

Inhaltsverzeichnis

Vorwort	17
1 Einleitung	19
1.1 Java – mehr als nur kalter Kaffee?	19
1.2 Java für Anfänger – das Konzept dieses Buches	20
1.3 Zusatzmaterial und Kontakt zu den Autoren	22
1.4 Verwendete Schreibweisen	22
2 Einige Grundbegriffe aus der Welt des Programmierens	23
2.1 Computer, Software, Informatik und das Internet	23
2.2 Was heißt Programmieren?	26
I Einstieg in das Programmieren in Java	29
3 Aller Anfang ist schwer	31
3.1 Mein erstes Programm	31
3.2 Formeln, Ausdrücke und Anweisungen	32
3.3 Zahlenbeispiele	33
3.4 Verwendung von Variablen	34
3.5 „Auf den Schirm!“	34
3.6 Das Programmgerüst	35
3.7 Eingeben, übersetzen und ausführen	37
3.8 Übungsaufgaben	38
4 Grundlagen der Programmierung in Java	39
4.1 Grundelemente eines Java-Programms	39
4.1.1 Kommentare	41
4.1.2 Bezeichner und Namen	43
4.1.3 Literale	44
4.1.4 Reservierte Wörter, Schlüsselwörter	44

4.1.5	Trennzeichen	45
4.1.6	Interpunktionszeichen	46
4.1.7	Operatorsymbole	46
4.1.8	import -Anweisungen	47
4.1.9	Zusammenfassung	48
4.1.10	Übungsaufgaben	48
4.2	Erste Schritte in Java	49
4.2.1	Grundstruktur eines Java-Programms	50
4.2.2	Ausgaben auf der Konsole	51
4.2.3	Eingaben von der Konsole	52
4.2.4	Schöner programmieren in Java	53
4.2.5	Zusammenfassung	54
4.2.6	Übungsaufgaben	54
4.3	Einfache Datentypen	55
4.3.1	Ganzzahlige Datentypen	55
4.3.2	Gleitkommatypen	57
4.3.3	Der Datentyp char für Zeichen	59
4.3.4	Zeichenketten	60
4.3.5	Der Datentyp boolean für Wahrheitswerte	60
4.3.6	Implizite und explizite Typumwandlungen	60
4.3.7	Zusammenfassung	62
4.3.8	Übungsaufgaben	62
4.4	Der Umgang mit einfachen Datentypen	63
4.4.1	Variablen	63
4.4.2	Operatoren und Ausdrücke	67
4.4.2.1	Arithmetische Operatoren	68
4.4.2.2	Bitoperatoren	70
4.4.2.3	Zuweisungsoperator	72
4.4.2.4	Vergleichsoperatoren und logische Operatoren	73
4.4.2.5	Inkrement- und Dekrementoperatoren	75
4.4.2.6	Priorität und Auswertungsreihenfolge der Operatoren	76
4.4.3	Allgemeine Ausdrücke	77
4.4.4	Ein- und Ausgabe	78
4.4.4.1	Statischer Import der IOTools-Methoden	79
4.4.5	Zusammenfassung	81
4.4.6	Übungsaufgaben	81
4.5	Anweisungen und Ablaufsteuerung	84
4.5.1	Anweisungen	85
4.5.2	Blöcke und ihre Struktur	85
4.5.3	Entscheidungsanweisung	86

