Overview of How Concurrent Programs are Developed in Java (Part 2)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

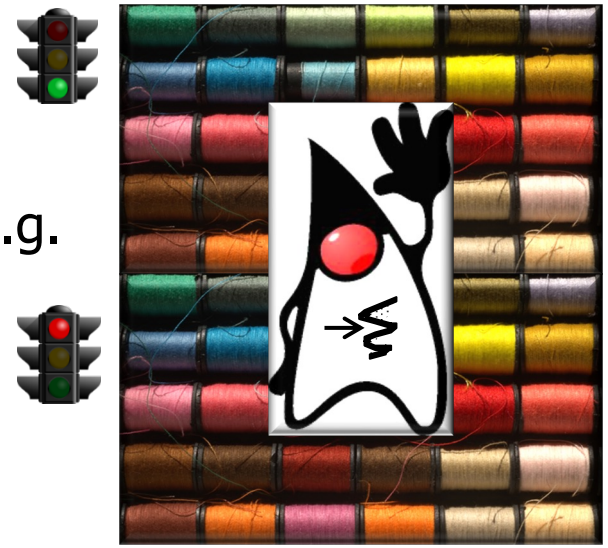
Professor of Computer Science

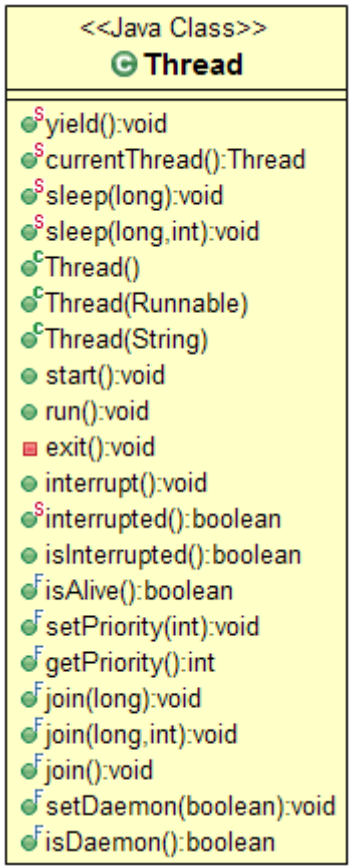
Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand the meaning of key concurrent programming concepts
- Recognize how Java supports concurrent programming concepts, e.g.
  - Thread objects
  - Interaction mechanisms
    - i.e., shared objects & message passing

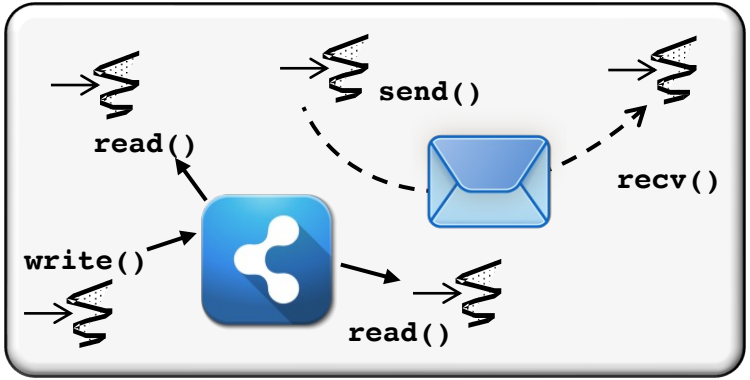




An Overview of Java Thread Interaction Mechanisms

An Overview of Java Thread Interaction Mechanisms

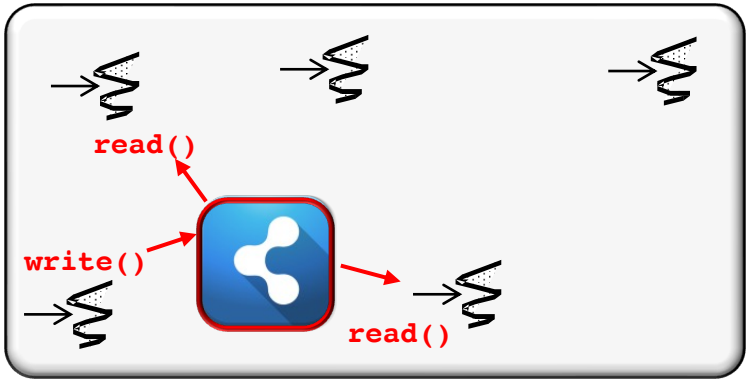
- Java threads interact via shared objects and/or message passing

