

HANSER



Inhaltsverzeichnis

Dietmar Ratz, Jens Scheffler, Detlef Seese, Jan Wiesenberger

Grundkurs Programmieren in Java

ISBN: 978-3-446-41655-0

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-41655-0>

sowie im Buchhandel.

Inhaltsverzeichnis

Vorwort	15
1 Einleitung	17
1.1 Java – mehr als nur kalter Kaffee?	17
1.2 Java für Anfänger – das Konzept dieses Buches	18
1.3 Weitere Infos und Kontakt zu den Autoren	20
1.4 Verwendete Schreibweisen	20
2 Einige Grundbegriffe aus der Welt des Programmierens	21
2.1 Computer, Software, Informatik und das Internet	21
2.2 Was heißt Programmieren?	24
I Einstieg in das Programmieren in Java	27
3 Aller Anfang ist schwer	29
3.1 Mein erstes Programm	29
3.2 Formeln, Ausdrücke und Anweisungen	30
3.3 Zahlenbeispiele	31
3.4 Verwendung von Variablen	32
3.5 „Auf den Schirm!“	32
3.6 Das Programmgerüst	33
3.7 Eingeben, übersetzen und ausführen	35
3.8 Übungsaufgaben	36
4 Grundlagen der Programmierung in Java	37
4.1 Grundelemente eines Java-Programms	37
4.1.1 Kommentare	39
4.1.2 Bezeichner und Namen	41
4.1.3 Literale	42
4.1.4 Reservierte Wörter, Schlüsselwörter	42
4.1.5 Trennzeichen	43
4.1.6 Interpunktionszeichen	44
4.1.7 Operatorsymbole	44
4.1.8 import -Anweisungen	45

4.1.9	Zusammenfassung	46
4.1.10	Übungsaufgaben	46
4.2	Erste Schritte in Java	47
4.2.1	Grundstruktur eines Java-Programms	48
4.2.2	Ausgaben auf der Konsole	49
4.2.3	Eingaben von der Konsole	50
4.2.4	Schöner programmieren in Java	51
4.2.5	Zusammenfassung	52
4.2.6	Übungsaufgaben	52
4.3	Einfache Datentypen	53
4.3.1	Ganzzahlige Datentypen	53
4.3.2	Gleitkommatypen	55
4.3.3	Der Datentyp char für Zeichen	57
4.3.4	Zeichenketten	58
4.3.5	Der Datentyp boolean für Wahrheitswerte	58
4.3.6	Implizite und explizite Typumwandlungen	58
4.3.7	Zusammenfassung	60
4.3.8	Übungsaufgaben	60
4.4	Der Umgang mit einfachen Datentypen	61
4.4.1	Variablen	61
4.4.2	Operatoren und Ausdrücke	65
4.4.2.1	Arithmetische Operatoren	66
4.4.2.2	Bitoperatoren	68
4.4.2.3	Zuweisungsoperator	70
4.4.2.4	Vergleichsoperatoren und Logische Operatoren	71
4.4.2.5	Inkrement- und Dekrementoperatoren	73
4.4.2.6	Priorität und Auswertungsreihenfolge der Operatoren	74
4.4.3	Allgemeine Ausdrücke	75
4.4.4	Ein- und Ausgabe	76
4.4.4.1	Statischer Import der IOTools-Methoden	77
4.4.5	Zusammenfassung	79
4.4.6	Übungsaufgaben	79
4.5	Anweisungen und Ablaufsteuerung	82
4.5.1	Anweisungen	83
4.5.2	Blöcke und ihre Struktur	83
4.5.3	Entscheidungsanweisung	84
4.5.3.1	Die if -Anweisung	84
4.5.3.2	Die switch -Anweisung	85
4.5.4	Wiederholungsanweisungen, Schleifen	87
4.5.4.1	Die for -Anweisung	87
4.5.4.2	Vereinfachte for -Schleifen-Notation	88
4.5.4.3	Die while -Anweisung	89
4.5.4.4	Die do -Anweisung	89
4.5.4.5	Endlosschleifen	90
4.5.5	Sprungbefehle und markierte Anweisungen	91
4.5.6	Zusammenfassung	93

