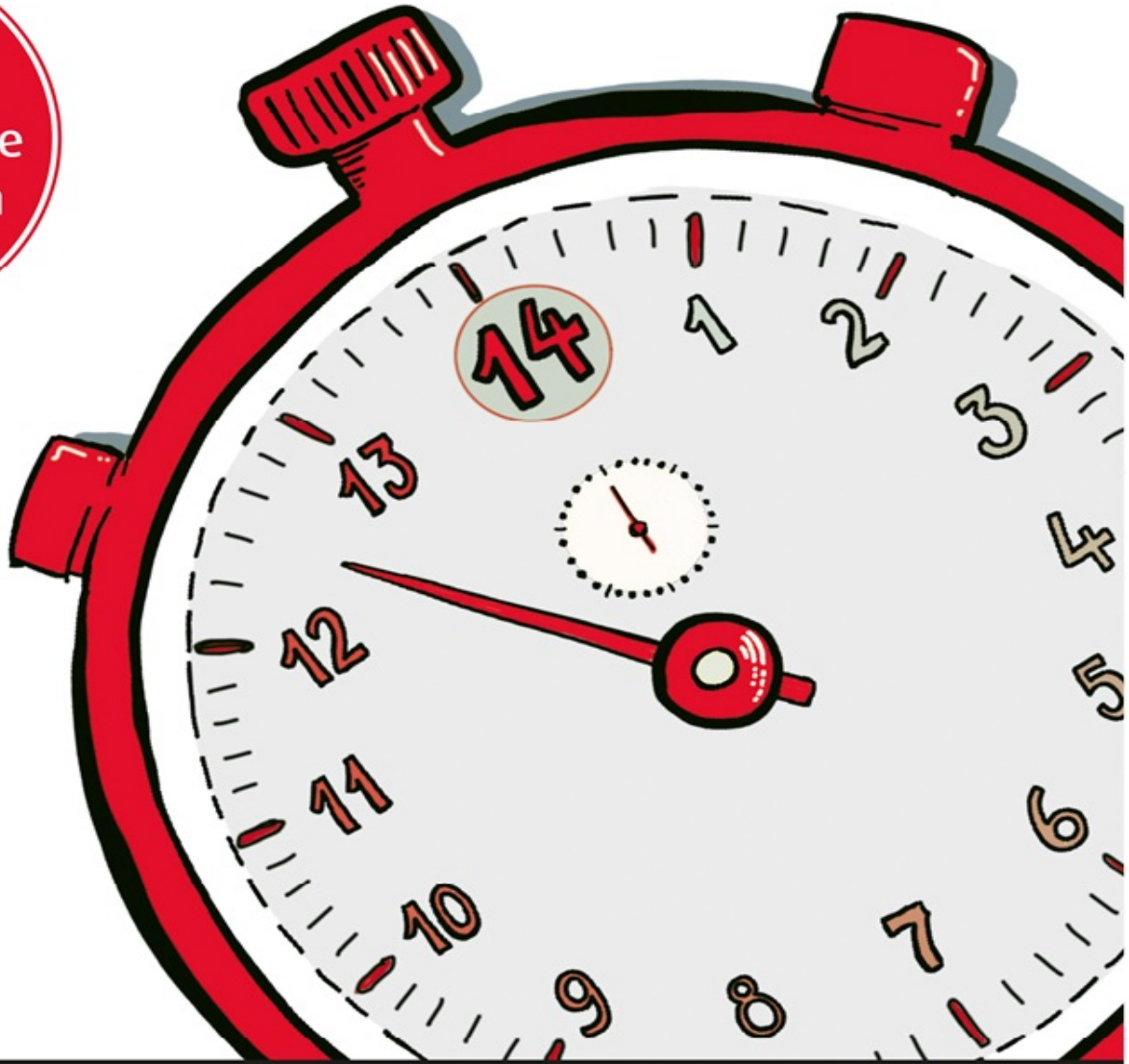
Daniel Braun

JAVA Schnelleinstieg

Programmieren lernen in 14 Tagen

Einfach und ohne Vorkenntnisse

Zahlreiche Praxisbeispiele und Übungen



mitp

Daniel Braun

JAVA Schnelleinstieg

Programmieren lernen in 14 Tagen

Einfach und ohne Vorkenntnisse

Zahlreiche Praxisbeispiele und Übungen

mitp

Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

Ihr mitp-Verlagsteam

Daniel Braun

Java

Schnelleinstieg

Programmieren lernen in 14 Tagen
Einfach und ohne Vorkenntnisse

Impressum

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 978-3-7475-0394-2
1. Auflage 2022

www.mitp.de

E-Mail: mitp-verlag@sigloch.de
Telefon: +49 7953 / 7189 - 079
Telefax: +49 7953 / 7189 - 082

© 2022 mitp Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Janina Bahlmann
Sprachkorrektorat: Petra Heubach-Erdmann
Covergestaltung: Janina Bahlmann, Christian Kalkert
Covergrafik & Icons: Tanja Wehr, sketchnotelovers
Electronic Publishing: Petra Kleinwegen

Dieses E-Book verwendet das ePub-Format und ist optimiert für die Nutzung mit dem iBooks-reader auf dem iPad von Apple. Bei der Verwendung anderer Reader kann es zu Darstellungsproblemen kommen.

Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Der Verlag räumt Ihnen mit dem Kauf des E-Books das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheherrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen,

Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine E-Books vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen E-Book-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Inhalt

Impressum

Einleitung

E.1 Programmieren lernen in 14 Tagen

E.2 Der Aufbau des Buches

E.3 Programmtexte und Lösungen zum Download

E.4 Fragen und Feedback

Kapitel 1: Erste Schritte mit Java

1.1 Programmiersprachen

1.2 Besonderheiten von Java

1.3 Installation und Einrichtung

1.3.1 Linux

1.3.2 macOS

1.3.3 Windows

1.3.4 Installation testen

1.4 Entwicklungsumgebungen

1.4.1 Eclipse

1.4.2 IntelliJ IDEA

Kapitel 2: Das erste Programm – »Hallo Welt!«

2.1 Datei erstellen

2.1.1 Eclipse

2.1.2 IntelliJ IDEA

2.2 Quelltext
2.2.1 Hallo Welt
2.2.2 Syntax
2.2.3 Kommentare
2.3 Kompilieren
2.3.1 Ohne IDE
2.3.2 Eclipse
2.3.3 IntelliJ IDEA
2.4 Fehler finden
2.5 Ausführen
2.5.1 Ohne IDE
2.5.2 Eclipse
2.5.3 IntelliJ IDEA
2.6 Übungen

Kapitel 3: Variablen

3.1 Funktionsweise von Variablen
3.2 Typen
3.2.1 Zahlen
3.2.2 Texte
3.2.3 Wahrheitswerte
3.3 Operatoren
3.3.1 Arithmetische Operatoren
3.3.2 String-Konkatenation
3.3.3 Boolesche Operatoren
3.4 Typumwandlung

3.5 Nutzereingaben verarbeiten
3.6 Übungen

Kapitel 4: Verzweigungen und Schleifen
4.1 Verzweigungen
4.1.1 if-Verzweigung
4.1.2 if-else und else if
4.1.3 Vergleichsoperatoren
4.1.4 case-Verzweigung
4.2 Schleifen
4.2.1 while-Schleife
4.2.2 do-while-Schleife
4.2.3 for-Schleife
4.2.4 Verschachtelung von Schleifen
4.3 Übungen

Kapitel 5: Funktionen
5.1 Grundlagen
5.2 Parameter
5.3 Rückgabewerte
5.4 Rekursion
5.5 Übungen

Kapitel 6: Objektorientierung
6.1 Die ganze Welt ist ein Objekt
6.1.1 Definition von Klassen
6.1.2 Instanzen und Objekte erzeugen

6.1.3 Attribute und Funktionen
6.2 Statische und nicht-statische Funktionen und Variablen
6.3 Verwendung mehrerer Klassen
6.4 Zugriffsrechte
6.5 Vererbung
6.6 Umwandlung zwischen verwandten Klassen (Typecasting)
6.7 Abstrakte Methoden und Klassen
6.8 Einlesen von Personendaten
6.9 Übungen

Kapitel 7: Arrays und Listen

7.1 Arrays
7.1.1 Deklaration
7.1.2 Lesen und Schreiben
7.1.3 Arrays und Schleifen
7.1.4 Personenverwaltung mit Arrays
7.1.5 Umwandlung zwischen Strings und Arrays
7.1.6 Mehrdimensionalität
7.1.7 Start-Parameter
7.2 Listen
7.2.1 Arten von Listen
7.2.2 Deklaration
7.2.3 Elemente hinzufügen
7.2.4 Elemente lesen
7.2.5 Elemente löschen

7.2.6 Elemente ändern
7.2.7 Listen durchlaufen
7.2.8 Unterschied zwischen ArrayList und LinkedList
7.2.9 Personenverwaltung mit Liste
7.3 Übungen

Kapitel 8: Fehlerbehandlung

8.1 Arten von Fehlern
8.2 Fehler abfangen
8.2.1 Umwandlungsfehler
8.2.2 Index-Fehler
8.2.3 Andere Fehler
8.2.4 Fehlerbehandlung bei Funktionen
8.3 Mehrere Arten von Fehlern abfangen
8.4 finally-Block
8.5 Übungen

