| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/util/concurrent/TimeUnit.html) | **Use** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent//class-useTimeUnit.html)    [**NO FRAMES**](http://docs.google.com/TimeUnit.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Class**

**java.util.concurrent.TimeUnit**

| Packages that use [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) | |
| --- | --- |
| [**java.util.concurrent**](#3znysh7) | Utility classes commonly useful in concurrent programming. |
| [**java.util.concurrent.locks**](#2et92p0) | Interfaces and classes providing a framework for locking and waiting for conditions that is distinct from built-in synchronization and monitors. |
| [**javax.swing**](#tyjcwt) | Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms. |

| Uses of [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) | |
| --- | --- |

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) that return [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) | |
| --- | --- |
| static [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) | **TimeUnit.**[**valueOf**](http://docs.google.com/java/util/concurrent/TimeUnit.html#valueOf(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns the enum constant of this type with the specified name. |
| static [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html)[] | **TimeUnit.**[**values**](http://docs.google.com/java/util/concurrent/TimeUnit.html#values())()            Returns an array containing the constants of this enum type, in the order they are declared. |

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) | |
| --- | --- |
| int | **CyclicBarrier.**[**await**](http://docs.google.com/java/util/concurrent/CyclicBarrier.html#await(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Waits until all [parties](http://docs.google.com/java/util/concurrent/CyclicBarrier.html#getParties()) have invoked await on this barrier, or the specified waiting time elapses. |
| boolean | **CountDownLatch.**[**await**](http://docs.google.com/java/util/concurrent/CountDownLatch.html#await(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Causes the current thread to wait until the latch has counted down to zero, unless the thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()), or the specified waiting time elapses. |
| boolean | **ThreadPoolExecutor.**[**awaitTermination**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#awaitTermination(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| boolean | **ExecutorService.**[**awaitTermination**](http://docs.google.com/java/util/concurrent/ExecutorService.html#awaitTermination(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Blocks until all tasks have completed execution after a shutdown request, or the timeout occurs, or the current thread is interrupted, whichever happens first. |
| long | **TimeUnit.**[**convert**](http://docs.google.com/java/util/concurrent/TimeUnit.html#convert(long,%20java.util.concurrent.TimeUnit))(long sourceDuration, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) sourceUnit)            Convert the given time duration in the given unit to this unit. |
| [V](http://docs.google.com/java/util/concurrent/Exchanger.html) | **Exchanger.**[**exchange**](http://docs.google.com/java/util/concurrent/Exchanger.html#exchange(V,%20long,%20java.util.concurrent.TimeUnit))([V](http://docs.google.com/java/util/concurrent/Exchanger.html) x, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Waits for another thread to arrive at this exchange point (unless the current thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()) or the specified waiting time elapses), and then transfers the given object to it, receiving its object in return. |
| [V](http://docs.google.com/java/util/concurrent/FutureTask.html) | **FutureTask.**[**get**](http://docs.google.com/java/util/concurrent/FutureTask.html#get(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [V](http://docs.google.com/java/util/concurrent/Future.html) | **Future.**[**get**](http://docs.google.com/java/util/concurrent/Future.html#get(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Waits if necessary for at most the given time for the computation to complete, and then retrieves its result, if available. |
| long | **Delayed.**[**getDelay**](http://docs.google.com/java/util/concurrent/Delayed.html#getDelay(java.util.concurrent.TimeUnit))([TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Returns the remaining delay associated with this object, in the given time unit. |
| long | **ThreadPoolExecutor.**[**getKeepAliveTime**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#getKeepAliveTime(java.util.concurrent.TimeUnit))([TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Returns the thread keep-alive time, which is the amount of time that threads in excess of the core pool size may remain idle before being terminated. |
| | <T> [List](http://docs.google.com/java/util/List.html)<[Future](http://docs.google.com/java/util/concurrent/Future.html)<T>> | | --- | | **ExecutorService.**[**invokeAll**](http://docs.google.com/java/util/concurrent/ExecutorService.html#invokeAll(java.util.Collection,%20long,%20java.util.concurrent.TimeUnit))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Executes the given tasks, returning a list of Futures holding their status and results when all complete or the timeout expires, whichever happens first. |
| | <T> [List](http://docs.google.com/java/util/List.html)<[Future](http://docs.google.com/java/util/concurrent/Future.html)<T>> | | --- | | **AbstractExecutorService.**[**invokeAll**](http://docs.google.com/java/util/concurrent/AbstractExecutorService.html#invokeAll(java.util.Collection,%20long,%20java.util.concurrent.TimeUnit))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| | <T> T | | --- | | **ExecutorService.**[**invokeAny**](http://docs.google.com/java/util/concurrent/ExecutorService.html#invokeAny(java.util.Collection,%20long,%20java.util.concurrent.TimeUnit))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Executes the given tasks, returning the result of one that has completed successfully (i.e., without throwing an exception), if any do before the given timeout elapses. |
| | <T> T | | --- | | **AbstractExecutorService.**[**invokeAny**](http://docs.google.com/java/util/concurrent/AbstractExecutorService.html#invokeAny(java.util.Collection,%20long,%20java.util.concurrent.TimeUnit))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| boolean | **SynchronousQueue.**[**offer**](http://docs.google.com/java/util/concurrent/SynchronousQueue.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/SynchronousQueue.html) o, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element into this queue, waiting if necessary up to the specified wait time for another thread to receive it. |
| boolean | **PriorityBlockingQueue.**[**offer**](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element into this priority queue. |
| boolean | **LinkedBlockingQueue.**[**offer**](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element at the tail of this queue, waiting if necessary up to the specified wait time for space to become available. |
| boolean | **LinkedBlockingDeque.**[**offer**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| boolean | **DelayQueue.**[**offer**](http://docs.google.com/java/util/concurrent/DelayQueue.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/DelayQueue.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element into this delay queue. |
| boolean | **BlockingDeque.**[**offer**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element into the queue represented by this deque (in other words, at the tail of this deque), waiting up to the specified wait time if necessary for space to become available. |
| boolean | **BlockingQueue.**[**offer**](http://docs.google.com/java/util/concurrent/BlockingQueue.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/BlockingQueue.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element into this queue, waiting up to the specified wait time if necessary for space to become available. |
| boolean | **ArrayBlockingQueue.**[**offer**](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html#offer(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element at the tail of this queue, waiting up to the specified wait time for space to become available if the queue is full. |
| boolean | **LinkedBlockingDeque.**[**offerFirst**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#offerFirst(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| boolean | **BlockingDeque.**[**offerFirst**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#offerFirst(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element at the front of this deque, waiting up to the specified wait time if necessary for space to become available. |
