Java ConditionObject: Key Class Methods

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

- Understand what condition variables are
- Note a human known use of condition variables
- Know what pattern they implement
- Recognize common use cases where condition variables are applied
- Recognize the structure & functionality of Java ConditionObject
- Know the key methods defined by the Java ConditionObject class
Key Methods of Java ConditionObject
Its key methods allow threads to wait & notify each other

```java
public class ConditionObject
    implements Condition,
    java.io.Serializable {

    ... ...

    /** Implement interruptible condition wait. */
    public final void await()
        throws InterruptedException
    { ... }

    /** Wakeup the longest waiting thread. */
    public final void signal()
    { ... }

    /** Wakeup all waiting threads. */
    public final void signalAll()
    { ... }

    ... ...
```
Key Methods of Java ConditionObject

• Its key methods allow threads to wait & notify each other

```java
public class ConditionObject implements Condition, java.io.Serializable {

    ... 

    /** Implement interruptible condition wait. */
    public final void await() throws InterruptedException {
        ... 
    }

    /** Wakeup the longest waiting thread. */
    public final void signal() {
        ... 
    }

    /** Wakeup all waiting threads. */
    public final void signalAll() {
        ... 
    }

    ... 
}
```

Method names are similar to Java’s built-in monitor object methods, but these Java Object final methods can’t be overridden

See lesson on “Java Built-in Monitor Objects”
Key Methods of Java ConditionObject

- Its key methods allow threads to wait & notify each other

```java
public class ConditionObject
    implements Condition,
    java.io.Serializable {

    ...  

    /** Implement interruptible condition wait. */
    public final void await()
    throws InterruptedException
    { ... }

    /** Wakeup the longest waiting thread. */
    public final void signal()
    { ... }

    /** Wakeup all waiting threads. */
    public final void signalAll()
    { ... }

    ...  

    // Methods are implemented via the AbstractQueued Synchronizer framework

    See gee.cs.oswego.edu/dl/papers/aqs.pdf
```
Key Methods of Java ConditionObject

• Its key methods allow threads to wait & notify each other
  • await() suspends the calling thread until it’s signaled (or interrupted)

```java
public class ConditionObject
        implements Condition,
        java.io.Serializable {

    ... 
    /** Implement interruptible condition wait. */
    public final void await() {
        ... 
    }

    ... 

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#await
```
Key Methods of Java ConditionObject

• Its key methods allow threads to wait & notify each other
• `await()` suspends the calling thread until it’s signaled (or interrupted)
• The thread is “parked” on the condition object’s queue

```java
public class ConditionObject implements Condition, java.io.Serializable {
    ...
    /** Implement interruptible condition wait. */
    public final void await() {
        ...
    }
    ...
}
```

See [www.docjar.com/docs/api/sun/misc/Unsafe.html#park(boolean, long)](http://www.docjar.com/docs/api/sun/misc/Unsafe.html#park(boolean, long))
Key Methods of Java ConditionObject

- Its key methods allow threads to wait & notify each other
  - `await()` suspends the calling thread until it’s signaled (or interrupted)
  - `signal()` moves the longest waiting thread from the queue for this condition object to the queue for the owning lock

```java
public class ConditionObject implements Condition, java.io.Serializable {
    ...
    /** Wakeup longest waiting thread. */
    public final void signal() {
        ...
    }
    ...
}
```

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#signal
Key Methods of Java ConditionObject

- Its key methods allow threads to wait & notify each other
  - `await()` suspends the calling thread until it’s signaled (or interrupted)
  - `signal()` moves the longest waiting thread from the queue for this condition object to the queue for the owning lock
  - `signalAll()` moves all threads from the ConditionObject’s queue to owning lock’s queue

```java
public class ConditionObject implements Condition, java.io.Serializable {
    ...
    /** Wakeup all waiting threads. */
    public final void signalAll() {
        ...
    }
    ...
}
```

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#signalAll](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html#signalAll)
Key Methods of Java ConditionObject

- Its key methods allow threads to wait & notify each other
  - `await()` suspends the calling thread until it’s signaled (or interrupted)
  - `signal()` moves the longest waiting thread from the queue for this condition object to the queue for the owning lock
  - `signalAll()` moves all threads from the ConditionObject’s queue to owning lock’s queue
  - `signalAll()` may cause the “thundering herd” problem, so use it sparingly!!

```java
public class ConditionObject
    implements Condition,
    java.io.Serializable {

    // Wakeup all waiting threads.
    public final void signalAll() {
        ...
    }
}
```

See [en.wikipedia.org/wiki/Thundering_herd_problem](en.wikipedia.org/wiki/Thundering_herd_problem)
Other Methods of Java
ConditionObject
### Other Methods of Java `ConditionObject`

- **ConditionObject** has several `await()` methods

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Method Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void</code></td>
<td><code>await()</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted</td>
</tr>
<tr>
<td><code>boolean</code></td>
<td><code>await(long time, TimeUnit unit)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted, or the specified waiting time elapses</td>
</tr>
<tr>
<td><code>long</code></td>
<td><code>awaitNanos(long nanosTimeout)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted, or the specified waiting time elapses</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>awaitUninterruptibly()</code></td>
<td>Causes the current thread to wait until it is signalled</td>
</tr>
<tr>
<td><code>boolean</code></td>
<td><code>awaitUntil(Date deadline)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted, or the specified deadline elapses</td>
</tr>
</tbody>
</table>
### Other Methods of Java `ConditionObject`

- `ConditionObject` has several `await()` methods
  - e.g., interruptible, non-interruptible, & timed operations

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void await()</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted</td>
</tr>
<tr>
<td><code>boolean await(long time, TimeUnit unit)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted, or the specified waiting time elapses</td>
</tr>
<tr>
<td><code>long awaitNanos(long nanosTimeout)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted, or the specified waiting time elapses</td>
</tr>
<tr>
<td><code>void awaitUninterruptibly()</code></td>
<td>Causes the current thread to wait until it is signalled</td>
</tr>
<tr>
<td><code>boolean awaitUntil(Date deadline)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted, or the specified deadline elapses</td>
</tr>
</tbody>
</table>
### Other Methods of Java ConditionObject

- **ConditionObject** has several `await()` methods
- e.g., interruptible, non-interruptible, & timed operations

<table>
<thead>
<tr>
<th>return type</th>
<th>method signature</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><code>await()</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>await(long time, TimeUnit unit)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted.</td>
</tr>
<tr>
<td>long</td>
<td><code>awaitNanos(long nanosTimeout)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted.</td>
</tr>
<tr>
<td>void</td>
<td><code>awaitUninterruptibly()</code></td>
<td>Causes the current thread to wait until it is signalled.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>awaitUntil(Date deadline)</code></td>
<td>Causes the current thread to wait until it is signalled or interrupted.</td>
</tr>
</tbody>
</table>

*Unlike Java’s built-in monitor object timed wait() calls, these timed await*() calls gives a sensible return value.*

See [stackoverflow.com/questions/3397722/how-to-differentiate-when-waitlong-timeout-exit-for-notify-or-timeout](https://stackoverflow.com/questions/3397722/how-to-differentiate-when-waitlong-timeout-exit-for-notify-or-timeout)
End of Java ConditionObject: Key Class Methods