Java Challenge

Michael Inden

Fit für das Jobinterview und die Praxis –
mit mehr als 100 Aufgaben
und Musterlösungen

dpunkt.verlag

Dipl.-Inform. Michael Inden ist Oracle-zertifizierter Java-Entwickler. Nach seinem Studium in Oldenburg hat er bei diversen internationalen Firmen in verschiedenen Rollen etwa als Softwareentwickler, -architekt, Consultant, Teamleiter sowie Trainer gearbeitet. Zurzeit ist er als CTO und Leiter Academy in Zürich tätig.

Michael Inden hat über zwanzig Jahre Berufserfahrung beim Entwurf komplexer Softwaresysteme gesammelt, an diversen Fortbildungen und mehreren Java-One-Konferenzen teilgenommen. Sein besonderes Interesse gilt dem Design qualitativ hochwertiger Applikationen mit ergonomischen GUIs sowie dem Coaching. Sein Wissen gibt er gerne als Trainer in internen und externen Schulungen und auf Konferenzen weiter, etwa bei der Java User Group Switzerland, bei der JAX/W-JAX, ch.open und den IT-Tagen.

Michael Inden

Java Challenge

Fit für das Job-Interview und die Praxis – mit mehr als 100 Aufgaben und Musterlösungen

Michael Inden
michael_inden@hotmail.com

Lektorat: Michael Barabas
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Michael Inden
Herstellung: Stefanie Weidner
Umschlaggestaltung: Helmut Kraus, www.exclam.de

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:
Print 978-3-86490-756-2
PDF 978-3-96910-028-8
ePub 978-3-96910-029-5
mobi 978-3-96910-030-1

1. Auflage 2020

Wieblinger Weg 17
69123 Heidelberg

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: hallo@dpunkt.de.

5 4 3 2 1 0

Inhaltsverzeichnis

Vorwort

Zunächst einmal bedanke ich mich bei Ihnen, dass Sie sich für dieses Buch entschieden haben. Hierin finden Sie eine Vielzahl an Übungsaufgaben zu den unterschiedlichsten Themengebieten, die kurzweilige Unterhaltung durch Lösen und Implementieren der Aufgaben bieten und Sie so entweder auf Bewerbungsgespräche einstimmen oder aber einfach Ihre Problemlösungsfähigkeiten verbessern.

Übung macht den Meister

Wir alle kennen das Sprichwort: »Übung macht den Meister.« Im Handwerk und in diversen Bereichen des realen Lebens wird viel geübt und der Ernstfall ist eher selten, etwa im Sport, bei Musikern und in anderen Bereichen. Merkwürdigerweise ist dies bei uns Softwareentwicklern oftmals deutlich anders. Wir entwickeln eigentlich fast die gesamte Zeit und widmen uns dem Üben und Lernen bzw. Einstudieren eher selten, teilweise gar nicht. Wie kommt das?

Vermutlich liegt das neben dem in der Regel vorherrschenden Zeitdruck auch daran, dass nicht so viel geeignetes Übungsmaterial zur Verfügung steht – es gibt zwar Lehrbücher zu Algorithmen sowie Bücher zu Coding, aber meistens sind diese entweder zu theoretisch oder zu Sourcecode-lastig und beinhalten zu wenig Erklärungen der Lösungswege. Das will dieses Buch ändern.

Wieso dieses Buch?

Wie kam ich dazu, dieses Buchprojekt in Angriff zu nehmen? Das hat mehrere Gründe. Zum einen wurde ich immer wieder per Mail oder persönlich von Teilnehmern meiner Workshops gefragt, ob es nicht ein Übungsbuch als Ergänzung zu meinem Buch »Der Weg zum Java-Profi« [2] geben würde. Dadurch kam die erste Idee auf.

