

# Das Java-Praktikum

Aufgaben und Lösungen zum Programmierenlernen mit Java 7

VON

Reinhard Schiedermeier, Klaus Köhler

2., akt. u. erw. Aufl.

Das Java-Praktikum – Schiedermeier / Köhler

schnell und portofrei erhältlich bei [beck-shop.de](http://beck-shop.de) DIE FACHBUCHHANDLUNG

Thematische Gliederung:

Objektorientierte Programmierung

dpunkt.verlag 2011

Verlag C.H. Beck im Internet:

[www.beck.de](http://www.beck.de)

ISBN 978 3 89864 765 6

**Reinhard Schiedermeier · Klaus Köhler**

# **Das Java-Praktikum**

**Aufgaben und Lösungen  
zum Programmierenlernen mit Java 7**

2., aktualisierte und erweiterte Auflage



**dpunkt.verlag**

Reinhard Schiedermeier  
rs@cs.hm.edu

Klaus Köhler  
koehler@hm.edu

Lektorat: Dr. Michael Barabas  
Copy-Editing: Ursula Zimpfer, Herrenberg  
Satz: Science & More  
Herstellung: Birgit Bäuerlein  
Umschlaggestaltung: Helmut Kraus, [www.exclam.de](http://www.exclam.de)  
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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

ISBN 978-3-89864-765-6

2., aktualisierte und erweiterte Auflage 2011  
Copyright © 2011 dpunkt.verlag GmbH  
Ringstraße 19 B  
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

# Vorwort

*Learning by doing* – einen besseren Leitspruch könnte es zum Lernen von Programmieren kaum geben.

Programmieren lernen hat viel Ähnlichkeit mit Schwimmen lernen in Kindertagen. Man kann einem Kind noch so genau erklären, wie man Arme und Beine im Takt bewegt, im richtigen Moment ein- und ausatmet, den Kopf anhebt oder loslässt und so weiter – schwimmen kann das Kind mit diesen Einsichten ganz bestimmt nicht! Dazu muss es ins Wasser springen, nass werden, husten, nach Luft japsen, untergehen und wieder auftauchen. Die Begeisterung durchläuft dabei sicher einige Höhen und auch unvermeidliche Tiefen.

Das Gleiche gilt für das Programmieren: Man kann sich beliebig lang die Konzepte, Strukturen, Ausdrucksmittel und Paradigmen von Programmiersprachen durchlesen oder erklären lassen – programmieren lernt man damit nicht. Dazu muss man Code schreiben, Fehler machen, suchen und beheben, Hilfe holen, sich andere Programme ansehen und so weiter. Bezüglich der Begeisterung gilt dabei das oben Gesagte.

Dieses Buch will Ihnen helfen, Java programmieren zu lernen. Sie finden hier eine Sammlung von ausgewählten Aufgaben und dazu jeweils einen ausführlich erklärten Lösungsvorschlag. Betrachten Sie es im übertragenen Sinn als eine Kombination aus Sprungbecken, Schwimmreifen und Wellenmaschine.

## Inhalt

Die Aufgaben in diesem Buch setzen Kenntnisse von Java voraus. Dieses Wissen wird hier nicht erklärt, sondern praktisch angewendet. Es ist unbedingt zu empfehlen, dass Sie gleichzeitig eine einschlägige Vorlesung besuchen oder ein Java-Lehrbuch heranziehen, wie zum Beispiel [Mössenböck 11], [Schiedermeier 10] oder eines der vielen anderen.

Der Schwierigkeitsgrad der Aufgaben in diesem Buch steigt fortlaufend an und bezieht schrittweise immer neue Sprachmittel von Java ein. Die folgende Liste gibt Ihnen einen Überblick über die Auswahl der Aufgaben und die Voraussetzun-

gen zur Lösung.<sup>1</sup> Eine genauere Liste der einzelnen Aufgaben und Teilaufgaben mit ihrem Schwierigkeitsgrad und den jeweils im Fokus stehenden Sprachmitteln finden Sie in Anhang B (Seite 430).

