| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/lang/InterruptedException.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/lang//class-useInterruptedException.html)    [**NO FRAMES**](http://docs.google.com/InterruptedException.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Class**

**java.lang.InterruptedException**

| Packages that use [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| [**java.awt**](#3znysh7) | Contains all of the classes for creating user interfaces and for painting graphics and images. |
| [**java.awt.image**](#2et92p0) | Provides classes for creating and modifying images. |
| [**java.lang**](#tyjcwt) | Provides classes that are fundamental to the design of the Java programming language. |
| [**java.lang.ref**](#3dy6vkm) | Provides reference-object classes, which support a limited degree of interaction with the garbage collector. |
| [**java.util.concurrent**](#1t3h5sf) | Utility classes commonly useful in concurrent programming. |
| [**java.util.concurrent.locks**](#4d34og8) | Interfaces and classes providing a framework for locking and waiting for conditions that is distinct from built-in synchronization and monitors. |
| [**javax.swing**](#2s8eyo1) | Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms. |

| Uses of [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) in [java.awt](http://docs.google.com/java/awt/package-summary.html) | |
| --- | --- |

| Methods in [java.awt](http://docs.google.com/java/awt/package-summary.html) that throw [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) | **EventQueue.**[**getNextEvent**](http://docs.google.com/java/awt/EventQueue.html#getNextEvent())()            Removes an event from the EventQueue and returns it. |
| static void | **EventQueue.**[**invokeAndWait**](http://docs.google.com/java/awt/EventQueue.html#invokeAndWait(java.lang.Runnable))([Runnable](http://docs.google.com/java/lang/Runnable.html) runnable)            Causes runnable to have its run method called in the dispatch thread of [the system EventQueue](http://docs.google.com/java/awt/Toolkit.html#getSystemEventQueue()). |
| void | **MediaTracker.**[**waitForAll**](http://docs.google.com/java/awt/MediaTracker.html#waitForAll())()            Starts loading all images tracked by this media tracker. |
| boolean | **MediaTracker.**[**waitForAll**](http://docs.google.com/java/awt/MediaTracker.html#waitForAll(long))(long ms)            Starts loading all images tracked by this media tracker. |
| void | **MediaTracker.**[**waitForID**](http://docs.google.com/java/awt/MediaTracker.html#waitForID(int))(int id)            Starts loading all images tracked by this media tracker with the specified identifier. |
| boolean | **MediaTracker.**[**waitForID**](http://docs.google.com/java/awt/MediaTracker.html#waitForID(int,%20long))(int id, long ms)            Starts loading all images tracked by this media tracker with the specified identifier. |

| Uses of [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) in [java.awt.image](http://docs.google.com/java/awt/image/package-summary.html) | |
| --- | --- |

| Methods in [java.awt.image](http://docs.google.com/java/awt/image/package-summary.html) that throw [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| boolean | **PixelGrabber.**[**grabPixels**](http://docs.google.com/java/awt/image/PixelGrabber.html#grabPixels())()            Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered. |
| boolean | **PixelGrabber.**[**grabPixels**](http://docs.google.com/java/awt/image/PixelGrabber.html#grabPixels(long))(long ms)            Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered or until the specified timeout has elapsed. |

