

Inhalt

Vorwort	XI
1 Der Einstieg in Java	1
1.1 Erstellung und Ablauf von Programmen in Java	2
1.2 Das erste Java-Programm	3
1.3 Erstellung und Ablauf des ersten Programms	5
1.4 Ein erstes Applet	9
1.5 Zusammenfassung	11
1.6 Aufgaben	12
2 Elemente der Programmierung	13
2.1 Daten erklären und verarbeiten	14
2.1.1 Schreibweisen für Deklarationen und Wertzuweisungen	16
2.1.2 Beispiel: Elementare Ausdrücke	18
2.1.3 Beispiel: Bereichsüberschreitungen	20
2.1.4 Typumwandlungen	21
2.1.5 Deklarationen mit dem static-Modifizierer	22
2.1.6 Namen und ihre Gültigkeit	23
2.2 Kontrollfluss	24
2.2.1 Verzweigung	25
2.2.2 Mehrfachverzweigung	29
2.2.3 Schleifen mit Vorabprüfung	32
2.2.4 Schleife mit Prüfung am Ende	36
2.2.5 Verlassen von Schleifen	38
2.2.6 Programmausnahmen	39
2.3 Methoden	43
2.3.1 Definitionen	43
2.3.2 Beispiele zum Einsatz von Methoden	45
2.3.3 Rekursion	48
2.3.3.1 Beispiel: Berechnung der Fakultät	49
2.3.3.2 Beispiel: Die Türme von Hanoi	50

2.4	Felder	54
2.4.1	Eindimensionale Felder	55
2.4.1.1	Grundlegende Definitionen	55
2.4.1.2	Beispiel: Einlesen und Bearbeiten eines Felds	58
2.4.1.3	Behandlung von Indexfehlern	59
2.4.2	Suche in Feldern	60
2.4.2.1	Lineare Suche	60
2.4.2.2	Halbierungsmethode: binäre Suche in Feldern	62
2.4.3	Sortieren	63
2.4.4	Mehrdimensionale Felder	66
2.5	Operatoren in Java	68
2.6	ANSI-Escape-Sequenzen	72
2.7	Aufgaben	73
3	Objektorientierte Programmierung	77
3.1	Auf dem Weg zu Klassen	77
3.1.1	Wege zur Objektorientierung	77
3.1.2	Beziehungen zwischen Klassen	80
3.1.3	Oberklassen und Unterklassen	81
3.1.4	Klassen und Objekte	82
3.1.5	Abstrakte Klassen	82
3.1.6	Entwurf der Klassen	83
3.1.6.1	Typ 1: die vorgegebenen Objekte	83
3.1.6.2	Typ 2: Verwaltungsobjekte oder Sammlungen	83
3.1.6.3	Typ 3: Umgebungsobjekte	83
3.1.6.4	Typ 4: übersehene Klassen doch noch finden	84
3.1.6.5	Gemeinsame Oberklassen finden	84
3.2	Klassen in Java	84
3.2.1	Eine Klasse zum Verwalten von Mitarbeitern	86
3.2.2	Erzeugung von Objekten: Konstruktoren	89
3.2.3	Wertzuweisung und Übergabe als Parameter	90
3.2.4	Statische Klassenelemente	91
3.2.4.1	Grundlagen	92
3.2.4.2	Initialisierung der static-Variablen in einer Klasse	93
3.2.5	Eingeschachtelte Klassen, innere Klassen	94
3.2.6	Umwicklertypen	96
3.3	Unterklassen und Polymorphie in Java	98
3.3.1	Definition von Unterklassen in Java	98
3.3.2	Methoden der Klasse Object überschreiben	101
3.3.3	Lebenszyklus von Objekten	102
3.3.4	Wie funktioniert die Polymorphie?	104
3.3.5	Wertzuweisung und Cast-Anweisung	105
3.3.6	Klassen und Ausnahmen	107
3.3.7	Abstrakte Klassen: Design für Vererbung	109

