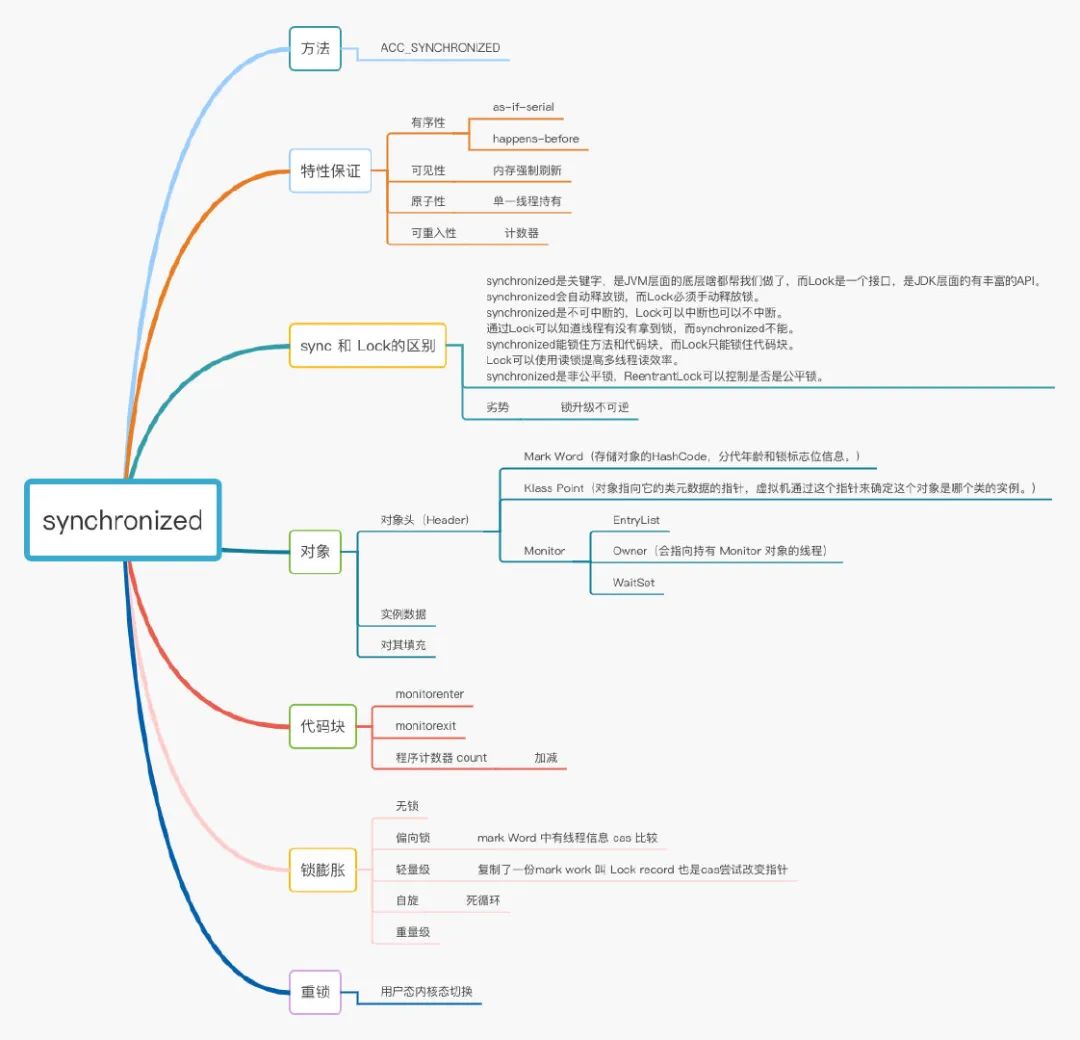
<https://mp.weixin.qq.com/s/2ka1cDTRyjsAGk_-ii4ngw>

# 死磕Synchronized底层实现

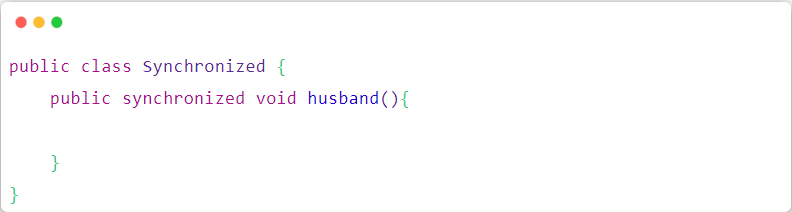


## **正文**

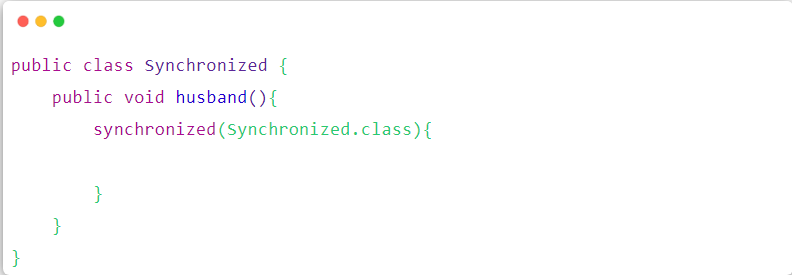
### **场景**

我们正常去使用Synchronized一般都是用在下面这几种场景：

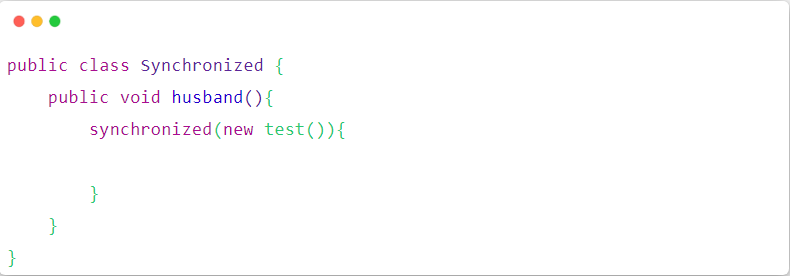
1、修饰实例方法，对当前实例对象this加锁



1. 修饰静态方法，对当前类的Class对象加锁



1. 修饰代码块，指定一个加锁的对象，给对象加锁



其实就是锁方法、锁代码块和锁对象，那他们是怎么实现加锁的呢？

在这之前，我就先跟大家聊一下我们Java对象的构成

在 JVM 中，对象在内存中分为三块区域：

* 对象头
* Mark Word（标记字段）：默认存储对象的HashCode，分代年龄和锁标志位信息。它会根据对象的状态复用自己的存储空间，也就是说在运行期间Mark Word里存储的数据会随着锁标志位的变化而变化。
* Class Point（类型指针）：对象指向它的类元数据的指针，虚拟机通过这个指针来确定这个对象是哪个类的实例。
* 实例数据
* 这部分主要是存放类的数据信息，父类的信息。
* 对齐填充
* 由于虚拟机要求对象起始地址必须是8字节的整数倍，填充数据不是必须存在的，仅仅是为了字节对齐。

Tip：不知道大家有没有被问过一个空对象占多少个字节？就是8个字节，是因为对齐填充的关系哈，不到8个字节对其填充会帮我们自动补齐。

