java.util.concurrency.atomic

This one is java.util.concurrency.atomic.

A small toolkit of classes that support **lock-free**, thread-safe programming on single variables.

Why is lock-free so important?

These classes can significantly improve the performance of concurrent applications, especially in high-throughput systems.

Atomic Classes

The java.util.concurrent.atomic package, has several atomic classes as shown on this slide, including atomic arrays.

In each of these cases, an instance of one of these classes can be updated atomically.

Let me encourage you to review this toolkit, if you're working on concurrent applications.

Single Element	Array of Elements
AtomicBoolean	n/a
AtomicInteger	AtomicIntegerArray
AtomicLong	AtomicLongArray
AtomicReference	AtomicReferenceArray