

Modul 01 - Willkommen & Einführung

Vorstellungsrunde

Erzählt uns kurz:

1. Euer Name und eure aktuelle Rolle
2. Erfahrung mit Spring Boot: Einsteiger / Fortgeschritten / Experte?
3. Hattet ihr bereits Berührungspunkte mit DDD?
4. Was ist eure größte architektonische Herausforderung im aktuellen Projekt?
5. Was erhofft ihr euch konkret von diesem Workshop?

Zeitraahmen

Beginn	09:00 Uhr
Kaffeepausen	ca. alle 90 Minuten
Mittagspause	12:00 - 13:00 Uhr
Ende	16:00 Uhr

Fragen jederzeit – bitte nicht aufsparen!

Tools & Setup

- IDE: IntelliJ IDEA (empfohlen) oder VS Code mit Java Extensions
- JDK: Java 21
- Build-Tool: Maven 3.9+
- Git-Repository: wird zu Beginn geteilt

Wir prüfen die Arbeitsumgebung direkt zu Beginn von Tag 1 gemeinsam.

Herzlich Willkommen!

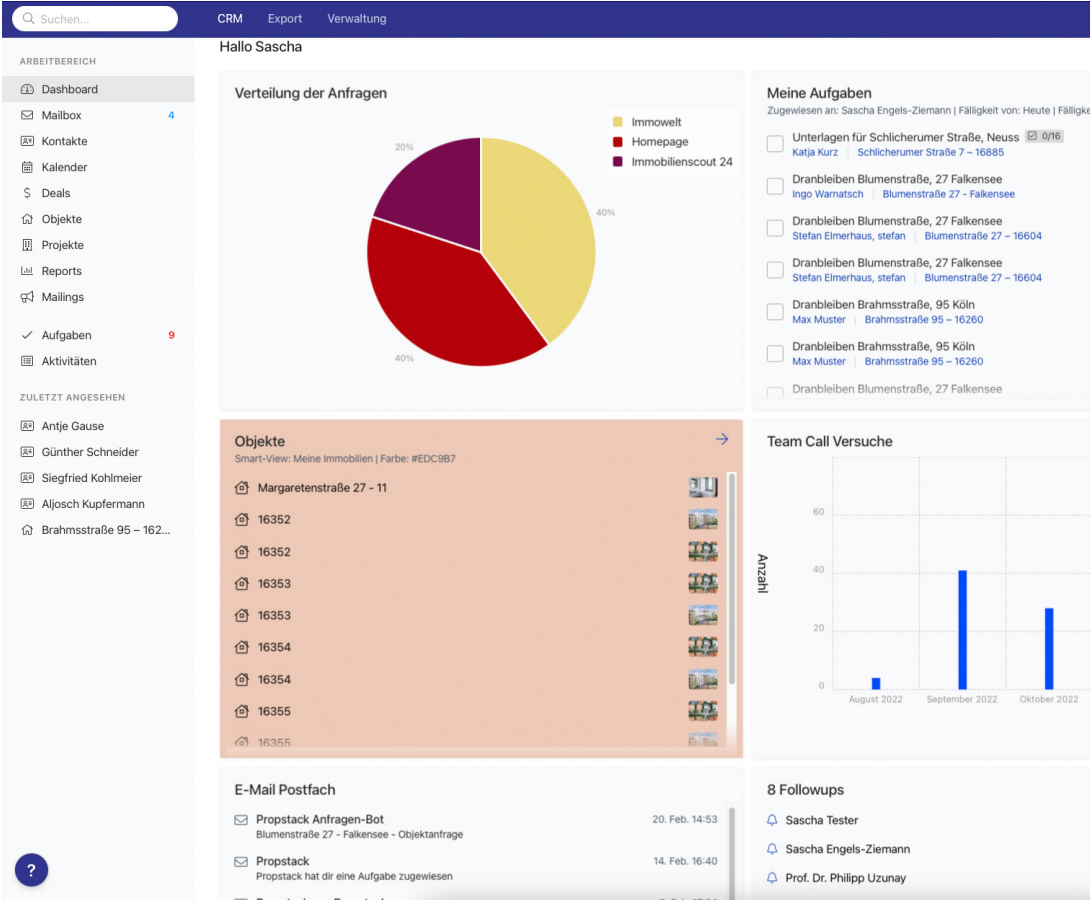
DDD & Clean Architecture mit Spring Boot 4

Ein praxisorientierter Workshop für Java-Entwickler und Software-Architekten, die ihre Spring-Boot-Projekte auf ein solides architektonisches Fundament stellen wollen.