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

| Methods in [java.lang](http://docs.google.com/java/lang/package-summary.html) that throw [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| void | **Thread.**[**join**](http://docs.google.com/java/lang/Thread.html#join())()            Waits for this thread to die. |
| void | **Thread.**[**join**](http://docs.google.com/java/lang/Thread.html#join(long))(long millis)            Waits at most millis milliseconds for this thread to die. |
| void | **Thread.**[**join**](http://docs.google.com/java/lang/Thread.html#join(long,%20int))(long millis, int nanos)            Waits at most millis milliseconds plus nanos nanoseconds for this thread to die. |
| static void | **Thread.**[**sleep**](http://docs.google.com/java/lang/Thread.html#sleep(long))(long millis)            Causes the currently executing thread to sleep (temporarily cease execution) for the specified number of milliseconds, subject to the precision and accuracy of system timers and schedulers. |
| static void | **Thread.**[**sleep**](http://docs.google.com/java/lang/Thread.html#sleep(long,%20int))(long millis, int nanos)            Causes the currently executing thread to sleep (cease execution) for the specified number of milliseconds plus the specified number of nanoseconds, subject to the precision and accuracy of system timers and schedulers. |
| void | **Object.**[**wait**](http://docs.google.com/java/lang/Object.html#wait())()            Causes the current thread to wait until another thread invokes the [Object.notify()](http://docs.google.com/java/lang/Object.html#notify()) method or the [Object.notifyAll()](http://docs.google.com/java/lang/Object.html#notifyAll()) method for this object. |
| void | **Object.**[**wait**](http://docs.google.com/java/lang/Object.html#wait(long))(long timeout)            Causes the current thread to wait until either another thread invokes the [Object.notify()](http://docs.google.com/java/lang/Object.html#notify()) method or the [Object.notifyAll()](http://docs.google.com/java/lang/Object.html#notifyAll()) method for this object, or a specified amount of time has elapsed. |
| void | **Object.**[**wait**](http://docs.google.com/java/lang/Object.html#wait(long,%20int))(long timeout, int nanos)            Causes the current thread to wait until another thread invokes the [Object.notify()](http://docs.google.com/java/lang/Object.html#notify()) method or the [Object.notifyAll()](http://docs.google.com/java/lang/Object.html#notifyAll()) method for this object, or some other thread interrupts the current thread, or a certain amount of real time has elapsed. |
| abstract  int | **Process.**[**waitFor**](http://docs.google.com/java/lang/Process.html#waitFor())()            causes the current thread to wait, if necessary, until the process represented by this Process object has terminated. |

| Uses of [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) in [java.lang.ref](http://docs.google.com/java/lang/ref/package-summary.html) | |
| --- | --- |

| Methods in [java.lang.ref](http://docs.google.com/java/lang/ref/package-summary.html) that throw [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| [Reference](http://docs.google.com/java/lang/ref/Reference.html)<? extends [T](http://docs.google.com/java/lang/ref/ReferenceQueue.html)> | **ReferenceQueue.**[**remove**](http://docs.google.com/java/lang/ref/ReferenceQueue.html#remove())()            Removes the next reference object in this queue, blocking until one becomes available. |
| [Reference](http://docs.google.com/java/lang/ref/Reference.html)<? extends [T](http://docs.google.com/java/lang/ref/ReferenceQueue.html)> | **ReferenceQueue.**[**remove**](http://docs.google.com/java/lang/ref/ReferenceQueue.html#remove(long))(long timeout)            Removes the next reference object in this queue, blocking until either one becomes available or the given timeout period expires. |

