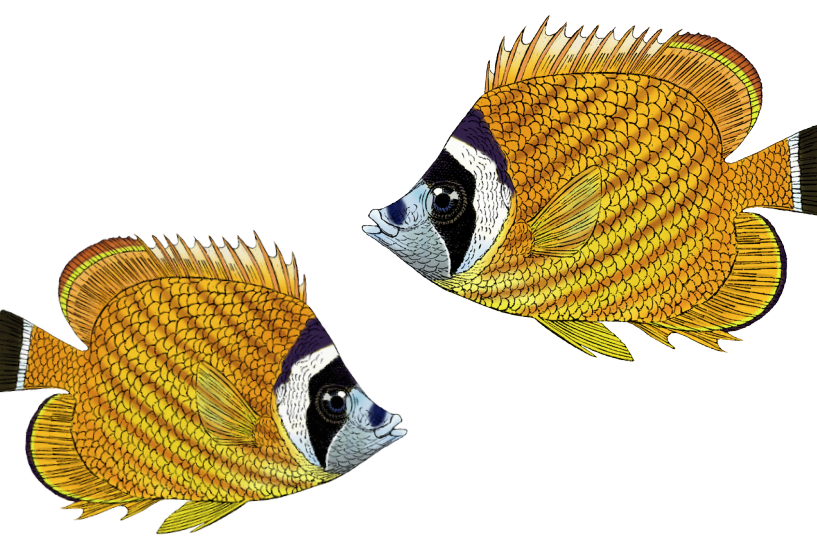
O'REILLY®

Java lernen

kurz & gut

O'Reillys Taschenbibliothek



Michael Inden

Java lernen

kurz & gut

Michael Inden

O'REILLY®

Michael Inden
michael_inden@hotmail.com

Lektorat: Dr. Benjamin Ziech
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Michael Inden
Herstellung: Stefanie Weidner, Frank Heidt
Umschlaggestaltung: Karen Montgomery, Michael Oréal, *www.oreal.de*
Druck und Bindung: BELTZ Grafische Betriebe GmbH, Bad Langensalza

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 978-3-96009-204-9
PDF 978-3-96010-708-8
ePub 978-3-96010-709-5
mobi 978-3-96010-710-1

1. Auflage 2024

Wieblinger Weg 17
69123 Heidelberg

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *kommentar@oreilly.de*.

5 4 3 2 1 0

Für unseren Sonnenschein und Wirbelwind Sophie Jelena.

Inhaltsverzeichnis

Vorwort

Zunächst einmal bedanke ich mich bei Ihnen, dass Sie sich für dieses Buch entschieden haben. Hierin finden Sie einen fundierten und interaktiven Einstieg in die Programmierung mit Java. Das Ganze startet bei den Grundlagen und basierend darauf wird Ihr Wissen immer weiter ausgebaut, sodass Sie nach der Lektüre bereit sind, eigene Experimente zu wagen, und bestenfalls Programmieren als neues Hobby lieben gelernt haben. Insbesondere die ungeheuren Möglichkeiten, kreativ zu werden und dabei immer wieder Neues zu entdecken, werden Sie bestimmt ähnlich faszinieren wie mich seit über 30 Jahren.

Zielgruppe

Dieses Buch richtet sich an Programmierneulinge und wendet sich somit insbesondere an

- **Schüler und Schülerinnen**, die ein paar Tipps und Hilfestellungen suchen, die das Nachvollziehen des Informatikunterrichts erleichtern,
- **Studierende**, die ergänzende Erklärungen zu denen aus den Vorlesungen suchen, um Gelerntes schneller anwenden zu können oder besser für die nächste Prüfung vorbereitet zu sein,
- und **alle Interessierten**, die einfach die wunderbare und vielfältige Welt der Programmierung mit Java kennenlernen möchten.