3.4	Generische Elemente in Java	113
3.5	Schnittstellen in Java	116
3.5.1	Vergleich von Objekten	118
3.5.2	Statische Methoden in Schnittstellen	121
3.5.3	Default-Methoden in Schnittstellen	121
3.5.4	Schnittstellen ohne abstrakte Methoden	123
3.5.5	Funktionale Schnittstellen	125
3.6	Funktionen in Java 8	126
3.6.1	Referenzen auf Funktionen in Java 8	126
3.6.2	Lambda-Ausdrücke	127
3.6.2.1	Syntax für Lambda-Ausdrücke	127
3.6.2.2	Bindung in Lambda-Ausdrücken	128
3.6.3	Beispiel: Anwendung der Comparator-Schnittstelle	129
3.6.4	Beispiel: Funktionen als Parameter	130
3.6.5	Zusammenfassung	132
3.7	Dynamische Erzeugung von Objekten	132
3.8	Aufzählung von Konstanten mit enum	137
3.9	Allgemeine Eigenschaften	140
3.9.1	Der final-Modifizierer	140
3.9.2	Packages mit package, import	140
3.9.3	Sichtbarkeit von Namen in Java	142
3.9.4	Wiederherstellung des Zustands eines Objekts: Serialisierung	142
3.9.5	Zusicherungen	144
3.10	Aufgaben	146
4	Grundlegende Klassen	151
4.1	Nützliche Klassen und Packages	151
4.1.1	Übersicht der Aufgaben des Package java.lang	152
4.1.2	Zeichenketten in Java: String	152
4.1.3	Die Klasse System	156
4.1.4	Die Klasse Math	156
4.1.5	Zeit und Datum in Java	157
4.1.6	Reflexion von Java-Programmen	160
4.1.7	Annotationen	162
4.1.8	Reguläre Ausdrücke	164
4.1.9	Protokollierung von Programmläufen: Logging	167
4.2	Verwalten von Objekten mit Sammlungen	169
4.2.1	Prinzip für die Aufbewahrung von Objekten	170
4.2.1.1	Schnittstellen für die Sammlungen in Java	171
4.2.1.2	Implementierungen für die Schnittstellen	172
4.2.2	Sequenzieller Zugriff: List, Set und SortedSet	173
4.2.2.1	Collection als Basisschnittstelle	173
4.2.2.2	Listen	175
4.2.2.3	Die ListIterator-Schnittstelle	178
4.2.2.4	Mengen	180

4.2.3	Assoziativer Zugriff: Map	184
4.2.3.1	Map als Basisschnittstelle	185
4.2.3.2	Die SortedMap-Schnittstelle	188
4.2.4	Nützliche Klassen und Methoden für Sammlungen	189
4.2.4.1	Die Klasse Collections	189
4.2.4.2	Implementierungen von Sammlungen für spezielle Zwecke	190
4.2.4.3	Sammlungen und Threads	191
4.2.4.4	Nützliche Klassen und Methoden	191
4.3	Streams in Java	192
4.3.1	Einstieg in die funktionale Programmierung	193
4.3.2	Ausgewählte Methoden	195
4.3.3	Fallbeispiele – Anwendungsfälle	197
4.3.3.1	Zeichenketten aneinanderhängen	197
4.3.3.2	Sortieren	198
4.3.3.3	Gruppieren von Objekten nach diversen Kriterien	199
4.3.3.4	Verarbeiten von Daten in Textdateien	200
4.3.3.5	Berechnungen mit Zwischenergebnissen durchführen	201
4.4	Aufgaben	202
5	Ein-/Ausgabe in Java	205
5.1	Prinzip der Ein-/Ausgabe in Java	208
5.1.1	Eingabe in Java	210
5.1.1.1	InputStream als Basisklasse für Eingaben	210
5.1.1.2	Reader als Brücke zwischen Bytes und Zeichen	212
5.1.2	Ausgabe in Java	214
5.1.2.1	OutputStream als Basisklasse für Ausgaben	214
5.1.2.2	Die Writer-Klassen in Java	219
5.2	Fallstudien zu Ein-/Ausgabe	220
5.2.1	Bearbeiten von Textdateien	220
5.2.2	Durchlaufen aller Dateien in einem Verzeichnis	222
5.2.3	Zugriff auf die Einträge in einem ZIP-Archiv	223
5.3	Aufgaben	224
6	Nebenläufigkeit in Java: Threads	227
6.1	Einstieg in Threads in Java	227
6.1.1	Implizite Programmierung von Threads	228
6.1.2	Streams parallel bearbeiten	229
6.1.3	Explizite Programmierung von Threads	230
6.2	Grundlagen zu Threads	232
6.2.1	Nutzen von Threads	232
6.2.2	Wettrennen	235
6.2.3	Zustände von Threads	239
6.2.4	Wichtige Methoden für Threads	240