Wirklich ausgelöst hat das Ganze, dass ein Recruiter von Google, mit einer Jobanfrage recht überraschend auf mich zukam. Als Vorbereitung für die dann bevorstehenden Jobinterviews und zur Auffrischung meiner Kenntnisse machte ich mich auf die Suche nach geeigneter Lektüre und entwickelte selbst schon einige Übungsaufgaben. Dabei entdeckte ich das großartige, aber auch teilweise recht anspruchsvolle Buch »Cracking the coding interview« von Gayle Laakmann McDowell [5], das mich weiter inspirierte, sodass ich ein paar der dort vorgestellten Ideen aufgreife.

An wen richtet sich dieses Buch?

Dieses Buch ist kein Buch für Programmierneulinge, sondern richtet sich an Leser, die bereits recht gutes Java-Know-how besitzen und dieses mithilfe von Übungen weiter vertiefen wollen. Anhand kleiner Programmieraufgaben erweitern Sie auf unterhaltsame Weise Ihr Wissen rund um Java, Algorithmen und gutes OO-Design.

Dieses Buch richtet sich im Speziellen an zwei Zielgruppen:

1. Zum einen sind dies engagierte Hobbyprogrammierer und Informatikstudierende, aber auch Berufseinsteiger, die Java als Sprache schon ganz gut beherrschen, und nun ihr Wissen anhand von Übungen vertiefen wollen.

2. Zum anderen ist das Buch für erfahrene Softwareentwickler und -architekten bestimmt, die ihr Wissen ergänzen oder auffrischen wollen, um einige althergebrachte Denkmuster aufzubrechen und neue Ideen zu entwickeln. Insbesondere zur Lösungsfindung und zu Algorithmen und Datenstrukturen sollte es das eine oder andere Aha-Erlebnis geben.

Generell verwende ich die maskuline Form, um den Text leichter lesbar zu halten. Natürlich beziehe ich damit alle weiblichen Leserinnen mit ein und freue mich über diese ganz besonders.

Was vermittelt dieses Buch?

Dieses Buch enthält einen bunten Mix an Übungsaufgaben zu verschiedenen Themengebieten. Mitunter gibt es auch einige Knobelaufgaben, die zwar nicht direkt für die Praxis wichtig sind, aber indirekt doch, weil Sie Ihre Fähigkeiten zur Kreativität und zur Lösungsfindung verbessern.

Neben Übungsaufgaben und dokumentierten Lösungen war es mir wichtig, dass jeder im Buch behandelte Themenbereich mit einer kurzen Einführung startet, damit auch diejenigen Leser abgeholt werden, die in einigen Gebieten vielleicht noch nicht so viel Know-how aufgebaut haben. Damit können Sie sich dann an die Aufgaben bis etwa zum mittleren Schwierigkeitsgrad wagen. In jedem Themengebiet finden sich immer auch einige leichtere Aufgaben zum Einstieg. Mit etwas Übung sollten Sie sich auch an etwas schwierigere Probleme wagen. Mitunter gibt es herausfordernde Knacknüsse, an denen sich dann Experten versuchen können oder solche, die es werden wollen.

Tipps und Hinweise aus der Praxis

Dieses Buch ist mit diversen Praxistipps gespickt. In diesen werden interessante Hintergrundinformationen präsentiert oder es wird auf Fallstricke hingewiesen.

Tipp: Praxistipp

In derart formatierten Kästen finden sich im späteren Verlauf des Buchs immer wieder einige wissenswerte Tipps und ergänzende Hinweise zum eigentlichen Text.

Schwierigkeitsgrad im Überblick

Für ein ausgewogenes, ansprechendes Übungsbuch bedarf es selbstverständlich einer Vielzahl an Aufgaben verschiedener Schwierigkeitsstufen, die Ihnen als Leser die Möglichkeit bieten, sich schrittweise zu steigern und Ihre Kenntnisse auszubauen. Dabei setze ich zwar ein gutes Java-Grundwissen voraus, allerdings erfordern die Lösungen niemals ganz tiefes Wissen über ein Themengebiet oder ganz besondere Sprachfeatures.

Damit der Schwierigkeitsgrad einfach und direkt ersichtlich ist, habe ich die von anderen Bereichen bekannte Sternekategorisierung genutzt, deren Bedeutung in diesem Kontext in nachfolgender Tabelle etwas genauer erläutert wird.

