

Douglas C. Schmidt <u>d.schmidt@vanderbilt.edu</u> 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 purpose of Java synchronizers





See <a href="mailto:en-wikipedia.org/wiki/Synchronization\_(computer\_science">en-wikipedia.org/wiki/Synchronization\_(computer\_science)</a>

Java synchronizers ensure interactions between threads obey certain properties





- Java synchronizers ensure interactions between threads obey certain properties, e.g.
  - Don't corrupt shared mutable state



See henrikeichenhardt.blogspot.com/2013/06/why-shared-mutable-state-is-root-of-all.html

}

 Java synchronizers ensure interactions between threads obey certain properties, e.g.
 class NonAtomicOps {



long mCounter = 0;

void increment() {

// Thread  $T_1$ 

Running increment() & decrement() concurrently yields undefined behavior since mCounter is shared mutable data

```
void decrement() {
   // Thread T<sub>2</sub>
   for (;;) mCounter--;
}
```

for (;;) mCounter++;

See tutorials.jenkov.com/java-concurrency/race-conditions-and-critical-sections.html

long mCounter = 0;

- Java synchronizers ensure interactions between threads obey certain properties, e.g.
   class AtomicOps {
  - Don't corrupt shared mutable state



Running increment() & decrement() concurrently yields correct behavior since mCounter is shared mutable data synchronized at the (coarse-grained) method level

See tutorials.jenkov.com/java-concurrency/synchronized.html

- Java synchronizers ensure interactions between threads obey certain properties, e.g.
  - Don't corrupt shared mutable state

```
class AtomicOps {
   long mCounter = 0;
   void increment() { // Thread T<sub>1</sub>
    for (;;) synchronized
    { mCounter++; }
  }
  void decrement() { // Thread T<sub>2</sub>
   for (;;) synchronized
```

{ mCounter--; }

Running increment() & decrement() concurrently yields correct behavior since mCounter is shared mutable data synchronized at the (fine-grained) statement level

See tutorials.jenkov.com/java-concurrency/synchronized.html

- Java synchronizers ensure interactions between threads obey certain properties, e.g.
  - Don't corrupt shared mutable state
  - Occur in the right order, at the right time, & under the right conditions



 Java synchronizers ensure interactions between threads obey certain properties, e.g.
 % java PingPongWrong



See upcoming lesson on "Java Semaphore: Coordinating Threads"

 Java synchronizers ensure interactions between threads obey certain properties, e.g.
 % java PlayPing



Pervasiveness of Synchronizers in Java

 Multiple layers of synchronizers are provided on the Java platform



- Multiple layers of synchronizers are provided on the Java platform, e.g.
  - The Java language contains some features that synchronize threads

e.g., volatile variables & built-in monitor objects





See <a href="mailto:en.wikipedia.org/wiki/Java\_(programming\_language">en.wikipedia.org/wiki/Java\_(programming\_language)</a>

- Multiple layers of synchronizers are provided on the Java platform, e.g.
  - The Java language contains some features that synchronize threads
  - Other synchronizers are provided by the Java Class Library

e.g., Java atomics, locks, & other synchronizers

Description of Java Conceptual Diagram

		Java Language	Java Language										
		Tools & Tool APIs	java	javac		javadoc	jar		javap	JPDA			
			JConsole	Java Vis	ualVM	Java DB	Security		Int'i	RMI		İ.	
			IDL	Deple	oy	Monitoring	Tro	ubleshoot	Scripting	JVM	TI W	eb Services	
[	JRE	Deployment	Java Web Start				Applet / Java Plug-in						
			JavaFX										
		User Interface Toolkits	Swing			Java 2D		AWT		Accessibility			
IDK			Drag and Drop		In	Input Methods		Image I/O		Print Service		Sound	1
JUN		Integration Libraries	IDL JDBC		JNDI		RMI RMI-I		RMI-IIOP	IIOP Scripting		ripting	
		Other Base Libraries	Beans	Int'l Support		t	Input/Output			JMX			
			JNI	Math			Networking			Override Mechanism			Jav
			Security	Serialization		n Exte	Extension Mechanism			XML JAXP			
		lang and util Base Libraries	lang and util Col		Colle	ctions	Concurrency Utilities			JAR			
			Logging Mana		Manag	jement	Preferences API			Ref Objects			
			Reflectio	n Reg	pressions	Versioning			Zip Instrumentation				
	Ja	va Virtual Machine				Java HotSpo	ot Cli	ent and Se	rver VM				



See en.wikipedia.org/wiki/Java\_Class\_Library

• We focus about equally on Java synchronization mechanisms & on Java threading mechanisms in this course







#### Synchronization coverage

• Synchronization complexity arises from coordinating the interactions of entities that run concurrently





Java's parallelism frameworks helps eliminate some of this complexity via "divide and conquer"

# End of Overview of Java Synchronizers