

## Clean Code Kochbuch

Rezepte für gutes Code-Design und  
bessere Softwarequalität

» Hier geht's  
direkt  
zum Buch

# DAS VORWORT

Code ist überall zu finden, von der Webentwicklung bis hin zu Smart Contracts, in eingebetteten Systemen, Blockchains, der Steuerungssoftware des James-Webb-Weltraumteleskops, chirurgischen Robotern und in vielen anderen Anwendungen. Software erobert tatsächlich die Welt, und wir erleben derzeit den Aufstieg professioneller KI-Tools zur Codeerzeugung. Das macht Clean Code wichtiger denn je. Da wir mit immer größeren proprietären oder Open-Source-Codebasen arbeiten, ist Clean Code der Schlüssel, um die Codebasis frisch und entwicklungsfähig zu halten.

## Für wen dieses Buch gedacht ist

Dieses Buch hilft Ihnen, häufig auftretende Probleme im Code zu identifizieren, erklärt die Konsequenzen dieser Probleme und bietet einfache, nachvollziehbare Rezepte, um sie zu vermeiden. Es unterstützt Programmierer, Code-Reviewer, Softwarearchitekten und Studierende dabei, ihre Programmierfähigkeiten zu verbessern und damit auch die bereits bestehende Software.

## Wie dieses Buch aufgebaut ist

Dieses Buch umfasst 25 Kapitel. Jedes Kapitel beginnt mit einigen Prinzipien und Grundlagen, die die Vorteile von Clean Code, seine Konsequenzen und die Nachteile aufzeigen, die bei falscher Anwendung entstehen. Das erste Kapitel behandelt die wesentliche Grundregel für Clean Code: Bilden Sie in Ihrem Design reale Entitäten 1:1 ab. Diese Regel dient als Basis, von der aus alle anderen Prinzipien abgeleitet werden können.

In jedem Kapitel finden Sie mehrere thematisch gruppierte Rezepte, die Werkzeuge und Tipps zur Änderung Ihres Codes enthalten. Alle Rezepten zielen darauf ab, positive Veränderungen und Verbesserungen an Ihrem aktuellen Code vorzunehmen. Neben den Rezepten und Beispielen werden verschiedene Designprinzipien, Heuristiken und Regeln vorgestellt. Die Rezepte enthalten Codebeispiele in verschiedenen Programmiersprachen, denn Clean Code ist keine Eigenschaft, die sich einer bestimmten Programmiersprache zuordnen lässt. Viele Bücher über Refactoring kon-

zentrieren sich auf eine einzige Sprache, und bei aktualisierten Neuauflagen beschäftigen sich die Autoren gerne mit der jeweils neuesten, angesagten Programmiersprache. Dieses Buch ist dagegen sprachagnostisch angelegt, und die meisten Rezepte gelten universell (es sei denn, es ist ausdrücklich etwas anderes angegeben).

Lesen Sie den Code als Pseudocode, auch wenn er größtenteils in der dargestellten Form lauffähig ist. Wenn ich mich zwischen Lesbarkeit und Performance entscheiden muss, wähle ich stets die Lesbarkeit. Im Laufe des Buchs definiere ich gängige Begriffe, aber Sie finden sie auch gesammelt im »Glossar«.

## Was Sie zur Verwendung dieses Buchs benötigen

Um die Codebeispiele auszuführen, benötigen Sie eine Arbeitsumgebung wie die O'Reilly-Sandboxes (<https://learning.oreilly.com/interactive>) oder Replit (<https://replit.com>). Ich möchte Sie ermutigen, die Codebeispiele in Ihre bevorzugte Programmiersprache zu übersetzen. Das lässt sich mit KI-Codegeneratoren heutzutage kostenlos erledigen. Um die Codebeispiele in diesem Buch zu erstellen, habe ich Tools wie GitHub Copilot, OpenAI Codex, Gemini, ChatGPT und viele andere verwendet. Dadurch konnte ich in diesem Buch mehr als 25 verschiedene Sprachen verwenden, auch wenn ich in vielen davon kein Experte bin.

