Software Developer's Knowledge Base

Papers by Aliaksei Bialiauski

Abstract:
This is a series of lectures related to mainly JVM Backend Software Development. It starts from basics like Git and UML and dives into Advanced Backend System Design and DevOps. The lectures provide basics and includes best practices for each topic.

What is the goal?
To unify knowledge and best practices for Java/JVM Software Developers in one place.

Knowledge Base Structure

- Software Design: Engineering Requirements
- Source control: Git
- Software Design: Textual documentation: Markdown, Wiki, LaTeX
- OOP
- FP
- Design Patterns
- UML
- Java
- Groovy
- Kotlin
- JVM Frameworks: Spring, Project Reactor, Ktor
- RDBMS: PostgreSQL, ORM
- NoSQL: MongoDB, DynamoDB, Cassandra
- Messaging: Apache Kafka RabbitMQ
- Software Testing: TDD, BDD, ATDD, JUnit, Testcontainers, JMeter, Mockito, PowerMock, JaCoCo, Codecov, Mutation coverage
- Dependencies, Build automation, CI/CD: Make, Maven, Gradle, GitHub Actions
- DevOps: Docker, K8s, Heroku, AWS, Terraform
- Advanced System Design
- Integration Development
- Big Data
- IoT

Learning Material

The following books are highly recommended to read (in no particular order):

Robert Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*

Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*

David Thomas et al., *The Pragmatic Programmer: Your Journey To Mastery*

Michael Feathers, *Working Effectively with Legacy Code*

Jez Humble et al., *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*

Michael T. Nygard, *Release It!: Design and Deploy Production-Ready Software*

Saurabh Shrivastava, *Solutions Architect's Handbook*

Martin Fowler, *UML Distilled*

Martin Kleppmann, *Designing Data-Intensive Applications*

Neha Narkhede, *Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale*

Yegor Bugayenko, *Code Ahead*

Project Management Institute, *A Guide to the Project Management Body of Knowledge*

Martin Fowler, *Refactoring: Improving the Design of Existing Code*

Yegor Bugayenko, *Elegant Objects, Volume 1*

Yegor Bugayenko, *Elegant Objects, Volume 2*

Mark Richards, Neal Ford, *Fundamentals of Software Architecture: An Engineering Approach*

Kent Beck, *Test-Driven Development: By Example*

Steve Freeman, Nat Pryce, *Growing Object-Oriented Software, Guided by Tests*

Eric Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*

Martin Fowler, *Patterns of Enterprise Application Architecture*