4.5.7	Übungsaufgaben	93
5	Referenzdatentypen	103
5.1	Felder	105
5.1.1	Was sind Felder?	107
5.1.2	Deklaration, Erzeugung und Initialisierung von Feldern	108
5.1.3	Felder unbekannter Länge	111
5.1.4	Referenzen	113
5.1.5	Ein besserer Terminkalender	117
5.1.6	Mehrdimensionale Felder	119
5.1.7	Mehrdimensionale Felder unterschiedlicher Länge	122
5.1.8	Vorsicht, Falle: Kopieren von mehrdimensionalen Feldern	124
5.1.9	Vereinfachte for -Schleifen-Notation	125
5.1.10	Zusammenfassung	127
5.1.11	Übungsaufgaben	127
5.2	Klassen	130
5.2.1	Was sind Klassen?	131
5.2.2	Deklaration und Instanziierung von Klassen	132
5.2.3	Komponentenzugriff bei Objekten	133
5.2.4	Ein erstes Adressbuch	134
5.2.5	Klassen als Referenzdatentyp	136
5.2.6	Felder von Klassen	139
5.2.7	Vorsicht, Falle: Kopieren von geschachtelten Referenzdatentypen	142
5.2.8	Auslagern von Klassen	143
5.2.9	Zusammenfassung	145
5.2.10	Übungsaufgaben	145
6	Methoden, Unterprogramme	147
6.1	Methoden	148
6.1.1	Was sind Methoden?	148
6.1.2	Deklaration von Methoden	149
6.1.3	Parameterübergabe und Ergebnisrückgabe	150
6.1.4	Aufruf von Methoden	152
6.1.5	Überladen von Methoden	153
6.1.6	Variable Argument-Anzahl bei Methoden	155
6.1.7	Vorsicht, Falle: Referenzen als Parameter	156
6.1.8	Sichtbarkeit und Verdecken von Variablen	158
6.1.9	Zusammenfassung	160
6.1.10	Übungsaufgaben	160
6.2	Rekursiv definierte Methoden	161
6.2.1	Motivation	161
6.2.2	Gute und schlechte Beispiele für rekursive Methoden	163
6.2.3	Zusammenfassung	166
6.3	Die Methode <code>main</code>	166
6.3.1	Kommandozeilenparameter	167
6.3.2	Anwendung der vereinfachten for -Schleifen-Notation	168

6.3.3	Zusammenfassung	169
6.3.4	Übungsaufgaben	169
6.4	Methoden aus anderen Klassen aufrufen	171
6.4.1	Klassenmethoden	171
6.4.2	Die Methoden der Klasse <code>java.lang.Math</code>	172
6.4.3	Statischer Import	173
6.5	Methoden von Objekten aufrufen	174
6.5.1	Instanzmethoden	174
6.5.2	Die Methoden der Klasse <code>java.lang.String</code>	175
6.6	Übungsaufgaben	178
 II Objektorientiertes Programmieren in Java		183
7	Die objektorientierte Philosophie	185
7.1	Die Welt, in der wir leben	185
7.2	Programmierparadigmen – Objektorientierung im Vergleich	186
7.3	Die vier Grundpfeiler objektorientierter Programmierung	188
7.3.1	Generalisierung	188
7.3.2	Vererbung	190
7.3.3	Kapselung	193
7.3.4	Polymorphismus	194
7.3.5	Weitere wichtige Grundbegriffe	195
7.4	Modellbildung – von der realen Welt in den Computer	196
7.4.1	Grafisches Modellieren mit UML	196
7.4.2	Entwurfsmuster	197
7.5	Zusammenfassung	198
7.6	Übungsaufgaben	199
8	Der grundlegende Umgang mit Klassen	201
8.1	Vom Referenzdatentyp zur Objektorientierung	201
8.2	Instanzmethoden	203
8.2.1	Zugriffsrechte	203
8.2.2	Was sind Instanzmethoden?	204
8.2.3	Instanzmethoden zur Validierung von Eingaben	207
8.2.4	Instanzmethoden als erweiterte Funktionalität	208
8.3	Statische Komponenten einer Klasse	209
8.3.1	Klassenvariablen und -methoden	210
8.3.2	Klassenkonstanten	212
8.4	Instantiierung und Initialisierung	213
8.4.1	Konstruktoren	213
8.4.2	Überladen von Konstruktoren	215
8.4.3	Der statische Initialisierer	217
8.4.4	Der Mechanismus der Objekterzeugung	220
8.5	Zusammenfassung	224
8.6	Übungsaufgaben	225