- Durchgängige Übungsdomäne: Immobilien-CRM für Makler
- Theorie und Praxis im stetigen Wechsel

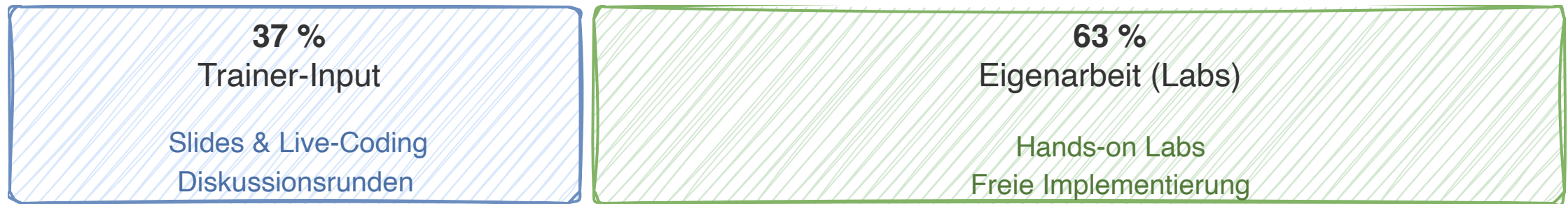
Übungsdomäne: Immobilien-CRM

Ein Maklerunternehmen benötigt ein CRM-System, das den gesamten Vermittlungsprozess abbildet – vom ersten Kontakt mit dem Eigentümer bis zum Notartermin.



Workshop-Methodik

Ausgewogener Mix aus Theorie und Praxis



Agenda - 5-Tage-Überblick

Tag	Schwerpunkt	Slides	Labs
1	Fundament legen	01 Intro · 02 Spring Boot · 03 DDD · 04 Event Storming	Lab 01-02b
2	Architektur gestalten	05 Strategic Design · 06 Building Blocks · 07 Clean Architecture	Lab 03-04
3	Implementierung starten	08 Paketstruktur · 09 Use Cases · 10 REST Adapter	Lab 05-07
4	Qualität sichern	11 ArchUnit · 12 Context Integration · 13 Modulith · 14 Querschnitt	Lab 08-11
5	Vertiefen & Reflektieren	15 Teststrategie	Lab 12-13

Lernziele des Gesamtworkshops

Nach diesen 5 Tagen könnt ihr:

- Die Prinzipien von DDD erklären und gezielt einsetzen
- Bounded Contexts identifizieren und über eine Context Map abgrenzen
- Eine Clean Architecture mit Spring Boot 4 umsetzen (Ports & Adapters)
- Building Blocks (Entity, Value Object, Aggregate) idiomatisch in Java implementieren
- Architekturregeln mit ArchUnit automatisiert durchsetzen
- Spring Modulith für modulare Monolithen nutzen

Diskussion: Eure Erwartungen

Welche konkreten Probleme in euren Projekten erhofft ihr euch durch DDD und Clean Architecture zu lösen?

- Nehmt euch 2 Minuten Zeit zum Nachdenken
- Teilt eure Gedanken in der Runde
- Wir sammeln die Punkte auf dem Whiteboard und greifen sie im Workshop auf

Lab 01 - Setup und Warmup

Ziel: Technische Basis

- Starter-Projekt aus `labs/lab-01-setup-und-warmup/initial-project` importieren
- Build und Start lokal prüfen: `mvn clean verify` und `mvn spring-boot:run`
- Danach eine einfache Immobilien-CRUD-API umsetzen
- Zusatzblöcke je nach Zeit: Geschäftsregeln, `Address` als Record, `@SoftDelete`, `ProblemDetail`, Virtual Threads