
Entwurfsprinzipien und Konstruktionskonzepte der Softwaretechnik

Joachim Goll

Entwurfsprinzipien und Konstruktionskonzepte der Softwaretechnik

Strategien für schwach gekoppelte,
korrekte und stabile Software

2., aktualisierte Auflage

 Springer Vieweg

Joachim Goll
IT-Designers Gruppe
Esslingen, Deutschland

ISBN 978-3-658-25974-7 ISBN 978-3-658-25975-4 (eBook)
<https://doi.org/10.1007/978-3-658-25975-4>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Vieweg

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2018, 2019

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Illustrationen: © Dominique Goll 2017 | www.gollinger.com. Alle Rechte vorbehalten.

Einbandabbildung: designed by eStudioCalamar © Fotolia_1639441/shutterstock_518160529

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Vorwort zur 2. Auflage

Dieses Buch beschreibt wichtige Entwurfsprinzipien und Konstruktionskonzepte für den Bau von Software. Entwurfsprinzipien sind bewährte, einfache und klare Denkkonzepte des Software Engineering, die dem Entwickler helfen, gute Softwaresysteme zu konstruieren.

Entwurfsprinzipien wurden aus dem Erfahrungsschatz der Softwareentwickler abgeleitet. Entwurfsprinzipien sind Handlungsanweisungen für die erfolgreiche Konstruktion von Systemen. Es sind abstrakte Grundsätze, die dem Handeln zugrunde gelegt und deren Konsequenzen überprüft werden können. Ein gutes Verständnis der Entwurfsprinzipien ist für den Bau einer hochwertigen Software unverzichtbar. Entwurfsprinzipien lenken den Blick eines Entwicklers in die richtige Richtung. Sie überzeugen durch ihre zwingende Logik und durch grundsätzlich positive Erfahrungen über Jahrzehnte hinweg.

Entwurfsprinzipien greifen insbesondere in die Konstruktion eines Systems ein und betreffen hierbei die für den Entwickler sichtbare Qualität des Quellcodes eines Programms. Der Schwerpunkt der meisten Entwurfsprinzipien liegt eindeutig auf der Entkopplung der Softwareteile bzw. der Abschwächung von Abhängigkeiten. Im Zuge der Clean-Code-Bewegung wird den Entwurfsprinzipien eine gesteigerte Aufmerksamkeit zuteil.

Die wesentlichen Entwurfsprinzipien und Konstruktionskonzepte zu verstehen und vor den Augen zu haben, ist Voraussetzung für das Schreiben einer guten Software. Inwieweit Entwurfsprinzipien berücksichtigt werden, hängt von der jeweiligen Projektsituation ab. So kann es sein, dass der Performance des Systems in einem Projekt eine höhere Priorität eingeräumt wird als bestimmten Entwurfsprinzipien. Verstöße gegen Entwurfsprinzipien müssen aber generell sehr sorgsam abgewogen werden, da die Nichtbeachtung von Entwurfsprinzipien zu erheblichen Mängeln der zu schreibenden Software führen kann und damit auch Qualitätsziele verfehlt werden können.

Qualitätsziele für den Entwurf sind beispielsweise

- die Entkopplung von Softwareteilen,
- Einfachheit und Verständlichkeit,
- Testbarkeit oder
- Stabilität.

Die genannten Qualitätsmerkmale betreffen die Güte der Konstruktion eines Softwaresystems und damit natürlich auch die Verantwortung der Softwareentwickler. Solche Merkmale werden innere¹ Qualitätsmerkmale genannt.

Es gibt keine allgemein anerkannten Kataloge von Entwurfsprinzipien. Daher enthält dieses Buch nur eine subjektive Auswahl, die dem Autor in der Praxis häufig begegnet ist und auch in der Clean-Code-Bewegung eine große Rolle spielt.

¹ Neben inneren gibt es auch äußere Qualitätsmerkmale wie Performance oder Bedienbarkeit, die für den Kunden direkt sichtbar und wichtig sind.

Kapitel 1 analysiert die historische Entwicklung des Konzepts von Software-Modulen. Kapitel 2 untersucht die Frage, wie Abhängigkeiten bei der Konstruktion von Systemen entstehen können. Kapitel 3 bis 9 diskutieren wichtige Entwurfsprinzipien und Konzepte für die Konstruktion schwach gekoppelter, korrekter und stabiler Software. Kapitel 3 betrachtet Entwurfsprinzipien zur modularen Struktur von Systemen, Kapitel 4 befasst sich mit Entwurfsprinzipien zur Reduktion der Komplexität, letztendlich um unnütze Abhängigkeiten zu vermeiden, Kapitel 5 mit Entwurfsprinzipien und Konzepten für die Konstruktion schwach gekoppelter Teilsysteme und Kapitel 6 mit Entwurfsprinzipien und Konzepten für korrekte Programme. Kapitel 7 untersucht Entwurfsprinzipien für die Stabilität und Erweiterbarkeit bei Programmänderungen, Kapitel 8 erörtert das Konzept "Inversion of Control" und Kapitel 9 gibt eine Übersicht über die vorgestellten Entwurfsprinzipien und Konzepte.

