

Inhaltsverzeichnis

I GRUNDLAGEN

1 Einleitung	1
1.1 Java	1
1.2 Vom Quellcode zum ausführbaren Programm	3
1.3 Entwicklungsumgebung	6
1.4 Zielsetzung und Aufbau des Buches	8
1.5 Programm- und Aufgabensammlung	10
1.6 Elementare Regeln und Konventionen	10
2 Datentypen und Variablen	13
2.1 Einfache Datentypen	13
2.2 Variablen	16
2.3 Aufgaben	18
3 Operatoren	19
3.1 Arithmetische Operatoren	19
3.2 Relationale Operatoren	22
3.3 Logische Operatoren	22
3.4 Bitoperatoren	24
3.5 Zuweisungsoperatoren	25
3.6 Bedingungsoperator	26
3.7 Cast-Operator	27
3.8 Aufgaben	28
4 Kontrollstrukturen	31
4.1 Verzweigungen	31
4.2 Schleifen	35
4.3 Sprunganweisungen	38
4.4 Die Java-Shell	40
4.5 Aufgaben	40
5 Arrays	45
5.1 Definition und Initialisierung	45
5.2 Zugriff auf Array-Elemente	46
5.3 Kommandozeilen-Parameter	49
5.4 Aufgaben	50

II OBJEKTORIENTIERUNG

6 Klassen, Objekte und Methoden	51
6.1 Klassen und Objekte	51

6.2 Methoden.....	56
6.3 Konstruktoren.....	60
6.4 Klassenvariablen und Klassenmethoden.....	63
6.5 Varargs.....	65
6.6 Iterative und rekursive Problemlösungen.....	66
6.7 Records.....	68
6.8 Aufgaben.....	69
7 Pakete.....	75
7.1 Pakete erzeugen und nutzen.....	75
7.2 Eigene Java-Bibliotheken erzeugen.....	78
7.3 Aufgaben.....	80
8 Vererbung.....	81
8.1 Klassen erweitern.....	81
8.2 Konstruktoren und Vererbung.....	84
8.3 Methodenauswahl zur Laufzeit.....	87
8.4 Pattern Matching.....	88
8.5 Abstrakte Klassen.....	91
8.6 Modifizierer.....	93
8.7 Versiegelte Klassen.....	95
8.8 Aufgaben.....	95
9 Interfaces.....	99
9.1 Interfaces definieren und einsetzen.....	99
9.2 Default-Methoden.....	105
9.3 Statische und private Methoden in Interfaces.....	110
9.4 Aufgaben.....	111
10 Vererbung vs. Delegation.....	115
10.1 Warum Vererbung problematisch sein kann.....	115
10.2 Delegation als Alternative.....	117
10.3 Aufgaben.....	119
11 Geschachtelte Klassen.....	121
11.1 Statische Klasse.....	121
11.2 Instanzklasse.....	122
11.3 Lokale Klasse.....	124
11.4 Anonyme Klasse.....	126
11.5 Aufgaben.....	128
12 Konstanten und enum-Aufzählungen.....	129
12.1 Verwendung von int-Konstanten.....	129
12.2 Einfache enum-Aufzählung.....	131

12.3 enum-Aufzählung mit Attributen und Methoden.....	132
12.4 Konstantenspezifische Implementierung von Methoden.....	134
12.5 Singleton.....	134
12.6 Aufgaben.....	136

13 Ausnahmebehandlung.....	137
13.1 Ausnahmetypen.....	137
13.2 Auslösung und Weitergabe von Ausnahmen.....	140
13.3 Abfangen von Ausnahmen.....	143
13.4 Verkettung von Ausnahmen.....	146
13.5 Aufgaben.....	148

III. NÜTZLICHE KLASSEN DER STANDARDBIBLIOTHEK

14 Zeichenketten.....	151
14.1 Die Klasse String.....	151
14.2 Mehrzeilige Texte.....	160
14.3 Die Klassen StringBuilder und StringBuffer.....	161
14.4 Die Klasse StringTokenizer.....	163
14.5 Aufgaben.....	164

15 Ausgewählte Klassen.....	167
15.1 Wrapper-Klassen für einfache Datentypen.....	167
15.2 Die Klasse Object.....	173
15.3 Die Klassen Vector, Hashtable und Properties.....	180
15.4 Die Klasse System.....	188
15.5 Die Klasse Class.....	191
15.6 Die Klasse Arrays.....	196
15.7 Mathematische Funktionen.....	199
15.8 Datum und Uhrzeit.....	204
15.9 Aufgaben.....	214

16 Internationalisierung.....	221
16.1 Die Klasse Locale.....	221
16.2 Zeit und Zahlen darstellen.....	224
16.3 Sprachspezifische Sortierung.....	227
16.4 Ressourcenbündel.....	228
16.5 Aufgaben.....	229

17 Services.....	231
17.1 Service Provider.....	231
17.2 Service Consumer.....	234
17.3 Aufgaben.....	234

X	Inhaltsverzeichnis
18 Javadoc	237
18.1 javadoc-Syntax.....	237
18.2 Das Werkzeug javadoc.....	239
18.3 Aufgaben.....	240
IV. GENERISCHE PROGRAMMIERUNG	
19 Generische Typen und Methoden	241
19.1 Einführung.....	241
19.2 Typeinschränkungen.....	244
19.3 Raw Types.....	248
19.4 Wildcards.....	249
19.5 Generische Methoden.....	254
19.6 Grenzen des Generics-Konzepts.....	255
19.7 Aufgaben.....	256
20 Collections	259
20.1 Die Interfaces Collection, Iterable und Iterator.....	259
20.2 Listen.....	261
20.3 Mengen.....	264
20.4 Schlüsseltabellen.....	265
20.5 Erzeugung von Collections für vordefinierte Werte.....	268
20.6 Sequenced Collections.....	269
20.7 Aufgaben.....	270
V. FUNKTIONALE PROGRAMMIERUNG	
21 Lambda-Ausdrücke	275
21.1 Funktionsinterfaces.....	276
21.2 Lambdas.....	278
21.3 Methodenreferenzen.....	282
21.4 Weitere Beispiele.....	285
21.5 Aufgaben.....	293
22 Streams	295
22.1 Das Stream-Konzept.....	295
22.2 Erzeugung von Streams.....	296
22.3 Transformation und Auswertung.....	297
22.4 Aufgaben.....	301
23 Optionale Werte	307
23.1 Motivation.....	307
23.2 Die Klasse Optional.....	308
23.3 Weitere Beispiele.....	313