Sterne (Bedeutung)	Einschätzung	Zeitaufwand
★☆☆☆☆ (sehr leicht)	Die Aufgaben sollten ohne große Vorkenntnisse mit einfachem Java-Wissen in wenigen Minuten lösbar sein.	< 15 min
★★☆☆☆ (leicht)	Die Aufgaben erfordern ein wenig Nachdenken, sind aber dann direkt zu lösen.	< 30 min
★★★☆☆ (mittel)	Die Aufgaben sind mit etwas Nachdenken, ein wenig Strategie und manchmal durch	~ 30 – 45 min

	die Betrachtung verschiedener Rahmenbedingungen gut schaffbar.	
★★★★☆ (schwierig)	Erprobte Problemlösungsstrategien, gutes Wissen zu Datenstrukturen und fundierte Java-Kenntnisse sind zur Lösung nötig.	~ 45 – 90 min
★★★★★ (sehr schwierig)	Die Aufgaben sind wirklich knifflig und schwierig zu lösen. Das sind erst dann Kandidaten, nachdem die anderen Aufgaben Ihnen keine größeren Schwierigkeiten mehr bereiten.	> 60 min

Dies sind jeweils nur Einschätzungen von meiner Seite und eher grobe Einordnungen. Bedenken Sie bitte, dass die von jedem Einzelnen wahrgenommene Schwierigkeit auch sehr von seinem Background und Wissensstand abhängt. Ich habe schon erlebt, dass sich Kollegen mit Aufgaben schwergetan haben, die ich als recht einfach empfand. Aber auch das Gegenteil kenne ich: Während andere eine Aufgabe anscheinend spielend lösen, ist man selbst am Verzweifeln, weil der Groschen einfach nicht fällt. Manchmal hilft dann eine Pause mit einem Kaffee oder ein kleiner Spaziergang. Lassen Sie sich auf keinen Fall demotivieren – jeder hat irgendwann mit irgendeiner Art von Aufgabe zu kämpfen.

Hinweis: Mögliche Alternativen zu den Musterlösungen

Beachten Sie bitte, dass es für Problemstellungen nahezu immer einige Varianten gibt, die für Sie vielleicht sogar eingängiger sind. Deswegen werde ich ab und an als Denkanstoß interessante Alternativen zur (Muster-)Lösung präsentieren.

Aufbau dieses Buchs

Nachdem Sie eine grobe Vorstellung über den Inhalt dieses Buchs haben, möchte ich die Themen der einzelnen Kapitel kurz vorstellen.

Wie bereits angedeutet, sind die Übungsaufgaben thematisch gruppiert. Dabei bilden die sechs Kapitel nach der Einleitung die Grundlage und die darauffolgenden drei Kapitel behandeln fortgeschrittenere Themengebiete.

Kapitel 1 – Einleitung Dieses Kapitel beschreibt den grundsätzlichen Aufbau der folgenden Kapitel mit den Abschnitten Einführung, Aufgaben und Lösungen. Zudem wird ein Grundgerüst für die oftmals zur Prüfung der Lösungen genutzten Unit Tests vorgestellt. Abschließend gebe ich Hinweise zum Ausprobieren der Beispiele und Lösungen.

Kapitel 2 – Mathematische Aufgaben Das zweite Kapitel widmet sich mathematischen Operationen sowie Aufgaben zu Primzahlen und dem römischen Zahlensystem. Darüber hinaus präsentiere ich ein paar Ideen zu Zahlenspielereien.

Kapitel 3 – Rekursion Rekursion ist ein wichtiger Basisbaustein bei der Formulierung von Algorithmen. Dieses Kapitel gibt einen kurzen Einstieg und die diversen Übungsaufgaben sollten dabei helfen, Rekursion zu verstehen.

Kapitel 4 – Strings Strings sind bekanntermaßen Zeichenketten, die eine Vielzahl an Methoden bieten. Ein solides Verständnis ist elementar wichtig, da nahezu kein

Programm ohne Strings auskommt. Deswegen werden wir in diesem Kapitel die Verarbeitung von Zeichenketten anhand verschiedener Übungsaufgaben kennenlernen.

