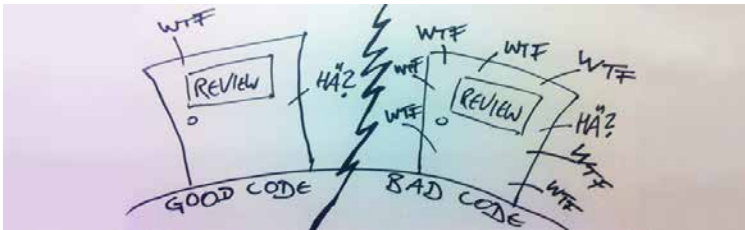


Course D10121

# Clean Code Workshop (Java)

- Write legible and comprehensible code
- Recognise bad code and refactor it step-by-step
- Numerous practical exercises



## Description

Clean code allows software to be continually adapted to new requirements. Although most developers know what clean code should look like, how to specifically recognise bad code and, step-by-step, convert it into good code is often less clear.

The course takes the form of a workshop with numerous practical exercises and involves working together in two-man teams. To facilitate transferral of what they have learnt to their everyday work, participants work through tasks in their own IDEs.

The focus is less about conveying knowledge and more about working through the topic together in order to create a culture in which code quality and collective code improvement are natural, everyday issues.

## Goals

Participants will develop a shared understanding of how to recognise bad code and how it can be restructured into better code.

They will learn to see their own code through the eyes of another developer and thus how to make it more readable and comprehensible.

## Participants

The course is aimed at Java developers who already have some practical development experience.

The course assumes that participants are thoroughly versed in the language, the IDE and standard unit testing frameworks.

## Duration

2 days

## Dates and Prices

[www.zuhlke.com/D10121](http://www.zuhlke.com/D10121)

## Level

Intermediate

## Content

- Code smells:
  - More than 25 different smells
  - Intra-class and inter-class
- Strategies and tactics for refactorisation
- Using automated IDE-mediated refactorings
- Principles of clean code:
  - Broken windows theory
  - DRY
  - KISS
  - YAGNI
  - The boy scout rule
- Clear, specific names
- Necessary and superfluous comments
- Numerous practical exercises

## Language

German

## Organisation

Public course or company course at Zühlke or at the client's premises.