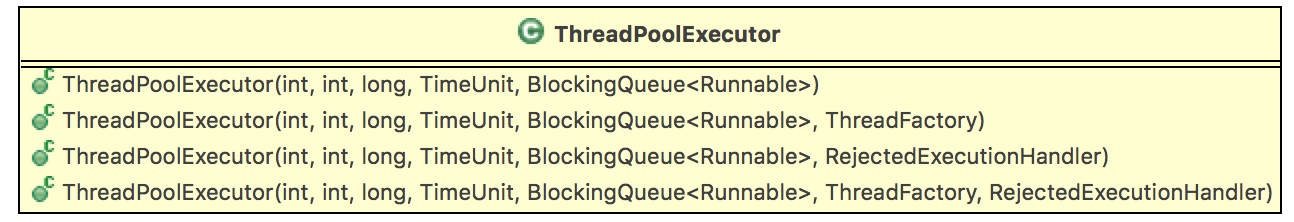
线程池之ThreadPoolExecutor使用

ThreadPoolExecutor提供了四个构造方法：



我们以最后一个构造方法，对其参数进行解释：

public ThreadPoolExecutor(int corePoolSize, // 1

int maximumPoolSize, // 2

long keepAliveTime, // 3

TimeUnit unit, // 4

BlockingQueue<Runnable> workQueue, // 5

ThreadFactory threadFactory, // 6

RejectedExecutionHandler handler ) { //7

if (corePoolSize < 0 ||

maximumPoolSize <= 0 ||

maximumPoolSize < corePoolSize ||

keepAliveTime < 0)

throw new IllegalArgumentException();

if (workQueue == null || threadFactory == null || handler == null)

throw new NullPointerException();

this.corePoolSize = corePoolSize;

this.maximumPoolSize = maximumPoolSize;

this.workQueue = workQueue;

this.keepAliveTime = unit.toNanos(keepAliveTime);

this.threadFactory = threadFactory;

this.handler = handler;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 含义 |
| 1 | corePoolSize | int | 核心线程池大小 |
| 2 | maximumPoolSize | int | 最大线程池大小 |
| 3 | keepAliveTime | long | 线程最大空闲时间 |
| 4 | unit | TimeUnit | 时间单位 |
| 5 | workQueue | BlockingQueue<Runnable> | 线程等待队列 |
| 6 | threadFactory | ThreadFactory | 线程创建工厂 |
| 7 | handler | RejectedExecutionHandler | 拒绝策略 |

知道了各个参数的作用后，我们开始构造符合我们期待的线程池。首先看JDK给我们预定义的几种线程池：

FixedThreadPool

CachedThreadPool

SingleThreadExecutor

ScheduledThreadPool

自定义线程池