- I. Arithmetik, Variablen, Wertzuweisungen** Die ersten Aufgaben kommen mit elementaren Kenntnissen von Java aus. Sie erwarten nicht mehr als Arithmetik, Variablen und Ausgabeanweisungen.
- II. Kontrollstrukturen** Die Aufgaben in diesem Abschnitt führen Alternativen und Schleifen ein. Diese Sprachmittel werden als Kontrollstrukturen bezeichnet. Praktisch alle verbreiteten Programmiersprachen bieten sie in ganz ähnlicher Form an.
- III. Algorithmen** In diesem Abschnitt werden Kontrollstrukturen zur Lösung algorithmisch anspruchsvollerer Aufgaben eingesetzt. Zum Teil werden geschachtelte Schleifen verwendet, ebenso wie Arrays und Rekursion.
- IV. Klassen** Der nächste Abschnitt beschäftigt sich mit der Definition und dem Einsatz von Klassen. Die Algorithmik tritt dabei etwas in den Hintergrund. Das Augenmerk liegt hier auf dem Umgang mit Objekten, Methoden und der Abgrenzung von Zuständigkeiten. Alle diese Maßnahmen zielen auf die systematische Konstruktion größerer Programme ab.
- V. Interfaces und Vererbung** Hier kommen Interfaces und Vererbung als Kern der Objektorientierung ins Spiel. Der disziplinierte Einsatz von Vererbung in ihren verschiedenen Facetten führt zu eleganten und flexiblen Lösungen. Aber ebenso kann Vererbung bei unangemessener Verwendung heilloses Chaos auslösen. Diese Aufgaben erfordern sorgfältige Modellierung des Problems in aufeinander abgestimmte Typen und den Entwurf passender Methoden.
- VI. Containerklassen** Eine wichtige Art von Klassen sind Containerklassen, das heißt Klassen, die andere Elemente organisieren und verwalten. Hier wird das *Collection-Framework* benutzt und es werden auch neue Containerklassen erstellt.
- VII. I/O, Netzwerke, Nebenläufigkeit** Die Aufgaben dieses Abschnittes widmen sich pragmatischen Themen wie der Ein- und Ausgabe. Der Umgang mit externen Daten und dem Filesystem ist nicht ganz einfach, aber in der Praxis unverzichtbar. Weitere Aufgaben haben Netzwerkprogrammierung zum Gegenstand. Diese Art von Problemen ist von Natur aus nebenläufig und legt den Einsatz von Threads nahe.
- VIII. Generics** Die Aufgaben im letzten Abschnitt befassen sich mit generischen Klassen und Typen. Diese Sprachmittel sind äußerst nützlich und führen zu robusten und kompakten Programmen.

---

<sup>1</sup>Viele Aufgaben bestehen aus mehreren Teilaufgaben, die die ursprüngliche Problemstellung in der einen oder anderen Richtung ausweiten. In solchen Fällen sind gelegentlich Vorgriffe auf Sprachmittel nötig, die erst weiter hinten in den Mittelpunkt rücken. Nehmen Sie sich bitte die Freiheit, solche Vertiefungen im ersten Anlauf zu übergehen. Sie können später jederzeit wieder darauf zurückkommen.

Eine Standardinstallation von Java beinhaltet eine umfangreiche Laufzeitbibliothek, die wohl kaum noch jemand in allen Einzelheiten überblickt.<sup>2</sup> Programmieren mit Java erfordert sicher nicht, die komplette Bibliotheksreferenz im Kopf zu haben. Die meisten Aufgaben sind deshalb so gestaltet, dass sie sich mit möglichst wenigen und elementaren Bibliotheksklassen lösen lassen. Ausnahmen sind Containerklassen im *Collection-Framework* sowie die Themen I/O und Netzwerkkommunikation, die ohne Bibliotheksunterstützung nicht sinnvoll zu behandeln sind.<sup>3</sup> Im Mittelpunkt steht allerdings immer die *Sprache* Java. Es reicht völlig aus, wenn Sie aus diesem Buch mitnehmen, dass es für gewisse Teilprobleme Hilfe in der Bibliothek gibt. Zu gegebener Zeit schlagen Sie in der ausgezeichneten Onlinereferenz nach oder besuchen eine der zahlreichen Quellen im Internet, wie zum Beispiel [Almanac 02] oder [Tutorial].