Kapitel 9: Dateisystem

9.1 Textdateien schreiben
9.2 Textdateien lesen
9.3 Dateien löschen und umbenennen
9.4 Ordner
9.4.1 Erstellen
9.4.2 Löschen
9.4.3 Umbenennen
9.4.4 Inhalt anzeigen
9.5 Objekte speichern

9.5.1 CSV-Dateien
9.5.2 Binäre Dateien
9.6 Übungen

Kapitel 10: Externe Bibliotheken

10.1 JSON-Bibliothek
10.1.1 Das Format
10.1.2 Bibliothek einbinden
10.1.3 Bibliothek verwenden
10.2 Eigene .jar-Dateien erstellen
10.2.1 Bibliotheken
10.2.2 Ausführbare .jar-Dateien
10.3 Abhängigkeiten in großen Softwareprojekten
10.4 Übungen

Kapitel 11: Grafische Benutzerschnittstellen

11.1 Fenster
11.2 Textfelder
11.3 Buttons
11.4 Layout
11.4.1 BorderLayout
11.4.2 FlowLayout
11.4.3 GridLayout
11.5 Grafische Benutzerschnittstelle für die Personenverwaltung
11.5.1 Personendaten einlesen
11.5.2 Personendaten anzeigen

11.5.3 Personendaten bearbeiten
11.5.4 Personendaten exportieren
11.6 Übungen

Kapitel 12: Multitasking

12.1 Threads
12.2 Data Race
12.3 Synchronisierung
12.4 Locks
12.5 Conditions
12.6 Threads beenden
12.7 Übungen

Kapitel 13: Datenbanken

13.1 Datenbank einrichten
13.2 Verbindung herstellen
13.3 Tabelle erstellen
13.4 Daten hinzufügen
13.5 Daten lesen
13.6 Daten ändern
13.7 Nachteile und Alternativen
13.8 Übungen

Kapitel 14: REST-Schnittstellen

14.1 Funktionsweise von REST-Schnittstellen
14.2 Beispiele für das Ansprechen einer REST-Schnittstelle
14.2.1 GET

14.2.2 POST
14.2.3 PUT
14.2.4 DELETE
14.3 Implementierung in Java
14.3.1 Server-Konfiguration
14.3.2 Router
14.3.3 REST-Schnittstellen testen
14.4 Schnittstelle für Personendaten
14.4.1 Initialisierung
14.4.2 POST
14.4.3 GET
14.4.4 DELETE
14.4.5 PUT
14.5 Übungen
14.6 Nächste Schritte

Einleitung

E.1 Programmieren lernen in 14 Tagen

Mit diesem Buch haben Sie sich für einen einfachen, praktischen und fundierten Einstieg in die Welt der Programmierung entschieden. Sie lernen ohne unnötigen Ballast alles, was Sie wissen müssen, um Java effektiv für Projekte in Ihrem Berufs- und Interessensgebiet einzusetzen.

Wenn Sie Zeit genug haben, können Sie jeden Tag ein neues Kapitel durcharbeiten und so innerhalb von zwei Wochen Programmieren lernen. Alle Erklärungen sind leicht verständlich formuliert und setzen keine Vorkenntnisse voraus. Am besten lesen Sie das Buch neben der Computer-Tastatur und probieren die Programmbeispiele und Übungen gleich aus. Sie werden schnell erste Erfolge erzielen und Freude an der Programmierung finden.

E.2 Der Aufbau des Buches

Das Buch beginnt mit den Grundlagen: Installation von Java, Nutzung verschiedener Entwicklungsumgebungen und Formulierung einfacher Anweisungen. Die Kapitel bauen aufeinander auf. Sie lernen Schritt für Schritt, wie man Daten lädt, verarbeitet und speichert, sowohl in Dateien als auch in Datenbanken, und erhalten eine Einführung in die Verwendung von Funktionen, objektorientierte Programmierung, die Gestaltung von grafischen Benutzeroberflächen und Programmieren von nebenläufigen Anwendungen. Das letzte Kapitel schließlich beschäftigt sich mit der Implementierung einer Web-Schnittstelle in Java und

zeigt Ihnen einige Möglichkeiten, wie Sie nach dem Schnelleinstieg Ihre Programmierkenntnisse weiterentwickeln können.

Ihr Tagespensum schließt mit praktischen Programmier-Übungen, in denen Sie Ihr neu gewonnenes Wissen vertiefen können. Die Lösungen zu diesen Übungen, die relativ viel Programmtext enthalten, stehen in einem Online-Kapitel zum Download zur Verfügung. Mehr dazu im nächsten Abschnitt.