4.5.3.1	Die if -Anweisung	86
4.5.3.2	Die switch -Anweisung	87
4.5.4	Wiederholungsanweisungen, Schleifen	89
4.5.4.1	Die for -Anweisung	89
4.5.4.2	Vereinfachte for -Schleifen-Notation	90
4.5.4.3	Die while -Anweisung	91
4.5.4.4	Die do -Anweisung	91
4.5.4.5	Endlosschleifen	92
4.5.5	Sprungbefehle und markierte Anweisungen	93
4.5.6	Zusammenfassung	95
4.5.7	Übungsaufgaben	95
5	Referenzdatentypen	105
5.1	Felder	107
5.1.1	Was sind Felder?	109
5.1.2	Deklaration, Erzeugung und Initialisierung von Feldern	110
5.1.3	Felder unbekannter Länge	113
5.1.4	Referenzen	115
5.1.5	Ein besserer Terminkalender	119
5.1.6	Mehrdimensionale Felder	121
5.1.7	Mehrdimensionale Felder unterschiedlicher Länge	124
5.1.8	Vorsicht, Falle: Kopieren von mehrdimensionalen Feldern	126
5.1.9	Vereinfachte for -Schleifen-Notation	127
5.1.10	Zusammenfassung	129
5.1.11	Übungsaufgaben	129
5.2	Klassen	132
5.2.1	Was sind Klassen?	133
5.2.2	Deklaration und Instantiierung von Klassen	134
5.2.3	Komponentenzugriff bei Objekten	135
5.2.4	Ein erstes Adressbuch	136
5.2.5	Klassen als Referenzdatentyp	138
5.2.6	Felder von Klassen	141
5.2.7	Vorsicht, Falle: Kopieren von geschachtelten Referenzdatentypen	144
5.2.8	Auslagern von Klassen	145
5.2.9	Zusammenfassung	147
5.2.10	Übungsaufgaben	147
6	Methoden, Unterprogramme	149
6.1	Methoden	150
6.1.1	Was sind Methoden?	150
6.1.2	Deklaration von Methoden	151

6.1.3	Parameterübergabe und ErgebnISRückgabe	152
6.1.4	Aufruf von Methoden	154
6.1.5	Überladen von Methoden	155
6.1.6	Variable Argument-Anzahl bei Methoden	157
6.1.7	Vorsicht, Falle: Referenzen als Parameter	158
6.1.8	Sichtbarkeit und Verdecken von Variablen	160
6.1.9	Zusammenfassung	162
6.1.10	Übungsaufgaben	162
6.2	Rekursiv definierte Methoden	163
6.2.1	Motivation	163
6.2.2	Gute und schlechte Beispiele für rekursive Methoden	165
6.2.3	Zusammenfassung	168
6.3	Die Methode <code>main</code>	168
6.3.1	Kommandozeilenparameter	169
6.3.2	Anwendung der vereinfachten <code>for</code> -Schleifen-Notation . . .	170
6.3.3	Zusammenfassung	171
6.3.4	Übungsaufgaben	171
6.4	Methoden aus anderen Klassen aufrufen	173
6.4.1	Klassenmethoden	173
6.4.2	Die Methoden der Klasse <code>java.lang.Math</code>	174
6.4.3	Statischer Import	175
6.5	Methoden von Objekten aufrufen	176
6.5.1	Instanzmethoden	176
6.5.2	Die Methoden der Klasse <code>java.lang.String</code>	177
6.6	Übungsaufgaben	180

II Objektorientiertes Programmieren in Java 185

7	Die objektorientierte Philosophie	187
7.1	Die Welt, in der wir leben	187
7.2	Programmierparadigmen – Objektorientierung im Vergleich	188
7.3	Die vier Grundpfeiler objektorientierter Programmierung	190
7.3.1	Generalisierung	190
7.3.2	Vererbung	192
7.3.3	Kapselung	195
7.3.4	Polymorphismus	196
7.3.5	Weitere wichtige Grundbegriffe	197
7.4	Modellbildung – von der realen Welt in den Computer	198
7.4.1	Grafisches Modellieren mit UML	198
7.4.2	Entwurfsmuster	199
7.5	Zusammenfassung	200

