blankfactor

Java RoadMap and Role Description

By:
Johana Saldarriaga, *Engineer - Tech Lead*
Blankfactor Colombia
Lead of the InternShips

Johannes Correa, *Engineer - Tech Lead*
Blankfactor Colombia

Supervised by:
Melody Pak*, VP for Product Development, Blankfactor*
Sabrina Solano*, VP of Engineering, Blankfactor*
Denise D'Angelo*, SVP Transformation Programs*

Jun 1, 2023

blankfactor

JAVA RoadMap and Role Description

Skills

Junior Java Developer:

- Basic understanding of Java syntax and object-oriented programming concepts.
- Knowledge of commonly used data structures and algorithms.
- Familiarity with IDEs such as Eclipse, IntelliJ IDEA, or NetBeans.
- Understanding of version control systems such as Git.
- Ability to write and execute unit tests using testing frameworks such as JUnit.
- Basic knowledge of SQL and relational databases.
- Basic understanding of frameworks such as Spring or Hibernate.
- Understanding of basic software design patterns.
- Ability to work in a team environment and collaborate effectively.

Experienced Java Developer:

- Strong understanding of Java syntax and object-oriented programming concepts.
- Advanced knowledge of commonly used data structures and algorithms.
- Proficient with IDEs such as Eclipse, IntelliJ IDEA, or NetBeans.
- Strong understanding of version control systems such as Git.
- Proficient in writing and executing unit and integration tests using testing frameworks such as JUnit.
- Good understanding of SQL and relational databases.
- Proficient in using web development frameworks such as Spring or Hibernate.
- Proficient in software design patterns and the ability to apply them in projects.
- Ability to lead a team and mentor junior developers.

Senior Java Developer:

- Expert-level understanding of Java syntax and object-oriented programming concepts.
- Advanced knowledge of data structures and algorithms and ability to design and implement complex algorithms.
- Expert-level proficiency with IDEs such as Eclipse, IntelliJ IDEA, or NetBeans.
- Expert-level understanding of version control systems such as Git.
- Expert-level proficiency in writing and executing unit, integration and functional tests using testing frameworks such as JUnit, Mockito or related libraries as RestAssured.
- Advanced knowledge of SQL and experience with performance tuning and optimization.

- Expert-level proficiency in using web development frameworks such as Spring or Hibernate.
- Expert-level proficiency in software design patterns and ability to design and implement complex software systems.
- Ability to lead and mentor a team of developers, collaborate with other teams, and participate in software architecture discussions
- Knowledge of security best practices and common vulnerabilities in software development.
- Experience with performance tuning and optimization.
- Ability to lead large teams of developers and provide technical direction.
- Excellent communication and collaboration skills

Responsibilities

Junior Java Developer:

- Participating in software development projects and assisting senior developers in coding, testing, and debugging tasks.
- Learning and applying basic software development concepts and practices, such as object-oriented programming, design patterns, and version control.
- Developing, testing, and deploying simple applications with guidance from senior developers.
- Writing and executing unit tests to ensure application functionality and performance.
- Troubleshooting and debugging code issues as they arise.
- Collaborating with team members to ensure project goals are met on time and within budget.
- Participating in code reviews and providing feedback to other developers.
- Documenting code and contributing to project documentation.

Experienced Java Developer:

- Collaborating with project stakeholders to understand requirements and design software solutions.
- Designing and implementing software applications using best practices and appropriate software design patterns.
- Mentoring junior developers and providing guidance on software development best practices.
- Developing, testing, and deploying complex software applications, ensuring scalability, reliability, and maintainability.
- Designing and implementing database schemas and optimizing database performance.
- Writing and executing unit tests and integration tests to ensure application functionality and performance.

