# Java Thread wait, notify and notifyAll Example

The Object class in java contains three final methods that allows threads to communicate about the lock status of a resource. These methods are **wait()**, **notify()**and **notifyAll()**. So today we will look into wait, notify and notifyAll in java program.

**Table of Contents**

[wait, notify and notifyAll in Java](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#wait-notify-and-notifyall-in-java)

[wait](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#wait)

[notify](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#notify)

[notifyAll](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#notifyall)

[Message](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#message)

[Waiter](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#waiter)

[Notifier](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#notifier)

[WaitNotifyTest](https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example#waitnotifytest)

**wait, notify and notifyAll in Java**

The current thread which invokes these methods on any object should have the object **monitor** else it throws **java.lang.IllegalMonitorStateException** exception.

**wait**

Object wait methods has three variance, one which waits indefinitely for any other thread to call notify or notifyAll method on the object to wake up the current thread. Other two variances puts the current thread in wait for specific amount of time before they wake up.

**notify**

notify method wakes up only one thread waiting on the object and that thread starts execution. So if there are multiple threads waiting for an object, this method will wake up only one of them. The choice of the thread to wake depends on the OS implementation of thread management.

**notifyAll**

notifyAll method wakes up all the threads waiting on the object, although which one will process first depends on the OS implementation.

These methods can be used to implement [producer consumer problem](https://www.journaldev.com/1034/java-blockingqueue-example) where consumer threads are waiting for the objects in Queue and producer threads put object in queue and notify the waiting threads.

Let’s see an example where multiple threads work on the same object and we use wait, notify and notifyAll methods.

### **Message**

A java bean class on which threads will work and call wait and notify methods.

Message.java

**Waiter**

A class that will wait for other threads to invoke notify methods to complete it’s processing. Notice that Waiter thread is owning monitor on Message object using synchronized block.

Waiter.java

**Notifier**

A class that will process on Message object and then invoke notify method to wake up threads waiting for Message object. Notice that synchronized block is used to own the monitor of Message object.

Notifier.java

### WaitNotifyTest

Test class that will create multiple threads of Waiter and Notifier and start them.

WaitNotifyTest.java

When we will invoke the above program, we will see below output but program will not complete because there are two threads waiting on Message object and notify() method has wake up only one of them, the other thread is still waiting to get notified.

waiter waiting to get notified at time:1356318734009

waiter1 waiting to get notified at time:1356318734010

All the threads are started

notifier started

waiter waiter thread got notified at time:1356318735011

waiter processed: notifier Notifier work done

If we comment the notify() call and uncomment the notifyAll() call in Notifier class, below will be the output produced.

waiter waiting to get notified at time:1356318917118

waiter1 waiting to get notified at time:1356318917118

All the threads are started

notifier started

waiter1 waiter thread got notified at time:1356318918120

waiter1 processed: notifier Notifier work done

waiter waiter thread got notified at time:1356318918120

waiter processed: notifier Notifier work done

Since notifyAll() method wake up both the Waiter threads and program completes and terminates after execution. That’s all for wait, notify and notifyAll in java.