"Lernkästchen", auf die grafisch durch eine kleine Glühlampe (💡) aufmerksam gemacht wird, stellen eine Zusammenfassung des behandelten Inhalts in kurzer Form dar. Sie erlauben eine rasche Wiederholung des Stoffes.

"Warnkästchen", die durch ein Vorsichtssymbol (⚠️) gekennzeichnet sind, zeigen Fallen und typische, gern begangene Fehler an, die in der Praxis oft zu einer langwierigen Fehlersuche führen oder – noch schlimmer – erst im Endprodukt beim Kunden erkannt werden.

Alle Beispielprogramme des Buches sind auf dem begleitenden Webauftritt unter www.springer.com zu finden, damit Sie diese bequem selbst ausführen und nachvollziehen können. Die Errata-Liste mit Korrekturen zum Buch ist ebenfalls unter www.springer.com zu finden.

Schreibweisen

In diesem Buch sind der Quellcode sowie die Ein- und Ausgabe der Beispielprogramme in der Schriftart *Courier New* geschrieben. Dasselbe gilt für Programmteile wie Klassennamen, Namen von Operationen und Variablen etc., die im normalen Text erwähnt werden.

Wichtige Begriffe im normalen Text sind **fett** gedruckt, um sie hervorzuheben.

Danksagung

Frau Patricia Maier, Herrn Prof. Dr. Manfred Dausmann, Herrn Timo Acquistapace, Herrn Christoph Gschrey, Herrn Lukas Jaeckle, Herrn Benjamin Jester, Herrn Markus Just, Herrn Marcel Kilian, Herrn Maximilian Schall und Herrn Michael Watzko danken wir für zahlreiche wertvolle Diskussionen sowie Herrn Jordanis Andreanidis für die professionelle Durchführung des Konfigurationsmanagements.

Esslingen, im Februar 2019

Joachim Goll

Inhaltsverzeichnis

1	Software-Module	1
1.1	Modularisierung von Systemen	2
1.2	Konzepte zur Konstruktion von Modulen	5
1.3	Vorteile der Modularisierung	8
1.4	Zusammenfassung	9
2	Abhängigkeiten	11
2.1	Entstehung von Abhängigkeiten	13
2.2	Abschwächung von Abhängigkeiten	31
2.3	Zusammenfassung	33
3	Entwurfsprinzipien zur modularen Struktur von Systemen	35
3.1	Teile und Herrsche	36
3.2	Design to Test	37
3.3	Zusammenfassung	38
4	Entwurfsprinzipien zur Reduktion der Komplexität	39
4.1	Keep it simple, stupid	40
4.2	You aren't gonna need it	42
4.3	Don't repeat yourself	44
4.4	Single Level of Abstraction	47
4.5	Zusammenfassung	49
5	Entwurfsprinzipien und Konzepte für schwach gekoppelte Teilsysteme	51
5.1	Loose Coupling and Strong Cohesion	53
5.2	Information Hiding	59
5.3	Separation of Concerns	61
5.4	Law of Demeter	63
5.5	Dependency Inversion Principle	73
5.6	Interface Segregation Principle	81
5.7	Single Responsibility Principle	85
5.8	Die Konzepte "Dependency Lookup" und "Dependency Injection"	88
5.9	Zusammenfassung	103
6	Entwurfsprinzipien und Konzepte für korrekte Programme	105
6.1	Das Konzept "Design by Contract"	107
6.2	Liskovsches Substitutionsprinzip	114
6.3	Principle of Least Astonishment	119
6.4	Zusammenfassung	121
7	Prinzipien für die Stabilität und Erweiterbarkeit bei Programmänderungen	123
7.1	Open-Closed Principle	125
7.2	Ziehe Objektkomposition der Klassenvererbung vor	132
7.3	Programmiere gegen Schnittstellen, nicht gegen Implementierungen	138
7.4	Zusammenfassung	140
8	Das Konzept "Inversion of Control"	143
8.1	Historie	145
8.2	Ziele	145
8.3	Ereignisorientierte Programmierung anstelle von Pollen	145
8.4	"Inversion of Control" bei Frameworks	148

8.5	Bewertung	148
8.6	Beispielprogramm.....	149
8.7	Zusammenfassung	154
9	Übersicht über die vorgestellten Entwurfsprinzipien und Konzepte	155
9.1	Entwurfsprinzipien zur modularen Struktur von Systemen	156
9.2	Entwurfsprinzipien zur Reduktion der Komplexität	156
9.3	Entwurfsprinzipien und Konzepte für schwach gekoppelte Teilsysteme.....	157
9.4	Entwurfsprinzipien und Konzepte für korrekte Programme.....	160
9.5	Prinzipien für die Stabilität und Erweiterbarkeit bei Programmänderungen.	160
9.6	Das Konzept "Inversion of Control"	161
Anhang: Problem- und Lösungsbereich.....		163
Literaturverzeichnis.....		165
Abkürzungsverzeichnis		171
Begriffsverzeichnis.....		173
Index		183