# 线程封闭

* 躲避并发问题，除了设计不可变对象，还有就是线程封闭，将对象封装成一个线程中，只有这一个线程能够看到该对象，不管这个对象是否是安全的，都不会出现线程安全问题，因为只能在一个线程里访问，那么如何实现线程封闭，第一种：通过Ad-hoc线程封闭，是通过程序控制实现的，很脆弱，最糟糕的，忽略不推荐；第二种堆栈封闭，其实就是局部变量，是最常用的，多个线程访问一个方法时，方法中的局部变量放在线程栈里，所以不会出现线程并发问题，能用局部变量就不用全局变量，平时自己写代码的时候不会出现并发问题；第三种是threadlocal线程封闭，特别好，推荐，threadlocal内部维护着一个threadlocalMap对象，其中的key对应线程名称，value对应我们封闭的对象，即每个封闭对象对应map中的一个value值，threadlocal通过threadlocalmap封装对象，所以线程是安全的，也不会引起并发问题，但是要注意使用remove方法，使用完之后要将thread remove，防止出现内存溢出。

## 要了解更多信息并获取 OneNote，请访问 [www.onenote.com](http://go.microsoft.com/fwlink/?LinkID=523891)。