Am Ende des Buches finden Sie ein Stichwortverzeichnis, das Ihnen hilft, bestimmte Themen im Buch schneller zu finden.

E.3 Programmtexte und Lösungen zum Download

In diesem Buch wird eine vollständige Personenverwaltung mit Anbindung an eine Datenbank, grafischer Benutzungsoberfläche und Web-Schnittstelle entwickelt, um die gezeigten Techniken zu demonstrieren und anzuwenden.

Der Code der Personenverwaltung in den unterschiedlichen Entwicklungsstadien sowie die Lösungen zu den Übungen stehen Ihnen auf der Webseite des Verlags unter www.mitp.de/0392 zum Download zur Verfügung.

Dort finden Sie außerdem ein praktisches Glossar mit den wichtigsten Fachbegriffen.

E.4 Fragen und Feedback

Unsere Verlagsprodukte werden mit großer Sorgfalt erstellt. Sollten Sie trotzdem einen Fehler bemerken oder eine

andere Anmerkung zum Buch haben, freuen wir uns über eine direkte Rückmeldung an lektorat@mitp.de.

Falls es zu diesem Buch bereits eine Errata-Liste gibt, finden Sie diese unter www.mitp.de/0392 im Reiter Downloads.

Wir wünschen Ihnen viel Erfolg und Spaß bei der Programmierung mit Java!

Daniel Braun und das mitp-Lektorat

Erste Schritte mit Java

Eine wichtige Entscheidung auf dem Weg zum Programmierer haben Sie bereits mit der Wahl dieses Buches getroffen: die Entscheidung für die Programmiersprache Java. Obwohl viele wichtige Grundkonzepte des Programmierens unabhängig von der gewählten Programmiersprache sind, gibt es doch Unterschiede. Dieses Kapitel wird einige der Besonderheiten von Java erklären und warum die Sprache eine gute Wahl für Programmiereinsteiger ist. Zuvor erhalten Sie aber zunächst eine Einführung in einige der allgemeinen Grundbegriffe und Konzepte des Programmierens.

1.1 Programmiersprachen

Computer können uns beim Lösen zahlreicher Aufgaben helfen, oft sind sie dabei sogar besser als der Mensch, zum Beispiel beim Rechnen oder dem Sortieren von Daten. Um eine Aufgabe erledigen zu können, benötigt der Computer eine genaue Anleitung zur Lösung eben jener Aufgabe. Eine solche Anleitung mit exakten Handlungsvorschriften für einen Computer nennt man auch *Algorithmus*.

Handlungsvorschriften gibt es nicht nur für Computer, sondern auch für Menschen, zum Beispiel in Form von Rezepten in Kochbüchern oder Aufbauanleitungen für

Möbelstücke. Allerdings sprechen Mensch und Maschine unterschiedliche Sprachen, und das aus gutem Grund. Unsere menschlichen Sprachen, egal ob Deutsch, Englisch oder Plansprachen wie Esperanto, sind vage. Eine »Prise Salz«, zum Beispiel, ist eine vollkommen normale Angabe in einem Rezept und bringt keinen Koch ins Schwitzen. Für einen Backroboter aber wäre eine solche Angabe nutzlos, sie ist nicht eindeutig. Unsere menschlichen Sprachen sind voll von solch mehrdeutigen Aussagen. Denken Sie zum Beispiel an den Satz: »Da vorne ist eine Bank.« Ein vollständiger und korrekter deutscher Satz, aber was ist gemeint? Befindet sich »da vorne« eine Parkbank zum Sitzen oder ein Kreditinstitut zum Einzahlen und Abheben von Geld?

Ein Algorithmus muss aus *eindeutigen* Handlungsvorschriften bestehen. Daher sind natürliche Sprachen, wie Deutsch und Englisch, ungeeignet zur Beschreibung von Computerprogrammen. Die »Muttersprache« von Computern sind Einsen und Nullen, Strom an oder aus. Eine Sprache unverständlich für Menschen. Daher wurden Programmiersprachen entwickelt, als Zwischenrepräsentation. Im Gegensatz zu unserer natürlichen Sprache sind Programmiersprachen nicht doppeldeutig, gleichzeitig sind sie, obwohl sie auf den ersten Blick kompliziert scheinen können, relativ einfach für den Menschen zu verstehen.

Damit der Computer die Programmiersprache versteht, muss sie trotzdem übersetzt werden, in sogenannte *Maschinensprache*. Diese Übersetzung geschieht automatisch mit einem Programm, das den Namen *Compiler* trägt, abgeleitet vom englischen Verb »to compile«, was so viel bedeutet wie »zusammensetzen« oder »erstellen«. Das Ergebnis des *Kompilierens* sind sogenannte *Binärdateien*, die die Maschinensprache, codiert in Nullen und Einsen, enthalten, wie in Abbildung 1.1 gezeigt.

