**Overview Bootcamp Refactorings**

**Rough schedule per training day**

**▶️ 09:30 – 11:00 CONTENT I**

**▶️ 11:00 – 11:15 PAUSE / BREAK**

**▶️ 11:15 – 12:45 CONTENT II**

**▶️ 12:45 – 13:45 MITTAG / LUNCH**

**▶️ 13:45 – 15:15 CONTENT III  
▶️ 15:15 – 15:30 PAUSE / BREAK**

**▶️ 15:30 – 17:00 CONTENT IV**

**DAY 1 ▶️/✅**

* **Intro**  **9:30 – 10:00**
  + Schedule
  + Rough Overview
* **PART 1: Bad Smells and Internal Quality ~1 h 10:00 – 11:00**
  + Getting to know typical problems
  + Getting a first idea of quality and design principles
* **PART 2a: Typical Smells + Basic Refactorings ~1 ½ h 11:15 – 12:00**
  + Why software rots
  + Learn about typical smells and pitfalls
  + Get to know (Martin Fowlers) basic refactorings
* **PART 2: Exercises 1 + 2 ~40 min 12:00 – 12:45**
* **PART 2b: Typical Smells + Basic Refactorings ~1 ½ h 13:45 – 14:30**
  + Learning how to combine refactorings
  + Pitfalls when performing refactorings
* **PART 2: Exercises 3 + 4 ~½ h 14:30 – 15:00**
* **PART 3: Programming Style and Coding Conventions ~½ h 15:00 – 15:15**
  + Getting to know good programming style and coding conventions applied
* **PART 4: Testing + JUnit Refresher ~1 h 15:30 – 16:30**
  + Repetition of Junit 5 basics and parameterized tests as preparation for more secure refactorings (accompanied / guided by tests)
* **PART 4: Exercises 4 – 6 ~45 min 16:30 – 17:15**

**DAY 2 ▶️/✅**

* **Recap Day 1 9:30 – 9:45**
  + What do you remember? What was most impressive?
* **PART 2: Exercise 5 ~½ h 9:45 – 10:15**
  + Recap of basic refactorings techiques to improve the design
* **PART 5a: Advanced Refactorings ~1 ½ h 10:15 – 11:00**
  + Checklist
  + Bigger refactorings explained step by step
* **PART 5b: Advanced Refactorings ~1 ½ h 11:15 – 12:00**
  + Checklist
  + Bigger refactorings explained step by step
* **PART 5: Exercises 5 ~1 h 11:45 – 12:45**
* **PART 5: Exercises 5 ~1 h 13:45 – 14:15**
* **PART 6: OO-Design-Principles (optional) ca. ½ h 14:15 – 14:45**
  + Get to know good design and common principle to get an idea of a target direction in which the refactorings should lead.

**Introduction round**

* **Name? Experience in Java? Which versions? Experience in Spring? Experience in Spring Boot? Expectations? Special Topics to cover?**
* **Matea –** PO PDU TU, 30, Master Computer Science, AI + Security, proficient Java 8 /11,
* **Hatim,** computer science, Java 7, 8, some 11 experience, worked on monolith before ☺
* **Kevin**, Java 8/11, Spring Boot Experience, REST
* **Kresimir**, Java 8/11,
* **Timo,** never worked with Spring Boot before, new to Java
* **Marin,** 32, studied, sw engineer, Java 8/11, JEE, Spring Boot, JSF,.., REST, SOAP, DB, ..
* **Alan, 34,** java 8/11, never worked with spring  
  Exp: get an overview of Spring