I'll create a 14-day Java mastery plan focusing on practical implementations and micro-topics, structured for 1.5 hours daily.

Day 1: Java Fundamentals Advanced
Time Split:

- 20 min: Concept review
- 40 min: Coding practice
- 30 min: Mini-project

Micro Topics:

1. Data Types Deep Dive
   - Wrapper classes internals
   - AutoBoxing/Unboxing
   - String pool mechanics
   - BigDecimal for precision
   - Type erasure understanding
2. Memory Management
   - Stack vs Heap
   - GC algorithms
   - Memory leaks
   - Soft/Weak references
   - Reference queues

Practice Project: Memory-efficient cache implementation

Day 2: Collections Framework Deep Dive
Micro Topics:

1. List Implementations
   - ArrayList internals
   - LinkedList use cases
   - Vector vs ArrayList
   - CopyOnWrite collections
   - Custom List implementation
2. Map Deep Dive
   - HashMap internals
   - TreeMap balancing
   - WeakHashMap usage

- ConcurrentHashMap features
- Custom Map implementation

Practice Project: Build a custom collection with iteration support

Day 3: Advanced Collections
Micro Topics:

1. Set Implementations
    - HashSet internals
    - TreeSet balancing
    - LinkedHashSet ordering
    - EnumSet performance
    - Custom Set implementation
2. Queue Types
    - PriorityQueue implementation
    - BlockingQueue patterns
    - Deque operations
    - Queue ordering
    - Custom Queue build

Practice Project: Priority task scheduler system

Day 4: Object-Oriented Concepts
Micro Topics:

1. Inheritance Patterns
    - Abstract classes
    - Interface evolution
    - Default methods
    - Diamond problem
    - Composition patterns
2. Polymorphism
    - Runtime vs Compile-time
    - Method overloading rules
    - Covariant returns
    - Generic type bounds
    - Bridge methods

Practice Project: Plugin system using interfaces

Day 5: Java Generics
Micro Topics:

1. Generic Types
   - Type parameters
   - Wildcards (? extends/super)
   - Type inference
   - Generic methods
   - Type erasure implications
2. Generic Constraints
   - Multiple bounds
   - Recursive type bounds
   - Raw types
   - Type capture
   - Generic constructors

Practice Project: Generic data structure implementation

Day 6: Exception Handling Advanced
Micro Topics:

1. Exception Mechanisms
   - Custom exceptions
   - try-with-resources
   - Suppressed exceptions
   - Exception chaining
   - Finally block patterns
2. Best Practices
   - Exception hierarchy
   - Logging patterns
   - Error handling strategies
   - Performance impact
   - Testing exceptions

Practice Project: Robust error handling framework

Day 7: Java I/O & NIO
Micro Topics:

1. I/O Streams

    - Buffered operations
    - Character vs Byte streams
    - File operations
    - Serialization
    - Custom serialization
2. NIO Features
    - Channels and Buffers
    - Selector patterns
    - Memory-mapped files
    - Path operations
    - Watch service

Practice Project: File processing system with monitoring

Day 8: Multithreading Basics
Micro Topics:

1. Thread Management
    - Thread lifecycle
    - Thread pools
    - Thread local
    - Daemon threads
    - Thread priorities
2. Synchronization
    - synchronized keyword
    - Volatile variables
    - Object locking
    - ReentrantLock usage
    - Condition objects

Practice Project: Thread pool implementation

Day 9: Advanced Concurrency
Micro Topics:

1. Concurrent Collections
    - ConcurrentHashMap
    - CopyOnWriteArrayList
    - BlockingQueue types
    - Concurrent Sets

   - Skip Lists
2. Synchronizers
   - CountDownLatch
   - CyclicBarrier
   - Phaser
   - Semaphore
   - Exchanger

Practice Project: Producer-consumer system

Day 10: Java Memory Model
Micro Topics:

1. Memory Architecture
   - Happens-before
   - Memory barriers
   - Cache coherence
   - False sharing
   - Memory leaks
2. Atomic Operations
   - Atomic variables
   - CAS operations
   - Memory ordering
   - Lock-free algorithms
   - ABA problem

Practice Project: Lock-free data structure

Day 11: Functional Programming
Micro Topics:

1. Lambda Expressions
   - Method references
   - Closure scope
   - Type inference
   - SAM conversion
   - Exception handling
2. Stream API
   - Stream operations
   - Parallel streams

- Custom collectors
- Spliterator
- Stream performance

Practice Project: Data processing pipeline

Day 12: Reflection & Annotations
Micro Topics:

1. Reflection API
    - Class loading
    - Dynamic proxy
    - Method invocation
    - Field access
    - Security concerns
2. Custom Annotations
    - Retention policies
    - Target types
    - Annotation processors
    - Runtime processing
    - Documentation

Practice Project: Dependency injection framework

Day 13: Modern Java Features
Micro Topics:

1. Module System
    - Module types
    - Service loading
    - Module resolution
    - Cyclic dependencies
    - Migration strategies
2. New APIs
    - Optional class
    - CompletableFuture
    - Process API
    - HTTP Client
    - Records

Practice Project: Modular application architecture

Day 14: Testing & Best Practices
Micro Topics:

1. Testing
   - JUnit features
   - Mockito advanced
   - Test containers
   - Performance testing
   - Code coverage
2. Best Practices
   - Design patterns
   - Code organization
   - Documentation
   - Logging strategies
   - Security practices

Practice Project: Complete test suite with CI setup

Daily Structure (1.5 hours):

1. 20 min: Theory and examples
2. 40 min: Hands-on coding
3. 30 min: Project implementation

Key Tools to Use:

- IDE: IntelliJ IDEA/Eclipse
- Build: Maven/Gradle
- Testing: JUnit/Mockito
- Profiling: JProfiler/VisualVM
- Version Control: Git

Would you like detailed exercises or specific project requirements for any of these days?