Kapitel 5 – Arrays Neben Strings sind Arrays ebenfalls Grundbausteine beim Programmieren. Arrays sind – wie Sie wissen – einfache Datenstrukturen zur Speicherung von Werten, allerdings ohne allzu viel Komfort. In der Praxis empfiehlt sich chrodeswegen oftmals, auf die Datenstrukturen des Collections-Frameworks zurückzugreifen. Diese behandelt dann Kapitel 7 eingehend.

Kapitel 6 – Datumsverarbeitung Mit Java 8 wurde das JDK um einige Funktionalitäten zur Datumsverarbeitung erweitert. Diese sollte jeder Java-Entwickler kennen. Die Übungsaufgaben verschaffen Ihnen einen guten Einstieg in die Thematik, sodass der Transfer in die Praxis leichtfallen sollte.

Kapitel 7 – Basisdatenstrukturen: Listen, Sets und Maps Im Collections-Framework werden Listen, Mengen und Schlüssel-Wert-Abbildungen durch verschiedene Containerklassen realisiert. Für den Programmieralltag ist ein sicherer Einsatz von großem Vorteil, was durch die Übungsaufgaben trainiert wird.

Kapitel 8 – Rekursion Advanced Kapitel 3 hat das Thema Rekursion einleitend behandelt. In diesem Kapitel beschäftigen wir uns mit einigen fortgeschritteneren Aspekten rund um das Thema Rekursion. Wir starten mit der Optimierungstechnik namens Memoization. Im Anschluss schauen wir uns Backtracking als eine Problemlösungsstrategie an, die auf Versuch-und-Irrtum beruht und mögliche Lösungswege durchprobiert. Damit

lassen sich diverse Algorithmen ziemlich verständlich und elegant halten.

Kapitel 9 – Bäume Baumstrukturen spielen in der Informatik sowohl in der Theorie als auch der Praxis eine wichtige Rolle. In vielen Anwendungskontexten lassen sich Bäume gewinnbringend einsetzen, etwa für die Verwaltung eines Dateisystems, die Darstellung eines Projekts mit Teilprojekten und Aufgabenpaketen oder eines Buchs mit Kapiteln, Unterkapiteln und Abschnitten.

Kapitel 10 – Suchen und Sortieren Suchen und Sortieren sind zwei elementare Themen der Informatik im Bereich der Algorithmen und Datenstrukturen. Das Collections-Framework setzt beide um und nimmt einem dadurch viel Arbeit ab. Jedoch lohnt sich auch ein Blick hinter die Kulissen, etwa auf verschiedene Sortierverfahren und deren spezifische Stärken und Schwächen.

Kapitel 11 – Schlusswort und ergänzende Literatur In diesem Kapitel fasse ich das Buch zusammen und gebe vor allem einen Ausblick auf ergänzende Literatur. Um Ihr Können zu erweitern, ist neben dem Programmiertraining auch das Studium von weiteren Büchern empfehlenswert. Eine Auswahl an hilfreichen Titeln bildet den Abschluss des Hauptteils dieses Buchs.

Anhang A – Schnelleinstieg JShell In diesem Buch werden diverse Beispiele direkt auf der Konsole ausprobiert. Der Grund ist vor allem, dass Java seit Version 9 die interaktive Kommandozeilenapplikation JShell als REPL (Read Evaluate Print Loop) bietet, die wir in diesem Anhang kurz kennenlernen wollen.

Anhang B – Schnelleinsteg JUnit Zum Prüfen kleinerer Programmbausteine haben sich Unit Tests bewährt. Mit JUnit 5 ist das Ganze insbesondere beim Formulieren von Testfällen für mehrere Eingabekombinationen ziemlich komfortabel. Weil viele der hier im Buch erstellten Lösungen mit Unit Tests geprüft werden, gibt dieser Anhang einen Einstieg in die Thematik.