9	Vererbung und Polymorphismus	245
9.1	Wozu braucht man Vererbung?	245
9.1.1	Aufgabenstellung	245
9.1.2	Analyse des Problems	246
9.1.3	Ein erster Ansatz	246
9.1.4	Eine Klasse für sich	247
9.1.5	Stärken der Vererbung	248
9.1.6	Vererbung verhindern durch final	251
9.1.7	Übungsaufgaben	252
9.2	Die super -Referenz	253
9.3	Überschreiben von Methoden und Variablen	255
9.3.1	Dynamisches Binden	255
9.3.2	Überschreiben von Methoden verhindern durch final	257
9.4	Die Klasse <code>java.lang.Object</code>	258
9.5	Übungsaufgaben	261
9.6	Abstrakte Klassen und Interfaces	261
9.7	Übungsaufgaben	265
9.8	Weiteres zum Thema Objektorientierung	270
9.8.1	Erstellen von Paketen	270
9.8.2	Zugriffsrechte	271
9.8.3	Innere Klassen	272
9.8.4	Anonyme Klassen	278
9.9	Zusammenfassung	280
9.10	Übungsaufgaben	280
10	Exceptions und Errors	291
10.1	Eine Einführung in Exceptions	292
10.1.1	Was ist eine Exception?	292
10.1.2	Übungsaufgaben	294
10.1.3	Abfangen von Exceptions	294
10.1.4	Ein Anwendungsbeispiel	295
10.1.5	Die <code>RuntimeException</code>	298
10.1.6	Übungsaufgaben	299
10.2	Exceptions für Fortgeschrittene	301
10.2.1	Definieren eigener Exceptions	301
10.2.2	Übungsaufgaben	303
10.2.3	Vererbung und Exceptions	303
10.2.4	Vorsicht, Falle!	307
10.2.5	Der finally -Block	309
10.2.6	Die Klassen <code>Throwable</code> und <code>Error</code>	313
10.2.7	Zusammenfassung	315
10.2.8	Übungsaufgaben	315
10.3	Assertions	316
10.3.1	Zusicherungen im Programmcode	316
10.3.2	Compilieren des Programmcodes	317
10.3.3	Ausführen des Programmcodes	318
10.3.4	Zusammenfassung	318

