Einleitung

Das *Java Übungsbuch für die Versionen Java 8 bis Java 13* ist wie alle meine Übungsbücher aus der Erkenntnis entstanden, dass zu umfangreiche Beispiele mit komplizierten Algorithmen beim Lernen von Java am Anfang keine echte Hilfe bieten. Darum liegt der Schwerpunkt des Buches nicht auf der Umsetzung von komplizierten Vorgängen, sondern auf der Art und Weise. Stattdessen konzentriert es sich darauf, die in der Dokumentation nicht immer verständlich formulierten Erläuterungen zu Java-Klassen und -Interfaces mit einfachen Beispielen zu erklären und gleichzeitig die zugrunde liegenden Konzepte zu erörtern.

Dieses Buch wendet sich in erster Linie an Lehrer, Schüler und Studenten als Begleitliteratur zum Lernen der Programmiersprache Java, ist aber auch zum Selbststudium für alle Interessenten an dem Erlernen der Programmiersprache geeignet.

Durch die Einfachheit und Vollständigkeit der Aufgabenlösungen sowie die unterschiedlichen Lösungsmöglichkeiten erhält der Leser ein fundiertes Verständnis für die Aufgabenstellungen und deren Lösungen.

Durch das Lösen von Aufgaben soll der in Referenz- und Lehrbüchern von Java angebotene Stoff vertieft werden, und die dabei erzielten Ergebnisse können anhand der Lösungsvorschläge überprüft werden. Die Beispiele im Buch sind eher selten von zu komplexer Natur, sodass der eigentliche Zweck nicht in den Hintergrund tritt, und alle beschriebenen Themen können tiefgehend und präzise damit eingeübt werden.

Vorkenntnisse

Es ist Voraussetzung, dass der Leser zusätzlich mit einem Lehrbuch zu Java arbeitet bzw. bereits damit gearbeitet hat. Die grundlegenden Erläuterungen zu Java in diesem Buch können lediglich als Wiederholung des bereits vorhandenen Wissens dienen, reichen aber nicht aus, um die Sprache Java erst neu zu lernen.

Als weitere Voraussetzung sind Grundlagen im Bereich der Programmierung und im Umgang mit dem Betriebssystem erforderlich. Ein paralleler Zugriff auf die Java-Online-Dokumentation kann Hilfe zu den Java-Standard-Klassen bieten.

Aufbau des Buches

Jedes Kapitel beginnt mit einer kurzen und knappen Wiederholung des Stoffes, der in den Übungsaufgaben dieses Kapitels verwendet wird. Danach folgen alle

Aufgabenstellungen der Übungen. Am Ende des Kapitels finden Sie gesammelt die Lösungen der Übungsaufgaben mit Kommentaren, Erläuterungen und Hinweisen.

Die Aufgaben haben unterschiedliche Schwierigkeitsgrade. Dieser wird im Aufgabenkopf durch ein bis drei Sternchen gekennzeichnet:

1 Sternchen für besonders einfache Aufgaben, die auch von Anfängern leicht bewältigt werden können

2 Sternchen für etwas kompliziertere Aufgaben, die einen durchschnittlichen Aufwand benötigen

3 Sternchen für Aufgaben, die sich an geübte Programmierer richten und einen wesentlich höheren Aufwand oder die Kenntnis von speziellen Details erfordern

Die Programme aus früheren Übungen werden teilweise in späteren Übungen gebraucht und es wird auch immer wieder auf theoretische Zusammenhänge zurückgekommen oder hingewiesen.

Die Lösungsvorschläge haben umfangreiche Kommentare, sodass ein Verständnis für die durchgeführte Aufgabe auch daraus abgeleitet werden kann und dadurch jede einzelne Aufgabe im Gesamtkontext unabhängig erscheint.

In den Kapiteln 1, 2 und 3 liegt das Hauptmerkmal auf den Eigenheiten der objektorientierten Programmierung mit Java. Durch eine Vielzahl von Beispielen wird gezeigt, was die Java-Standard-Klassen und Interfaces an Funktionalitäten bieten und wie diese sinnvoll in die Definition von eigenen Klassen eingebettet werden können. Diese Kapitel enthalten zusätzlich Informationen zur Reflection-API von Java, der Definition von Annotationen und inneren Klassen sowie Neuerungen aus den Versionen 8 bis 13, die sich auf die neue Date&Time-API, Textblöcke, Compact Strings und die Weiterentwicklung von Interfaces beziehen. Mit Java 8 wurden sogenannte Default-Methoden eingeführt. Diese werden in der Literatur auch als »virtual extension«- bzw. »defender«-Methoden bezeichnet und Schnittstellen, die über derartige Methoden verfügen, als erweiterte Schnittstellen. Damit können Interfaces zusätzlich zu abstrakten Methoden konkrete Methoden in Form von Standard-Implementierungen definieren und in Java wird die Mehrfachvererbung von Funktionalität ermöglicht. Neben Default-Methoden können Interfaces in Java

nun auch statische Methoden enthalten. Anders als die statischen Methoden von Klassen werden diese jedoch nicht von abgeleiteten Typen geerbt.

