

volatile

概念：

每个线程都拥有自己的工作内存，1.7以前当线程开启之前会先将主内存的数据拷贝一份，然后当执行完毕线程之后再将数据写入主线程，中途若主线程数据被改变，线程是不知道的。

所以就要使用volatile

特点：

若主线程数据被volatile修饰后，当线程对volatile变量进行读操作时，会先将自己工作内存中的变量置为无效，之后再通过主内存拷贝新值到工作内存中使用。

volatile 并不能保证数据的原子性，只能保证线程之间的数据来自于主内存而不是工作内存中

的缓存数据，若想保证数据的原子性，则使用同步Synchronized.

synchronized 修饰 当有线程获取锁则将主内存数据读取到线程的工作内存中，其他线程无法访问主内存. 直到线程

执行完毕释放锁时，其他线程才能访问主内存[前提 主内存数据被完好锁住]

总结(来源于：laosu)：

1. volatile重要工作是避免线程脏读：当线程对volatile变量进行读操作时，会先将自己工作内存中的变量置为无效，之后再通过主内存拷贝新值到工作内存中使用。

2. volatile解决的是变量在多个线程之间的可见性，但不能完全保证数据的原子性。

3. 现在JVM经过优化，已不会出现liveness failure 。所以没事别用volatile。

关键字volatile主要使用的场合是在多个线程中可以感知实例变量被更改了，并且可以获得最新的值使用，也就是用多线程读取共享变量时可以获得最新值使用。

4. 关键字volatile提示线程每次从共享内存中读取变量，而不是从私有内存中读取，这样就保证了同步数据的可见性。

2. volatile和synchronized比较

通过使用volatile关键字，强制的从公共内存中读取变量的值，增加了实例变量在多个线程之间的可见性。但volatile关键字最致命的缺点是不支持原子性。

- 1.关键字volatile是线程同步的轻量级实现，所以volatile性能肯定比synchronized要好，并且volatile只能修饰变量，而synchronized可以修饰方法，以及代码块。随着新版本的发布，

2.多线程访问volatile不会发生阻塞，而synchronized会出现阻塞。

3.volatile能保证数据的可见性，但不能保证原子性；而synchronized可以保证原子性，也可以间接保证可见性，因为它会将私有内存和公共内存中的数据做同步

4.关键字volatile解决的是变量在多个线程之间的可见性；而synchronized关键字解决的是多个线程之间访问资源的同步性