Anhang C – Schnelleinstieg O-Notation In diesem Buch verwende ich manchmal zur Abschätzung des Laufzeitverhaltens und zur Einordnung der Komplexität von Algorithmen die sogenannte O-Notation. Dieser Anhang stellt Wesentliches dazu vor.

Konventionen und ausführbare Programme

Verwendete Zeichensätze

Im gesamten Text gelten folgende Konventionen bezüglich der Schriftart: Der normale Text erscheint in der vorliegenden Schriftart. Dabei werden wichtige Textpassagen *kursiv* oder ***kursiv und fett*** markiert. Englische Fachbegriffe werden eingedeutscht groß geschrieben. Zusammensetzungen aus englischen und deutschen (oder eingedeutschten) Begriffen werden mit Bindestrich verbunden, z. B. Plugin-Manager. Sourcecode-Listings sind in der Schrift `courier` gesetzt, um zu verdeutlichen, dass dieser Text einen Ausschnitt aus einem Java-Programm wiedergibt. Auch im normalen Text werden Klassen, Methoden, Konstanten und Übergabeparameter in dieser Schriftart dargestellt.

Verwendete Abkürzungen

Im Buch verwende ich die in der nachfolgenden Tabelle aufgelisteten Abkürzungen. Weitere Abkürzungen werden im laufenden Text in Klammern nach ihrer ersten Definition aufgeführt und anschließend bei Bedarf genutzt.

Abkürzung	Bedeutung
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
(G)UI	(Graphical) User Interface
IDE	Integrated Development Environment
JDK	Java Development Kit
JLS	Java Language Specification
JRE	Java Runtime Environment
JSR	Java Specification Request

JVM	Java Virtual Machine

Verwendete Java-Version(en)

Nahezu alle Programme nutzen Java 11 als Basis, da dies die letzte LTS-Version (Long Term Support) ist, die mittlerweile eine gute Verbreitung besitzt. Die neueren Java-Versionen bringen einige hilfreiche Syntaxänderungen und API-Erweiterungen, die jedoch zur Lösung der Aufgaben nicht von Bedeutung sind – außer für ganz spezielle Aufgaben zu `switch` mit Java 14.

Verwendete Klassen aus dem JDK

Werden Klassen des JDKs zum ersten Mal im Text erwähnt, so wird deren voll qualifizierter Name, d. h. inklusive der Package-Struktur, angegeben: Für die Klasse `String` würde dann etwa `java.lang.String` notiert. Dies erleichtert ein Auffinden im JDK. Im darauffolgenden Text wird zur besseren Lesbarkeit auf diese Angabe verzichtet und nur der Klassenname genannt. Zudem sind aus Platzgründen in den Listings nur selten `import`-Anweisungen abgebildet.

Im Text beschriebene Methodenaufrufe enthalten in der Regel die Typen der Übergabeparameter, etwa `substring(int, int)`. Sind die Parameter in einem Kontext nicht entscheidend, wird auf deren Angabe aus Gründen der besseren Lesbarkeit verzichtet oder aber durch die Zeichenfolge … abgekürzt.

Download, Sourcecode und ausführbare Programme

Der Sourcecode der Beispiele steht auf der Webseite

www.dpunkt.de/java-challenge

zum Download bereit und ist in ein Eclipse-Projekt integriert. Weil dies ein Buch zum Mitmachen ist, sind einige der Programme mithilfe von Gradle-Tasks ausführbar. Deren Name wird in Kapitälchenschrift, etwa LocaleExample, angegeben –

alternativ ist natürlich eine Ausführung in der IDE bzw. als Unit Test möglich.

Viele Codeschnipsel lassen sich aber auch hervorragend in der JShell ausprobieren. Um das zu gewährleisten, sind mitunter bereits entwickelte Methoden an geeigneter Stelle nochmals abgebildet.

Nacharbeiten nach Projekt-Import Nach dem erstmaligen Import müssen die Abhängigkeiten auf die externen Bibliotheken im Eclipse-Projekt mit dem Kommando `gradle cleanEclipse eclipse` neu initialisiert und auf Ihren Rechner aktualisiert werden.