Barrier Synchronization: Overview of Java Barrier Synchronizers

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 Lesson**

- Understand what barrier synchronization is & know three different ways of using barrier synchronizers
- Note a human known use of barrier synchronization
- Recognize the three types of Java barrier synchronizers

### Java Class: Phaser
- `Phaser()`
- `Phaser(int)`
- `Phaser(Phaser)`
- `Phaser(Phaser,int)`
- `register():int`
- `bulkRegister(int):int`
- `arrive():int`
- `arriveAndDeregister():int`
- `arriveAndAwaitAdvance():int`
- `awaitAdvance(int):int`
- `awaitAdvanceInterruptibly(int):int`
- `awaitAdvanceInterruptibly(int,long,TimeUnit):int`
- `forceTermination():void`
- `getPhase():int`
- `getRegisteredParties():int`
- `getArrivedParties():int`
- `getUnarrivedParties():int`
- `getParent():Phaser`
- `getRoot():Phaser`
- `isTerminated():boolean`
- `onAdvance(int,int):boolean`
- `toString()`

### Java Class: CountDownLatch
- `CountDownLatch(int)`
- `await():void`
- `await(long,TimeUnit):boolean`
- `countDown():void`

### Java Class: CyclicBarrier
- `CyclicBarrier(int,Runnable)`
- `CyclicBarrier(int)`
- `getParties():int`
- `await():int`
- `await(long,TimeUnit):int`
- `isBroken():boolean`
- `reset():void`
Learning Objectives in this Lesson

- Understand what barrier synchronization is & know three different ways of using barrier synchronizers
- Note a human known use of barrier synchronization
- Recognize the three types of Java barrier synchronizers
- Know how to categorize various type of Java barrier synchronizers

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDownLatch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Types of Java Barrier Synchronizers
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - **CountDownLatch**
    - Allows one or more threads to wait on the completion of operations in other threads

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/CountDownLatch.html](http://docs.oracle.com/javase/8/docs/api/java/util/concurrent/CountDownLatch.html)

E.g., a race can’t begin until all horses are at the starting gate
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - **CountDownLatch**
    - Allows one or more threads to wait on the completion of operations in other threads
    - Supports entry & exit barriers, but not cyclic barriers

<table>
<thead>
<tr>
<th>&lt;&lt;Java Class&gt;&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CountDownLatch</strong></td>
</tr>
<tr>
<td>CountDownLatch(int)</td>
</tr>
<tr>
<td>await():void</td>
</tr>
<tr>
<td>await(long, TimeUnit):boolean</td>
</tr>
<tr>
<td>countDown():void</td>
</tr>
<tr>
<td>getCount():long</td>
</tr>
<tr>
<td>toString()</td>
</tr>
</tbody>
</table>
Java supports 3 types of barrier synchronizers

- **CountDownLatch**
  - Allows one or more threads to wait on the completion of operations in other threads
  - Supports entry & exit barriers, but not cyclic barriers
  - The CountDownLatch API is very simple

```java
public class CountDownLatch {
    public void await()
    public boolean await(long timeout, TimeUnit unit)
    public void countDown()
    public long getCount()
    public String toString()
}
```
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - CountDownLatch
  - CyclicBarrier
    - Allows a set of threads to all wait for each other to reach a common barrier point

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CyclicBarrier.html
e.g., a team begins their work when the next car arrives on the assembly line
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - **CountDownLatch**
  - **CyclicBarrier**
    - Allows a set of threads to all wait for each other to reach a common barrier point
    - Supports entry, exit, & cyclic barriers for a fixed # of threads

```java
<<Java Class>>
G CyclicBarrier

- CyclicBarrier(int,Runnable)
- CyclicBarrier(int)
- getParties():int
- await():int
- await(long,TimeUnit):int
- isBroken():boolean
- reset():void
- getNumberOfWaiting():int
```
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - CountDownLatch
  - CyclicBarrier
    - Allows a set of threads to all wait for each other to reach a common barrier point
    - Supports entry, exit, & cyclic barriers for a fixed # of threads
  - The CyclicBarrier API is also very simple

![CyclicBarrier API](image)

![IT'S SIMPLE](image)
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - CountDownLatch
  - CyclicBarrier
  - Phaser

  - A more flexible, reusable, & dynamic barrier synchronizer that subsumes CyclicBarrier & CountDownLatch

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/Phaser.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/Phaser.html)

e.g., crews begin their work when all the team members arrive
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - CountDownLatch
  - CyclicBarrier
  - Phaser
    - A more flexible, reusable, & dynamic barrier synchronizer that subsumes CyclicBarrier & CountDownLatch
    - Supports entry, exit, & cyclic barriers for a variable # of threads
Types of Java Barrier Synchronizers

- Java supports 3 types of barrier synchronizers
  - CountDownLatch
  - CyclicBarrier
  - Phaser

  - A more flexible, reusable, & dynamic barrier synchronizer that subsumes CyclicBarrier & CountDownLatch
  - Supports entry, exit, & cyclic barriers for a variable # of threads
  - The Phaser API is more complex..
Categorizing Java Barrier Synchronizers
Java’s barrier synchronizers can be categorized in several ways. Here is a table summarizing the different types:

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Java’s barrier synchronizers can be categorized in several ways:

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDownLatch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Categorizing Java Barrier Synchronizers

Java’s barrier synchronizers can be categorized in several ways:

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Java’s barrier synchronizers can be categorized in several ways:

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDownLatch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>

A CountDownLatch can be used with a variable number of parties, but it’s uncommon.
Categorizing Java Barrier Synchronizers

- Java’s barrier synchronizers can be categorized in several ways

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Java’s barrier synchronizers can be categorized in several ways. These categories are not mutually exclusive, i.e., Phaser appears multiple times.
End of Barrier Synchronization: Overview of Java Barrier Synchronizers