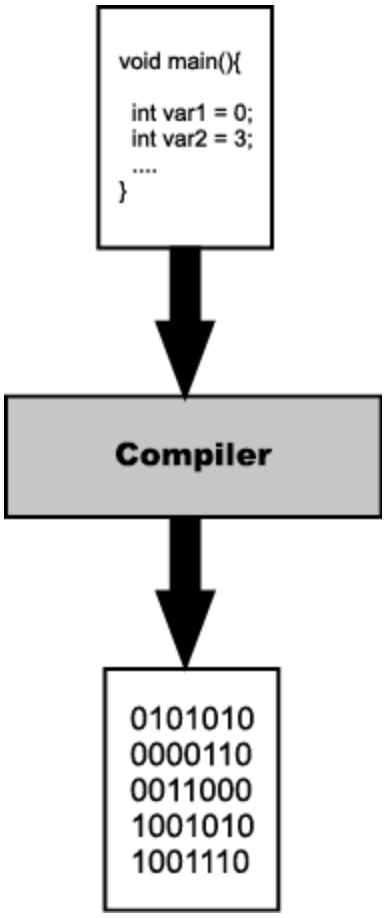


Abb.1.1: Funktionsweise eines Compilers

Für die Übersetzung von menschlicher Sprache in Programmiersprache gibt es leider noch keine automatische Übersetzung, dafür sind Sie als Programmierer zuständig, und wie das funktioniert, lernen Sie mit diesem Buch.

1.2 Besonderheiten von Java

Als Programmierer haben Sie die Wahl zwischen Hunderten Programmiersprachen. Einige davon, wie zum Beispiel C, Python oder JavaScript, sind weit verbreitet und werden häufig eingesetzt, andere eher obskur und nur für spezielle Einsatzzwecke oder Hardware geeignet. Java gehört zur Gruppe der populären Sprachen mit zahlreichen

Einsatzgebieten. Dabei bietet jede Programmiersprache ihre eigenen Vor-, aber auch Nachteile. Eine »richtige« oder »beste« Programmiersprache gibt es nicht – je nach Anwendungsfall kann der Einsatz einer anderen Programmiersprache sinnvoll sein.

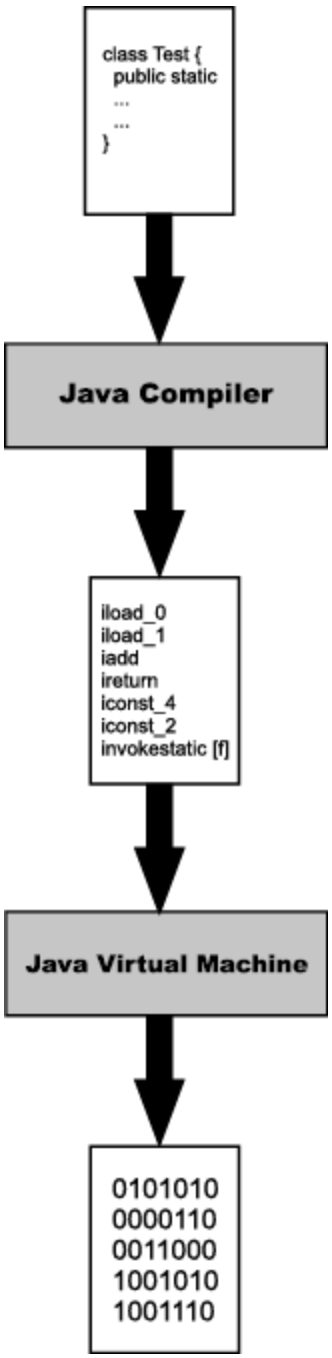


Abb.1.2: Funktionsweise des Java-Compilers

Eine der wichtigsten Eigenschaft von Java ist die Tatsache, dass Programme, die in Java geschrieben wurden, auf allen gängigen Plattformen und Betriebssystemen, also insbesondere Windows, Linux und macOS, ausgeführt werden können. Im Vergleich dazu kann zum Beispiel ein C-Programm, das unter Windows kompiliert wurde, auch nur unter Windows ausgeführt werden. Um diese Plattformunabhängigkeit zu ermöglichen, verwendet Java eine weitere Abstraktionsebene zwischen der Programmiersprache und der Maschinensprache, den sogenannten *Bytecode*.