## Bedienungsanleitung

Neben den Aufgaben enthält dieses Buch auch alle Lösungen. Wiederkehrende und nebensächliche Teile der Lösungen, wie zum Beispiel der Kopf der `main`-Methode oder `import`-Klauseln, sind zum Teil nicht abgedruckt. Die vollständig ausformulierten, sofort übersetzbaren Lösungen finden Sie auf der Webseite zum Buch: <http://so1.cs.hm.edu/dpunkt-java-praktikum>.

Wir sind uns bewusst, dass die unmittelbar auf den Aufgabentext folgende Lösung recht verführerisch ist. Dennoch sollten Sie, lieber Leser<sup>4</sup>, versuchen, den maximalen Nutzen aus diesem Buch zu ziehen, und die mitgelieferte Lösung so lange wie möglich ignorieren. Blättern oder lesen Sie bitte erst dann weiter, wenn Sie selbst eine befriedigende Lösung entwickelt haben oder wenn Sie unbedingt einen Hinweis brauchen.

Zur Lösung der Aufgaben implementieren Sie lauffähige Programme. Konzepte oder Ideen reichen leider nicht aus, denn deren Tragfähigkeit bleibt ungewiss, bis sie mit einer konkreten Realisierung nachgewiesen wird. Geben Sie sich deshalb bitte nicht mit der Annahme zufrieden, dass Sie die Lösung im Ernstfall schon finden würden. Einige Aufgaben weisen verborgene Klippen auf, die erst bei der praktischen Umsetzung zutage treten.

<sup>2</sup>Einige Java-Versionen unterscheiden sich überwiegend im Umfang der Bibliothek. Merkliche Änderungen der *Sprache* Java gab es in Java 1.5 und 1.7. Java 1.5 wurde, mehr aus Marketing-strategischen Gründen, in »Java 5« umbenannt. Damit begann eine neue Versionszählung, in der die vormalige *Major version* 1 weggefallen ist.

<sup>3</sup>Eine kompakte, aber für den Zweck dieser Aufgaben ausreichende Darstellung finden Sie beispielsweise in [Mössenböck 11], Kap. 21.

<sup>4</sup>Wo immer in diesem Buch auf Personen Bezug genommen wird, sprechen wir immer Frauen und Männer in bestimmten Rollen gleichermaßen an. Wir verzichten um der Lesbarkeit willen auf sperrige Formulierungen wie »Leser/Leserin«.

Wenn Ihnen die richtige Idee einfach nicht kommen will, möchten wir Ihnen empfehlen, wenigstens eine Nacht über einer Aufgabe zu schlafen (oder schlaflos über dem Problem zu brüten), bevor Sie die Lösung nachschlagen.

Betrachten Sie die abgedruckten Lösungen im Buch nur als Vorschläge, aber nicht als das Maß aller Dinge. Die Aufgaben lassen zum Teil viel Spielraum bei der Wahl des Lösungsweges. Deshalb ist es vollkommen in Ordnung, wenn Sie zu einer ganz anderen Lösung kommen.

In der Java-Programmierung haben sich im Laufe der Zeit einige Konventionen entwickelt, die den *test of time* bestanden haben und deshalb allgemein akzeptiert werden. In Anhang C haben wir einige Regeln zusammengestellt, die wir bei der Entwicklung unserer Lösungen beachtet haben. Es wäre sicher hilfreich, wenn Sie diesen Anhang durchsehen, bevor Sie die Aufgaben angehen. Es macht nichts, wenn Sie zunächst noch nicht alles verstehen. Wichtig ist nur, dass Sie sich wieder daran erinnern, zu diesem oder jenem Punkt etwas gelesen zu haben. Sie werden auch feststellen, dass wir selbst gelegentlich gegen die Konventionen verstoßen. Konventionen sind nicht in Marmor gemeißelt, sondern spielen mehr die Rolle von Empfehlungen, von denen man durchaus mit guten Gründen abweichen kann.