Kapitel 4 beschäftigt sich im Detail mit Generics und dem Collection Framework mit all seinen generischen Klassen und Interfaces sowie mit der Definition von Enumerationen. Die Typinferenz für Methoden und beim Erzeugen von generischen Typen (der Diamond-Operator) sowie das Subtyping von parametrisierten und Wildcard-parametrisierten Typen sind ebenfalls Gegenstand der Themen aus diesem Kapitel.

Kapitel 5 erläutert das Exception-Handling.

Kapitel 6 beschäftigt sich mit den neuen Sprachmitteln von Java 8, Lambdas und Streams sowie mit weiteren Neuerungen aus den Versionen 8 bis 13, wie Switch-Expressions und Local Variable Type Inference.

Mit der Java-Version 8.0 haben sich ganz neue Betrachtungsweisen und Programmiertechniken in der Entwicklung von Applikationen mit Java eröffnet. Eine der wichtigsten Neuerungen in Java 8 sind neue Sprachmittel, die sogenannten Lambda-Ausdrücke, eine Art anonyme Methoden, die auf funktionalen Interfaces basieren. Diese besitzen jedoch eine viel kompaktere Syntax als Methoden. Das resultiert daraus, dass in ihrer Benutzung auf Namen, Modifikatoren, Rückgabetyp, throws-Klausel und in vielen Fällen auch auf Parameter verzichtet werden kann. Mit ihnen kann Funktionalität ausgeführt, gespeichert und übergeben werden, wie dies bisher nur von Instanzen in Java bekannt war.

Damit verbundene Themen wie die Gegenüberstellung zu anonymen Klassen, Syntax und Semantik, Behandlung von Exceptions, Scoping und Variable Capture, Methoden- und Konstruktor-Referenzen werden in den ersten Unterkapiteln des 6. Kapitels dieses Buches beschrieben und anhand von vielen Beispielen erläutert.

Des Weiteren finden Sie hier die Beschreibung aller neuen funktionalen Interfaces und deren Methoden. Die nachfolgenden Unterkapitel beschäftigen sich im Detail mit der Definition und Nutzung von Streams. Ein Stream besteht aus einer Folge von Werten (in der Literatur wird auch von Sequenzen von Elementen gesprochen), die nur teilweise von mehreren in einer Pipeline dazwischenliegenden Operationen ausgewertet und durch eine abschließende Operation bereitgestellt werden. Diese Operationen werden in Java als Methodenaufrufe formuliert, die Funktionalität in Form von Lambdas und Methoden-Referenzen entgegennehmen können und diese auf alle Elemente der Folge anwenden.

Mit einer Vielzahl von Aufgaben basierend auf Lambdas, Streams und Kollektoren (in denen Stream-Elemente angesammelt und reduziert werden können) werden die neuen Techniken angewandt und alle neuen Begriffe erklärt.

Kapitel 7 präsentiert das neue Java-Modulsystem. Mit dem neuen Modulsystem wurde Java selbst modular gemacht und es können eigene Applikationen und Bibliotheken modularisiert werden.

Java 9 führt das Modul als eine neue Programmkomponente ein. Das Erzeugen von Modulen und deren Abhängigkeiten führt dazu, dass der Zugriffsschutz in Java 9 restriktiver ist. Das Anlegen der erforderlichen Verzeichnisstrukturen für modulbasierte Applikationen, das Packaging von Modul-Code sowie die Implementierung von Services werden ebenfalls im Detail erklärt. Eine Vielzahl von Applikationen mit ausführlichen `.cmd`-Dateien für deren Ausführung ergänzen die theoretischen Erläuterungen aus diesem Kapitel.

Weitere Sprachänderungen und Neuigkeiten aus den Versionen Java 10 bis Java 13 sind in allen Kapiteln integriert. Dazu gehören der Diamond-Operator für innere anonyme Klassen, Default- und private Methoden in Interfaces, Erweiterungen der `switch`-Anweisung und die Einführung von Switch-Expressions, die Typinferenz für lokale Variablen und nicht zuletzt die mit Java 13 neu eingeführten Textblöcke, die bereits im ersten Kapitel im Vergleich zu den klassischen String-Literalen ausführlich und mit vielen Beispielen präsentiert werden.