7.6	Übungsaufgaben	201
8	Der grundlegende Umgang mit Klassen	203
8.1	Vom Referenzdatentyp zur Objektorientierung	203
8.2	Instanzmethoden	205
8.2.1	Zugriffsrechte	205
8.2.2	Was sind Instanzmethoden?	206
8.2.3	Instanzmethoden zur Validierung von Eingaben	209
8.2.4	Instanzmethoden als erweiterte Funktionalität	210
8.3	Statische Komponenten einer Klasse	211
8.3.1	Klassenvariablen und -methoden	212
8.3.2	Klassenkonstanten	214
8.4	Instantiierung und Initialisierung	215
8.4.1	Konstruktoren	215
8.4.2	Überladen von Konstruktoren	217
8.4.3	Der statische Initialisierer	219
8.4.4	Der Mechanismus der Objekterzeugung	222
8.5	Zusammenfassung	226
8.6	Übungsaufgaben	227
9	Vererbung und Polymorphismus	247
9.1	Wozu braucht man Vererbung?	247
9.1.1	Aufgabenstellung	247
9.1.2	Analyse des Problems	248
9.1.3	Ein erster Ansatz	248
9.1.4	Eine Klasse für sich	249
9.1.5	Stärken der Vererbung	250
9.1.6	Vererbung verhindern durch final	253
9.1.7	Übungsaufgaben	254
9.2	Die super -Referenz	255
9.3	Überschreiben von Methoden und Variablen	257
9.3.1	Dynamisches Binden	257
9.3.2	Überschreiben von Methoden verhindern durch final	259
9.4	Die Klasse <code>java.lang.Object</code>	260
9.5	Übungsaufgaben	263
9.6	Abstrakte Klassen und Interfaces	263
9.7	Übungsaufgaben	267
9.8	Weiteres zum Thema Objektorientierung	272
9.8.1	Erstellen von Paketen	272
9.8.2	Zugriffsrechte	273
9.8.3	Innere Klassen	274
9.8.4	Anonyme Klassen	280

9.9 Zusammenfassung	282
9.10 Übungsaufgaben	282
10 Exceptions und Errors	293
10.1 Eine Einführung in Exceptions	294
10.1.1 Was ist eine Exception?	294
10.1.2 Übungsaufgaben	296
10.1.3 Abfangen von Exceptions	296
10.1.4 Ein Anwendungsbeispiel	297
10.1.5 Die RuntimeException	300
10.1.6 Übungsaufgaben	301
10.2 Exceptions für Fortgeschrittene	303
10.2.1 Definieren eigener Exceptions	303
10.2.2 Übungsaufgaben	305
10.2.3 Vererbung und Exceptions	305
10.2.4 Vorsicht, Falle!	309
10.2.5 Der finally -Block	311
10.2.6 Die Klassen Throwable und Error	315
10.2.7 Zusammenfassung	317
10.2.8 Übungsaufgaben	317
10.3 Assertions	318
10.3.1 Zusicherungen im Programmcode	318
10.3.2 Compilieren des Programmcodes	319
10.3.3 Ausführen des Programmcodes	320
10.3.4 Zusammenfassung	320
11 Fortgeschrittene objektorientierte Programmierung	321
11.1 Aufzählungstypen	322
11.1.1 Deklaration eines Aufzählungstyps	322
11.1.2 Instanzmethoden der enum -Objekte	323
11.1.3 Selbstdefinierte Instanzmethoden für enum -Objekte	323
11.1.4 Übungsaufgaben	325
11.2 Generische Datentypen	327
11.2.1 Generizität in alten Java-Versionen	327
11.2.2 Generizität ab Java 5.0	330
11.2.3 Einschränkungen der Typ-Parameter	332
11.2.4 Wildcards	334
11.2.5 Bounded Wildcards	335
11.2.6 Generische Methoden	337
11.2.7 Ausblick	339
11.2.8 Übungsaufgaben	339
11.3 Sortieren von Feldern und das Interface Comparable	344