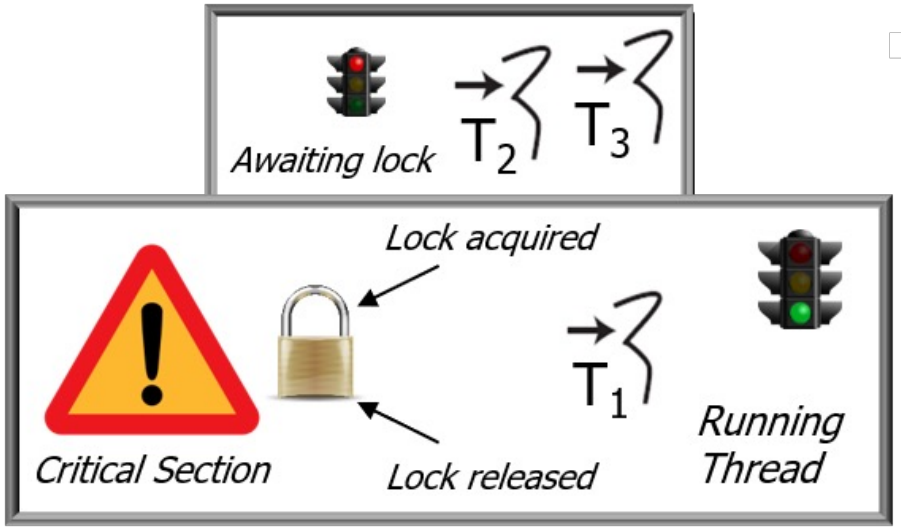


See docs.oracle.com/javase/8/docs/api/?java/util/concurrent/package-summary.html

An Overview of Java Thread Interaction Mechanisms

- Java threads interact via shared objects and/or message passing
  - **Shared objects**
    - Synchronize concurrent operations to ensure certain properties

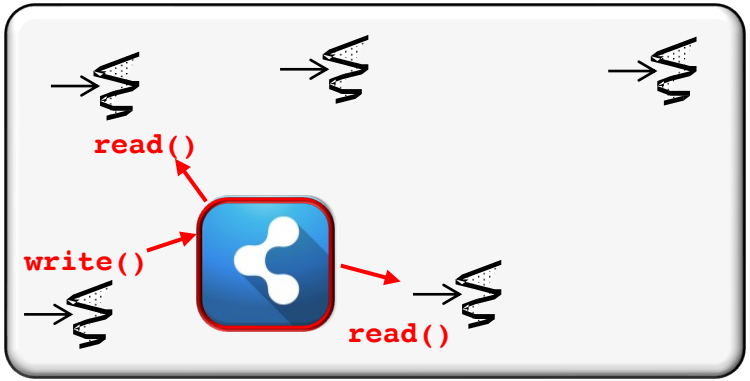




See en.wikipedia.org/wiki/Synchronization_(computer_science)

An Overview of Java Thread Interaction Mechanisms

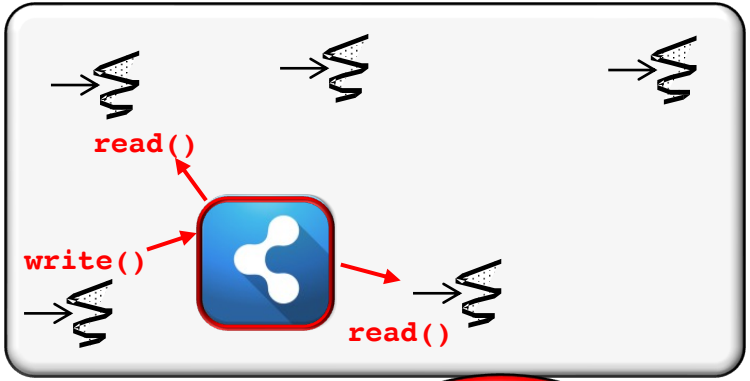
- Java threads interact via shared objects and/or message passing
  - **Shared objects**
    - Synchronize concurrent operations to ensure certain properties, e.g.
      - *Atomicity*
        - Ensures an action either happens completely or doesn't happen at all