Voraussetzungen

Zum Einstieg sind Programmiererfahrungen keine zwingende Voraussetzung – natürlich schaden diese nicht. Selbst dann

nicht, wenn Sie sich vielleicht eher mit Python, C#, TypeScript oder JavaScript beschäftigt haben. Für die Lektüre des Buchs ist es aber hilfreich, wenn Sie

- einigermaßen fit im Installieren von Programmen sind und
- wissen, was die Kommandozeile ist und sie grundlegend bedienen können.

Was vermittelt dieses Buch?

Sie als Leser erhalten in diesem Buch einen Einstieg in Java. Allerdings ist die trockene Theorie auf ein Minimum reduziert und wir legen immer mit kleinen Beispielen los. Deshalb ist es ein Buch zum Mitmachen. Ich ermutige Sie, parallel zum Lesen auch immer ein paar Dinge auszuprobieren, vielleicht sogar mal das eine oder andere abzuwandeln. Man lernt Programmieren einfach am besten, wenn man es praktiziert. Somit bietet es sich an, die abgebildeten Codeschnipsel abzutippen. Das kann in der Konsolenapplikation JShell erfolgen oder im Editor Ihrer Entwicklungsumgebung/IDE (Integrated Development Environment). Beide Varianten werden im Verlauf des Buchs genauer beschrieben.

Damit Sie nicht über einfache Probleme stolpern, führt das Buch jeweils behutsam und schrittweise in die jeweilige Thematik ein und gibt Ihnen immer auch ein paar Hinweise, auf was man achten oder was man vielleicht sogar vermeiden sollte. Dazu dienen diverse Hinweise mit Hintergrundinformationen.

Hinweis: Hintergrundinformation

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

Aufbau dieses Buchs

Dieses Buch besteht aus jeweils in sich abgeschlossenen, aber aufeinander aufbauenden Kapiteln zu elementar wichtigen Bereichen der Programmiersprache Java. Für Ihren erfolgreichen Weg zur Java-Programmierung gliedert sich das Buch in die beiden Teile Einstieg und Aufstieg.

Im Teil »Einstieg« werden Grundlagen behandelt. Hier empfiehlt es sich, die Kapitel in der Reihenfolge des Buchs zu lesen, da mit jedem Kapitel neue Wissensbausteine und Themen hinzukommen, die im Anschluss vorausgesetzt und verwendet werden.

Dann folgt der Teil »Aufstieg«. Dort beschäftigen wir uns mit leicht fortgeschrittenen Themen. Hier können Sie zwar nach Lust und Laune eins der Kapitel zur Lektüre auswählen, aber auch hier bauen einige Themen aufeinander auf.

Einstieg

Kapitel 1 – Einführung Dieses Kapitel gibt einen Überblick über Javas mittlerweile über 25-jährige Geschichte. Zudem wird die Installationen von Java an sich und einer Entwicklungsumgebung/IDE sowie das Ausprobieren der Beispiele beschrieben.

Kapitel 2 – Schnelleinstieg Dieses Kapitel bietet einen Schnelleinstieg und stellt viele wesentliche Elemente von Java vor. Dabei wird behutsam Fahrt aufgenommen: Wir beginnen mit einer Ausgabe eines Textes, traditionell »Hello World«, und lernen dann, wie wir das mithilfe von Variablen variieren. Zudem schauen wir uns bedingte Ausführungen mit Fallunterscheidungen sowie Wiederholungen mit Schleifen an.

Kapitel 3 – Strings Variablen vom Typ `String` repräsentieren Zeichenketten und dienen zur Verwaltung von textuellen Informationen. Diese sind aus kaum einem Programm wegzudenken und werden in diesem Kapitel genauer behandelt.

Kapitel 4 – Arrays Ebenso wie Strings sind auch Arrays recht gebräuchliche Datenstrukturen und helfen dabei, mehrere gleichartige Dinge zu speichern, etwa eine Menge von Zahlen, Namen, Personen usw. Insbesondere bilden Arrays die Grundlage für viele andere Datenstrukturen. In diesem Kapitel lernen wir Arrays im Detail kennen.