11 Fortgeschrittene objektorientierte Programmierung	319
11.1 Aufzählungstypen	320
11.1.1 Deklaration eines Aufzählungstyps	320
11.1.2 Instanzmethoden der enum -Objekte	321
11.1.3 Selbstdefinierte Instanzmethoden für enum -Objekte	321
11.1.4 Übungsaufgaben	323
11.2 Generische Datentypen	325
11.2.1 Generizität in alten Java-Versionen	325
11.2.2 Generizität ab Java 5.0	328
11.2.3 Einschränkungen der Typ-Parameter	330
11.2.4 Wildcards	332
11.2.5 Bounded Wildcards	333
11.2.6 Generische Methoden	335
11.2.7 Ausblick	337
11.2.8 Übungsaufgaben	337
11.3 Sortieren von Feldern und das Interface <code>Comparable</code>	342
12 Einige wichtige Hilfsklassen	345
12.1 Die Klasse <code>StringBuffer</code>	345
12.1.1 Arbeiten mit <code>String</code> -Objekten	345
12.1.2 Arbeiten mit <code>StringBuffer</code> -Objekten	348
12.1.3 Übungsaufgaben	350
12.2 Die Wrapper-Klassen (Hüll-Klassen)	351
12.2.1 Arbeiten mit „eingepackten“ Daten	351
12.2.2 Aufbau der Wrapper-Klassen	352
12.2.3 Ein Anwendungsbeispiel	355
12.2.4 Automatische Typwandlung für die Wrapper-Klassen	356
12.2.5 Übungsaufgaben	358
12.3 Die Klassen <code>BigInteger</code> und <code>BigDecimal</code>	359
12.3.1 Arbeiten mit langen Ganzzahlen	359
12.3.2 Aufbau der Klasse <code>BigInteger</code>	361
12.3.3 Übungsaufgaben	363
12.3.4 Arbeiten mit langen Gleitkommazahlen	363
12.3.5 Aufbau der Klasse <code>BigDecimal</code>	366
12.3.6 Viele Stellen von Nullstellen gefällig?	369
12.3.7 Übungsaufgaben	370
12.4 Die Klasse <code>DecimalFormat</code>	371
12.4.1 Standard-Ausgaben in Java	371
12.4.2 Arbeiten mit <code>Format</code> -Objekten	372
12.4.3 Vereinfachte formatierte Ausgabe	374
12.4.4 Übungsaufgaben	375
12.5 Die Klassen <code>Date</code> und <code>Calendar</code>	375
12.5.1 Arbeiten mit „Zeitpunkten“	376
12.5.2 Auf die Plätze, fertig, los!	377
12.5.3 Spezielle <code>Calendar</code> -Klassen	378
12.5.4 Noch einmal: Zeitmessung	380
12.5.5 Übungsaufgaben	382

12.6	Die Klassen <code>SimpleDateFormat</code> und <code>DateFormat</code>	382
12.6.1	Arbeiten mit Format-Objekten für Datum/Zeit-Angaben . .	382
12.6.2	Übungsaufgaben	387
12.7	Die <code>Collection</code> -Klassen	387
12.7.1	„Sammlungen“ von Objekten – Der Aufbau des Interface <code>Collection</code>	387
12.7.2	„Sammlungen“ durchgehen – Der Aufbau des Interface <code>Iterator</code>	390
12.7.3	Mengen	391
12.7.3.1	Das Interface <code>Set</code>	391
12.7.3.2	Die Klasse <code>HashSet</code>	391
12.7.3.3	Das Interface <code>SortedSet</code>	393
12.7.3.4	Die Klasse <code>TreeSet</code>	394
12.7.4	Listen	395
12.7.4.1	Das Interface <code>List</code>	396
12.7.4.2	Die Klassen <code>ArrayList</code> und <code>LinkedList</code>	396
12.7.4.3	Suchen und Sortieren – Die Klassen <code>Collections</code> und <code>Arrays</code>	398
12.7.5	Übungsaufgaben	401
12.8	Die Klasse <code>StringTokenizer</code>	402
12.8.1	Übungsaufgaben	404
 III Grafische Oberflächen in Java		405
 13 Aufbau grafischer Oberflächen in Frames – von AWT nach Swing . . .		407
13.1	Grundsätzliches zum Aufbau grafischer Oberflächen	407
13.2	Ein einfaches Beispiel mit dem AWT	409
13.3	Let's swing now!	411
13.4	Etwas „Fill-in“ gefällig?	413
13.5	Die AWT- und Swing-Klassenbibliothek im Überblick	415
13.6	Übungsaufgaben	417
 14 Swing-Komponenten		419
14.1	Die abstrakte Klasse <code>Component</code>	419
14.2	Die Klasse <code>Container</code>	420
14.3	Die abstrakte Klasse <code>JComponent</code>	421
14.4	Layout-Manager, Farben und Schriften	422
14.4.1	Die Klasse <code>Color</code>	423
14.4.2	Die Klasse <code>Font</code>	425
14.4.3	Layout-Manager	426
14.4.3.1	Die Klasse <code>FlowLayout</code>	427
14.4.3.2	Die Klasse <code>BorderLayout</code>	429
14.4.3.3	Die Klasse <code>GridLayout</code>	430
14.5	Einige Grundkomponenten	432
14.5.1	Die Klasse <code>JLabel</code>	434
14.5.2	Die abstrakte Klasse <code>AbstractButton</code>	434