See en.wikipedia.org/wiki/Linearizability

An Overview of Java Thread Interaction Mechanisms

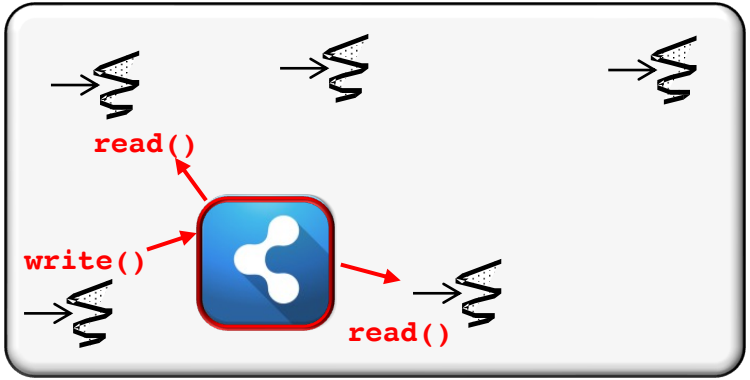
- Java threads interact via shared objects and/or message passing
  - **Shared objects**
    - Synchronize concurrent operations to ensure certain properties, e.g.
      - *Atomicity*
      - *Mutual exclusion*
        - Interactions between threads does not corrupt shared mutable data



See en.wikipedia.org/wiki/Monitor_(synchronization)#Mutual_exclusion

An Overview of Java Thread Interaction Mechanisms

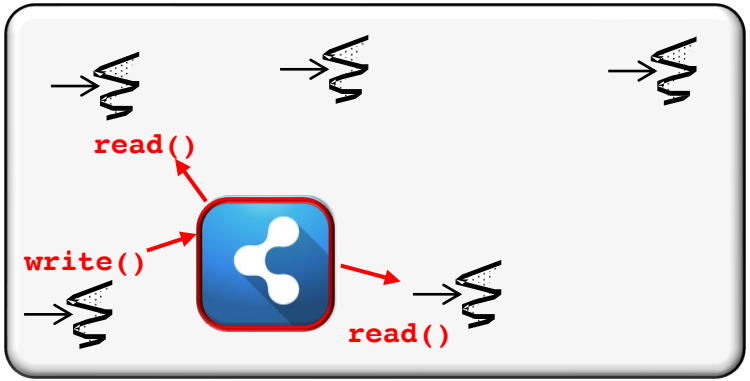
- Java threads interact via shared objects and/or message passing
  - **Shared objects**
    - Synchronize concurrent operations to ensure certain properties, e.g.
      - *Atomicity*
      - *Mutual exclusion*
      - *Coordination*
        - Operations occur in the right order, at the right time, & under the right conditions



See en.wikipedia.org/wiki/Monitor_(synchronization)#Condition_variables

An Overview of Java Thread Interaction Mechanisms

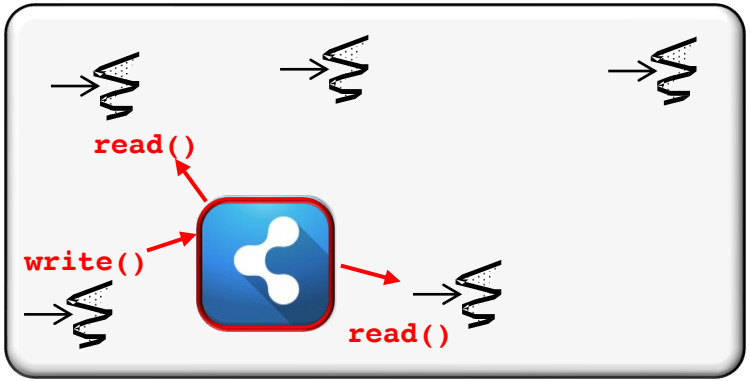
- Java threads interact via shared objects and/or message passing
  - **Shared objects**
    - Synchronize concurrent operations to ensure certain properties, e.g.
      - *Atomicity*
      - *Mutual exclusion*
      - *Coordination*
      - *Entry & exit barriers*
        - Enable a group of threads to wait for each other to reach a common execution point before proceeding



See en.wikipedia.org/wiki/Barrier_(computer_science)

