# Executors类源码阅读

## 类的概述

Factory and utility methods for [Executor](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executor.html), [ExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html), [ScheduledExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html), [ThreadFactory](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html), and [Callable](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) classes defined in this package. This class supports the following kinds of methods:

此程序包中定义的Executor，ExecutorService，ScheduledExecutorService，ThreadFactory和Callable类的工厂和实用程序方法。此类支持以下方法：

* Methods that create and return an [ExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) set up with commonly useful configuration settings.

使用常用的配置设置创建和返回ExecutorService的方法。

* Methods that create and return a [ScheduledExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) set up with commonly useful configuration settings.

使用常用的配置设置创建和返回ScheduledExecutorService的方法。

* Methods that create and return a "wrapped" ExecutorService, that disables reconfiguration by making implementation-specific methods inaccessible.

创建并返回“包装的” ExecutorService的方法，该方法通过使特定于实现的方法不可访问来禁用重新配置

* Methods that create and return a [ThreadFactory](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) that sets newly created threads to a known state.

创建并返回将新创建的线程设置为已知状态的ThreadFactory的方法。

* Methods that create and return a [Callable](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) out of other closure-like forms, so they can be used in execution methods requiring Callable.

从其他类似于闭包的形式创建并返回Callable的方法，因此它们可用于需要Callable的执行方法中。

## 方法的概述

|  |  |
| --- | --- |
| **Modifier and Type** | **Method and Description** |
| static [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<[**Object**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)> | [**callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#callable-java.security.PrivilegedAction-)([**PrivilegedAction**](https://docs.oracle.com/javase/8/docs/api/java/security/PrivilegedAction.html)<?> action)  Returns a [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) object that, when called, runs the given privileged action and returns its result.  返回一个Callable对象，该对象在被调用时将运行给定的特权操作并返回其结果。 |
| static [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<[**Object**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)> | [**callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#callable-java.security.PrivilegedExceptionAction-)([**PrivilegedExceptionAction**](https://docs.oracle.com/javase/8/docs/api/java/security/PrivilegedExceptionAction.html)<?> action)  Returns a [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) object that, when called, runs the given privileged exception action and returns its result.  返回一个Callable对象，该对象在被调用时运行给定的特权异常操作并返回其结果。 |
| static [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<[**Object**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)> | [**callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#callable-java.lang.Runnable-)([**Runnable**](https://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html) task)  Returns a [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) object that, when called, runs the given task and returns null.  返回一个Callable对象，该对象在被调用时将运行给定任务并返回null。 |
| static <T> [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<T> | [**callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#callable-java.lang.Runnable-T-)([**Runnable**](https://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html) task, T result)  Returns a [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) object that, when called, runs the given task and returns the given result.  返回一个Callable对象，该对象在被调用时将运行给定任务并返回给定结果。 |
| static [**ThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) | [**defaultThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#defaultThreadFactory--)()  Returns a default thread factory used to create new threads.  返回用于创建新线程的默认线程工厂。 |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newCachedThreadPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newCachedThreadPool--)()  Creates a thread pool that creates new threads as needed, but will reuse previously constructed threads when they are available. 创建一个线程池，该线程池根据需要创建新线程，但是将在先前构造的线程可用时重用它们。 |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newCachedThreadPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newCachedThreadPool-java.util.concurrent.ThreadFactory-)([**ThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) threadFactory)  Creates a thread pool that creates new threads as needed, but will reuse previously constructed threads when they are available, and uses the provided ThreadFactory to create new threads when needed.  创建一个线程池，该线程池根据需要创建新线程，但是将在可用之前重用以前构造的线程，并在需要时使用提供的ThreadFactory创建新线程。 |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newFixedThreadPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newFixedThreadPool-int-)(int nThreads)  Creates a thread pool that reuses a fixed number of threads operating off a shared unbounded queue.  创建一个线程池，该线程池重用在共享的无边界队列上运行的固定数量的线程。 |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newFixedThreadPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newFixedThreadPool-int-java.util.concurrent.ThreadFactory-)(int nThreads, [**ThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) threadFactory)  Creates a thread pool that reuses a fixed number of threads operating off a shared unbounded queue, using the provided ThreadFactory to create new threads when needed.  创建一个线程池，该线程池重用固定数量的线程，这些线程在共享的无界队列之外运行，并在需要时使用提供的ThreadFactory创建新线程。 |
| static [**ScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) | [**newScheduledThreadPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newScheduledThreadPool-int-)(int corePoolSize)  Creates a thread pool that can schedule commands to run after a given delay, or to execute periodically.  创建一个线程池，该线程池可以安排命令在给定的延迟后运行或定期执行。 |