| Uses of [InterruptedException](http://docs.google.com/java/lang/InterruptedException.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 throw [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| void | **Semaphore.**[**acquire**](http://docs.google.com/java/util/concurrent/Semaphore.html#acquire())()            Acquires a permit from this semaphore, blocking until one is available, or the thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| void | **Semaphore.**[**acquire**](http://docs.google.com/java/util/concurrent/Semaphore.html#acquire(int))(int permits)            Acquires the given number of permits from this semaphore, blocking until all are available, or the thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| int | **CyclicBarrier.**[**await**](http://docs.google.com/java/util/concurrent/CyclicBarrier.html#await())()            Waits until all [parties](http://docs.google.com/java/util/concurrent/CyclicBarrier.html#getParties()) have invoked await on this barrier. |
| void | **CountDownLatch.**[**await**](http://docs.google.com/java/util/concurrent/CountDownLatch.html#await())()            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()). |
| 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. |
| [V](http://docs.google.com/java/util/concurrent/Exchanger.html) | **Exchanger.**[**exchange**](http://docs.google.com/java/util/concurrent/Exchanger.html#exchange(V))([V](http://docs.google.com/java/util/concurrent/Exchanger.html) x)            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())), and then transfers the given object to it, receiving its object in return. |
| [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())() |
| [V](http://docs.google.com/java/util/concurrent/Future.html) | **Future.**[**get**](http://docs.google.com/java/util/concurrent/Future.html#get())()            Waits if necessary for the computation to complete, and then retrieves its result. |
| [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. |
| | <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))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks)            Executes the given tasks, returning a list of Futures holding their status and results when all complete. |
| | <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))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks) |
| | <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))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks)            Executes the given tasks, returning the result of one that has completed successfully (i.e., without throwing an exception), if any do. |
| | <T> T | | --- | | **AbstractExecutorService.**[**invokeAny**](http://docs.google.com/java/util/concurrent/AbstractExecutorService.html#invokeAny(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [Callable](http://docs.google.com/java/util/concurrent/Callable.html)<T>> tasks) |
| | <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 | **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 | **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. |
| void | **SynchronousQueue.**[**put**](http://docs.google.com/java/util/concurrent/SynchronousQueue.html#put(E))([E](http://docs.google.com/java/util/concurrent/SynchronousQueue.html) o)            Adds the specified element to this queue, waiting if necessary for another thread to receive it. |
| void | **LinkedBlockingQueue.**[**put**](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html#put(E))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html) e)            Inserts the specified element at the tail of this queue, waiting if necessary for space to become available. |
| void | **LinkedBlockingDeque.**[**put**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#put(E))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) e) |
| void | **BlockingDeque.**[**put**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#put(E))([E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) e)            Inserts the specified element into the queue represented by this deque (in other words, at the tail of this deque), waiting if necessary for space to become available. |
| void | **BlockingQueue.**[**put**](http://docs.google.com/java/util/concurrent/BlockingQueue.html#put(E))([E](http://docs.google.com/java/util/concurrent/BlockingQueue.html) e)            Inserts the specified element into this queue, waiting if necessary for space to become available. |
| void | **ArrayBlockingQueue.**[**put**](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html#put(E))([E](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html) e)            Inserts the specified element at the tail of this queue, waiting for space to become available if the queue is full. |
| void | **LinkedBlockingDeque.**[**putFirst**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#putFirst(E))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) e) |
