Java “Happens-Before” Relationships:

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 what “happens-before” relationships mean in Java

Grokking these relationships is essential to mastering concurrent programming.
Overview of Java “Happens-Before” Relationships
Overview of Java “Happens-Before” Relationships

• Each action in a single thread “happens-before” every action in that thread that occurs later in the program order

See earlier lesson on “Java Volatile Variables”
Overview of Java “Happens-Before” Relationships

- In a multi-threaded program, however, actions in different threads can occur in different orders, which can cause problems without proper synchronization.

See en.wikipedia.org/wiki/Thread_safety
Overview of Java “Happens-Before” Relationships

- A “happens-before” relationship is a guarantee by a computing system that if one action “happens before” another action, the results must reflect that ordering.

See en.wikipedia.org/wiki/Happened-before

There’s a need for programs to properly “signal” the order in which certain actions need to occur.

See en.wikipedia.org/wiki/Happened-before
Overview of Java “Happens-Before” Relationships

• A “happens-before” relationship is a guarantee by a computing system that if one action “happens before” another action, the results must reflect that ordering
• Even if those actions actually execute out of order to optimize program flow & performance

Overview of Java “Happens-Before” Relationships

- A “happens-before” relationship in Java is a guarantee by the execution environment that an action performed by one thread is visible to another action in a different thread.

With proper use of synchronization Thread\(_1\) will complete its writes to answer \& ready before Thread\(_2\) starts its reads of ready \& answer

Overview of Java “Happens-Before” Relationships

• A “happens-before” relationship in Java is a guarantee by the execution environment that an action performed by one thread is visible to another action in a different thread

• Java atomic operations, variables, & classes enforce “happens-before” relationships on single entities

See earlier lessons on “Atomic Operations, Variables, & Classes”
Overview of Java “Happens-Before” Relationships

- A “happens-before” relationship in Java is a guarantee by the execution environment that an action performed by one thread is visible to another action in a different thread.
- Java atomic operations, variables, & classes enforce “happens-before” relationships on single entities.

See earlier lesson on "Java Volatile Variables"
Overview of Java “Happens-Before” Relationships

• A “happens-before” relationship in Java is a guarantee by the execution environment that an action performed by one thread is visible to another action in a different thread
  • Java atomic operations, variables, & classes enforce “happens-before” relationships on single entities
  • Our focus here is on “happens-before” relationships involving objects with multiple fields
A “happens-before” relationship in Java is a guarantee by the execution environment that an action performed by one thread is visible to another action in a different thread.

Java atomic operations, variables, & classes enforce “happens-before” relationships on single entities.

Our focus here is on “happens-before” relationships involving objects with multiple fields.

These relationships often require more elaborate synchronization mechanisms.
End of Java “Happens-Before” Relationships: Introduction