Java Monitor Objects:
Overview

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 monitors are & know how Java built-in monitor objects can ensure mutual exclusion & coordination between threads

1. Enter monitor object
2. Acquire lock
3. wait()
4. notifyAll()
5. Release lock
6. Leave monitor object

Critical Section

See www.artima.com/insidejvm/ed2/threadsynch.html
Learning Objectives in this Part of the Lesson

- Understand what monitors are & know how Java built-in monitor objects can ensure mutual exclusion & coordination between threads
- Recognize a human known use of monitors
Overview of Monitors
Overview of Monitors

• A monitor is a synchronization mechanism designed in the early 1970s.

See en.wikipedia.org/wiki/Monitor_(synchronization)
Overview of Monitors

• A monitor provides three capabilities to concurrent programs

![Image of statues and a critical section]

- Critical Section

- T_1
- T_2
- T_3
- T_4

- A monitor provides three capabilities to concurrent programs

- Critical Section
Overview of Monitors

• A monitor provides three capabilities to concurrent programs

1. Only one thread at a time has mutually exclusive access to a critical section

See en.wikipedia.org/wiki/Critical_section
Overview of Monitors

- A monitor provides three capabilities to concurrent programs

1. Only one thread at a time has mutually exclusive access to a critical section

2. Threads running in a monitor can block awaiting certain conditions to become true
Overview of Monitors

• A monitor provides three capabilities to concurrent programs

1. Only one thread at a time has mutually exclusive access to a critical section

2. Threads running in a monitor can block awaiting certain conditions to become true

3. A thread can notify one or more threads that conditions they’re waiting on have been met
Overview of Built-in Java Monitor Objects
Overview of Java Built-in Monitor Objects

- All objects in Java can be used as built-in monitor objects, which support two types of thread synchronization.

See [en.wikipedia.org/wiki/Monitor_(synchronization)](en.wikipedia.org/wiki/Monitor_(synchronization))

#Implicit_condition_variable_monitors
Overview of Java Built-in Monitor Objects

• All objects in Java can be used as built-in monitor objects, which support two types of thread synchronization

• **Mutual exclusion** – allows concurrent access & updates to shared data without race conditions
All objects in Java can be used as built-in monitor objects, which support two types of thread synchronization:

- **Mutual exclusion** - allows concurrent access & updates to shared data without race conditions.

Overview of Java Built-in Monitor Objects:

- **A Java Monitor Object**
  - `synchronized m1()`
  - `synchronized m2()`

Java’s execution environment supports mutual exclusion via an entrance queue & synchronized methods.
Overview of Java Built-in Monitor Objects

- All objects in Java can be used as built-in monitor objects, which support two types of thread synchronization
  - **Mutual exclusion** - allows concurrent access & updates to shared data without race conditions
  - **Coordination** - Ensures computations run properly, e.g., in the right order, at the right time, under the right conditions, etc.

![Diagram showing mutual exclusion and coordination in Java monitors](image-url)
Overview of Java Built-in Monitor Objects

- All objects in Java can be used as built-in monitor objects, which support two types of thread synchronization
  - **Mutual exclusion** – allows concurrent access & updates to shared data without race conditions
  - **Coordination** – Ensures computations run properly, e.g., in the right order, at the right time, under the right conditions, etc.

Java’s execution environment supports coordination via a wait queue & notification mechanisms
Overview of Java Built-in Monitor Objects

- These mechanisms implement a variant of the Monitor Object pattern

[Diagram showing the interaction between Threads 1 and 2 with Monitor Object, Monitor Lock, and Monitor Condition]
Overview of Java Built-in Monitor Objects

• These mechanisms implement a variant of the *Monitor Object* pattern

• **Intent** – Ensure that only one method runs within an object & allow an object’s methods to cooperatively schedule their execution sequences

![Diagram showing the interaction between threads and a monitor object using synchronized methods, monitor lock, monitor condition, wait(), notify(), and notifyAll() methods.](image-url)
Human Known Use of Monitors
Human Know Use of Monitors

- A human known use of a monitor is an operating room in a hospital
End of Java Monitor
Objects: Overview