| void | **BlockingDeque.**[**putFirst**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#putFirst(E))([E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) e)            Inserts the specified element at the front of this deque, waiting if necessary for space to become available. |
| void | **LinkedBlockingDeque.**[**putLast**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#putLast(E))([E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) e) |
| void | **BlockingDeque.**[**putLast**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#putLast(E))([E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) e)            Inserts the specified element at the end of this deque, waiting if necessary for space to become available. |
| void | **TimeUnit.**[**sleep**](http://docs.google.com/java/util/concurrent/TimeUnit.html#sleep(long))(long timeout)            Performs a Thread.sleep using this unit. |
| [E](http://docs.google.com/java/util/concurrent/SynchronousQueue.html) | **SynchronousQueue.**[**take**](http://docs.google.com/java/util/concurrent/SynchronousQueue.html#take())()            Retrieves and removes the head of this queue, waiting if necessary for another thread to insert it. |
| [E](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html) | **PriorityBlockingQueue.**[**take**](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html#take())() |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html) | **LinkedBlockingQueue.**[**take**](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html#take())() |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) | **LinkedBlockingDeque.**[**take**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#take())() |
| [Future](http://docs.google.com/java/util/concurrent/Future.html)<[V](http://docs.google.com/java/util/concurrent/ExecutorCompletionService.html)> | **ExecutorCompletionService.**[**take**](http://docs.google.com/java/util/concurrent/ExecutorCompletionService.html#take())() |
| [E](http://docs.google.com/java/util/concurrent/DelayQueue.html) | **DelayQueue.**[**take**](http://docs.google.com/java/util/concurrent/DelayQueue.html#take())()            Retrieves and removes the head of this queue, waiting if necessary until an element with an expired delay is available on this queue. |
| [Future](http://docs.google.com/java/util/concurrent/Future.html)<[V](http://docs.google.com/java/util/concurrent/CompletionService.html)> | **CompletionService.**[**take**](http://docs.google.com/java/util/concurrent/CompletionService.html#take())()            Retrieves and removes the Future representing the next completed task, waiting if none are yet present. |
| [E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) | **BlockingDeque.**[**take**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#take())()            Retrieves and removes the head of the queue represented by this deque (in other words, the first element of this deque), waiting if necessary until an element becomes available. |
| [E](http://docs.google.com/java/util/concurrent/BlockingQueue.html) | **BlockingQueue.**[**take**](http://docs.google.com/java/util/concurrent/BlockingQueue.html#take())()            Retrieves and removes the head of this queue, waiting if necessary until an element becomes available. |
| [E](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html) | **ArrayBlockingQueue.**[**take**](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html#take())() |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) | **LinkedBlockingDeque.**[**takeFirst**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#takeFirst())() |
| [E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) | **BlockingDeque.**[**takeFirst**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#takeFirst())()            Retrieves and removes the first element of this deque, waiting if necessary until an element becomes available. |
| [E](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html) | **LinkedBlockingDeque.**[**takeLast**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html#takeLast())() |
| [E](http://docs.google.com/java/util/concurrent/BlockingDeque.html) | **BlockingDeque.**[**takeLast**](http://docs.google.com/java/util/concurrent/BlockingDeque.html#takeLast())()            Retrieves and removes the last element of this deque, waiting if necessary until an element becomes available. |
| void | **TimeUnit.**[**timedJoin**](http://docs.google.com/java/util/concurrent/TimeUnit.html#timedJoin(java.lang.Thread,%20long))([Thread](http://docs.google.com/java/lang/Thread.html) thread, long timeout)            Performs a timed Thread.join using this time unit. |
| void | **TimeUnit.**[**timedWait**](http://docs.google.com/java/util/concurrent/TimeUnit.html#timedWait(java.lang.Object,%20long))([Object](http://docs.google.com/java/lang/Object.html) obj, long timeout)            Performs a timed Object.wait using this time unit. |
| 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()). |

