

# Principles of Java Multithreaded Programming

## Table Of Contents

### Overview

- Why Java
  - Parallelism vs. Asynchronous
  - Process vs. Thread
  - Quick note about Java 8 Lambda Expressions
- 

### Java Threads

- Runnable Interface
  - Example
- Sleep, Interrupts, Join
  - Sleep
  - Interrupts
  - Example
  - Creating Interrupts
  - Join
  - Example
- Thread Pools
  - What Do Thread Pools Do?
  - The Optimal Thread Pool Size
  - Little's Law
- Executor Interface
  - Types of Executors
  - Fixed Thread Pool
  - Fork/Join Pool

### Java Memory Models

- Thread Sate
- Shared Vs. Local Variables
- Java Object Models
- One Thread / One Object
  - Example
- Multiple Threads / One Object
  - Example
- Locks and Atomic Variables in Java
- Synchronized
- Synchronized Methods

- Example
  - Memory Model
  - Synchronized Statements
    - Example
  - Memory Model
  - Example
  - Dangers of Improper Locking
  - Final Note about Synchronized
  - Volatile
  - Volatile Cons
    - Example
  - Volatile Performance v. Synchronized Performance
    - The Cost of a Volatile Cache flush
- 

## Atomic Operations

- Atomic Primitives
- Atomic Class Methods
- LongAdder, LongAccumulator

## Futures (Java 8)

- Futures Continued
- Batching Futures

## Concurrent Collections

- Common Concurrent Collections
  - Java Strings
  - Different String Alternatives
  - String Alternatives Comparison
- 

## Resources

- References
  - Thank You
- 

# Excluded

## Implementation Details of the Java Compiler

- Ordering of Events
- Happens Before
- Expectations
- Compiler Transformations (Practical Example)
  - Example
  - Transformed Example
- How Volatile can help

## **Immutability**

- Final Fields
- Final Field Compiler Optimizations
- Final Field Initialization
  - Example

## **Software Design Patterns, Tips, and Tricks**

- Thread Safe Lazy Loading
- Properties that Must hold True for a Multi-Threaded Singleton
- Double Check Locking
  - Example
  - Example
  - Example
- Multi-threaded lazy loading with Java Language Tricks