Synchronized, Transient, Volatile

Synchronized：声明的方法同一时间只能被一个线程访问。synchronized 修饰符可以应用于四个访问修饰符。

？Transient：只能用于方法变量。表示不能被序列化。序列化的对象包含被 transient 修饰的实例变量时，java 虚拟机(JVM)跳过该特定的变量。

该修饰符包含在定义变量的语句中，用来预处理类和变量的数据类型。

Volatile：被修饰的变量可以被多个线程访问。volatile 修饰的成员变量在每次被线程访问时，都强制从共享内存中重新读取该成员变量的值。而且，当成员变量发生变化时，会强制线程将变化值回写到共享内存。这样在任何时刻，两个不同的线程总是看到某个成员变量的同一个值。

public class MyRunnable implements Runnable

{

private volatile boolean active;

public void run() {

active = true;

while (active) // 第一行 { // 代码 }

}

public void stop() {

active = false; // 第二行

}

}

通常情况下，在一个线程调用 run() 方法（在 Runnable 开启的线程），在另一个线程调用 stop() 方法。 如果 ***第一行*** 中缓冲区的 active 值被使用，那么在 ***第二行*** 的 active 值为 false 时循环不会停止。

但是以上代码中我们使用了 volatile 修饰 active，所以该循环会停止。

Private：当前类访问权限

Default：包访问权限

Protected：子类访问权限，可以被同一个包中的其他类访问，也可以被不同包中的子类访问

Public：公共访问权限