14.5.3	Die Klasse <code>JButton</code>	436
14.5.4	Die Klasse <code>JToggleButton</code>	437
14.5.5	Die Klasse <code>JCheckBox</code>	438
14.5.6	Die Klassen <code>JRadioButton</code> und <code>ButtonGroup</code>	439
14.5.7	Die Klasse <code>JComboBox</code>	441
14.5.8	Die Klasse <code>JList</code>	444
14.5.9	Die abstrakte Klasse <code>JTextComponent</code>	447
14.5.10	Die Klassen <code>JTextField</code> und <code>JPasswordField</code>	448
14.5.11	Die Klasse <code>JTextArea</code>	450
14.5.12	Die Klasse <code>JScrollPane</code>	452
14.5.13	Die Klasse <code>JPanel</code>	454
14.6	Spezielle Container, Menüs und Toolbars	456
14.6.1	Die Klasse <code>JFrame</code>	456
14.6.2	Die Klasse <code>JWindow</code>	457
14.6.3	Die Klasse <code>JDialog</code>	457
14.6.4	Die Klasse <code>JMenuBar</code>	461
14.6.5	Die Klasse <code>JToolBar</code>	463
14.7	Übungsaufgaben	466
15	Ereignisverarbeitung	469
15.1	Zwei einfache Beispiele	470
15.1.1	Zufällige Grautöne als Hintergrund	470
15.1.2	Ein interaktiver Bilderrahmen	473
15.2	Programmvarianten für die Ereignisverarbeitung	477
15.2.1	Innere Klasse als Listener-Klasse	477
15.2.2	Anonyme Klasse als Listener-Klasse	477
15.2.3	Container-Klasse als Listener-Klasse	478
15.2.4	Separate Klasse als Listener-Klasse	479
15.3	Event-Klassen und -Quellen	481
15.4	Listener-Interfaces und Adapter-Klassen	485
15.5	Listener-Registrierung bei den Event-Quellen	490
15.6	Auf die Plätze, fertig, los!	494
15.7	Übungsaufgaben	498
16	Einige Ergänzungen zu Swing-Komponenten	503
16.1	Zeichnen in Swing-Komponenten	503
16.1.1	Grafische Darstellung von Komponenten	503
16.1.2	Das Grafik-Koordinatensystem	504
16.1.3	Die abstrakte Klasse <code>Graphics</code>	505
16.1.4	Ein einfaches Zeichenprogramm	508
16.1.5	Layoutveränderungen und der Einsatz von <code>revalidate</code>	510
16.2	Noch mehr Swing gefällig?	513
16.3	Übungsaufgaben	514
17	Applets	517
17.1	Erstellen und Ausführen von Applets	517
17.1.1	Vom Frame zum Applet am Beispiel	517
17.1.2	Applet in HTML-Datei einbetten	519

