The Specific Notification Pattern: Introduction

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 the *Specific Notification* pattern
Overview of the Specific Notification Pattern
Overview of the Specific Notification Pattern

- **Context**
  - A family of threads in Java that need to cooperate by synchronizing access to shared resources in a specific way
  - e.g., FIFO, LIFO, priority, etc.
Overview of the Specific Notification Pattern

**Problem**
- Java built-in monitor objects provide apps no control over the order in which threads acquire a resource

*a.k.a., “Haphazard Notification”*
Overview of the Specific Notification Pattern

- **Problem**
  - Java built-in monitor objects provide apps no control over the order in which threads acquire a resource, e.g.
  - Selection of a thread after notify() or notifyAll()
Overview of the Specific Notification Pattern

• Problem
  • Java built-in monitor objects provide apps no control over the order in which threads acquire a resource, e.g.
    • Selection of a thread after notify() or notifyAll()
    • Scheduling of threads after notify() or notifyAll()
Overview of the Specific Notification Pattern

• Solution – Apply the **Specific Notification** pattern

See [www.dre.vanderbilt.edu/~schmidt/PDF/specific-notification.pdf](http://www.dre.vanderbilt.edu/~schmidt/PDF/specific-notification.pdf) (especially Listing 3)
Overview of the Specific Notification Pattern

• Solution – Apply the *Specific Notification* pattern
  • Provide a non-haphazard mechanism for selecting/scheduling threads
Overview of the Specific Notification Pattern

- **Solution – Apply the Specific Notification pattern**
  - Provide a non-haphazard mechanism for selecting/scheduling threads
  - Designate *exactly* which thread in a family of threads should proceed after notify() or notifyAll()
Overview of the Specific Notification Pattern

- **Solution Outline**
  - Put threads to sleep via `wait()` calls in monitor objects

Can also use ReentrantLock & ConditionObject
Overview of the Specific Notification Pattern

• Solution Outline
  • Put threads to sleep via wait() calls in monitor objects
  • One monitor object is used for each thread that must be individually notified
Overview of the Specific Notification Pattern

- **Solution Outline**
  - Put threads to sleep via `wait()` calls in monitor objects
  - A thread waiting on its monitor object is notified in a specific order
    - e.g., FIFO, LIFO, priority, etc.
End of the Specific Notification Pattern: Introduction