Weil der Schwerpunkt des Buches nicht auf der Umsetzung von aufwendigen Algorithmen liegen soll, verwende ich einfache Beispiele mit Zahlen, Buchstaben, Wörtern, Büchern, Wochentagen, geometrischen Figuren etc. und teilweise auch mit ganz abstrakten Klassennamen wie `Klasse1`, `Klasse2`, `KlasseA`, `KlasseB` etc.

An dieser Stelle möchte ich auf das dem Buch zugrunde liegende Konzept hinweisen, das parallel zu einfachen Aufgaben, die zu allen eingeführten Definitionen und Begriffen gebracht werden, auch Aufgaben von einem höheren Schwierigkeitsgrad präsentiert werden. Dabei werden anhand von inhaltlichen Zusammenhängen zwischen den Beispielen viele Basiskonzepte von Java erläutert.

Ich habe generell versucht, keine Begriffe, Klassen und Komponenten zu benutzen, die nicht schon in vorangehenden Beispielen und Kapiteln definiert oder erläutert wurden. In den wenigen Fällen, wo es sich nicht vermeiden ließ, wird darauf hingewiesen und auf die entsprechenden Stellen verwiesen.

Das Buch soll möglichst parallel zu einer Vielzahl von Java-Lehrbüchern eingesetzt werden können und einen Beitrag dazu leisten, die große Fülle von Informationen, die auf uns über die API-Dokumentation zukommt, besser einzuordnen und korrekt anwenden zu können.

Schrecken Sie nicht davor zurück, von Anfang an (gerade bei den schwierigeren Aufgaben) die Anforderungen aus dem Aufgabentext mit den Ergebnissen aus dem Lösungsvorschlag zu vergleichen (zumindest anfangs und vielleicht auch nur teilweise). Nahe an der Programmiersprache formuliert, sollen diese Aufgaben in erster Linie dazu dienen, das Programmieren mit Java zu erlernen, ohne sich gleichzeitig auf aufwendige Algorithmen zu konzentrieren. Es ist ja auch mit ein Grund, warum die Bücher vollständige Lösungsvorschläge beinhalten. Sie sind gerade dafür gedacht, die Theorie besser zu verstehen, aber auch gleichzeitig mit Beispielen einzuüben.

Andererseits verpflichten diese Bücher nicht, selbst auf die gleiche Lösung zu kommen, und enthalten nur Vorschläge zu den Lösungen von Aufgaben.

Anhand eines umfangreichen Index können Sie beim selbstständigen Programmieren im Buch nachschlagen, wenn Sie nach einem Lösungsansatz oder Hilfe beim Beseitigen von Fehlern suchen sollten.

Benötigte Software

Das aktuelle Java Development Kit der Java Standard Edition können Sie sich kostenlos von der Java-Homepage von Oracle `http://www.oracle.com/technetwork/java/javase/downloads/index.html` herunterladen. Das JDK umfasst sowohl die Software zur Programmerstellung als auch das JRE (Java Runtime Environment) für die Programmausführung.

Grafische Entwicklungsoberflächen (wie z. B. Eclipse) sind keine Voraussetzung und auch nicht Bestandteil dieses Buches. Die Programme lassen sich grundsätzlich mit einem Texteditor wie z. B. Notepad++ oder auch Wordpad eingeben und über die Kommandozeile durch den Aufruf der Programme javac, jar und java übersetzen, paketieren und starten. Die vollständigen Programmaufrufe sind bei jeder Aufgabe angegeben.

Sollte Ihnen beim Erlernen der Programmiersprache selbst eine Entwicklungsumgebung nicht zu aufwendig oder unübersichtlich erscheinen, steht Ihnen nichts im Wege, die zu den Aufgaben zugehörigen Programm- und Klassendateien zum Testen oder auch Ergänzen in eine solche einzubetten.

Website

Die Website zum Buch unter `www.mitp.de/0123` beinhaltet den plattformunabhängigen Quellcode der Lösungsvorschläge und die für Windows kompilierte ausführbare Version als Download-Archiv. Diese Archivdatei enthält alle Java-Quellcodes, übersetzten Klassen und Bilddateien in einer Verzeichnisstruktur, die mit der im Buch beschriebenen übereinstimmt. Zusätzlich finden Sie auf dieser Webseite die Datei `Java9Migration.pdf`, in der die Migration von Anwendungen beschrieben wird, die durch verschiedene Kategorien von Modulen unterstützt werden kann.

Ich wünsche Ihnen viel Erfolg beim Programmieren mit Java.

Elisabeth Jung