12 Einige wichtige Hilfsklassen	347
12.1 Die Klasse <code>StringBuffer</code>	347
12.1.1 Arbeiten mit <code>String</code> -Objekten	347
12.1.2 Arbeiten mit <code>StringBuffer</code> -Objekten	350
12.1.3 Übungsaufgaben	352
12.2 Die Wrapper-Klassen (Hüll-Klassen)	353
12.2.1 Arbeiten mit „eingepackten“ Daten	353
12.2.2 Aufbau der Wrapper-Klassen	354
12.2.3 Ein Anwendungsbeispiel	357
12.2.4 Automatische Typwandlung für die Wrapper-Klassen	358
12.2.5 Übungsaufgaben	360
12.3 Die Klassen <code>BigInteger</code> und <code>BigDecimal</code>	361
12.3.1 Arbeiten mit langen Ganzzahlen	361
12.3.2 Aufbau der Klasse <code>BigInteger</code>	363
12.3.3 Übungsaufgaben	365
12.3.4 Arbeiten mit langen Gleitkommazahlen	365
12.3.5 Aufbau der Klasse <code>BigDecimal</code>	368
12.3.6 Viele Stellen von Nullstellen gefällig?	371
12.3.7 Übungsaufgaben	372
12.4 Die Klasse <code>DecimalFormat</code>	373
12.4.1 Standard-Ausgaben in Java	373
12.4.2 Arbeiten mit <code>Format</code> -Objekten	374
12.4.3 Vereinfachte formatierte Ausgabe	376
12.4.4 Übungsaufgaben	377
12.5 Die Klassen <code>Date</code> und <code>Calendar</code>	377
12.5.1 Arbeiten mit „Zeitpunkten“	378
12.5.2 Auf die Plätze, fertig, los!	379
12.5.3 Spezielle <code>Calendar</code> -Klassen	380
12.5.4 Noch einmal: Zeitmessung	382
12.5.5 Übungsaufgaben	384
12.6 Die Klassen <code>SimpleDateFormat</code> und <code>DateFormat</code>	384
12.6.1 Arbeiten mit <code>Format</code> -Objekten für Datum/Zeit-Angaben	384
12.6.2 Übungsaufgaben	389
12.7 Die <code>Collection</code> -Klassen	389
12.7.1 „Sammlungen“ von Objekten – Der Aufbau des Interface <code>Collection</code>	389
12.7.2 „Sammlungen“ durchgehen – Der Aufbau des Interface <code>Iterator</code>	392
12.7.3 Mengen	393
12.7.3.1 Das Interface <code>Set</code>	393
12.7.3.2 Die Klasse <code>HashSet</code>	393

12.7.3.3	Das Interface SortedSet	395
12.7.3.4	Die Klasse TreeSet	396
12.7.4	Listen	397
12.7.4.1	Das Interface List	398
12.7.4.2	Die Klassen ArrayList und LinkedList	398
12.7.4.3	Suchen und Sortieren – Die Klassen Collections und Arrays	400
12.7.5	Übungsaufgaben	403
12.8	Die Klasse StringTokenizer	404
12.8.1	Übungsaufgaben	406

III Grafische Oberflächen in Java 407

13	Aufbau grafischer Oberflächen in Frames – von AWT nach Swing	409
13.1	Grundsätzliches zum Aufbau grafischer Oberflächen	409
13.2	Ein einfaches Beispiel mit dem AWT	411
13.3	Let's swing now!	413
13.4	Etwas „Fill-in“ gefällig?	415
13.5	Die AWT- und Swing-Klassenbibliothek im Überblick	417
13.6	Übungsaufgaben	419
14	Swing-Komponenten	421
14.1	Die abstrakte Klasse Component	421
14.2	Die Klasse Container	422
14.3	Die abstrakte Klasse JComponent	423
14.4	Layout-Manager, Farben und Schriften	424
14.4.1	Die Klasse Color	425
14.4.2	Die Klasse Font	427
14.4.3	Layout-Manager	428
14.4.3.1	Die Klasse FlowLayout	429
14.4.3.2	Die Klasse BorderLayout	431
14.4.3.3	Die Klasse GridLayout	432
14.5	Einige Grundkomponenten	434
14.5.1	Die Klasse JLabel	436
14.5.2	Die abstrakte Klasse AbstractButton	436
14.5.3	Die Klasse JButton	438
14.5.4	Die Klasse JToggleButton	439
14.5.5	Die Klasse JCheckBox	440
14.5.6	Die Klassen JRadioButton und ButtonGroup	441
14.5.7	Die Klasse JComboBox	443
14.5.8	Die Klasse JList	446