An Overview of Java Thread Interaction Mechanisms

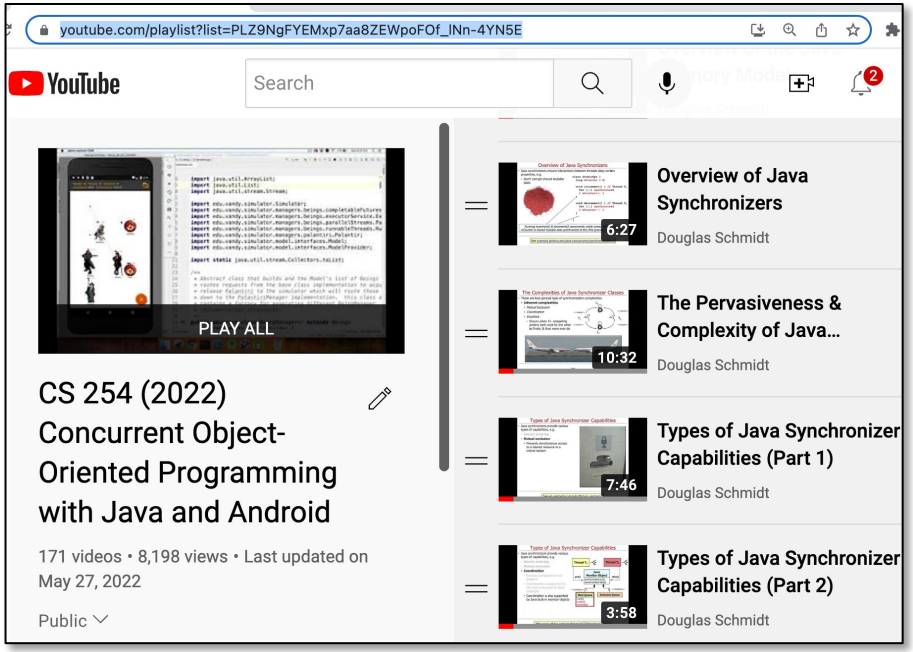
- Java threads interact via shared objects and/or message passing
  - **Shared objects**
    - Synchronize concurrent operations to ensure certain properties
    - Examples of Java synchronizers:
      - Atomic operations & volatile
      - Synchronized statements/methods
      - Reentrant & readers-writer locks
      - Semaphores
      - Condition objects
      - Barriers



See dzone.com/articles/the-java-synchronizers

An Overview of Java Thread Interaction Mechanisms

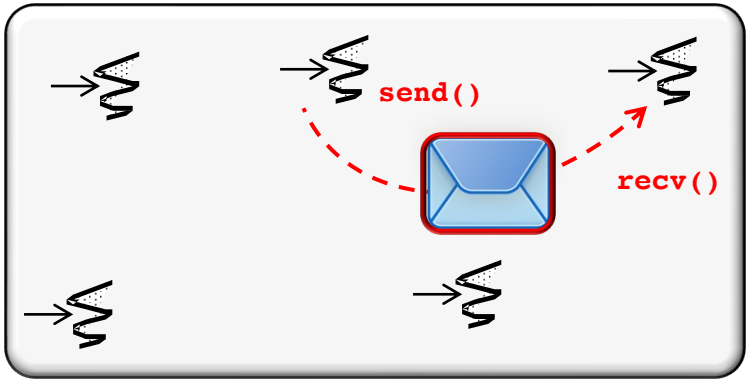
- Java threads interact via shared objects and/or message passing
  - **Shared objects**
    - Synchronize concurrent operations to ensure certain properties
    - Examples of Java synchronizers
    - Java synchronizers are covered in depth at this YouTube playlist

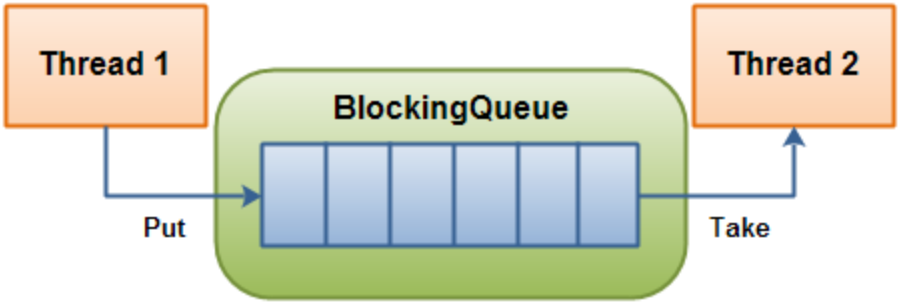


See www.youtube.com/playlist?list=PLZ9NgFYEMxp7aa8ZEWpoFOf_lNn-4YN5E

An Overview of Java Thread Interaction Mechanisms

- Java threads interact via shared objects and/or message passing
  - Shared objects
  - **Message passing**
    - Send message(s) from producer thread(s) to consumer thread(s)
      - e.g., via a blocking queue