Kapitel 5 – Klassen und Objektorientierung Immer wieder hört man, Java ist eine objektorientierte Sprache. Doch was bedeutet das? Zum Verständnis gibt dieses Kapitel einen Einblick in den objektorientierten Entwurf von Software. Dazu vermittle ich die grundlegenden Ideen von Zustand (Daten) in Kombination mit Verhalten (Funktionen auf diesen Daten) und wie man dies in Java formuliert.

Kapitel 6 – Collections Während Arrays ziemlich elementar sind, bieten die in Java integrierten Collections oder Containerklassen mehr Flexibilität und Komfort bei der Verwaltung von Daten. Insbesondere unterstützen die vordefinierten Listen, Mengen und Schlüssel-Wert-Abbildungen bei der Verwaltung anderer Objekte.

Kapitel 7 – Ergänzendes Wissen In diesem Kapitel werden Themen angesprochen, die zuvor aus didaktischen Gründen bewusst ausgelassen wurden, weil das Ganze sonst zu tief in die Details gegangen wäre und zu viel anderes Wissen vorausgesetzt hätte. Hier angelangt lohnt es sich, Ihre Java-Kenntnisse zu primitiven Typen, dem Ternary-Operator, Fallunterscheidungen mit `switch` usw. zu komplettieren.

Aufstieg

Kapitel 8 – Lambdas und Streams Dieses Kapitel stellt sowohl Lambda-Ausdrücke (kurz Lambdas) als auch das damit eng verbundene Stream-API vor. Beides sind essenzielle Bausteine von modernem Java und ermöglichen es, Lösungen oftmals elegant zu formulieren.

Kapitel 9 – Verarbeitung von Dateien Dieses Kapitel beschäftigt sich mit der Verarbeitung von Informationen aus Dateien. Dies ist für viele Anwendungen von großer Bedeutung, da Informationen oftmals nicht nur während der Programmlaufzeit von Interesse sind, sondern vor allem auch darüber hinaus – denken Sie etwa an die Highscore-Liste Ihres Lieblingsspiels.

Kapitel 10 – Fehlerbehandlung mit Exceptions Vielleicht kennen Sie es schon: Manchmal tritt ein Programmfehler auf und das Programm stürzt ab. Wichtige Daten gehen potenziell verloren. So etwas ist ärgerlich. Daher gehört auch die Behandlung von Fehlern zum guten Ton beim Programmieren. Dieses Kapitel führt in die Thematik ein.

Kapitel 11 – Datumsverarbeitung Während früher die Datumsverarbeitung eher stiefmütterlich in Java unterstützt wurde, bietet modernes Java eine Vielzahl praktischer Funktionalitäten zur Datumsverarbeitung, die in diesem Kapitel einführend dargestellt werden.

Kapitel 12 – Schlusswort Hier rekapitulieren wir kurz, was Sie durch die Lektüre dieses Buchs gelernt haben sollten und wie Sie möglicherweise weitermachen können.

Anhang

Anhang A – Schlüsselwörter im Überblick In Java existiert eine Reihe von Schlüsselwörtern, die reserviert sind und nicht als Bezeichner für Variablen, Methoden, Klassen oder anderes verwendet werden dürfen. Hier erhalten Sie einen Überblick.

Anhang B – Schnelleinstieg JShell In diesem Buch werden diverse Beispiele direkt auf der Konsole ausprobiert. Dabei hilft die interaktive Kommandozeilenapplikation JShell als REPL (Read-Eval-Print-Loop).

Anhang C – Grundlagen zur JVM In diesem Anhang vermittle ich Grundwissen zur JVM (Java Virtual Machine).