Der Java-Compiler wandelt den Programmcode, wie in Abbildung 1.2 gezeigt, zunächst in diese Zwischenrepräsentation um. Dieser Bytecode kann dann von einem weiteren Programm, einem sogenannten *Interpreter*, ausgeführt werden. Der Java-Interpreter wird auch *Java Virtual Machine* (*JVM*) genannt. So muss nur ein Programm, nämlich die JVM, auf das jeweilige Betriebssystem angepasst werden und sofort können sämtliche Java-Programme ausgeführt werden.

Allerdings hat dieser Vorteil auch seinen Preis. Die JVM muss den Bytecode in Echtzeit – das heißt, während das Programm ausgeführt wird – in Maschinensprache übersetzen. Bei einem klassischen Compiler ist das nicht notwendig. Das führt dazu, dass Java, aber auch beispielsweise Python, das nach demselben Prinzip funktioniert, in der Ausführung weniger schnell ist als zum Beispiel die Programmiersprache C. Für sehr ressourcenintensive Anwendungen, zum Beispiel 3D-Grafik, ist Java deshalb häufig nicht die beste Wahl.

Eine weitere Besonderheit von Java ist die strikte *Typisierung* und *Objektorientierung*. Was genau es damit auf sich hat, das werden Sie in den folgenden Kapiteln erfahren. Das ist ein Nachteil für Programmiereinsteiger, den Java mit

sich bringt. Gerade am Anfang gibt es ein gewisses Grundgerüst für jedes Programm, das Sie einfach als gegeben hinnehmen müssen, bis Sie Schritt für Schritt besser verstehen werden, was dahintersteckt. Dafür lernen Sie aber von Beginn an wichtige Grundlagen explizit kennen, die andere Programmiersprachen, wie zum Beispiel Python oder JavaScript, vor Ihnen verstecken. Deshalb ist Java vielleicht nicht die einfachste Sprache zum Programmierenlernen, aber trotzdem eine sehr gute Wahl für den Einstieg. Nicht umsonst ist Java elementarer Bestandteil des Informatikstudiums an den meisten Universitäten.

Außer dem Namen haben die beiden Programmiersprachen Java und JavaScript nicht viel gemein. JavaScript, eine Programmiersprache, die ursprünglich hauptsächlich zum Einsatz auf Websites entwickelt wurde, hieß zunächst auch LiveScript. Durch eine Partnerschaft zwischen dem ursprünglichen Entwickler von LiveScript und dem Entwickler von Java, Sun Microsystems, kam es zur namentlichen Annäherung.

Java ist universell einsetzbar. Es wird zum Beispiel häufig zur Programmierung von betrieblichen Anwendungen verwendet, unter anderem, da es einfach ist, ein Programm über mehrere Rechner zu verteilen, was zum Beispiel dabei hilft, steigende Hardwareanforderungen bei wachsenden Nutzerzahlen zu erfüllen. Ebenso lassen sich Android-Apps mit Java programmieren – ja selbst die im Juni 2003 von der NASA gestartete Raumsonde »Spirit« bewegte sich mithilfe von Java über die Oberfläche des Mars.

1.3 Installation und Einrichtung

Bevor Sie damit beginnen können, Java-Programme zu entwickeln – egal ob für Raumsonden oder Smartphones –, müssen Sie zunächst Ihren Rechner so einrichten, dass er Java-Programme ausführen und kompilieren kann. Die JVM, die benötigt wird, um Java-Programme auszuführen, ist als Teil des sogenannten *Java Runtime Environments (JRE)* bereits auf den meisten Computern installiert. Um selbst Java-Programme erstellen zu können, reicht das JRE allerdings nicht aus, dafür wird das *Java Development Kit (JDK)* benötigt, das unter anderem den Java-Compiler beinhaltet.

1.3.1 Linux

Unter Linux lässt sich das JDK am einfachsten über den Paketmanager apt installieren. Dazu müssen Sie lediglich den Befehl `sudo apt-get install openjdk-17-jdk` im Terminal eingeben, wobei die Zahl 17 die Versionsnummer beschreibt und je nach gewünschter Version angepasst werden muss. Welche Version gerade aktuell ist, können Sie zum Beispiel unter https://jdk.java.net einsehen.

1.3.2 macOS

Um das JDK auf Ihrem Apple-Rechner zu installieren, müssen Sie zunächst die gewünschte Version unter https://jdk.java.net herunterladen. Achten Sie dabei darauf, die macOS-Version auszuwählen. Danach öffnen Sie das Terminal und wechseln mit dem Befehl `cd ~/Downloads` in den Download-Ordner, wo sich die noch gepackten Dateien des JDK befinden. Mit dem Befehl `tar xf openjdk-17_osx-x64_bin.tar.gz` können Sie die Dateien entpacken, wobei die Zahl 17 der Versionsnummer entspricht und je nach Version, die Sie heruntergeladen haben, angepasst werden muss.

