

# Clean Code Training

2-tägiger Workshop



Code-Änderungen sind Alltag. Und mit Clean Code leichter.

Intuitiv verständlich ist Code dann, wenn er mit wenig Aufwand und in kurzer Zeit erfassbar und logisch klar ist, unabhängig davon, wer ihn programmiert hat. Diese Art der Verständlichkeit ist mehr als ein ehrgeiziges Ziel einiger Entwicklerinnen und Entwickler, die besonderen Wert auf handwerkliche Fertigkeiten legen. Denn „sauberer“, sprich verständlicher Code verkürzt die Entwicklungszeit dramatisch, wenn Erweiterungen anstehen oder Fehler behoben werden sollen. Und das sind zwei Situationen, die absolut zum Entwickler-Alltag gehören. Code wird weitaus häufiger gelesen als geschrieben. Schätzungsweise 20% der Lebensdauer einer Software entfallen auf die primäre Entwicklung, der Rest ist Wartung und Erweiterung.

## Das Training: Clean Code

Wenn Sie Code erzeugen möchten, der

- lesbar
- verständlich
- anpassbar und
- wartbar

ist, dann ist dieser Workshop ideal für Sie. Robert C. Martin prägte den Begriff des „Clean Code“ in seinem Standardwerk\*. Seitdem hat die Bewegung der Software Craftsmanship weltweit immer mehr Anhänger gefunden. Das Training bietet Ihnen in zwei Tagen einen kompakten Überblick über die Grundlagen und Techniken, um intuitiv verständlichen Code zu schreiben.

### Was Sie lernen



Das Training vermittelt, wie Sie

- guten von schlechtem Code unterscheiden
- guten Code schreiben
- Namen, Objekte, Funktionen und Klassen „sauber“ gestalten
- schlechten Code verbessern
- Unittests und Refactoring-Techniken einsetzen
- mit Bugs effizient umgehen

### Wer Sie sind



Der Clean Code-Workshop richtet sich besonders an Interessentinnen und Interessenten, die

- generelle Prinzipien für sauberen Code lernen möchten
- bereits Erfahrung im Programmieren mit Java haben, Kenntnisse spezieller Java-Frameworks sind nicht erforderlich
- den bewussten Umgang mit den Prinzipien und Methoden des „Clean Code“ üben möchten
- dauerhaft Zeit und Aufwand sparen wollen

\* Robert C. Martin: Clean Code, a Handbook of Agile Software Craftsmanship, 2008

# Clean Code Training

## 2-tägiger Workshop

### Das Clean Code Training

Das Training umfasst folgende Kapitel:

- Wann sind Namen aussagekräftig und wann tragen sie eher zur Verwirrung bei, sei es bei Klassen, Methoden oder Variablen?
- Klein ist relativ - wann ist eine Klasse wirklich klein?
- Mit welchen Eigenschaften werden Funktionen verständlich?
- Alles schon gehabt - warum Clean Code fordert, sich nicht zu wiederholen
- Mehr Information oder mehr Ballast - wann es sich lohnt, einen Kommentar zu schreiben
- Bring' nicht zusammen, was nicht zusammen gehört - warum Fehler-Handling und Code-Logik getrennt sein sollten
- Drei gute Dinge - wie lauten die drei Regeln des Test Driven Development?
- Refactoring - was deutet darauf hin, das ein Refactoring das Mittel der Wahl ist?
- Was bringen automatisierte Tests?
- Wenn früher doch nicht alles besser war - wie befreie ich mich aus der Komplexitätsfalle eines Legacy Systems?

Das Training kombiniert die theoretischen Prinzipien für Clean Code und einer praktischen Übung zu jedem Kapitel. Die Übungen sind alle im Technologiestacks des Trainings. Sie tragen dazu bei, die neuen Kenntnisse sofort zu vertiefen und auf ihre Anwendbarkeit hin zu testen.

Wir bieten das Clean Code Training als Inhouse-Training an und klären daher die Termine individuell mit Ihnen ab.

### Technologiestacks

Aktuell bieten wir das Clean Code Training in diesen Technologien an:



Ihre Ansprechpartner finden Sie unter:



<https://www.andrena.de/kontakt>