Sourcecode und ausführbare Programme

Beim Erlernen des Programmierens ist es hilfreich, Beispiele selbst auszuprobieren und abzutippen. Um Ihnen ein wenig Tipparbeit und Mühe zu ersparen, finden Sie viele der Beispiele als Programme in einem Eclipse-Projekt. Dieses steht unter `https://oreilly.de/produkt/java-lernen-kurz-gut/` zur Verfügung. Weitere Informationen zum genauen Vorgehen finden Sie auf der Download-Seite.

Verwendete Java-Version(en)

Viele Beispiele wurden auf Basis von Java 17 entwickelt und ausprobiert. Mittlerweile ist bereits Java 21 erschienen. Hilfreiche Features dieser Version setze ich passend ein. Für dieses Buch sind die brandaktuellen Java-Features zwar von Interesse, aber nicht von entscheidender Bedeutung, da es ja um die Grundlagen der Sprache geht.

Nach der Lektüre dieses Buchs sind Sie bestens gerüstet für modernes Java und können nach Lust und Laune eigene Experimente und Hobbyprojekte starten.

Konventionen

Verwendete Zeichensätze

Neben der vorliegenden Schriftart sind wichtige Textpassagen *kursiv* oder ***kursiv und fett*** markiert. Englische Fachbegriffe werden eingedeutscht großgeschrieben, etwa Event Handling. Zusammensetzungen aus englischen und deutschen (oder eingedeutschten) Begriffen werden mit Bindestrich verbunden, z. B. Plugin-Manager. Listings mit Sourcecode sind in der Schrift `Courier` gesetzt, um zu verdeutlichen, dass diese einen Ausschnitt aus einem Java-Programm darstellen. Auch im normalen Text wird für Klassen, Methoden, Konstanten und Parameter diese Schriftart genutzt.

Schreibweise von Methodenaufrufen

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 mitunter auf deren Angabe aus Gründen der besseren Lesbarkeit verzichtet.

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
IDE	Integrated Development Environment
JDK	Java Development Kit
JEP	JDK Enhancement Proposal
JLS	Java Language Specification
JRE	Java Runtime Environment
JVM	Java Virtual Machine

Gerne möchte ich noch den Begriff API (Application Programming Interface) kurz erklären: Dabei handelt es sich um die von einem Programm bereitgestellten Funktionalitäten in Form von Methoden.

Danksagung

Dieses Buch basiert auf meinem Buch »Einfach Java« [3], das in weiten Teilen leicht überarbeitet wurde und somit ein Facelift erfahren hat. Dazu wurden diverse Passagen und Beispiele ergänzt und derart modifiziert, dass sie noch besser verständlich sind. Auch potenziell missverständliche Formulierungen

wurden korrigiert und sogar kleinere Teile entfernt, um dem Text mehr Stringenz zu verleihen und sich in die Kurz-und-gut-Serie einzupassen. Aufgrund der Basis »Einfach Java« möchte ich hier nochmals folgenden Leuten danken: Michael Kulla für das Aufdecken von Tippfehlern bis hin zu diversen inhaltlichen Hinweisen, Prof. Dr. Dominik Gruntz für eine Vielzahl an hilfreichen Anmerkungen, Jean-Claude Brantschen und Christian Heitzmann für ihre Kommentare und Tipps und schließlich Maria Herdt für einen kritischen Blick auf einige Kapitel.

Zudem geht ein Dankeschön an das Team des dpunkt.verlags (Dr. Michael Barabas, Dr. Benjamin Ziech, Julia Griebel und Stefanie Weidner) für die tolle Zusammenarbeit. Außerdem möchte ich mich bei Torsten Horn für die fundierte fachliche Durchsicht sowie bei Ursula Zimpfer für ihre Adleraugen beim Copy-Editing bedanken.

Abschließend geht ein lieber Dank an meine Frau Lilija für ihr Verständnis und die Unterstützung. Ihren ganz besonderen Anteil hat unser kleiner Sonnenschein Sophie Jelena dazu beigetragen, indem sie den Papa immer wieder zum Lachen gebracht hat.