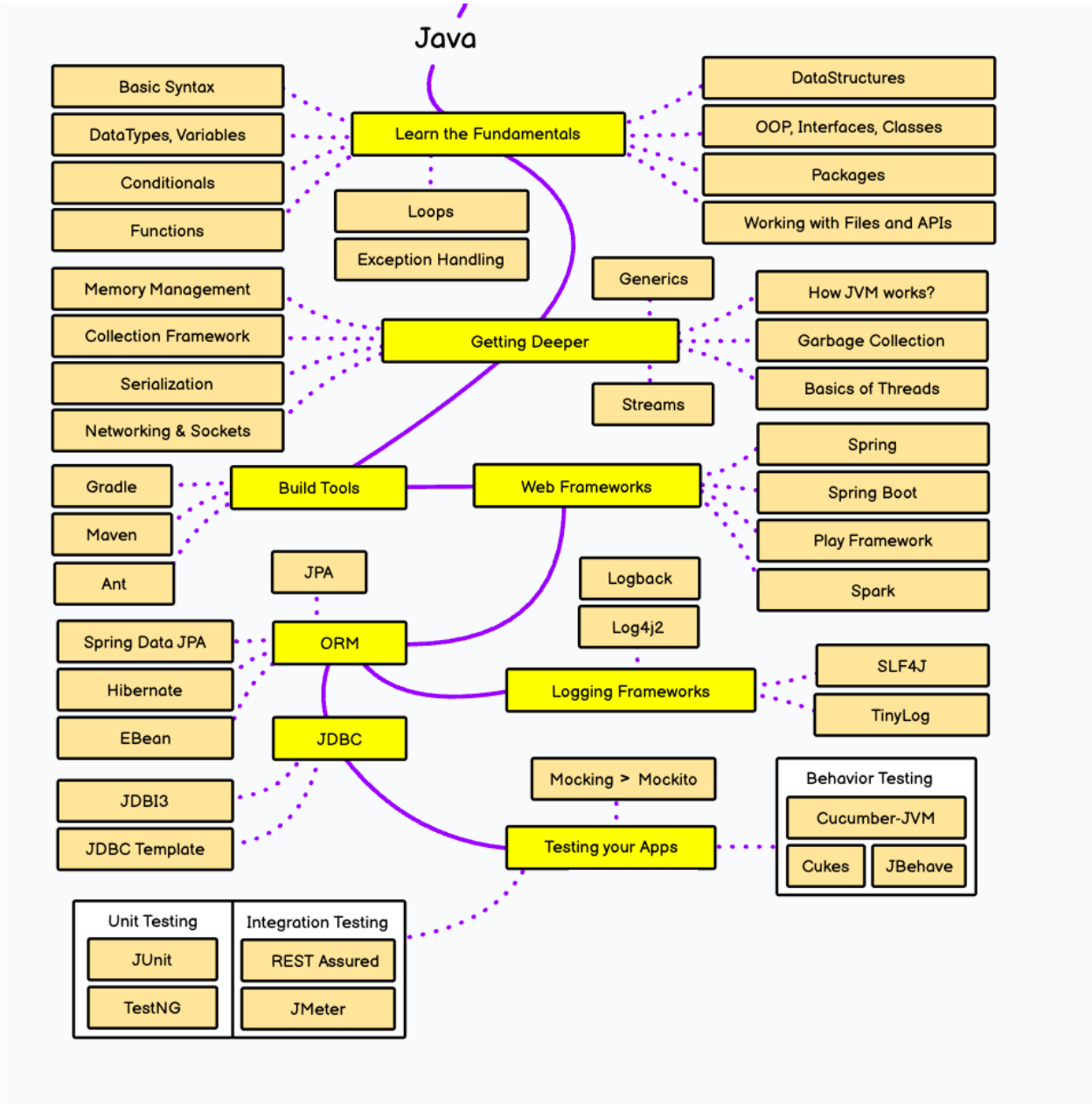
- Identifying and resolving complex software issues and optimizing application performance.
- Participating in code reviews and providing feedback to other developers.
- Documenting code and contributing to project documentation.
- Collaborating with team members to ensure project goals are met on time and within budget.

Senior Java Developer:

- Leading software development projects and collaborating with project stakeholders to understand requirements and design software solutions.
- Architecting and implementing software applications using best practices and appropriate software design patterns.
- Mentoring and leading a team of developers, providing guidance on software development best practices and ensuring code quality.
- Designing and implementing database schemas and optimizing database performance.
- Writing and executing unit tests, integration tests, and performance tests to ensure application functionality and performance.
- Experience with containerization and orchestration technologies such as Docker and Kubernetes.
- Familiarity with message-oriented middleware (MOM) systems such as RabbitMQ or Apache Kafka.
- Identifying and resolving complex software issues and optimizing application performance.
- Participating in code reviews and providing feedback to other developers.
- Documenting code and contributing to project documentation.
- Collaborating with other teams to ensure successful integration of software solutions.
- Participating in software architecture discussions and providing input on best practices and technologies.