## Zugang zur Digitalversion dieses Buchs

Eine kostenlose englischsprachige Version als stets verfügbare, durchsuchbare Onlineausgabe finden Sie unter <https://www.cleancodecookbook.com>.

## Konventionen, die in diesem Buch verwendet werden

In diesem Buch werden die folgenden typografischen Konventionen verwendet:

### *Kursiv*

Zeigt neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen an.

### **Nichtproportionalschrift**

Wird für Programmlistings verwendet, aber auch innerhalb von Absätzen, um dort auf Programmelemente wie Variablen- oder Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter zu verweisen.

### **Fette Nichtproportionalschrift**

Zeigt Befehle oder anderen Text an, der vom Benutzer wortgetreu eingegeben werden muss.

### *Kursive Nichtproportionalschrift*

Zeigt Text an, der durch Benutzereingaben oder durch kontextabhängige Werte ersetzt werden soll.



Dieses Element weist auf einen Tipp oder Vorschlag hin.



Dieses Element kennzeichnet einen allgemeinen Hinweis.



Dieses Element weist auf eine Warnung hin.

## Verwendung von Codebeispielen

Zusätzliches Material wie Codebeispiele und Übungen finden Sie zum Download unter: <https://github.com/mcsee/clean-code-cookbook>.

Bei technischen Fragen oder Problemen mit den Codebeispielen wenden Sie sich bitte per E-Mail an [bookquestions@oreilly.com](mailto:bookquestions@oreilly.com).

Dieses Buch soll Ihnen bei Ihren Aufgaben helfen. Sie dürfen den in diesem Buch bereitgestellten Beispielcode in Ihren Programmen und Ihrer Dokumentation verwenden, ohne uns um Erlaubnis bitten zu müssen, außer Sie reproduzieren einen wesentlichen Teil des Codes. Wenn Sie beispielsweise ein Programm schreiben, das mehrere Teile des Codes aus diesem Buch nutzt, benötigen Sie keine Genehmigung. Der Verkauf oder die Verbreitung von Beispielen aus O'Reilly-Büchern erfordert dagegen eine Genehmigung. Die Beantwortung einer Frage durch das Zitieren dieses Buchs und von Beispielcode ist nicht genehmigungspflichtig. Das Einbinden einer erheblichen Menge an Beispielcode aus diesem Buch in die Dokumentation Ihres Produkts erfordert dagegen eine Genehmigung.

Wir freuen uns über eine Quellenangabe, verlangen sie aber im Allgemeinen nicht. Eine Quellenangabe sollte in der Regel den Titel, den Autor, den Verlag und die ISBN umfassen. Zum Beispiel: Clean Code Kochbuch von Maximiliano Contieri (O'Reilly). Copyright 2024 Maximiliano Contieri.

Wenn Sie der Meinung sind, dass Ihre Verwendung von Codebeispielen nicht unter die Fair-Use-Regelung oder die oben genannte Erlaubnis fällt, können Sie uns bitte unter [permissions@oreilly.com](mailto:permissions@oreilly.com) kontaktieren.

# Danksagungen

Dieses Buch ist meiner Frau Maria Virginia gewidmet, die mich stets liebevoll unterstützt hat, sowie meinen geliebten Töchtern Malena und Miranda und meinen Eltern Juan Carlos und Alicia.

Ich bin Maximo Prieto und Hernan Wilkinson, die wesentlich zu den Ideen beigetragen haben, die in diesem Buch vorgestellt werden, zu großem Dank für ihre wertvollen Einsichten und ihr Fachwissen verpflichtet. Mein Dank gilt auch meinen Kollegen von Ingenieria de Software dafür, dass sie ihre Ideen mit mir geteilt haben, und meinen Kolleginnen und Kollegen an der Ciencias Exactas, der Fakultät für Exakte und Naturwissenschaften der Universidad de Buenos Aires, für ihr geteiltes Wissen und ihrer Erfahrung über viele Jahre.

Abschließend möchte ich den technischen Gutachtern Luben Alexandrov, Daniel Moka und Carlos E. Ferro sowie meiner Lektorin Sara Hunter danken, deren Betreuung und Ratschläge dieses Buch erheblich verbessert haben.