17.1.3	Applet über HTML-Datei ausführen	521
17.2	Die Methoden der Klasse <code>JApplet</code>	522
17.3	Zwei Beispiele	524
17.3.1	Auf die Plätze, fertig, los!	525
17.3.2	Punkte verbinden im Applet	528
17.4	Details zur HTML-Einbettung	529
17.4.1	Der Applet-Tag	529
17.4.2	Die Methode <code>showDocument</code>	532
17.5	Sicherheitseinschränkungen bei Applets	534
17.6	Übungsaufgaben	538
IV	Threads, Datenströme und Netzwerk-Anwendungen	541
18	Parallele Programmierung mit Threads	543
18.1	Ein einfaches Beispiel	543
18.2	Threads in Java	545
18.2.1	Die Klasse <code>Thread</code>	546
18.2.2	Das Interface <code>Runnable</code>	550
18.2.3	Threads vorzeitig beenden	552
18.3	Wissenswertes über Threads	554
18.3.1	Lebenszyklus eines Threads	554
18.3.2	Thread-Scheduling	556
18.3.3	Dämon-Threads und Thread-Gruppen	556
18.4	Thread-Synchronisation und -Kommunikation	557
18.4.1	Das Leser/Schreiber-Problem	558
18.4.2	Das Erzeuger/Verbraucher-Problem	562
18.5	Threads in Frames und Applets	569
18.5.1	Auf die Plätze, fertig, los!	569
18.5.2	Spielereien	573
18.5.3	Swing-Komponenten sind nicht Thread-sicher	575
18.6	Übungsaufgaben	576
19	Ein- und Ausgabe über Streams	579
19.1	Grundsätzliches zu Streams in Java	580
19.2	Dateien und Verzeichnisse – Die Klasse <code>File</code>	580
19.3	Ein- und Ausgabe über Character-Streams	583
19.3.1	Einfache <code>Reader</code> - und <code>Writer</code> -Klassen	584
19.3.2	Gepufferte <code>Reader</code> - und <code>Writer</code> -Klassen	587
19.3.3	Die Klasse <code>StreamTokenizer</code>	589
19.3.4	Die Klasse <code>PrintWriter</code>	590
19.3.5	Die Klassen <code>IOTools</code> und <code>Scanner</code>	592
19.3.5.1	Was machen eigentlich die <code>IOTools</code> ?	592
19.3.5.2	Konsoleneingabe über ein <code>Scanner</code> -Objekt	594
19.4	Ein- und Ausgabe über Byte-Streams	594
19.4.1	Einige <code>InputStream</code> - und <code>OutputStream</code> -Klassen	595
19.4.2	Die Serialisierung und Deserialisierung von Objekten	597

19.4.3 Die Klasse <code>PrintStream</code>	599
19.5 Einige abschließende Bemerkungen	600
19.6 Übungsaufgaben	601
20 Client/Server-Programmierung in Netzwerken	603
20.1 Wissenswertes über Netzwerk-Kommunikation	604
20.1.1 Protokolle	604
20.1.2 IP-Adressen	606
20.1.3 Ports und Sockets	607
20.2 Client/Server-Programmierung	608
20.2.1 Die Klassen <code>ServerSocket</code> und <code>Socket</code>	609
20.2.2 Ein einfacher Server	611
20.2.3 Ein einfacher Client	614
20.2.4 Ein Server für mehrere Clients	615
20.2.5 Ein Mehrzweck-Client	618
20.3 Wissenswertes über URLs	621
20.3.1 Client/Server-Kommunikation über URLs	621
20.3.2 Netzwerkverbindungen in Applets	622
20.4 Übungsaufgaben	623
V Ausblick und Anhang	627
21 Blick über den Tellerrand	629
21.1 Der Vorhang fällt	629
21.2 A fool with a tool	630
21.3 Alles umsonst?	631
21.4 Und fachlich?	632
21.5 Zu guter Letzt	634
A Der Weg zum guten Programmierer	635
A.1 Die goldenen Regeln der Code-Formatierung	636
A.2 Die goldenen Regeln der Namensgebung	639
A.3 Zusammenfassung	641
B Die Klasse <code>IOTools</code> – Tastatureingaben in Java	643
B.1 Kurzbeschreibung	643
B.2 Anwendung der <code>IOTools</code> -Methoden	644
C Der Umgang mit der API-Spezifikation	647
C.1 Der Aufbau der API-Spezifikation	647
C.2 Der praktische Einsatz der API-Spezifikation	648
D Glossar	653
Literaturverzeichnis	667
Stichwortverzeichnis	671