Einige Aufgaben erfordern den Einsatz von Bibliotheksklassen und -methoden, deren Beherrschung zwar nützlich, aber aus konzeptioneller Sicht wenig tieferschürfend ist. Um Ihnen die Arbeit zu erleichtern, finden Sie in Anhang D kurze Programme, die einfache, wiederkehrende Aufgaben erledigen, wie zum Beispiel das Einlesen einer Textdatei oder den Transport von Daten über ein Netzwerk. Benutzen Sie diese Programme nach Bedarf als Bausteine für Ihre eigenen Lösungen.

### **Technische Voraussetzung**

Die minimalen Hilfsmittel zur Bearbeitung der Aufgaben sind ein einfacher Texteditor und ein Java-Entwicklungssystem. Ein Texteditor<sup>5</sup> wird mit jedem Betriebssystem geliefert. Etwas komfortablere Editoren, die oft die Entwicklung von Java-Programmen unterstützen, sind für die meisten populären Plattformen frei verfügbar, wie zum Beispiel *jEdit* oder *Joe*.

Ein Java-Entwicklungssystem kann kostenlos von der Website der Firma Oracle heruntergeladen werden (<http://java.oracle.com>). Sie finden auf dieser Website eine ganze Anzahl von Downloads. Die folgenden Kriterien führen Sie zum passenden Paket:

---

<sup>5</sup>Ein *Textverarbeitungssystem*, wie Microsoft Word oder OpenOffice, eignet sich schlecht.

*Version* Es gibt verschiedene Java-Versionen. Sie brauchen Java Version 7 oder eine nachfolgende Version.<sup>6</sup>

*Edition* Java wird für unterschiedliche Einsatzzwecke angeboten. Für den persönlichen Gebrauch vorgesehen ist die *Standard Edition* (SE). Die ebenfalls verfügbare *Enterprise Edition* (EE) und die *Micro Edition* (ME) sind für andere Systeme ausgelegt und bieten hier keinen Vorteil.

#### *Entwicklungssystem*

Die Standard Edition ist als *Java SE Development Kit* (JDK) und als *Java SE Runtime Environment* (JRE) verfügbar. Mit dem *Runtime Environment* JRE können zwar fertige Java-Programme ausgeführt, aber keine neuen entwickelt werden. Sie brauchen deshalb das *Development Kit* JDK, das das JRE komplett enthält.

Elegant lässt sich mit einer integrierten Entwicklungsumgebung (IDE) arbeiten. Derzeit populär sind beispielsweise *Eclipse* und *Netbeans*, die beide kostenlos und für praktisch alle Systeme verfügbar sind. Diese Umgebungen kapseln Editor, Compiler, Laufzeitsystem und weitere Werkzeuge in einer ansprechenden grafischen Oberfläche, die viele Aufgaben automatisch oder mit Mausclick erledigt. Darin liegt allerdings auch eine gewisse Gefahr, denn ohne Kenntnis der tatsächlichen Abläufe ist der Benutzer hilflos, wenn die IDE einmal nicht so reagiert wie erwartet.

IDEs sind selbst umfangreiche Programme, deren Arbeitsweise verstanden werden will und entsprechende Einarbeitung erfordert. Insgesamt sehen sich gerade Anfänger mit einer Komplexität konfrontiert, die ohne IDE vielleicht leichter und schneller zu bewältigen wäre.

Behalten Sie schließlich im Auge, dass eine hübsche IDE zwar technische Abläufe vereinfacht, aber bei konzeptioneller Arbeit keine Unterstützung leistet. So reduziert sich die Zeiteinsparung auf die Codierung, während die vorausgehende, oft entscheidende Denkarbeit unverändert überwiegend im Kopf stattfindet, allenfalls unterstützt von Papier und Bleistift.<sup>7</sup>

Insgesamt sollten Sie abwägen, ob Sie die erste Zeit Ihres Vorhabens nicht vielleicht auf eine IDE verzichten sollten, um die tatsächlichen Abläufe und Werkzeuge ungefiltert kennenzulernen. Sobald Ihre Programme aus mehreren Quelltextdateien bestehen, ist die Zeit möglicherweise reif für eine IDE.