| boolean | **LinkedBlockingDeque.**[**offerLast**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#offerLast(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| boolean | **BlockingDeque.**[**offerLast**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#offerLast(E,%20long,%20java.util.concurrent.TimeUnit))([E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) e, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Inserts the specified element at the end of this deque, waiting up to the specified wait time if necessary for space to become available. |
| [E](http://docs.google.com/java/util/concurrent/SynchronousQueue.html) | **SynchronousQueue.**[**poll**](http://docs.google.com/java/util/concurrent/SynchronousQueue.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Retrieves and removes the head of this queue, waiting if necessary up to the specified wait time, for another thread to insert it. |
| [E](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html) | **PriorityBlockingQueue.**[**poll**](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html) | **LinkedBlockingQueue.**[**poll**](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) | **LinkedBlockingDeque.**[**poll**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [Future](http://docs.google.com/java/util/concurrent/Future.html)<[V](http://docs.google.com/java/util/concurrent/ExecutorCompletionService.html)> | **ExecutorCompletionService.**[**poll**](http://docs.google.com/java/util/concurrent/ExecutorCompletionService.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [E](http://docs.google.com/java/util/concurrent/DelayQueue.html) | **DelayQueue.**[**poll**](http://docs.google.com/java/util/concurrent/DelayQueue.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Retrieves and removes the head of this queue, waiting if necessary until an element with an expired delay is available on this queue, or the specified wait time expires. |
| [Future](http://docs.google.com/java/util/concurrent/Future.html)<[V](http://docs.google.com/java/util/concurrent/CompletionService.html)> | **CompletionService.**[**poll**](http://docs.google.com/java/util/concurrent/CompletionService.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Retrieves and removes the Future representing the next completed task, waiting if necessary up to the specified wait time if none are yet present. |
| [E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) | **BlockingDeque.**[**poll**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Retrieves and removes the head of the queue represented by this deque (in other words, the first element of this deque), waiting up to the specified wait time if necessary for an element to become available. |
| [E](http://docs.google.com/java/util/concurrent/BlockingQueue.html) | **BlockingQueue.**[**poll**](http://docs.google.com/java/util/concurrent/BlockingQueue.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Retrieves and removes the head of this queue, waiting up to the specified wait time if necessary for an element to become available. |
| [E](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html) | **ArrayBlockingQueue.**[**poll**](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html#poll(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) | **LinkedBlockingDeque.**[**pollFirst**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#pollFirst(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) | **BlockingDeque.**[**pollFirst**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#pollFirst(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Retrieves and removes the first element of this deque, waiting up to the specified wait time if necessary for an element to become available. |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) | **LinkedBlockingDeque.**[**pollLast**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#pollLast(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) | **BlockingDeque.**[**pollLast**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#pollLast(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Retrieves and removes the last element of this deque, waiting up to the specified wait time if necessary for an element to become available. |
| | <V> [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<V> | | --- | | **ScheduledThreadPoolExecutor.**[**schedule**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html#schedule(java.util.concurrent.Callable,%20long,%20java.util.concurrent.TimeUnit))([Callable](http://docs.google.com/java/util/concurrent/Callable.html)<V> callable, long delay, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| | <V> [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<V> | | --- | | **ScheduledExecutorService.**[**schedule**](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html#schedule(java.util.concurrent.Callable,%20long,%20java.util.concurrent.TimeUnit))([Callable](http://docs.google.com/java/util/concurrent/Callable.html)<V> callable, long delay, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Creates and executes a ScheduledFuture that becomes enabled after the given delay. |
| [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<?> | **ScheduledThreadPoolExecutor.**[**schedule**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html#schedule(java.lang.Runnable,%20long,%20java.util.concurrent.TimeUnit))([Runnable](http://docs.google.com/java/lang/Runnable.html) command, long delay, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<?> | **ScheduledExecutorService.**[**schedule**](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html#schedule(java.lang.Runnable,%20long,%20java.util.concurrent.TimeUnit))([Runnable](http://docs.google.com/java/lang/Runnable.html) command, long delay, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Creates and executes a one-shot action that becomes enabled after the given delay. |
| [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<?> | **ScheduledThreadPoolExecutor.**[**scheduleAtFixedRate**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html#scheduleAtFixedRate(java.lang.Runnable,%20long,%20long,%20java.util.concurrent.TimeUnit))([Runnable](http://docs.google.com/java/lang/Runnable.html) command, long initialDelay, long period, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<?> | **ScheduledExecutorService.**[**scheduleAtFixedRate**](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html#scheduleAtFixedRate(java.lang.Runnable,%20long,%20long,%20java.util.concurrent.TimeUnit))([Runnable](http://docs.google.com/java/lang/Runnable.html) command, long initialDelay, long period, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Creates and executes a periodic action that becomes enabled first after the given initial delay, and subsequently with the given period; that is executions will commence after initialDelay then initialDelay+period, then initialDelay + 2 \* period, and so on. |
| [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<?> | **ScheduledThreadPoolExecutor.**[**scheduleWithFixedDelay**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html#scheduleWithFixedDelay(java.lang.Runnable,%20long,%20long,%20java.util.concurrent.TimeUnit))([Runnable](http://docs.google.com/java/lang/Runnable.html) command, long initialDelay, long delay, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit) |
| [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<?> | **ScheduledExecutorService.**[**scheduleWithFixedDelay**](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html#scheduleWithFixedDelay(java.lang.Runnable,%20long,%20long,%20java.util.concurrent.TimeUnit))([Runnable](http://docs.google.com/java/lang/Runnable.html) command, long initialDelay, long delay, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Creates and executes a periodic action that becomes enabled first after the given initial delay, and subsequently with the given delay between the termination of one execution and the commencement of the next. |
| void | **ThreadPoolExecutor.**[**setKeepAliveTime**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#setKeepAliveTime(long,%20java.util.concurrent.TimeUnit))(long time, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Sets the time limit for which threads may remain idle before being terminated. |
| boolean | **Semaphore.**[**tryAcquire**](http://docs.google.com/java/util/concurrent/Semaphore.html#tryAcquire(int,%20long,%20java.util.concurrent.TimeUnit))(int permits, long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Acquires the given number of permits from this semaphore, if all become available within the given waiting time and the current thread has not been [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| boolean | **Semaphore.**[**tryAcquire**](http://docs.google.com/java/util/concurrent/Semaphore.html#tryAcquire(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Acquires a permit from this semaphore, if one becomes available within the given waiting time and the current thread has not been [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |

| Constructors in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) | |
| --- | --- |
| [**ThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#ThreadPoolExecutor(int,%20int,%20long,%20java.util.concurrent.TimeUnit,%20java.util.concurrent.BlockingQueue))(int corePoolSize, int maximumPoolSize, long keepAliveTime, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit, [BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<[Runnable](http://docs.google.com/java/lang/Runnable.html)> workQueue)            Creates a new ThreadPoolExecutor with the given initial parameters and default thread factory and rejected execution handler. |
| [**ThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#ThreadPoolExecutor(int,%20int,%20long,%20java.util.concurrent.TimeUnit,%20java.util.concurrent.BlockingQueue,%20java.util.concurrent.RejectedExecutionHandler))(int corePoolSize, int maximumPoolSize, long keepAliveTime, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit, [BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<[Runnable](http://docs.google.com/java/lang/Runnable.html)> workQueue, [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) handler)            Creates a new ThreadPoolExecutor with the given initial parameters and default thread factory. |
| [**ThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#ThreadPoolExecutor(int,%20int,%20long,%20java.util.concurrent.TimeUnit,%20java.util.concurrent.BlockingQueue,%20java.util.concurrent.ThreadFactory))(int corePoolSize, int maximumPoolSize, long keepAliveTime, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit, [BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<[Runnable](http://docs.google.com/java/lang/Runnable.html)> workQueue, [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            Creates a new ThreadPoolExecutor with the given initial parameters and default rejected execution handler. |
| [**ThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#ThreadPoolExecutor(int,%20int,%20long,%20java.util.concurrent.TimeUnit,%20java.util.concurrent.BlockingQueue,%20java.util.concurrent.ThreadFactory,%20java.util.concurrent.RejectedExecutionHandler))(int corePoolSize, int maximumPoolSize, long keepAliveTime, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit, [BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<[Runnable](http://docs.google.com/java/lang/Runnable.html)> workQueue, [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory, [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) handler)            Creates a new ThreadPoolExecutor with the given initial parameters. |

