Java ConditionObject (Part 2)

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
- Know what pattern they implement
- Recognize how condition variables are often applied in practice
Learning Objectives in this Part of the Lesson

- Understand what condition variables are
- Know what pattern they implement
- Recognize how condition variables are often applied in practice
- Be aware of a human known use of condition variables
Applying Condition Variables in Practice
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly

See en.wikipedia.org/wiki/Grok
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly, e.g.
- The protocol for using CVs involves several “moving parts”

CAUTION
BE ALERT!!
MOVING PARTS
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly, e.g.
- The protocol for using CVs involves several "moving parts"
  - i.e., a condition variable & a lock
CVs are powerful, but can be hard to grok & apply correctly, e.g.

- The protocol for using CVs involves several “moving parts”

- The non-determinism of concurrency is tricky

See en.wikipedia.org/wiki/Nondeterministic_algorithm
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly, e.g.
  - The protocol for using CVs involves several “moving parts”
  - The non-determinism of concurrency is tricky
  - i.e., a loop may be needed to ensure a resource is available

See stackoverflow.com/a/38313778
Applying Condition Variables in Practice

- CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions
Applying Condition Variables in Practice

- CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions.
- CVs form the basis for higher-level synchronizers in Java.

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html](http://docs.oracle.com/javase/8/docs/api/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html)
• CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions
• CVs form the basis for higher-level synchronizers in Java, e.g.
  • Blocking queues & deques in java.util.concurrent* packages

See docs.oracle.com/javase/tutorial/collections/implementations/queue.html
CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions.

CVs form the basis for higher-level synchronizers in Java, e.g.

- Blocking queues & deques in java.util.concurrent* packages
  - e.g., ArrayBlockingQueue

Applying Condition Variables in Practice

See upcoming discussion in part 5 of “Java ConditionObject”
Applying Condition Variables in Practice

- CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions.
- CVs form the basis for higher-level synchronizers in Java, e.g.
  - Blocking queues & deques in java.util.concurrent* packages
  - Java built-in monitor objects

See upcoming lesson on “Java Built-in Monitor Objects"
Applying Condition Variables in Practice

- CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions
- CVs form the basis for higher-level synchronizers in Java, e.g.
  - Blocking queues & deques in java.util.concurrent* packages
  - Java built-in monitor objects
  - The *Monitor Object* pattern

See [www.dre.vanderbilt.edu/~schmidt/PDF/monitor.pdf](http://www.dre.vanderbilt.edu/~schmidt/PDF/monitor.pdf)
Human Known Use of Condition Variables
Human Known Uses of Condition Variables

• A human known use is a pizza delivery protocol
  • Must acquire both the pizza & the keys to deliver the pizza
End of Java ConditionObject (Part 2)