

## Middle-Level Java Developer Roadmap

### 1. Core Java Fundamentals

- Language Syntax & Basics
- Primitive types, literals, variables, arrays
- Operators, control flow (if/for/while/switch)
- Methods (overloading, varargs), recursion
- Object-Oriented Programming
- Classes, objects, fields, methods
- Inheritance, polymorphism, encapsulation, abstraction
- this, super, inner/nested/static classes
- Interfaces vs. abstract classes, default/static interface methods
- Memory Model & JVM
- Stack vs. heap, method area, metaspace
- Class loading, bytecode, JIT compilation
- Garbage collection algorithms (Serial, G1, CMS), tuning basics
- Exception Handling
- Checked vs. unchecked exceptions
- try/catch/finally, multi-catch, try-with-resources
- Creating custom exceptions

### 2. Collections & Data Structures

- Java Collections Framework
- List, Set, Map, Queue interfaces
- Concrete implementations: ArrayList, LinkedList, HashSet, TreeSet, HashMap, ConcurrentHashMap, etc.
- Iterators, forEach, fail-fast vs. fail-safe
- Generics
- Generic classes/methods, bounds (extends, super), wildcard types
- Algorithms & Complexity
- Sorting, searching basics, Big-O notation
- When to choose array vs. linked vs. tree vs. hash

### 3. Java 8+ Modern Features

- Lambda Expressions & Functional Interfaces
- Streams API (intermediate/final operations, parallel streams)
- Optionals, CompletableFuture
- Date & Time API (java.time)
- Module System (JPMS): module-info.java

### 4. Concurrency & Multithreading

- Threads & Runnables, thread lifecycle
- Synchronization, locks, volatile, synchronized blocks/methods
- High-level Concurrency Utilities (java.util.concurrent)
- ExecutorService, ForkJoinPool, CountdownLatch, Semaphore, CyclicBarrier
- Concurrent Collections (ConcurrentHashMap, BlockingQueue)
- Thread safety, deadlock, livelock, starvation
- Atomic variables, java.util.concurrent.atomic

### 5. I/O & Networking

- Java I/O (java.io): File, streams, readers/writers, serialization
- Java NIO (java.nio): buffers, channels, selectors, Path API
- Networking: sockets (TCP/UDP), URL & HTTP connections
- APIs: working with JSON (e.g. Jackson, Gson), XML (JAXP, JAXB)

## 6. Build & Dependency Management

- Maven: POM, lifecycle, plugins, dependency scopes, parent/child projects
- Gradle: build scripts (Groovy/Kotlin DSL), tasks, configurations
- Repositories: Maven Central, Nexus/Artifactory

## 7. Frameworks & Libraries

- Spring Ecosystem
- Core Spring (IoC, DI), Beans, Profiles
- Spring MVC / WebFlux
- Spring Boot (starters, auto-configuration, CLI)
- Spring Data JPA, JDBC, MongoDB
- Spring Security fundamentals
- Spring Cloud components (Config, Eureka, Circuit Breaker)
- Hibernate / JPA: mappings (OneToOne, OneToMany, etc.), JPQL, Criteria API
- Web Services
- RESTful APIs (Spring MVC, JAX-RS), data formats
- SOAP (JAX-WS), WSDL basics
- API documentation (Swagger/OpenAPI)
- Testing Libraries
- JUnit 5, TestNG
- Mockito, PowerMock, AssertJ
- Integration testing (Spring Test, TestContainers)

## 8. Databases & Persistence

- Relational Databases: SQL, schema design, indexing, transactions, isolation levels
- JDBC: drivers, DataSource, connection pooling
- NoSQL: basics of MongoDB, Redis, Cassandra
- Caching: Ehcache, Redis, Spring Cache abstraction

## 9. Design Principles & Patterns

- SOLID Principles
- GRASP
- Common Patterns: Singleton, Factory, Builder, Adapter, Decorator, Observer, Strategy, Proxy
- Anti-patterns (God Object, Spaghetti Code)

## 10. Quality, Monitoring & Performance

- Logging: SLF4J, Logback, Log4J2, log levels, structured logging
- Profiling & Monitoring: JVisualVM, JConsole, YourKit, Flight Recorder
- Metrics: Micrometer, Prometheus, Grafana basics
- Static Analysis & Code Quality: SonarQube, Checkstyle, PMD, SpotBugs

## 11. Security Fundamentals

- Authentication & Authorization (JWT, OAuth2, OAuth2.0, OpenID Connect)
- OWASP Top 10 awareness
- Basic Encryption: hashing, symmetric/asymmetric crypto, SSL/TLS basics
- Securing Spring Apps: CSRF, CORS, method-level security

## 12. DevOps & Deployment

- Version Control: Git workflows (feature branch, GitFlow), merging, rebasing
- CI/CD Pipelines: Jenkins, GitHub Actions, GitLab CI, Azure DevOps basics
- Containerization: Dockerfile, Docker Compose, basic Kubernetes concepts
- Cloud Platforms: AWS/GCP/Azure fundamentals (EC2, S3, RDS, IAM)

## 13. Architecture & Microservices

- Monolith vs. Microservices trade-offs
- Service Discovery, load balancing, API gateway
- Resiliency Patterns: Circuit Breaker, Bulkhead, Retry, Timeout
- Event-driven architectures, messaging (RabbitMQ, Kafka)

#### 14. Soft Skills & Practices

- Agile Methods: Scrum/Kanban rituals, user stories, estimation techniques
- Code Reviews: best practices, constructive feedback
- Collaboration Tools: Jira, Confluence, Slack
- Communication: clear problem statements, design docs, tech specs
- Time Management: task breakdown, prioritization

#### 15. Continuous Learning

- Stay Current: JDK release notes, Java Enhancement Proposals (JEPs)
- Community: participate in meetups, Stack Overflow, GitHub open-source
- Certifications (optional): Oracle OCP, Spring Professional