## **Aufbau der Lösungen**

Viele Lösungen werden schrittweise aufgebaut. Dabei wird das endgültige Programm aus kurzen Abschnitten zusammengesetzt, die jeweils einzeln diskutiert

---

<sup>6</sup>Mit Versionen 5 und 6 können die Aufgaben gelöst werden, allerdings sind nicht alle Lösungen im Buch und auf der Webseite damit übersetzbar. Version 1.4 oder noch ältere Versionen reichen nicht aus.

<sup>7</sup>Vom zu frühen Einsatz bildschirmgestützter Entwurfswerkzeuge (UML-Editoren und dergleichen) raten wir ab. Sie leisten gute Dienste zu Dokumentationszwecken, lenken aber von der eigentlich wichtigen Denkarbeit ab.



und erklärt werden. Um nicht beim Einfügen jedes neuen Fragments alle zuvor entwickelten Fragmente wiederholen zu müssen, werden die einzelnen Auszüge mit Kommentartiteln versehen, auf die von anderer Stelle Bezug genommen wird. Hier ein Beispiel für ein Codefragment:

```
// Berechnen der Summe s aller Zahlen von 1 bis n
int s = 0;
int i = 1;
while(i <= n) {
    s = s + i;
    i = i + 1;
}
```

Später wird anstelle dieses Codeabschnitts nur noch der Kommentar genannt und von drei Punkten gefolgt. Dieser Kommentar ist ein Platzhalter für das vorher gezeigte Codefragment, das hier nicht mehr abgedruckt ist:

```
// Festlegen der Obergrenze n
int n = 10;

// Berechnen der Summe s aller Zahlen von 1 bis n
...

// Summe ausgeben
System.out.println(s);
```

Manche Lösungen werden sequenziell entwickelt. Ein Codefragment endet dann mit drei Punkten, um anzuzeigen, dass das Programm noch nicht komplett ist und weiter unten fortgesetzt wird.

```
// Variablen zur Berechnung der Zahlensumme
int n = 10;    // Obergrenze, bis zu der addiert wird
int s = 0;    // akkumulierte Summe
int i = 1;    // Zahl, die gerade addiert wird
...
```

Das nachfolgende Fragment setzt das oben begonnene Fragment fort:

```
// die Zahlen von 1 bis n addieren
while(i <= n) {
    s = s + i;
    i = i + 1;
}

// Summe ausgeben
System.out.println(s);
```

Wenn die einzelnen Abschnitte hintereinander zusammengesetzt werden, ergibt sich das vollständige Programm.

## Auswahl der Aufgaben

Sie finden in diesem Buch ein breites Spektrum von Themen. Die zur Lösung erforderlichen Sprachmittel überspannen einen weiten Bereich. In dem Maße, wie Sie Ihre Kenntnisse erweitern, werden immer neue Aufgaben zugänglich. Um Ihnen die Orientierung zu erleichtern, finden Sie im Anhang B die Sprachmittel, die jeweils im Mittelpunkt der verschiedenen Aufgaben stehen.

Die Auswahl der Aufgaben ist unbestreitbar subjektiv. Einige Aspekte der Sprache Java und weite Bereiche der Bibliothek von Java werden nicht erwähnt, obwohl sie zweifellos wichtig sind. Wir haben uns auf Themen beschränkt, die nach unserer Erfahrungen bei der Softwareentwicklung immer wieder auftauchen oder grundsätzliche Strukturen der Programmiersprache betreffen, deren Verständnis für einen sicheren Umgang mit Java notwendig ist.

Wir hoffen natürlich, lieber Leser, dass möglichst *alle* Aufgaben in diesem Buch Sie ansprechen. Allerdings ist diese Hoffnung wahrscheinlich vermessen, deshalb wären wir bereits sehr zufrieden, wenn Ihnen ein großer Teil der Aufgaben zusagt.