| static [**ScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) | [**newScheduledThreadPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newScheduledThreadPool-int-java.util.concurrent.ThreadFactory-)(int corePoolSize, [**ThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) threadFactory)  Creates a thread pool that can schedule commands to run after a given delay, or to execute periodically. |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newSingleThreadExecutor**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newSingleThreadExecutor--)()  Creates an Executor that uses a single worker thread operating off an unbounded queue.  创建一个执行程序，该执行程序使用在不受限制的队列上操作的单个工作线程。 |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newSingleThreadExecutor**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newSingleThreadExecutor-java.util.concurrent.ThreadFactory-)([**ThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) threadFactory)  Creates an Executor that uses a single worker thread operating off an unbounded queue, and uses the provided ThreadFactory to create a new thread when needed. |
| static [**ScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) | [**newSingleThreadScheduledExecutor**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newSingleThreadScheduledExecutor--)()  Creates a single-threaded executor that can schedule commands to run after a given delay, or to execute periodically. |
| static [**ScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) | [**newSingleThreadScheduledExecutor**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newSingleThreadScheduledExecutor-java.util.concurrent.ThreadFactory-)([**ThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) threadFactory)  Creates a single-threaded executor that can schedule commands to run after a given delay, or to execute periodically. |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newWorkStealingPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newWorkStealingPool--)()  Creates a work-stealing thread pool using all [**available processors**](https://docs.oracle.com/javase/8/docs/api/java/lang/Runtime.html#availableProcessors--) as its target parallelism level. |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**newWorkStealingPool**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#newWorkStealingPool-int-)(int parallelism)  Creates a thread pool that maintains enough threads to support the given parallelism level, and may use multiple queues to reduce contention. |
| static <T> [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<T> | [**privilegedCallable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#privilegedCallable-java.util.concurrent.Callable-)([**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<T> callable)  Returns a [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) object that will, when called, execute the given callable under the current access control context. |
| static <T> [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<T> | [**privilegedCallableUsingCurrentClassLoader**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#privilegedCallableUsingCurrentClassLoader-java.util.concurrent.Callable-)([**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)<T> callable)  Returns a [**Callable**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html) object that will, when called, execute the given callable under the current access control context, with the current context class loader as the context class loader. |
| static [**ThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html) | [**privilegedThreadFactory**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#privilegedThreadFactory--)()  Returns a thread factory used to create new threads that have the same permissions as the current thread. |
| static [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) | [**unconfigurableExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#unconfigurableExecutorService-java.util.concurrent.ExecutorService-)([**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) executor)  Returns an object that delegates all defined [**ExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) methods to the given executor, but not any other methods that might otherwise be accessible using casts. |
| static [**ScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) | [**unconfigurableScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#unconfigurableScheduledExecutorService-java.util.concurrent.ScheduledExecutorService-)([**ScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) executor)  Returns an object that delegates all defined [**ScheduledExecutorService**](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) methods to the given executor, but not any other methods that might otherwise be accessible using casts. |

## 重点方法详细

|  |  |
| --- | --- |
| **方法** | public static [ExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) newCachedThreadPool() |
| **描述** | Creates a thread pool that creates new threads as needed, but will reuse previously constructed threads when they are available. These pools will typically improve the performance of programs that execute many short-lived asynchronous tasks. Calls to execute will reuse previously constructed threads if available. If no existing thread is available, a new thread will be created and added to the pool. Threads that have not been used for sixty seconds are terminated and removed from the cache. Thus, a pool that remains idle for long enough will not consume any resources. Note that pools with similar properties but different details (for example, timeout parameters) may be created using [ThreadPoolExecutor](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadPoolExecutor.html)constructors.  新建一个线程池，该线程池会在必要的时候创建线程，优先使用池中存在的空闲线程。该线程池将提高需要频繁执行短期异步任务的程序的性能。  如果可用，执行调用将重用以前构造的线程。如果没有可用的现有线程，则将创建一个新线程并将其添加到池中。六十秒内未使用的线程将终止并从池中删除。因此长时间空闲的线程池将不会持有任何资源。注意，可以只用ThreadPoolExecutor类的构造函数创建更加定制化的线程池。 |
| **细节** | public static ExecutorService newCachedThreadPool() {  return new ThreadPoolExecutor(0, Integer.MAX\_VALUE,  60L, TimeUnit.SECONDS,  new SynchronousQueue<Runnable>()); }  以上为newCachedThreadPool()方法的源码，方法中使用ThreadPoolExecutor类的构造方法创建线程池。该构造方法的参数情况为：  public ThreadPoolExecutor(int corePoolSize,int maximumPoolSize,long keepAliveTime,TimeUnit unit,BlockingQueue<Runnable> workQueue)  maximumPoolSize参数表示池中能够存在的线程的最大数量，由此可见，newCachedThreadPool()方法创建的线程池可以容纳的线程是没有上限的，使用过程中可能造成池中线程的不断创建，有内存泄漏的风险。 |