Anregungen und Kritik

Trotz großer Sorgfalt und mehrfachen Korrekturlesens lassen sich missverständliche Formulierungen oder teilweise sogar Fehler leider nicht vollständig ausschließen. Falls Ihnen etwas Derartiges auffallen sollte, so zögern Sie bitte nicht, mir dies mitzuteilen. Gerne nehme ich auch Anregungen oder Verbesserungsvorschläge entgegen. Kontaktieren Sie mich bitte per Mail unter:

michael_inden@hotmail.com

Zürich, im Oktober 2023
Michael Inden

Einstieg

1 Einführung

1.1 Java im Überblick

Die Programmierspache Java wurde Mitte der 1990er-Jahre von der Firma Sun entwickelt und später von Oracle übernommen. Mittlerweile hat Java zwar mehr als 25 Jahre auf dem Buckel, wird aber nicht altersschwach, sondern kontinuierlich gepflegt und besitzt ein extrem breitgefächertes und professionelles Angebot an externen Bibliotheken und Entwicklungstools. Insbesondere gibt es mit Eclipse, IntelliJ IDEA und NetBeans sowie Visual Studio Code mehrere hervorragende sogenannte IDEs (Integrated Development Environments) zum komfortablen Programmieren.

Java eignet sich zur Entwicklung unterschiedlichster Applikationen, etwa für Businessapplikationen, Webapplikationen und sogar (einfachere) Datenbanken. Es wird aber auch im Bereich Mobile in Form von Android-Apps oder gar im Bereich von Spielen, z. B. Minecraft, verwendet. Java ist also eine vielseitige Programmiersprache mit breitem Einsatzspektrum. Ein guter Grund, sich damit ein wenig zu beschäftigen.

Außerdem ist Programmieren ein wunderbares Hobby sowie ein faszinierender Beruf und es macht zudem noch jede Menge Spaß, fördert die Kreativität und den Gestaltungswillen.

Darüber hinaus ist Java laut TIOBE-Index[1] seit Jahren eine der populärsten Programmiersprachen. Eine wichtige Rolle spielte vermutlich lange die freie Verfügbarkeit, was zwischenzeitlich für die Originalvariante von Oracle nur noch für private, aber nicht für kommerzielle Zwecke galt. Mit Java 17 wurde das Releasemodell im September 2021 wieder auf eine freie Verfügbarkeit zurück geändert. Praktischerweise existieren einige

[1] `https://www.tiobe.com/tiobe-index/`

frei verwendbare Alternativen wie etwa das AdoptOpenJDK – das mittlerweile als Eclipse Adoptium weitergeführt wird: `https://projects.eclipse.org/projects/adoptium`. Zudem ist Java recht einfach zu erlernen (schwieriger als Python, aber deutlich leichter als C++) und bietet eine große Vielfalt an Tools, Bibliotheken und Literatur sowie Hilfestellungen wie StackOverflow im Internet. Weiterhin zeichnet sich Java durch seine gute Performance aus. Das Ganze ist die Folge von jahrelangen Optimierungen und Verbesserungen. Dadurch ist die Ausführung von Java beispielsweise nur geringfügig langsamer als die von C++, aber um Längen schneller als die Ausführung von Python. Schließlich ermöglicht Java je nach Einsatzzweck entweder einen imperativen Stil oder die objektorientierte sowie die funktionale Programmierung, sodass man geeignet wählen kann.

Wie Sie sehen, sprechen viele gute Gründe für einen Einstieg in die Programmierung mit Java. Das Wichtigste ist jedoch der Spaß am Programmieren, Tüfteln und Ausprobieren. Lassen Sie uns starten!