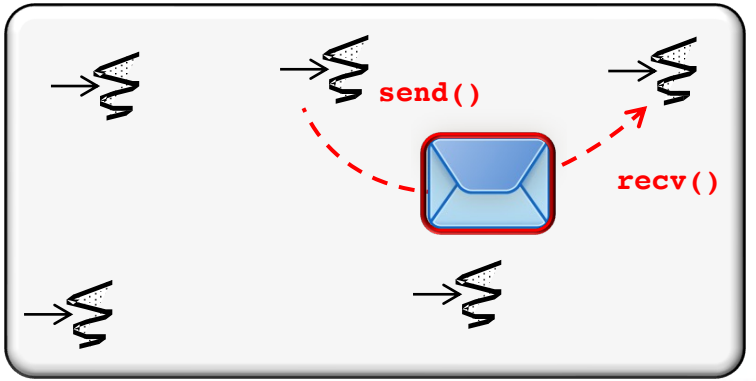




See en.wikipedia.org/wiki/Message_passing

An Overview of Java Thread Interaction Mechanisms

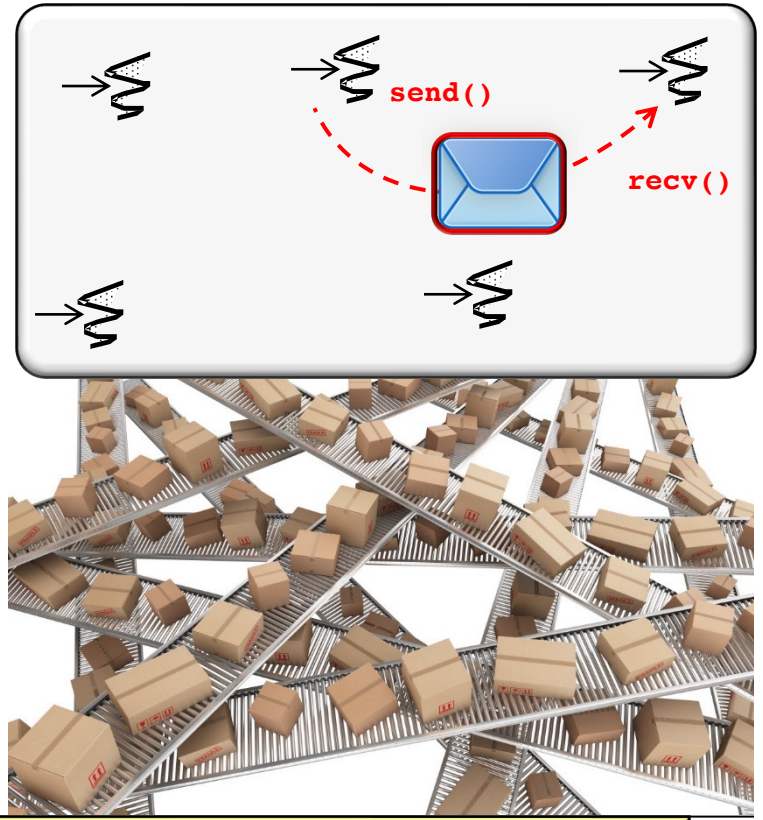
- Java threads interact via shared objects and/or message passing
  - **Shared objects**
  - **Message passing**
    - Send message(s) from producer thread(s) to consumer thread(s)
    - Decouples producer(s) & consumer(s)



See en.wikipedia.org/wiki/Message_passing

An Overview of Java Thread Interaction Mechanisms

- Java threads interact via shared objects and/or message passing
  - **Shared objects**
  - **Message passing**
    - Send message(s) from producer thread(s) to consumer thread(s)
    - Decouples producer(s) & consumer(s)
    - Examples of Java thread-safe queues
      - Array & linked blocking queues
      - Priority blocking queue
      - Synchronous queue
      - Concurrent linked queue



See docs.oracle.com/javase/tutorial/collections/implementations/queue.html

End of Overview of How Concurrent Programs are Developed in Java (Part 2)