1. ThreadPoolExector常用线程池
2. Executors.newSingleThreadExecutor()

为什么单线程的线程池？

有任务队列可以进行任务管理；线程池生命周期管理

1. Executors.newCacheThreadPool()

非核心线程数为int最大值，任务队列（SynchronousQueue）大小为0，来一个任务马上起线程处理

场景：用于新任务数不太平稳，但是需要即使处理。

1. Executors.newFixedThreadPool()

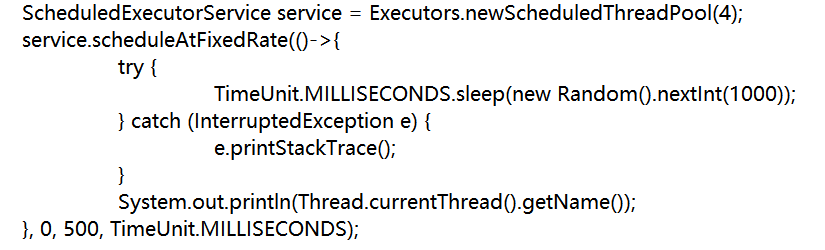
都是核心线程，数量为指定的。

场景：用于新任务数比较稳定。

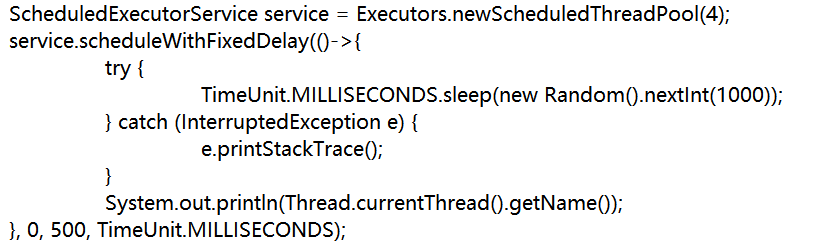
1. Executors.newScheduledThreadPool()

定时任务线程池，定时执行任务

* 定时执行任务，scheduleAtFixedRate（两次任务不能同时执行）

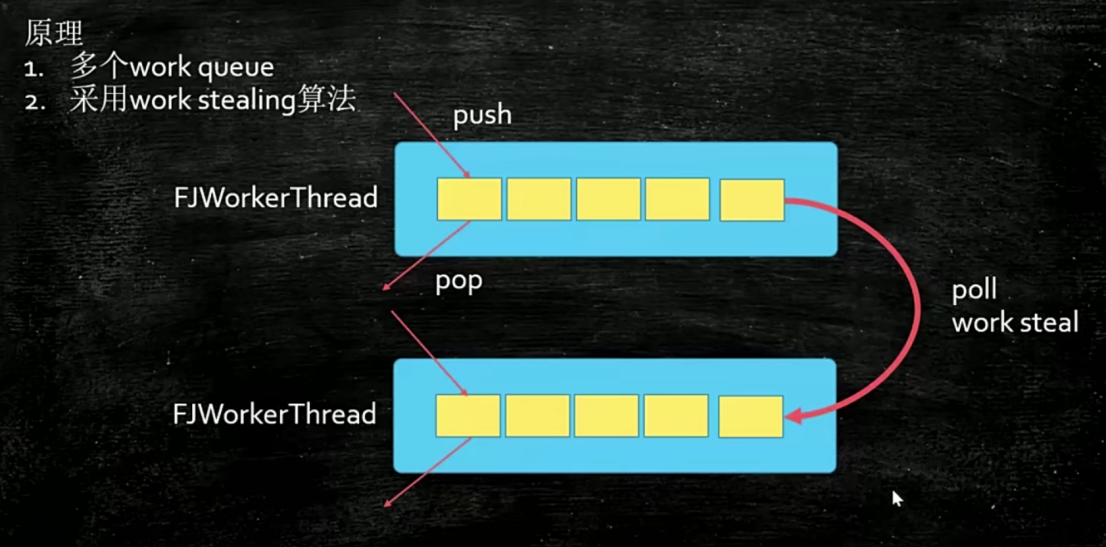
如果上一次任务未执行完成则想要阻塞等待，如果上一次任务执行完成等时间到才执行 

* 本次任务执行完成后，延迟多久执行下一次任务，scheduleWithFixedDelay



1. ForkJoin常用线程池
2. Executors.newWorkStealingPool()

每个线程有一个队列，底层是ForkJoinThreadPool



1. new ForkJoinThreadPool()

Fork：把大任务分为许多小任务

Join：把所有小任务的结果汇总

* 定义任务

有返回值：继承RecursiveTask，并重写compute方法

无返回值：继承RecuriveAction，并重写compute方法