Bestandteile von Java-Programmen

Java als Programmiersprache besitzt wie eine natürliche Sprache auch eine Grammatik und feststehende Begriffe/Wörter. Man spricht dabei von Syntax und Schlüsselwörtern (vgl. Anhang A).

Java-Programme werden textuell verfasst. Das wird ***Sourcecode*** genannt. Schauen wir uns zum Einstieg ein einfaches Java-Programm an:

```
public class MyFirstJavaProgram
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

Keine Sorge, Sie müssen das Ganze noch nicht vollständig verstehen, wir werden das alles Stück für Stück erlernen. Hier

ist zunächst nur wichtig, dass Sie elementare Bestandteile von Java-Programmen grob einordnen können. Dazu gehören die ***Schlüsselwörter***, also Java-Befehle oder -Anweisungen, hier etwa `public`, `class`, `static` und `void`. Wie die Begriffe in einer Sprache tragen diese reservierten Wörter eine besondere Bedeutung, ganz analog etwa zu Auto, Haus, Tür usw. im Deutschen.

Ebenso wie im Deutschen können (oder besser sollten) die Begriffe nicht einfach wahllos miteinander verknüpft werden, um einen gültigen Satz zu formulieren. Das wird durch die Grammatik geregelt. Auch in Java existiert eine solche. Damit wird etwa festgelegt, dass es `static void`, aber nicht `void static` heißen muss. Man spricht hier auch von ***Syntax***.

Zudem sehen wir geschweifte Klammern. Diese kann man sich wie Absätze in einem Text vorstellen. In Java bündeln diese Klammern Anweisungen. Man spricht dann auch von Blöcken und Sichtbarkeitsbereichen.

Genug der Vielzahl an Informationen. Nachfolgend werden wir die Dinge schön gründlich und detailliert besprechen und didaktisch immer ein neues Themengebiet ergründen, bis wir schließlich einen guten Einstieg in die Java-Programmierung vollzogen haben werden.

Vorab sollten wir erst einmal Java und Eclipse installieren, um erste Schritte machen zu können und für unsere weitere Entdeckungsreise bereit zu sein.

1.2 Los geht's – Installation

Im ersten Teil dieses Buchs wird ein Hands-on-Ansatz verfolgt, bei dem wir Dinge in Form kleinerer Java-Codeschnipsel direkt ausprobieren. Sie benötigen hierfür keine tiefgreifenden Programmiererfahrungen.

Damit Sie die nachfolgend beschriebenen Java-Programme ausführen können, benötigen Sie ein sogenanntes JDK (Java Development Kit). Dort finden sich alle für den Moment benötigten Tools. Beginnen wir also mit der Installation von Java.

1.2.1 Java-Download

Java ist frei auf der folgenden Oracle-Webseite verfügbar: `https://www.oracle.com/java/technologies/downloads/`

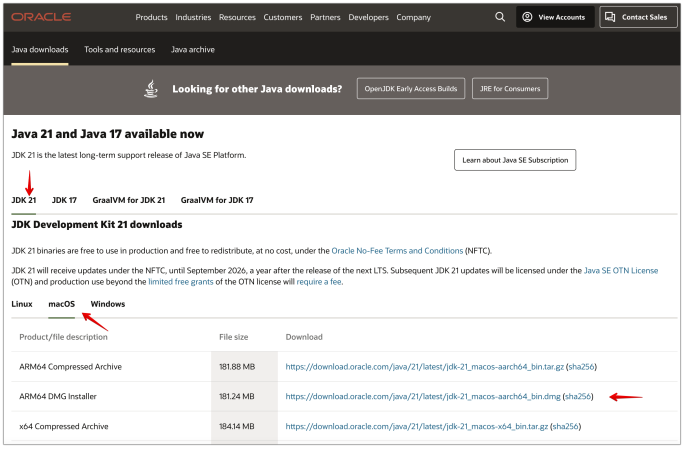


Abb. 1–1: Java-Download-Seite

Im unteren Bereich finden Sie verschiedene Links für unterschiedliche Betriebssysteme. Wählen Sie den für Sie passenden Link und laden die entsprechende Installationsdatei herunter.

1.2.2 Installation des JDKs