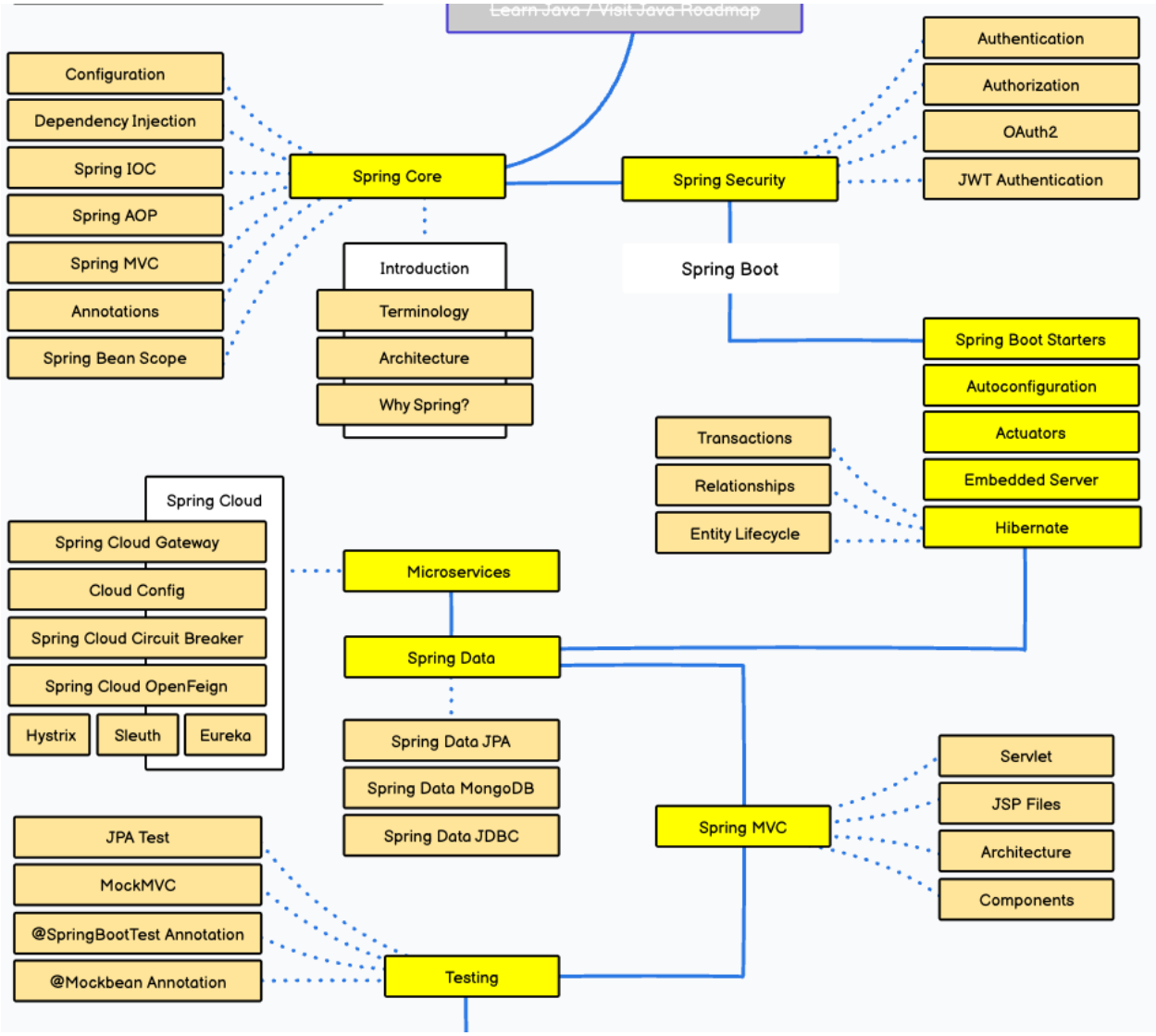
blankfactor

RoadMap



blankfactor

Spring/Spring Boot complementary roadmap



blankfactor

Resources

Junior Java Developer

- What are Data Structures?
- Data Structures and Algorithms
- Data Structures Illustrated
- OOPs Concept in Java
    - Class
    - Interfaces
    - Object
    - Inheritance
    - Polymorphism
    - Abstraction
    - Encapsulation
- Java OOPs Concepts (Article)
- Java complete OOPs playlist
- Java OOPs Concepts (Video)
- Basic Java Syntax
- Java - Basic Syntax
- Java Tutorial for Beginners (Video 1)
- Java Tutorial for Beginners. (Video 2)
- Java Basics
- Java Programming Language Tutorial
- Oracle - Learning the Java Language. This extensive tutorial covers the following topics:
    - Object-Oriented Programming Concepts
    - Language Basics
    - Classes and Objects
    - Annotations
    - Interfaces and Inheritance
    - Numbers and Strings
    - Generics
    - Packages
- Oracle - Essential Java Classes: Covers:
    - Exceptions
    - Basic I/O
    - Concurrency