Die entpackten Dateien müssen dann noch in den richtigen Ordner verschoben werden, damit sie vom System auch

gefunden werden können. Dies geschieht über den Befehl `sudo cp -Rv jdk-17.jdk /Library/Java/JavaVirtualMachines/`, wobei auch hier wieder die Versionsnummer entsprechend angepasst werden muss.

1.3.3 Windows

Um das JDK unter Windows zu installieren, müssen Sie zunächst die gewünschte Version unter https://jdk.java.net herunterladen. Achten Sie dabei darauf, die Windows-Version auszuwählen, die Sie an der Endung `.zip` erkennen.

Nach dem Download muss das ZIP-Archiv zunächst entpackt werden. Hierzu klicken Sie mit der rechten Maustaste auf das Archiv und wählen den Eintrag Alle Extrahieren aus. Im extrahierten Archiv befindet sich dann ein Ordner, der den Namen `jdk-17` trägt, wobei die Zahl 17 der Versionsnummer entspricht und entsprechend abweichen kann, je nachdem, welche Version Sie heruntergeladen haben. Diesen Ordner können Sie nun an einen beliebigen Platz verschieben. Es empfiehlt sich standardmäßig, einen Ordner `C:\Programme\Java` anzulegen, falls dieser noch nicht existiert, und den `jdk`-Ordner dorthin zu verschieben.

Nun muss dem Betriebssystem noch mitgeteilt werden, wo das JDK zu finden ist. Dafür muss die sogenannte *Path-Variable* angepasst werden, die in den erweiterten Systemeinstellungen zu finden ist. Diese lassen sich auf verschiedene Arten öffnen: Durch Drücken der Windows-Taste und des Buchstabens R öffnet sich das Fenster Ausführen. Dort können Sie `sysdm.cpl` eingeben und durch einen Klick auf OK die Systemeigenschaften öffnen. Hier können Sie dann die Kategorie Erweitert auswählen.

Unter Windows 8 und 10 können Sie alternativ auch den Begriff Erweiterte Systemeinstellungen in die Windows-Suche eingeben und so direkt in das entsprechende Fenster springen.

Egal, welchen Weg Sie wählen, danach sehen Sie das in Abbildung 1.3 gezeigte Fenster auf Ihrem Bildschirm.