| Uses of [InterruptedException](http://docs.google.com/java/lang/InterruptedException.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) that throw [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| void | **AbstractQueuedSynchronizer.**[**acquireInterruptibly**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.html#acquireInterruptibly(int))(int arg)            Acquires in exclusive mode, aborting if interrupted. |
| void | **AbstractQueuedLongSynchronizer.**[**acquireInterruptibly**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.html#acquireInterruptibly(long))(long arg)            Acquires in exclusive mode, aborting if interrupted. |
| void | **AbstractQueuedSynchronizer.**[**acquireSharedInterruptibly**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.html#acquireSharedInterruptibly(int))(int arg)            Acquires in shared mode, aborting if interrupted. |
| void | **AbstractQueuedLongSynchronizer.**[**acquireSharedInterruptibly**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.html#acquireSharedInterruptibly(long))(long arg)            Acquires in shared mode, aborting if interrupted. |
| void | **AbstractQueuedLongSynchronizer.ConditionObject.**[**await**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.ConditionObject.html#await())()            Implements interruptible condition wait. |
| void | **Condition.**[**await**](http://docs.google.com/java/util/concurrent/locks/Condition.html#await())()            Causes the current thread to wait until it is signalled or [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| void | **AbstractQueuedSynchronizer.ConditionObject.**[**await**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#await())()            Implements interruptible condition wait. |
| 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. |
| long | **AbstractQueuedLongSynchronizer.ConditionObject.**[**awaitNanos**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.ConditionObject.html#awaitNanos(long))(long nanosTimeout)            Implements timed condition wait. |
| long | **Condition.**[**awaitNanos**](http://docs.google.com/java/util/concurrent/locks/Condition.html#awaitNanos(long))(long nanosTimeout)            Causes the current thread to wait until it is signalled or interrupted, or the specified waiting time elapses. |
| long | **AbstractQueuedSynchronizer.ConditionObject.**[**awaitNanos**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#awaitNanos(long))(long nanosTimeout)            Implements timed condition wait. |
| boolean | **AbstractQueuedLongSynchronizer.ConditionObject.**[**awaitUntil**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.ConditionObject.html#awaitUntil(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) deadline)            Implements absolute timed condition wait. |
| boolean | **Condition.**[**awaitUntil**](http://docs.google.com/java/util/concurrent/locks/Condition.html#awaitUntil(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) deadline)            Causes the current thread to wait until it is signalled or interrupted, or the specified deadline elapses. |
| boolean | **AbstractQueuedSynchronizer.ConditionObject.**[**awaitUntil**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#awaitUntil(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) deadline)            Implements absolute timed condition wait. |
| void | **ReentrantReadWriteLock.ReadLock.**[**lockInterruptibly**](http://docs.google.com/java/util/concurrent/locks/ReentrantReadWriteLock.ReadLock.html#lockInterruptibly())()            Acquires the read lock unless the current thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| void | **ReentrantReadWriteLock.WriteLock.**[**lockInterruptibly**](http://docs.google.com/java/util/concurrent/locks/ReentrantReadWriteLock.WriteLock.html#lockInterruptibly())()            Acquires the write lock unless the current thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| void | **Lock.**[**lockInterruptibly**](http://docs.google.com/java/util/concurrent/locks/Lock.html#lockInterruptibly())()            Acquires the lock unless the current thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| void | **ReentrantLock.**[**lockInterruptibly**](http://docs.google.com/java/util/concurrent/locks/ReentrantLock.html#lockInterruptibly())()            Acquires the lock unless the current thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()). |
| boolean | **AbstractQueuedSynchronizer.**[**tryAcquireNanos**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.html#tryAcquireNanos(int,%20long))(int arg, long nanosTimeout)            Attempts to acquire in exclusive mode, aborting if interrupted, and failing if the given timeout elapses. |
| boolean | **AbstractQueuedLongSynchronizer.**[**tryAcquireNanos**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.html#tryAcquireNanos(long,%20long))(long arg, long nanosTimeout)            Attempts to acquire in exclusive mode, aborting if interrupted, and failing if the given timeout elapses. |
| boolean | **AbstractQueuedSynchronizer.**[**tryAcquireSharedNanos**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedSynchronizer.html#tryAcquireSharedNanos(int,%20long))(int arg, long nanosTimeout)            Attempts to acquire in shared mode, aborting if interrupted, and failing if the given timeout elapses. |
| boolean | **AbstractQueuedLongSynchronizer.**[**tryAcquireSharedNanos**](http://docs.google.com/java/util/concurrent/locks/AbstractQueuedLongSynchronizer.html#tryAcquireSharedNanos(long,%20long))(long arg, long nanosTimeout)            Attempts to acquire in shared mode, aborting if interrupted, and failing if the given timeout elapses. |
| 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 [InterruptedException](http://docs.google.com/java/lang/InterruptedException.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) that throw [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) | |
| --- | --- |
| [T](http://docs.google.com/javax/swing/SwingWorker.html) | **SwingWorker.**[**get**](http://docs.google.com/javax/swing/SwingWorker.html#get())()            Waits if necessary for the computation to complete, and then retrieves its result. |
| [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. |
| static void | **SwingUtilities.**[**invokeAndWait**](http://docs.google.com/javax/swing/SwingUtilities.html#invokeAndWait(java.lang.Runnable))([Runnable](http://docs.google.com/java/lang/Runnable.html) doRun)            Causes doRun.run() to be executed synchronously on the AWT event dispatching thread. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/lang/InterruptedException.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/lang//class-useInterruptedException.html)    [**NO FRAMES**](http://docs.google.com/InterruptedException.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).