Java Barrier Synchronizers: Usage Considerations

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

• Appreciate Java barrier synchronizer usage considerations
Java Barrier
Usage Considerations
Java Barrier Usage Considerations

• Java’s barrier synchronizers can be used for several purposes

See stackoverflow.com/questions/6830904/java-tutorials-explanations-of-jsr166y-phaser/6831171#6831171
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
- CountDownLatch focuses on events/actions
Java Barrier Usage Considerations

• Java’s barrier synchronizers can be used for several purposes
  • CountDownLatch focuses on events/actions, e.g.,
    • It can be used a simple on/off latch
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions, e.g.,
    - It can be used a simple on/off latch

  e.g., all video conversion Threads invoking await() block at the latch until the main Thread invokes countDown()
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions, e.g.,
    - It can be used a simple on/off latch
    - It can also be used for more sophisticated use cases
Java Barrier Usage Considerations

• Java’s barrier synchronizers can be used for several purposes
  • CountDownLatch focuses on events/actions, e.g.,
    • It can be used a simple on/off latch
  • It can also be used for more sophisticated use cases, e.g.
    • 1 thread waits until \( N \) threads have completed an action
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions, e.g.,
    - It can be used a simple on/off latch
  - It can also be used for more sophisticated use cases, e.g.
    - 1 thread waits until N threads have completed an action
    - 1 thread waits until some action has completed N times
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions, e.g.,
    - It can be used a simple on/off latch
    - It can also be used for more sophisticated use cases, e.g.
      - 1 thread waits until N threads have completed an action
      - 1 thread waits until some action has completed N times

- e.g., the main thread waits until the worker Threads are finished converting the video
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions
  - CyclicBarrier focuses on threads
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions
  - CyclicBarrier focuses on threads, e.g.
    - It enables group of threads to wait for each other to complete
      - e.g., complete the processing of one or more algorithm iterations
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions
  - CyclicBarrier focuses on threads, e.g.
    - It enables group of threads to wait for each other to complete
    - It requires a fixed # of threads
Java Barrier Usage Considerations

- Java’s barrier synchronizers can be used for several purposes
  - CountDownLatch focuses on events/actions
  - CyclicBarrier focuses on threads, e.g.
    - It enables group of threads to wait for each other to complete
    - It requires a fixed # of threads
      - This may be overly restrictive
Java Barrier Usage Considerations

- CountDownLatch is used throughout Android

```java
frameworks/base/core/java/android/app/SharedPreferencesImpl.java
frameworks/base/core/java/android/content/AsyncTaskLoader.java
frameworks/base/core/java/android/content/SyncManager.java
frameworks/base/core/java/android/inputmethodservice/IInputMethodWrapper.java
frameworks/base/core/java/android/view/inputmethod/InputMethodManager.java
frameworks/base/core/java/android/view/ViewDebug.java
frameworks/base/services/java/com/android/server/location/GpsLocationProvider.java
frameworks/base/services/java/com/android/server/NetworkManagementService.java
frameworks/ex/variablespeed/src/com/android/ex/variablespeed/VariableSpeed.java
packages/apps/Browser/src/com/android/browser/AutofillHandler.java
packages/apps/Browser/src/com/android/browser/NfcHandler.java
packages/apps/Browser/tests/src/com/android/browser/PopularUrlsTest.java
packages/apps/Contacts/src/com/android/contacts/model/AccountTypeManager.java
packages/apps/Contacts/tests/src/com/android/contacts/util/FakeAsyncTaskExecutor.java
packages/apps/Gallery/src/com/android/camera/CropImage.java
packages/apps/Gallery2/tests/src/com/android/gallery3d/data/LocalDataTest.java
packages/providers/ContactsProvider/src/com/android/providers/contacts/
    ContactsProvider2.java
```
Java Barrier Usage Considerations

• CountDownLatch is used throughout Android
Java Barrier Usage Considerations

- CyclicBarrier is not used in Android
End of Java Barrier Synchronizers: Usage Considerations