线程池总结

* 跨线程控制方法

t.interrupt 线程中断，线程中断标识设为true，当线程处在object.wait，thread.sleep,thread.join 调用线程中断操作，提示抛出InterruptedException异常，但是t的中断标识会被设为false。

* 静态方法

Thread.sleep 线程暂停。

Thread.currentThread返回当前线程引用

Thread.intettupted清除当前线程的中断状态，返回前一个状态，线程的中断状态是不允许其他线程清除

Thread.yield 线程谦让，用在单核jvm,让具有较低有限级别的线程可以被执行。

* 对象锁

Object 有所，标量类型不是对象类型，volatile 可以影响变量原子性，可见性，排序性

标量元素数组持有锁时其中的标量元素是不拥有锁，

Class 本质是一个锁，静态同步方法是类对象锁。

* 同步方法块，同步方法

synchronized关键字并不是方法签名的一部分，synchronized 不可被继承。构造方法不可被同步。

同步代码块被认为比同步方法更加的基础。如下两种声明方式是等同的：

|  |  |
| --- | --- |
| 2 | void f() { synchronized(this) { /\* body \*/ } } |

|  |  |
| --- | --- |
| 1 | synchronized void f() { /\* body \*/ } |

同步实例方法在其子类和父类中使用同样的锁。但是内部类方法的同步却独立于其外部类， 然而一个非静态的内部类方法可以通过下面这种方式锁住其外部类。

synchronized(OuterClass.this) { /\* body \*/ }