Zum Start der Installation doppelklicken Sie unter macOS auf die `.dmg`-Datei und folgen den Aufforderungen. Möglicherweise müssen Sie das Administrator-Passwort eingeben, um fortzufahren. Nach Abschluss der Installation können Sie die `.dmg`-Datei löschen, um Speicherplatz zu sparen.

Für Windows doppelklicken Sie bitte auf die `.exe`-Datei. Auch diese kann nach erfolgreicher Installation gelöscht werden. Führen Sie das heruntergeladene Installationsprogramm aus (z. B. `jdk-21_windows-x64_bin.exe`). Damit wird Java standardmäßig in das Verzeichnis `C:\Program Files\Java\jdk-21` installiert, wobei der Verzeichnisname von der gewählten Version abhängt. Akzeptieren Sie die Standardeinstellungen und befolgen Sie die Anweisungen während der Installation.

1.2.3 Installationsnacharbeiten

Damit Java bei Ihnen nach dem Download und der Installation in der Konsole korrekt funktioniert, sind ein paar Nacharbeiten nötig. Dazu sollten wir es zur leichteren Handhabung in den Pfad aufnehmen. Dies wird im Anschluss für die Betriebssysteme Windows und macOS beschrieben. Falls Sie ein Unix-Derivat nutzen, dann finden Sie ausführlichere Informationen auf dieser Seite: `https://www.java.com/de/download/help/download_options.html`. Für Windows und Mac gibt es dort auch noch einige ergänzende Informationen.

Nacharbeiten für Windows

Das Installationsverzeichnis sollte in die Umgebungsvariable `PATH` aufgenommen werden. Diese können Sie unter »Umgebungsvariablen« ändern. Drücken Sie die Win-Taste und geben Sie dann »umgeb« ein, bis »Systemumgebungsvariablen bearbeiten« erscheint. Mit Enter öffnet sich der Dialog »Systemvariablen«. Klicken Sie auf den Button »Bearbeiten« zum Öffnen eines Bearbeitungsdialogs. Fügen Sie in der Liste das Installationsverzeichnis gefolgt von `bin`, etwa `C:\Program Files\Java\jdk-21\bin`, hinzu. Legen Sie eine Umgebungsvariable namens `JAVA_HOME` an, die auf das Installationsverzeichnis verweist:

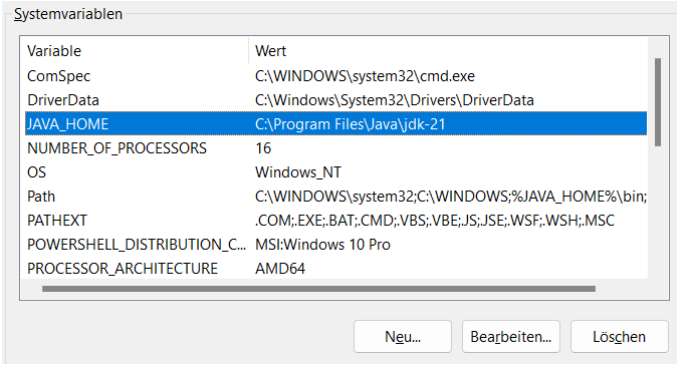


Abb. 1–2: Umgebungsvariablen bearbeiten

Ergänzen Sie die Variable `PATH` wie nachfolgend gezeigt um den Eintrag `%JAVA_HOME%\bin`. Außerdem sollte der Eintrag möglichst ganz oben stehen:

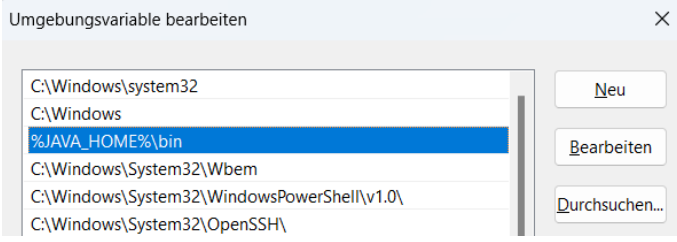


Abb. 1–3: Umgebungsvariablen ordnen

Beachten Sie bitte noch Folgendes: Bestätigen Sie die gesamten Dialoge immer mit OK, sodass die Variablen gesetzt sind. Eventuell geöffnete Eingabeaufforderungen müssen geschlossen und neu geöffnet werden, um die geänderten Variablen wirksam werden zu lassen.

Nacharbeiten für macOS

Auch unter macOS empfiehlt es sich, einen Verweis auf Java im Pfad in der jeweiligen Shell (dem Terminal) passend zu setzen bzw. in das Startskript Ihrer Shell einzutragen, etwa `~/.bash_profile` oder neuer `~/.zshrc`:

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/
      Contents/Home
export PATH=$JAVA_HOME/bin:$PATH
```