- Oracle - Java Date-Time API
- Oracle - Collections Framework
- JDBC and SQL
    - Oracle - JDBC Database Access
    - Introduction to JDBC
    - IBM: What is JDBC
    - Relational Databases
    - Introduction to Relational Databases
    - SQL Tutorial
- Understanding Java Exceptions
- Java + DSA + Interview Preparation Course
- Build tools and version control systems
    - Introduction to Gradle
    - Maven Getting Started
    - What is Version Control?
    - Git
- Spring/Spring Boot basics
    - Why Spring?
    - Spring Framework Documentation
    - Spring Dependency Injection
    - Spring Configuration Tutorial
    - Intro to Inversion of Control with Spring
- Testing Foundations
    - Testing Pyramid
    - Basic JUnit 5 tutorial
    - Learn JUnit
    - Testing with JUnit crash course
    - Learn Mockito
    - Mockito - Mocking Framework for Java

Junior Challenge

Experienced Java Developer:

- Java Language: Advanced topics
    - Memory Management in Java
    - Serialization and Deserialization in Java
    - JVM (Java Virtual Machine) Architecture
    - How JVM works?

    - Java Networking
    - Java Generics
    - Java Streams Tutorials
    - Java Optional
    - Java Garbage Collection
- Build tools
    - Working with Gradle
    - Gradle Tutorial
    - Maven Guide
- Logging
    - LogBack - Official Website
    - Log4j explained: Everything you need to know
    - Java Databases: An Overview of Libraries & APIs
    - Java Logger
    - Java Logging Frameworks
    - How to Do Logging In Java
- Spring and Spring Boot
    - Spring AOP Tutorial
    - Spring MVC
    - What is Spring Framework? An Unorthodox Guide
    - Learn Spring Boot
    - Spring Boot Tutorial
    - Spring Boot for Beginners
- ORM and Persistence:
    - ORM tutorial
    - JDBC Template tutorial
    - Hibernate
    - Hibernate Tutorial
    - Introduction to Spring Data JPA

Experienced Challenge

Senior Java Developer:

- Advanced topics Persistence:
    - Pro Jakarta Persistence in Jakarta EE 10
    - Java Persistence with Spring Data and Hibernate by Catalin Tudose
    - Spring Persistence Tutorial

    - MyBatis
    - Quick guide to MyBatis
    - MyBatis with Spring
- Advanced topics - Spring and Spring Boot
    - Spring Security
    - Spring Security Tutorials
    - What is Spring Boot Actuator
    - Spring Cloud
    - Spring Cloud Tutorial
    - Introduction to Spring Data JDBC
    - Introduction to Spring Data MongoDB
    - Spring Testing
    - Integration Testing in Spring
    - Spring Boot Testing
    - Microservices with Spring
    - REST API with Spring
    - Spring Reactive Guide
    - Udemy - Master Java Web Services and REST API with Spring Boot
- Advanced topics - Testing
    - Cucumber Documentation
    - Cucumber-JVM for Java
    - Cucumber-JVM 5 with Enhanced Cucumber Expression
    - Cukes Github
    - Getting Started with Cukes-REST
    - Jbehave
    - Jbehave Tutorial
    - Rest-assured
    - Rest-assured Documentation
    - Testng Documentation
    - Testng tutorial
    - Apache JMeter Website
- Additional topics
    - Docker
    - Kubernetes
    - Domain Driven Design
    - Microservice Architecture
    - Apache Kafka
    - RabbitMQ
    - What is Play Framework?
    - Intro to Play Framework

    - Intro to Spark Java Framework
    - What is Spark java?
-

Senior challenge

Other Resources:

The following list includes some other more general resources, either transversal or that comprehends contents for all seniority levels.

- Oracle - Java Language Changes
- Udemy - Máster Completo en Java de cero a experto 2023 (Spanish)
- Udemy - Java Programming Masterclass (English)
- Udemy - Master Microservices with Java, Spring, Docker, Kubernetes