|  |  |
| --- | --- |
| **方法** | public static [ExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) newFixedThreadPool(int nThreads) |
| **描述** | Creates a thread pool that reuses a fixed number of threads operating off a shared unbounded queue. At any point, at most nThreads threads will be active processing tasks. If additional tasks are submitted when all threads are active, they will wait in the queue until a thread is available. If any thread terminates due to a failure during execution prior to shutdown, a new one will take its place if needed to execute subsequent tasks. The threads in the pool will exist until it is explicitly [shutdown](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html#shutdown--).  创建一个基于无限队列的线程数可变的线程池，任意时刻，池中最多只有nThread个线程。如果在关闭之前执行期间由于执行失败导致任何线程终止，，如果需要执行后续任务，则将替换一个新线程。池中的线程会一直存在，直到明确将其关闭。 |
| **细节** | public static ExecutorService newFixedThreadPool*(*int nThreads*) {* return new ThreadPoolExecutor*(*nThreads, nThreads,  0L, TimeUnit.*MILLISECONDS*,  new LinkedBlockingQueue*<*Runnable*>())*; *}*    通过查看newFixedThreadPool方法以及ThreadPoolExecutor方法的源码，我发现：   1. newFixedThreadPool将线程池核心线程数和线程池最大线程数设置为相当，结合ThreadPoolExecutor类中的注解，我们可以知道，使用newFixedThreadPool方法得到的线程池有这样一个特殊的地方：当存在一个任务需要执行，如果池中的线程数少于nThread，即使存在空闲线程，依然会创建新的线程来执行任务。 2. 线程的最大空闲时间keepAliveTime在此处其实不发挥作用，因为线程最大空闲时间作用于池中线程数超过corePoolSize的情况。之前说过在这里corePoolSize的值和maximumPoolSize的值相等，所以池中的线程其实会一直存在，直到池被关闭。 3. 等待队列为无限队列，因此存在内存泄漏的风险。 |

|  |  |
| --- | --- |
| **方法** | public static [ScheduledExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html) newScheduledThreadPool(int corePoolSize) |
| **描述** | Creates a thread pool that can schedule commands to run after a given delay, or to execute periodically.  创建一个线程池，该线程池可以安排命令在给定的延迟后运行或定期执行。 |

|  |  |
| --- | --- |
| **方法** | public static [ExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) newSingleThreadExecutor() |
| **描述** | Creates an Executor that uses a single worker thread operating off an unbounded queue. (Note however that if this single thread terminates due to a failure during execution prior to shutdown, a new one will take its place if needed to execute subsequent tasks.) Tasks are guaranteed to execute sequentially, and no more than one task will be active at any given time. Unlike the otherwise equivalent newFixedThreadPool(1) the returned executor is guaranteed not to be reconfigurable to use additional threads.  基于无限队列创建单线程池。（需要注意的是，如果该线程在实行任务的时候因为失败而终止，将会创建一个新的线程执行后续的任务），使用这种线程池，能够确保任务按照提交顺序执行，且同一时间内只有一个任务被执行。和与之等效的newFixedThreadPool(1)不同，本方法生成的线程池能够保证程序不可重新配置为使用其他线程。 |

|  |  |
| --- | --- |
| **方法** | public static [ExecutorService](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html) newWorkStealingPool() |
| **描述** | Creates a work-stealing thread pool using all [available processors](https://docs.oracle.com/javase/8/docs/api/java/lang/Runtime.html#availableProcessors--) as its target parallelism level.  使用所有可用处理器作为目标并行度，创建一个“窃取工作”线程池。  available processors - Java虚拟机可用的处理器数量，在虚拟机的特定调用期间，此值可能会更改。因此，对可用处理器数量敏感的应用程序应该偶尔轮询此属性并适当地调整其资源使用情况。 |

|  |  |
| --- | --- |
| **方法** |  |
| **描述** |  |