1.2.4 Java-Installation prüfen

Nach dem Ausführen der obigen Schritte sollte Java auf Ihrem Rechner installiert und von der Konsole startbar sein und Sie damit bereit für die nächsten Schritte.

Öffnen Sie eine Konsole und geben Sie das Kommando `java --version` ein – im folgenden Text nutze ich immer $ zur Kennzeichnung von Eingaben auf der Konsole, also dem Terminal bei macOS bzw. der Windows-Eingabeaufforderung. Wenn die Ausgabe ähnlich zu dieser erfolgt, dann haben Sie Java erfolgreich installiert.

```
$ java --version
java 21 2023-09-19 LTS
Java(TM) SE Runtime Environment (build 21+35-LTS-2513)
Java HotSpot(TM) 64-Bit Server VM (build 21+35-LTS-2513, mixed mode
      , sharing)
```

JShell prüfen

Prüfen Sie der Vollständigkeit halber bitte auch noch den Aufruf sowie das Beenden des Tools JShell, das wir für den ersten Teil des Buchs intensiv nutzen werden:

```
$ jshell
|  Willkommen bei JShell - Version 21
|  Geben Sie für eine Einführung Folgendes ein: /help intro

jshell> /exit
|  Auf Wiedersehen
```

Wenn die Programme bzw. Tools starten und Sie ähnliche Meldungen erhalten (möglicherweise mit kleinen Abweichungen bei den Versionsangaben), so können wir uns auf die Entdeckungsreise zur Java-Programmierung machen.

1.3 Entwicklungsumgebungen

Wenn Sie nach den ersten Gehversuchen mit Java damit beginnen, auch umfangreichere Java-Programme zu schreiben (also viel Sourcecode zu bearbeiten), dann empfiehlt sich der Einsatz einer IDE anstelle von Texteditoren oder anstatt rein auf der Konsole in der JShell zu arbeiten. Für kleine Experimente und zum Einstieg ist aber gerade die JShell ein wunderbares Hilfsmittel (zumindest in modernen Java-Versionen).

Zwar kann man für Änderungen an Java-Programmen einen Texteditor nutzen, aber dieser bietet nicht die Annehmlichkeiten einer IDE: In IDEs laufen viele Dinge und Analysen automatisch und im Hintergrund ab, wodurch gewisse Softwaredefekte direkt noch während des Modifizierens von Sourcecode (auch Editieren genannt) erkannt und angezeigt werden, etwa in einer To-do-/Task-Liste. IDEs bereiten zudem vielfältige Informationen auf. Weiterhin werden Annehmlichkeiten wie Quick Fixes zur Korrektur kleinerer Probleme sowie automatische Transformationen und Änderungen von Sourcecode, sogenannte ***Refactorings***, unterstützt.