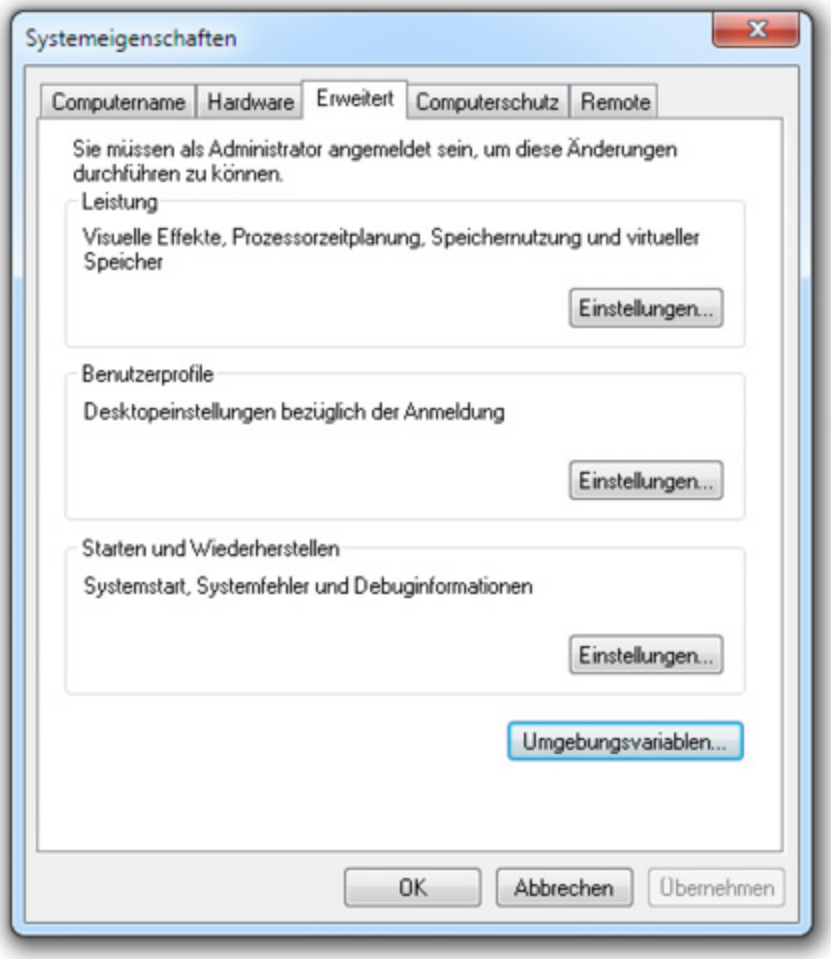


Abb.1.3: Erweiterte Systemeinstellungen

Dort klicken Sie dann auf den Button Umgebungsvariablen..., woraufhin sich das in Abbildung 1.4 gezeigte Fenster öffnet. Hier wählen Sie nun im oberen Bereich der Benutzervariablen den Eintrag PATH aus und klicken auf den Bearbeiten...-Button. Im sich daraufhin öffnenden Fenster legen Sie dann einen neuen Eintrag an, der den Pfad zum `bin`-Ordner innerhalb des JDK enthält. Also zum Beispiel `C:\Programme\Java\jdk-17\bin`, wobei auch hier die Zahl 17 wieder für die Versionsnummer steht und entsprechend der

von Ihnen verwendeten Version angepasst werden muss. Danach müssen die Änderungen nur noch mit einem Klick auf OK bestätigt werden.

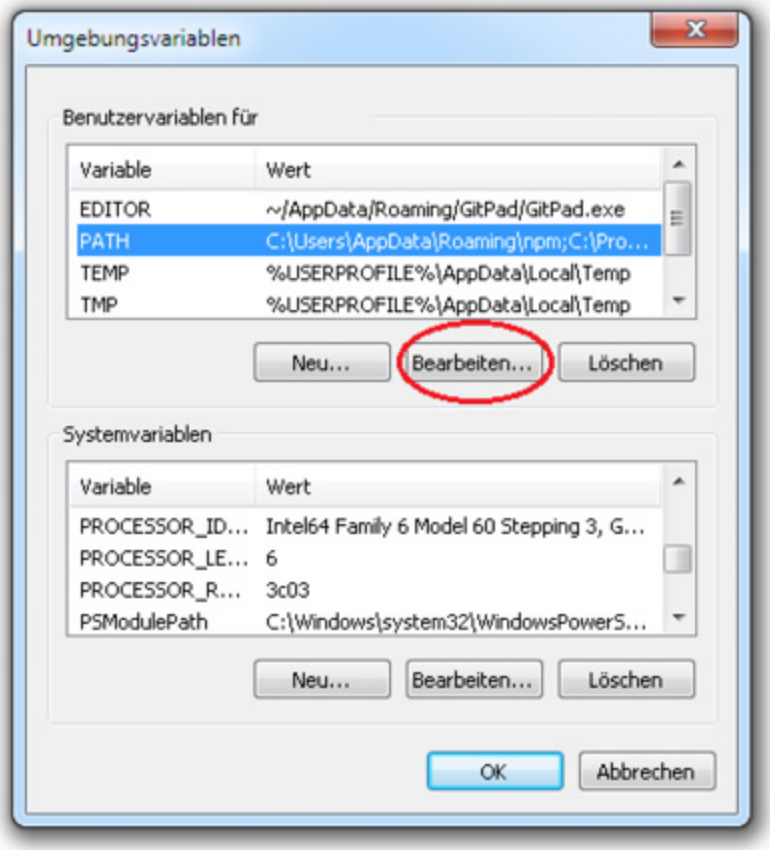


Abb.1.4: Umgebungsvariablen

1.3.4 Installation testen

Damit ist die Installation des JDK bereits abgeschlossen und Sie können testen, ob alles korrekt eingerichtet ist. Dazu öffnen Sie ein Terminal – beziehungsweise unter Windows die Eingabeaufforderung – und geben den Befehl `javac` ein. Hierdurch wird der Java-Compiler aufgerufen. Ist dieser korrekt installiert und eingerichtet, so erscheint als Ausgabe auf dem Bildschirm die Hilfe für den Befehl, die dessen Nutzung erklärt, wie in Abbildung 1.5 gezeigt. Erscheint dagegen eine Fehlermeldung, so ist bei der Installation

etwas schiefgelaufen, und Sie sollten noch einmal überprüfen, ob alle Eingaben und insbesondere Pfade korrekt waren, insbesondere auch, ob Sie überall die korrekte Versionsnummer angegeben haben.

Abb.1.5: Ausgabe bei korrekt installiertem Java-Compiler

1.4 Entwicklungsumgebungen

Damit haben Sie bereits alles zusammen, was Sie zum Programmieren mit Java unbedingt benötigen. Theoretisch können Sie Java-Programme nämlich in jedem Texteditor oder sogar direkt in der Eingabeaufforderung