6.3	Monitore in Java	241
6.3.1	Grundlagen des Monitorkonzepts in Java	241
6.3.2	Anwendung der Monitore in Java	241
6.4	Anwendungsfälle	243
6.4.1	Lang laufende Aktivitäten in Benutzungsoberflächen	244
6.4.2	Erzeuger-Verbraucher-Kopplung	247
6.4.3	Leser-Schreiber-Problem	249
6.4.4	Semaphoren	249
6.4.5	Verklemmungen und die fünf Philosophen	251
6.4.6	Animationen	253
6.5	Aufgaben	256
7	Grafikanwendungen in Java	259
7.1	Struktur von GUI-Anwendungen	259
7.1.1	Ein erstes Programm für Swing	260
7.1.2	Prinzip der ereignisgesteuerten Programmierung	261
7.1.3	Ereignissteuerung	262
7.1.3.1	Das „Delegation Event Model“	264
7.1.3.2	Listener-Schnittstellen und Adapter	266
7.1.4	Hierarchie der Swing-Klassen für Steuerelemente	267
7.1.5	Elementare Steuerelemente	268
7.1.6	Das Model-View-Controller-Paradigma und Swing	269
7.2	Anordnung der Komponenten	269
7.2.1	BorderLayout	271
7.2.2	FlowLayout	271
7.2.3	GridLayout	272
7.2.4	CardLayout	273
7.2.5	GridBagLayout	274
7.2.6	BoxLayout (nur Swing)	277
7.2.7	Schachtelung der Layouts	278
7.3	Steuerelemente in Benutzeroberflächen	279
7.3.1	Schaltflächen: JButton	279
7.3.2	Checkboxes und Radiobutton	280
7.3.3	Statischer Text zur Anzeige von Informationen	281
7.3.4	Listen zur Auswahl	282
7.3.5	Elementare Auswahl mit der Combobox	284
7.3.6	Textfelder	285
7.3.7	Menüs in Java	286
7.4	Steuerelemente unter der MVC-Architektur	288
7.4.1	Übersicht: Aufgabenverteilung Swing-Anwender	289
7.4.2	Vertiefung für JList und JComboBox	290
7.4.2.1	Eine MVC-Anwendung für JList	291
7.4.2.2	JComboBox	293

7.4.3	Tabellen und Baumsteuerelemente	294
7.4.3.1	Das Steuerelement für Tabellen JTable	294
7.4.3.2	JTree	298
7.5	Kurs: GUI-Anwendungen	304
7.5.1	Erstellung einer grafischen Komponente	305
7.5.2	Reaktion auf Mausklicks	307
7.5.3	Reaktion auf Mausbewegungen: ein Malprogramm	309
7.5.4	Turtle-Grafik	311
7.5.5	Dialoge in Java	315
7.5.6	Dialog zur Auswahl von Dateinamen	319
7.5.7	Die Türme von Hanoi	321
7.6	Aufgaben	325
8	Programmierung in Netzwerken	331
8.1	Programmierung von Sockets	331
8.1.1	Grundlagen von Netzwerken	332
8.1.2	Sockets in Java	333
8.1.2.1	Verbindungsorientierte Kommunikation mit TCP	333
8.1.2.2	Verbindungslose Kommunikation	336
8.1.2.3	Java-Klassen für Internetadressen und zur Darstellung von URLs	337
8.2	Verteilte Anwendungen	338
8.2.1	Der Additions-Dienst mit TCP	339
8.2.1.1	Problemanalyse: Datenaustausch	339
8.2.1.2	Problemanalyse: Aufbau der Anwendung	340
8.2.2	Beispiel: der Additions-Dienst mit UDP	342
8.2.2.1	Problemanalyse: Datenaustausch	342
8.2.2.2	Problemanalyse: Aufbau der Anwendung	342
8.2.3	RMI	344
8.2.3.1	Problemanalyse: Datenaustausch	344
8.2.3.2	Problemanalyse: Aufbau der Anwendung	345
8.2.4	Prinzip von RMI	345
8.3	Aufgaben	348
9	Anbindung von Datenbanken mit JDBC	349
9.1	Grundlagen von JDBC	350
9.1.1	Grundsätzlicher Ablauf beim Zugriff	350
9.1.2	Einstieg in relationale Datenbanken und SQL	353
9.1.3	Klassen und Schnittstellen im Package java.sql	355
9.2	Beispiel: Die Datenbank mit den Speisen in JDBC	362
9.2.1	Programmierung der Verbindung zu den Datenbanken	362
9.2.2	Vorbereitung: Datenbanken einrichten	363
9.2.3	Zugriffe mit JDBC	364
9.2.3.1	Die Speisedatenbank	364

9.2.3.2	Die Speisedatenbank: objektrelationale Zuordnung	364
9.2.3.3	Die Speisedatenbank: Vermeidung doppelter Einträge	365
9.3	Datentypen in Java und SQL	367
9.4	Metadaten	368
9.4.1	Metadaten und die Auskunft über die Datenbank	368
9.4.2	Anwendung	369
9.5	Aufgaben	371
10	Bearbeiten von XML in Java	373
10.1	Schreiben von XML und Lesen mittels JAXB	374
10.1.1	Zusammenhänge: Klasse und Objekt bzw. Schema und XML	375
10.1.2	Kochrezept: Anleitung zur Benützung von JAXB	377
10.1.2.1	Schritt: Kennzeichnung eines Objekts zum Speichern	377
10.1.2.2	Schritt: Programm zum Speichern und Laden	377
10.2	SAX-Parser	378
10.3	Aufgaben	381
11	Werkzeuge für die Java-Programmierung	383
11.1	Der Compiler javac	383
11.2	Der Interpreter java	384
11.3	Applet-Viewer	384
11.4	Der Generator für die Dokumentation	385
11.5	Der Disassembler	385
Literatur	387
Index	389