**Volatile关键字-内存可见性**

Java JUC 简介:

在 Java 5.0 提供了 java.util.concurrent （简称JUC ）包，在此包中增加了在并发编程中很常用的实用工具类，用于定义类似于线程的自定义子系统，包括线程池、异步 IO 和轻量级任务框架。提供可调的、灵活的线程池。还提供了设计用于多线程上下文中的 Collection 实现等

* **volatile 关键字-内存可见性**

内存可见性（Memory Visibility）是指当某个线程正在使用对象状态而另一个线程在同时修改该状态，需要确保当一个线程修改了对象状态后，其他线程能够看到发生的状态变化。可见性错误是指当读操作与写操作在不同的线程中执行时，我们无法确保执行读操作的线程能适时地看到其他线程写入的值，有时甚至是根本不可能的事情。我们可以通过同步来保证对象被安全地发布。除此之外我们也可以使用一种更加轻量级的 volatile 变量.

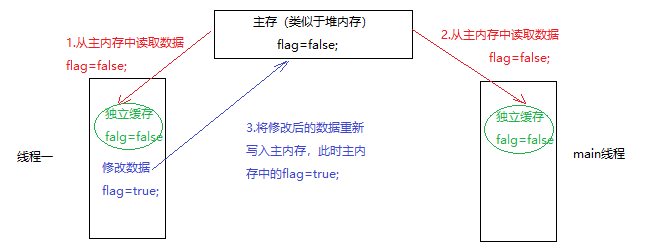
volatile 关键字:Java 提供了一种稍弱的同步机制，即 volatile 变量，用来确保将变量的更新操作通知到其他线程。可以将 volatile 看做一个轻量级的锁，但是又与锁有些不同：

1. 对于多线程，不是一种互斥关系
2. 不能保证变量状态的“原子性操作”

分析以下代码：

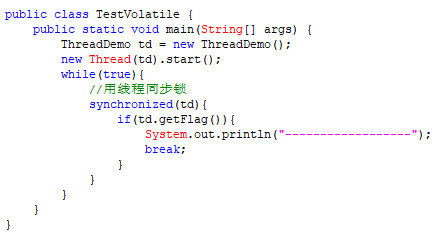


这是因为出现了内存可见性问题：当多个线程操作共享数据时，彼此不可见，如图：



解决方法：

1. 用线程同步锁：每次访问flag时先用同步更新主存中的数据，但是因为同步的效率



1. 使用 volatile 关键字

C:\Users\ADMINI~1\AppData\Local\Temp\1547296150(1).png