14.5.9	Die abstrakte Klasse JTextComponent	449
14.5.10	Die Klassen JTextField und JPasswordField	450
14.5.11	Die Klasse JTextArea	452
14.5.12	Die Klasse JScrollPane	454
14.5.13	Die Klasse JPanel	456
14.6	Spezielle Container, Menüs und Toolbars	458
14.6.1	Die Klasse JFrame	458
14.6.2	Die Klasse JWindow	459
14.6.3	Die Klasse JDialog	459
14.6.4	Die Klasse JMenuBar	463
14.6.5	Die Klasse JToolBar	465
14.7	Übungsaufgaben	468
15	Ereignisverarbeitung	471
15.1	Zwei einfache Beispiele	472
15.1.1	Zufällige Grautöne als Hintergrund	472
15.1.2	Ein interaktiver Bilderrahmen	475
15.2	Programmiervarianten für die Ereignisverarbeitung	479
15.2.1	Innere Klasse als Listener-Klasse	479
15.2.2	Anonyme Klasse als Listener-Klasse	479
15.2.3	Container-Klasse als Listener-Klasse	480
15.2.4	Separate Klasse als Listener-Klasse	481
15.3	Event-Klassen und -Quellen	483
15.4	Listener-Interfaces und Adapter-Klassen	487
15.5	Listener-Registrierung bei den Event-Quellen	492
15.6	Auf die Plätze, fertig, los!	496
15.7	Übungsaufgaben	500
16	Einige Ergänzungen zu Swing-Komponenten	505
16.1	Zeichnen in Swing-Komponenten	505
16.1.1	Grafische Darstellung von Komponenten	505
16.1.2	Das Grafik-Koordinatensystem	506
16.1.3	Die abstrakte Klasse Graphics	507
16.1.4	Ein einfaches Zeichenprogramm	510
16.1.5	Layoutveränderungen und der Einsatz von revalidate	512
16.2	Noch mehr Swing gefällig?	515
16.3	Übungsaufgaben	516
17	Applets	519
17.1	Erstellen und Ausführen von Applets	519
17.1.1	Vom Frame zum Applet am Beispiel	519
17.1.2	Applet in HTML-Datei einbetten	521

17.1.3	Applet über HTML-Datei ausführen	523
17.2	Die Methoden der Klasse JApplet	524
17.3	Zwei Beispiele	526
17.3.1	Auf die Plätze, fertig, los!	527
17.3.2	Punkte verbinden im Applet	530
17.4	Details zur HTML-Einbettung	531
17.4.1	Der Applet-Tag	531
17.4.2	Die Methode showDocument	534
17.5	Sicherheitseinschränkungen bei Applets	536
17.6	Übungsaufgaben	540
 IV Threads, Datenströme und Netzwerk-Anwendungen		543
18	Parallele Programmierung mit Threads	545
18.1	Ein einfaches Beispiel	545
18.2	Threads in Java	547
18.2.1	Die Klasse Thread	548
18.2.2	Das Interface Runnable	552
18.2.3	Threads vorzeitig beenden	554
18.3	Wissenswertes über Threads	556
18.3.1	Lebenszyklus eines Threads	556
18.3.2	Thread-Scheduling	558
18.3.3	Dämon-Threads und Thread-Gruppen	558
18.4	Thread-Synchronisation und -Kommunikation	559
18.4.1	Das Leser/Schreiber-Problem	560
18.4.2	Das Erzeuger/Verbraucher-Problem	564
18.5	Threads in Frames und Applets	571
18.5.1	Auf die Plätze, fertig, los!	571
18.5.2	Spielereien	575
18.5.3	Swing-Komponenten sind nicht Thread-sicher	577
18.6	Übungsaufgaben	578
19	Ein- und Ausgabe über Streams	581
19.1	Grundsätzliches zu Streams in Java	582
19.2	Dateien und Verzeichnisse – Die Klasse File	582
19.3	Ein- und Ausgabe über Character-Streams	585
19.3.1	Einfache Reader- und Writer-Klassen	586
19.3.2	Gepufferte Reader- und Writer-Klassen	589
19.3.3	Die Klasse StreamTokenizer	591
19.3.4	Die Klasse PrintWriter	592
19.3.5	Die Klassen IOTools und Scanner	594

