

# ThreadPoolExecutor

---

## 一、基本介绍

### 1、概念

java.util.concurrent.ThreadPoolExecutor类是线程池中最核心的一个类

### 2、构造方法

```
public ThreadPoolExecutor(int corePoolSize,int maximumPoolSize,long
keepAliveTime,TimeUnit unit,
    BlockingQueue<Runnable> workQueue);
```

```
public ThreadPoolExecutor(int corePoolSize,int maximumPoolSize,long
keepAliveTime,TimeUnit unit,
    BlockingQueue<Runnable> workQueue,ThreadFactory threadFactory);
```

```
public ThreadPoolExecutor(int corePoolSize,int maximumPoolSize,long
keepAliveTime,TimeUnit unit,
    BlockingQueue<Runnable> workQueue,RejectedExecutionHandler handler);
```

```
public ThreadPoolExecutor(int corePoolSize,int maximumPoolSize,long
keepAliveTime,TimeUnit unit,
    BlockingQueue<Runnable> workQueue,ThreadFactory
threadFactory,RejectedExecutionHandler handler);
```

### 3、参数

#### 1) corePoolSize:

核心池的大小，这个参数与后面讲述的线程池的实现原理有非常大的关系。在创建了线程池后，默认情况下，线程池中并没有任何线程，而是等待有任务到来才创建线程去执行任务，除非调用了

```
prestartAllCoreThreads()
```

或者

```
prestartCoreThread()
```

方法，从这2个方法的名字就可以看出，是预创建线程的意思，即在没有任何任务到来之前就创建corePoolSize个线程或者一个线程。默认情况下，在创建了线程池后，线程池中的线程数为0，当有任务来之后，就会创建一个线程去执行任务，当线程池中的线程数目达到corePoolSize后，就会把到达的任务放到缓存队列当中；

#### 2) maximumPoolSize:

线程池最大线程数，它表示在线程池中最多能创建多少个线程；

#### 3) keepAliveTime:

表示线程没有任务执行时最多保持多久时间会终止。默认情况下，只有当线程池中的线程数大于corePoolSize时，keepAliveTime才会起作用，直到线程池中的线程数不大于corePoolSize；即当线程池中的线程数大于corePoolSize时，如果一个线程空闲的时间达到keepAliveTime，则会终止，直到线程池中的线程数不超过corePoolSize；但是如果调用了allowCoreThreadTimeOut(boolean)方法，在线程池中的线程数不大于corePoolSize时，keepAliveTime参数也会起作用，直到线程池中的线程数为0；

#### 4) unit:

参数keepAliveTime的时间单位，有7种取值，在TimeUnit类中有7种静态属性，以及TimeUtil的源码展示：

TimeUnit.DAYS;	//天
TimeUnit.HOURS;	//小时
TimeUnit.MINUTES;	//分钟
TimeUnit.SECONDS;	//秒
TimeUnit.MILLISECONDS;	//毫秒
TimeUnit.MICROSECONDS;	//微妙
TimeUnit.NANOSECONDS;	//纳秒

#### 5) workQueue:

一个阻塞队列，用来存储等待执行的任务，这个参数的选择会对线程池的运行过程产生重大影响，一般来说，这里的阻塞队列有以下几种选择：

**ArrayBlockingQueue;**

**LinkedBlockingQueue;**

**SynchronousQueue;**

6)