Inhaltsverzeichnis	XI
23.4 Aufgaben.....	315
VI. DATENSTRÖME	
24 Dateien und Datenströme.....	317
24.1 Dateien und Verzeichnisse bearbeiten.....	317
24.2 Zeichensätze und Codierungen.....	325
24.3 Datenströme.....	328
24.4 Daten lesen und schreiben.....	334
24.5 Texte lesen und schreiben.....	341
24.6 Datenströme filtern.....	347
24.7 Wahlfreier Dateizugriff.....	349
24.8 Datenkomprimierung.....	352
24.9 Aufgaben.....	357
25 Serialisierung.....	363
25.1 Serialisieren und Deserialisieren.....	363
25.2 Sonderbehandlung bei Serialisierung.....	366
25.3 Das Datenformat JSON.....	369
25.4 Aufgaben.....	376
VII. NEBENLÄUFIGE PROGRAMMIERUNG	
26 Nebenläufige Programmierung.....	379
26.1 Threads erzeugen, starten und beenden.....	379
26.2 Lebenszyklus eines Threads.....	388
26.3 Synchronisation.....	388
26.4 Fallbeispiel Summenberechnung.....	394
26.5 Thread-Sicherheit.....	397
26.6 Kommunikation zwischen Threads.....	403
26.7 Shutdown-Threads.....	413
26.8 Thread-Pools und Executors.....	415
26.9 Asynchrone Verarbeitung mit CompletableFuture.....	418
26.10 Das Process-API.....	421
26.11 Virtuelle Threads.....	423
26.12 Aufgaben.....	426
VIII. GRAFISCHE OBERFLÄCHEN	
27 GUI-Programmierung mit Swing.....	435
27.1 Ein Fenster erzeugen mit JFrame.....	437
27.2 Ereignisbehandlung.....	440
27.3 Layouts.....	446
27.4 Buttons und Labels.....	458
27.5 Spezielle Container.....	463

27.6 Textkomponenten.....	467
27.7 Auswahlkomponenten.....	472
27.8 Menüs und Symbolleisten.....	476
27.9 Dialoge.....	482
27.10 Tabellen.....	488
27.11 Event-Queue und Event-Dispatcher.....	493
27.12 Aufgaben.....	499
 28 Einführung in JavaFX.....	 505
28.1 Installation und Konfiguration.....	505
28.2 Ein erstes Beispiel.....	507
28.3 Beispiel Brutto-Rechner.....	510
28.4 Asynchrone Verarbeitung.....	519
28.5 Diagramme.....	527
28.6 Tabellen.....	533
28.7 Aufgaben.....	536
 IX. DATENBANK- UND NETZANWENDUNGEN	
 29 Datenbankzugriffe mit JDBC.....	 541
29.1 Voraussetzungen.....	541
29.2 Datenbank erstellen.....	543
29.3 Daten einfügen.....	545
29.4 Daten ändern und löschen.....	547
29.5 Daten abrufen.....	549
29.6 Daten in einer Tabelle anzeigen.....	550
29.7 Daten in einem Diagramm präsentieren.....	557
29.8 Aufgaben.....	561
 30 Netzwerkkommunikation.....	 565
30.1 Dateien aus dem Netz laden.....	565
30.2 Eine einfache Client/Server-Anwendung.....	567
30.3 HTTP.....	571
30.4 Simple Web Server.....	577
30.5 Das HTTP-Client-API.....	579
30.6 Aufgaben.....	582
 31 Fallbeispiel.....	 585
31.1 Die Anwendung.....	585
31.2 Variante 1: Lokale Anwendung.....	588
31.3 Variante 2: Client/Server-Anwendung.....	601
31.4 Aufgaben.....	614

X. DESIGNVORGABEN UND MODULARISIERUNG

32 Prinzipien objektorientierten Designs	615
32.1 Single-Responsibility-Prinzip	615
32.2 Open-Closed-Prinzip	618
32.3 Liskovsches Substitutionsprinzip	621
32.4 Interface-Segregation-Prinzip	624
32.5 Dependency-Inversion-Prinzip	627
32.6 Zusammenfassung	630
32.7 Aufgaben	631
33 Einführung in das Modulsystem	635
33.1 Motivation	635
33.2 Grundlagen	636
33.3 Abhängigkeiten und Zugriffsschutz	637
33.4 Transitive Abhängigkeiten	643
33.5 Abhängigkeit von JDK-Modulen und anderen Modulen	643
33.6 Trennung von Schnittstelle und Implementierung	647
33.7 Modularisierung und Services	650
33.8 Einbindung nicht-modularer Bibliotheken	653
33.9 Modulkategorien	656
33.10 Aufgaben	658
XI. ALLGEMEINES	
34 IntelliJ IDEA, Debugging und Testen mit JUnit	661
34.1 Hinweise zu IntelliJ IDEA	661
34.2 Debugging	666
34.3 Testen mit JUnit	668
Literaturhinweise	673
Sachwortverzeichnis	675