19.3.5.1	Was machen eigentlich die IOTools?	594
19.3.5.2	Konsoleneingabe über ein Scanner-Objekt	596
19.4	Ein- und Ausgabe über Byte-Streams	596
19.4.1	Einige InputStream- und OutputStream-Klassen	597
19.4.2	Die Serialisierung und Deserialisierung von Objekten	599
19.4.3	Die Klasse PrintStream	601
19.5	Einige abschließende Bemerkungen	602
19.6	Übungsaufgaben	603
20	Client/Server-Programmierung in Netzwerken	605
20.1	Wissenswertes über Netzwerk-Kommunikation	606
20.1.1	Protokolle	606
20.1.2	IP-Adressen	608
20.1.3	Ports und Sockets	609
20.2	Client/Server-Programmierung	610
20.2.1	Die Klassen ServerSocket und Socket	611
20.2.2	Ein einfacher Server	613
20.2.3	Ein einfacher Client	616
20.2.4	Ein Server für mehrere Clients	617
20.2.5	Ein Mehrzweck-Client	620
20.3	Wissenswertes über URLs	623
20.3.1	Client/Server-Kommunikation über URLs	623
20.3.2	Netzwerkverbindungen in Applets	624
20.4	Übungsaufgaben	625
V	Aktuelles, Ausblick und Anhang	629
21	Neuerungen in Java 7	631
21.1	Spracherweiterungen	631
21.1.1	Elementare Datentypen und Anweisungen	631
21.1.1.1	Binäre ganzzahlige Literalkonstanten	631
21.1.1.2	Unterstrich als Trennzeichen in Literalkonstanten	632
21.1.1.3	Strings in der switch -Anweisung	633
21.1.2	Verkürzte Notation bei generischen Datentypen	636
21.1.3	Ausnahmebehandlung	640
21.1.3.1	Mehrere Ausnahme-Typen in einem catch -Block	640
21.1.3.2	try -Block mit Ressourcen	643
21.2	Erweiterungen der Klassenbibliothek	646
21.2.1	Dateien und Verzeichnisse	646
21.2.1.1	Das Interface Path und die Klasse Paths	646
21.2.1.2	Die Klasse Files	647

21.2.2 Grafische Oberflächen	650
22 Blick über den Tellerrand	653
22.1 Der Vorhang fällt	653
22.2 A fool with a tool	654
22.3 Alles umsonst?	655
22.4 Und fachlich?	656
22.5 Zu guter Letzt	658
A Der Weg zum guten Programmierer	659
A.1 Die goldenen Regeln der Code-Formatierung	660
A.2 Die goldenen Regeln der Namensgebung	663
A.3 Zusammenfassung	665
B Die Klasse <code>IOTools</code> – Tastatureingaben in Java	667
B.1 Kurzbeschreibung	667
B.2 Anwendung der <code>IOTools</code> -Methoden	668
C Der Umgang mit der API-Spezifikation	671
C.1 Der Aufbau der API-Spezifikation	671
C.2 Der praktische Einsatz der API-Spezifikation	672
D Glossar	677
Literaturverzeichnis	691
Stichwortverzeichnis	695