**并发编程**

ThreadPoolExecutor类

包含多个可调整的参数和可扩展的钩子。

**核心参数：**

corePoolSize：核心线程数

maximumPoolSize：最大线程数

keepAliveTime：线程存活时间；如果当前线程池中存在比corePoolSize数量多的线程，当空闲时间超过keepAliveTime，多余的线程将终止。

Queue：队列，三种类型的队列：

1. Direct handoffs(直接交换)，如：SynchronousQueue(不能保持task，只能传递task)
2. Unbounded queues(无界队列)，如：LinkedBlockingQueue
3. Bounded queues(有界队列)，如：ArrayBlockingQueue

**Rejected Task(拒绝接受task)：**

4种处理策略：

ThreadPoolExecutor.AbortPolicy：默认的处理策略，抛出运行时异常RejectedExecutionException。

ThreadPoolExecutor.CallerRunsPolicy：使用调用execute方法本身的线程执行该任务。

ThreadPoolExecutor.DiscardPolicy：丢弃该任务

ThreadPoolExecutor.DiscardOldestPolicy：当executor未关闭，删除工作队列的头部的task。

Hook Methods(钩子方法)

beforeExecute()和afterExecute()方法：在执行每个task之前和之后调用。

**RunState(线程池的运行状态):**

RUNNING：接受新的任务和处理排队任务

SHUTDOWN：不接受新的任务，但处理排队任务

STOP: 不在接受新的任务，不处理排队任务并且中断正在执行的任务

TIDYING(整理): 所有的任务都已终止，workerCount为0，转换到tidying状态的线程将运行terminate()钩子方法

TERMINATED: 所有的terminate()钩子方法都已经结束

Thread、ThreadGroup

NOTE：

1、对于corePoolSize和maximunPoolSize

当调用方法execute(Runnable task)提交新任务时，当运行的线程数少于corePoolSize数量的线程，即使其他的工作线程处于空闲状态，也会创建一个新线程来处理该任务；如果此时有超过corePoolSize但少于maximumPoolSize的线程在运行，当且当队列满了，才会创建一个新的线程来处理该任务。