Fassen Sie bitte die Aufgaben nicht als Test oder Prüfung auf, sondern als Anregung zum Grübeln und Tüfteln. Sicher wird Ihnen nicht jedes Mal in wenigen Minuten die vollständige Lösung gelingen. Wenn Sie stehenden Fußes alle Aufgaben lösen könnten, dann hätten Sie dieses Buch ohnedies nicht nötig :-). Nehmen Sie also in Kauf, dass es manchmal ein wenig dauert, bis Ihnen der entscheidende Einfall kommt.

## Webseite zum Buch

Unter der Internetadresse

<http://sol.cs.hm.edu/dpunkt-java-praktikum>

finden Sie weiteres Material zum Buch. In erster Linie stehen dort für Sie die vollständigen, lauffähigen Lösungen zum Download zur Verfügung.

## Dank

Unser Dank gilt den Mitarbeitern des *dpunkt.verlages*, die uns viel Freiheiten bei der Gestaltung des Buches gelassen haben. Besonders danken möchten wir Tschucky (Gudrun Schiedermeier), die sich in mühseliger Arbeit durch ständig neue, unausgegorene Fassungen dieses Buches gearbeitet hat und Aufgaben lösen musste, die nicht immer lösbar waren. Schließlich sollen auch die Studenten<sup>8</sup> unserer Fakultät 07 für Informatik und Mathematik der Hochschule München

<sup>8</sup>Wie bereits weiter vorne ausgeführt steht eine solche Bezeichnung für Personen beiderlei Geschlechts, die die betreffende Rolle innehaben.

gewürdigt werden, die über viele Semester hinweg in Prüfungen und Praktika mit früheren Versionen der Aufgaben<sup>9</sup> die Klinge kreuzten. Sie fanden manche überraschenden und kreativen Lösungen, die zum Teil in dieses Buch eingeflossen sind.

Anregungen, Kritik und Verbesserungsvorschläge sind uns jederzeit willkommen. Bitte richten Sie Ihre Nachricht an [koehler@hm.edu](mailto:koehler@hm.edu) oder [rs@cs.hm.edu](mailto:rs@cs.hm.edu).

München, Frühjahr 2008

Klaus Köhler und Reinhard Schiedermeier

### **Vorwort zur 2. Auflage**

Seit den Erscheinen der ersten Auflage in 2008 hat sich Java mit der Veröffentlichung von Version 7 im Sommer 2011 weiterentwickelt. Neben kleineren syntaktischen Vereinfachungen erlaubt vor allem das »automatische Ressourcenmanagement« (ARM, *try-with-resource*) robusteren und kompakteren Code. Die Codefragmente im Buchtext und der komplette Code auf der Webseite zum Buch wurden an die neuen Möglichkeiten angepasst. Zum Übersetzen der Lösungsbeispiele ist infolgedessen allerdings ein *Java SE Development Kit* (JDK) der Version 7 erforderlich.

Inhaltlich sind sowohl einige komplett neue Aufgaben dazugekommen wie auch eine Reihe von Erweiterungen bestehender Aufgaben. In vielen Lösungen wurden statische UML-Klassendiagramme<sup>10</sup> eingefügt, die das Zusammenspiel verschiedener Interfaces und Klassen veranschaulichen. Selbstredend wurden alle bekannten Fehler der ersten Auflage korrigiert.

Wir möchten an dieser Stelle den in der ersten Auflage ausgesprochenen Dank auf Rebecca Neigert ausdehnen, die findige und interessante Ideen zu diesem Buch beigetragen und mit scharfem Blick Schwachstellen im Text entdeckt hat.

Material zu diesem Buch finden Sie weiterhin auf der Internetseite

<http://sol.cs.hm.edu/dpunkt-java-praktikum>

Anregungen, Kritik und Verbesserungsvorschläge senden Sie bitte an [koehler@hm.edu](mailto:koehler@hm.edu) oder [rs@cs.hm.edu](mailto:rs@cs.hm.edu).

München, Sommer 2011

Klaus Köhler und Reinhard Schiedermeier

---

<sup>9</sup>Einige Aufgaben in diesem Buch beruhen auf Ideen, die als Übungsaufgaben in [Schiedermeier 10] zu finden sind.

<sup>10</sup>Eine kurze Einführung in Unified Modeling Language gibt beispielsweise [Inden 11].