**Thread 操作笔记**

1. **多线程的实现方式**

1.1 Runnable与Callable区别

Runnable无返回参数 调用run()方法

Callable 有返回参数 调用call()方法

1.2 Future

用于接收ExecutorService 执行Callable的返回值；调用Future.get()方法可得到返回值，如果子线程未超时，子线程没有执行完 则会主线程一直等待直至子线程走完并返回结果

1. **线程池的使用**

2.1 ExecutorService

根据配置的核心线程数、线程池最大线程数、存活时间、有界队列、线程工厂以及阻塞代理 初始化完成后，可通过执行submit方法提交多个线程任务（实现Runnable或者Callable）

2.2 CountDownLatch

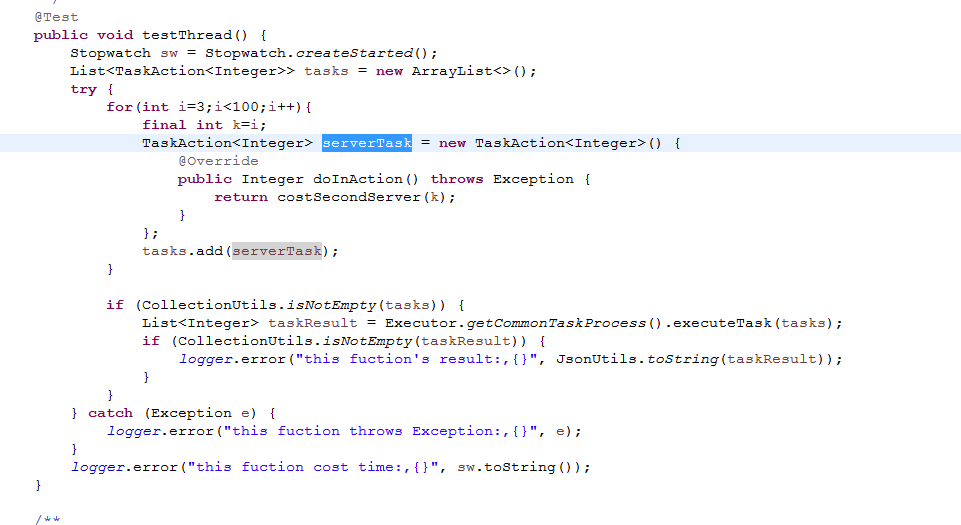
CountDownLatch为一个计数器类,可用于监控线程池内是否执行完所有任务。使用方式为：

Runnable或者Callable的try-catch-finally块中的finally块调用countDown()方法进行计数减一;ExecutorService全部提交完成后 执行await()方法，此方法会一直阻塞直至CountDownLatch计数为0（及线程池内的线程全部执行完）；

不明白为啥，反正可能分配不均匀

1. **多线程工具类**

3.1 测试实例



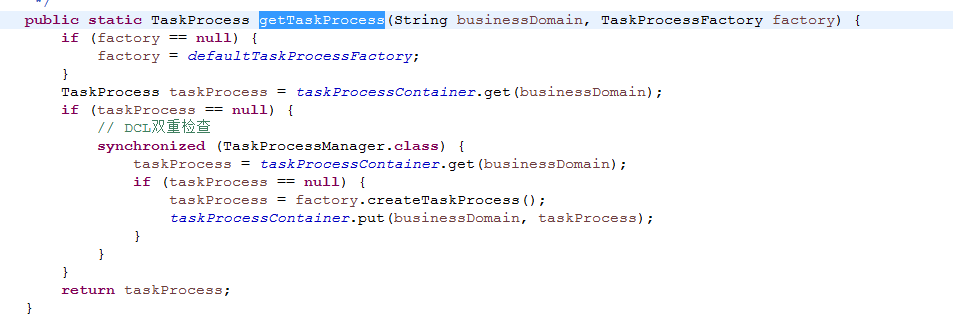


3.2 Executor

创建线程工厂并返回TaskProcess实例，依赖TaskProcessManager



3.3 TaskProcessManager



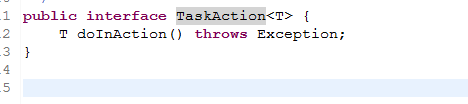


3.4 TaskProcessFactory

****

3.5 TaskAction

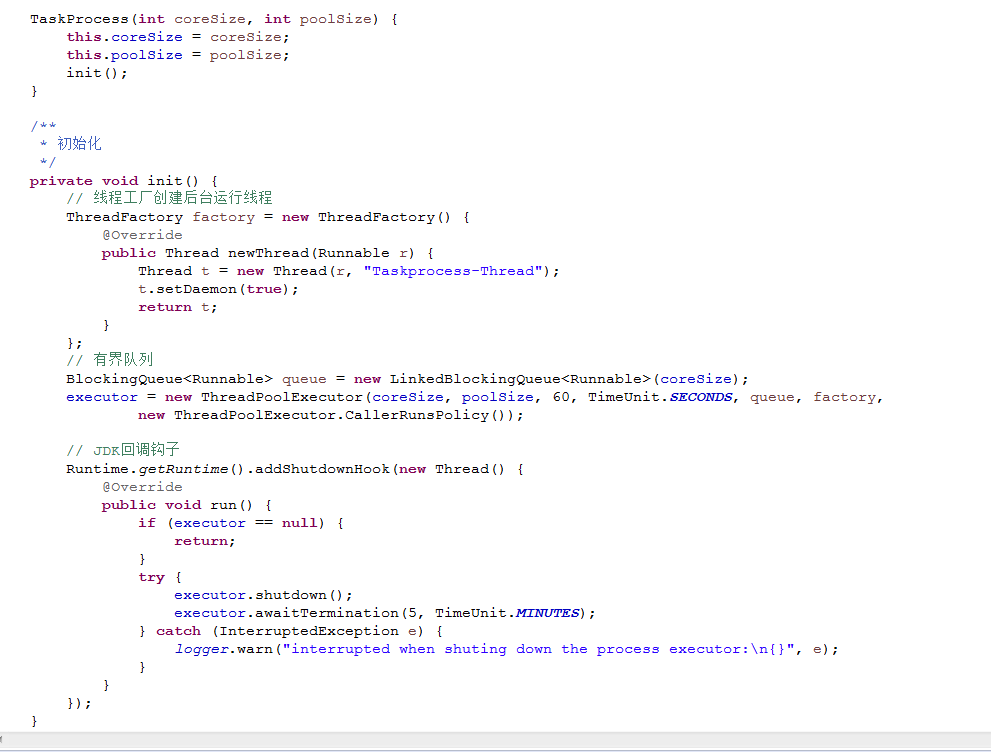
声明一个接口



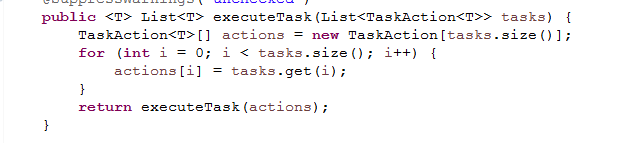
****

3.6 TaskProcess

Step.1初始化ExecutorService



Step.2 将List对象转换为数组对象



Step.3 执行线程任务



****