Training program:

Refactoring towards Clean Code

Info:

Name:	**Refactoring towards Clean Code**
Code:	**legacy-clean**
Category:	Working with legacy code
Target audience:	developers
Duration:	1 day
Format:	20% lecture, 80% ćwiczenia

Clean code defines what good code should look like and the principles for maintaining it.

Pure code consists of various architecture-related rules such as SOLID and attributes such as readability, extensibility, testability and much more.

Refactoring is nothing else but cleaning your code. But do we know how to approach it? Isn't it often a theory followed by an idea to re-write given part of code because we are afraid to touch legacy code any longer?

During the workshop we go carefully through the concept of Refactoring Pyramid that allows to clean your code in small steps:

- Clean flow Smaller methods
- Smaller classes
- Emerging design patterns
- Clean architecture

The workshop is focused on working with code using automated refactoring-features that are built into IntelliJ IDE. The source code is based on Java language. The trainer does all the refactorings live in small steps what allows the attendees to practice the same transformations on their laptops.

Remaining 25% of the training is dedicated to organizational and motivation part, how to make cleaning code a daily habit in order to keep technical debt under control.

Please note also that this is a training mainly about cleaning code – not necessary about one and the only correct design. Different teams might prefer different designs whereas the goal of this training is to show how easy we can keep the design emergent according to our architectural decisions and needs.

We start by acquiring new technical skills. Then we build awareness of our approach, which is reflected in the quality of the code, in actions to improve its quality or lack thereof. Awareness of the impact of work organization on quality is the first step to change that will allow continuous refactoring into clean code.

It's all about the content.

- One-day "detachment" of participants from daily duties
- Possibility to train another group of participants the next day
- Participants will use new skills immediately after the training
- Work with the live code together with the trainer

Training program

1. Cleaning Code in small steps

1.1. Review of Legacy Code project

1.2. It’s only one more "if" statement…

1.3. Review of automated refactorings in IntelliJ

1.4. Live refactoring in small steps

1.5. Habits of personal effectiveness for code quality

1.6. Clean Code and Pyramid of Refactoring

1.7. SOLID Principles and Pyramid of Refactoring

2. Clean Code to ease extendability and testability

2.1. Review of “Extended” Legacy Code project

2.2. New Requirements + No Cleaning = (More) Dirty Legacy Code

2.3. More Requirements are coming…

2.4. Live refactoring towards extraction of extension points

2.5. Clean code is extendable code

2.6. Habits of team effectiveness for code quality