| Uses of [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) in [java.util.concurrent.locks](http://docs.google.com/java/util/concurrent/locks/package-summary.html) | |
| --- | --- |

| Methods in [java.util.concurrent.locks](http://docs.google.com/java/util/concurrent/locks/package-summary.html) with parameters of type [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) | |
| --- | --- |
| boolean | **AbstractQueuedLongSynchronizer.ConditionObject.**[**await**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.ConditionObject.html#await(long,%20java.util.concurrent.TimeUnit))(long time, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Implements timed condition wait. |
| boolean | **Condition.**[**await**](http://docs.google.com/java/util/concurrent/locks/Condition.html#await(long,%20java.util.concurrent.TimeUnit))(long time, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Causes the current thread to wait until it is signalled or interrupted, or the specified waiting time elapses. |
| boolean | **AbstractQueuedSynchronizer.ConditionObject.**[**await**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#await(long,%20java.util.concurrent.TimeUnit))(long time, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Implements timed condition wait. |
| boolean | **ReentrantReadWriteLock.ReadLock.**[**tryLock**](http://docs.google.com/java/util/concurrent/locks/ReentrantReadWriteLock.ReadLock.html#tryLock(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Acquires the read lock if the write lock is not held by another thread within the given waiting time and the current thread has not been [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| boolean | **ReentrantReadWriteLock.WriteLock.**[**tryLock**](http://docs.google.com/java/util/concurrent/locks/ReentrantReadWriteLock.WriteLock.html#tryLock(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Acquires the write lock if it is not held by another thread within the given waiting time and the current thread has not been [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| boolean | **Lock.**[**tryLock**](http://docs.google.com/java/util/concurrent/locks/Lock.html#tryLock(long,%20java.util.concurrent.TimeUnit))(long time, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Acquires the lock if it is free within the given waiting time and the current thread has not been [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| boolean | **ReentrantLock.**[**tryLock**](http://docs.google.com/java/util/concurrent/locks/ReentrantLock.html#tryLock(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Acquires the lock if it is not held by another thread within the given waiting time and the current thread has not been [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |

| Uses of [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) | |
| --- | --- |

| Methods in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) with parameters of type [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) | |
| --- | --- |
| [T](http://docs.google.com/javax/swing/SwingWorker.html) | **SwingWorker.**[**get**](http://docs.google.com/javax/swing/SwingWorker.html#get(long,%20java.util.concurrent.TimeUnit))(long timeout, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit)            Waits if necessary for at most the given time for the computation to complete, and then retrieves its result, if available. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/util/concurrent/TimeUnit.html) | **Use** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent//class-useTimeUnit.html)    [**NO FRAMES**](http://docs.google.com/TimeUnit.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).