1. **几种线程池的区别**

* **newFixedThreadPool**

newFixedThreadPool创建一个指定工作线程数量的线程池。每当提交一个任务就创建一个工作线程，如果工作线程数量达到线程池初始的最大数，则将提交的任务存入到池队列中。

* **newCachedThreadPool**

1).工作线程的创建数量几乎没有限制(其实也有限制的,数目为Interger. MAX\_VALUE), 这样可灵活的往线程池中添加线程。   
2).如果长时间没有往线程池中提交任务，即如果工作线程空闲了指定的时间(默认为1分钟)，则该工作线程将自动终止。终止后，如果你又提交了新的任务，则线程池重新创建一个工作线程。

* **newSingleThreadExecutor**

newSingleThreadExecutor创建一个单线程化的Executor，即只创建唯一的工作者线程来执行任务，如果这个线程异常结束，会有另一个取代它，保证顺序执行(我觉得这点是它的特色)。单工作线程最大的特点是可保证顺序地执行各个任务，并且在任意给定的时间不会有多个线程是活动的 。

* **newScheduleThreadPool**

newScheduleThreadPool创建一个定长的线程池，而且支持定时的以及周期性的任务执行，类似于Timer。(这种线程池原理暂还没完全了解透彻)

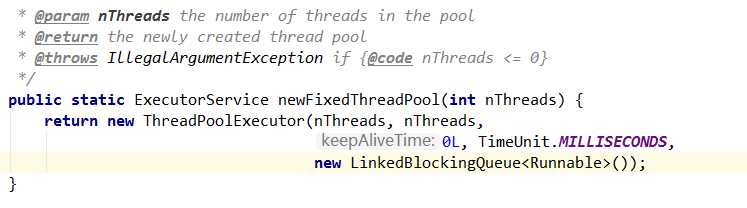
* **newWorkStealingPool**

**1.1** newFixedThreadPool

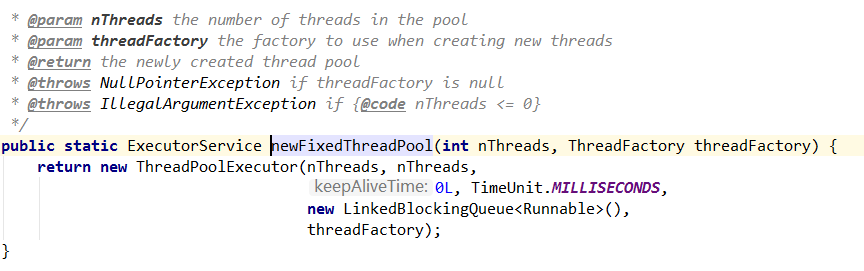
**查看**newFixedThreadPool的构造函数：

NewFixedThreadPool的构造函数有两个：

1)

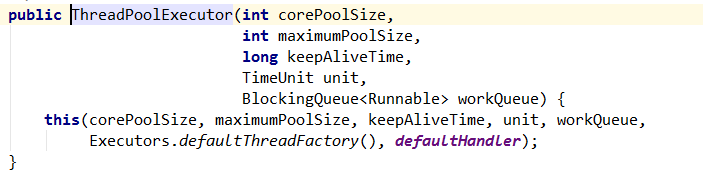


2)

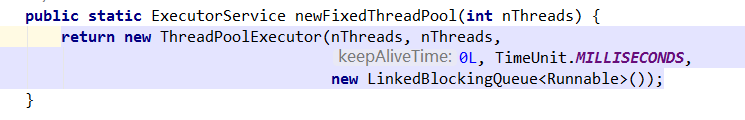


现在先看第一个构造函数，参数只有一个nThreads，线程池中的线程数，超出线程数的部分会加入LinkedBlockingQueue的等待队列，而LinkedBlockingQueue的理论值最大为int的最大值，那么现在思考一个问题，如果超出这个队列后，还有任务继续过来会怎样，可以查看跟踪查看Executors.newFixedThreadPool(num)和ThreadPoolExecutor的源码：

进行Executors.newFixedThreadPool(num)的时候，代码会返回一个ThreadPoolExecutor实例，查看ThreadPoolExecutor的构造函数，



对比newFixedThreadPool的构造函数：



ThreadPoolExecutor线程池几个参数如下：

核心线程数：nThreads；

最大线程数：nThreads；

keepAliveTime：0L；

时间单位：秒；

等待队列：BlockingQueue；