

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>15</b>
<b>1 Einleitung</b>	<b>17</b>
1.1 Java – definitiv kein kalter Kaffee! . . . . .	18
1.2 Java für Fortgeschrittene – Unser Konzept . . . . .	18
1.3 Was war ... . . . .	19
1.4 ... und was kommen wird . . . . .	20
1.5 Weitere Infos und Kontakt zu den Autoren . . . . .	22
1.6 Verwendete Schreibweisen . . . . .	22
<b>I Einige Vorbereitungen</b>	<b>23</b>
<b>2 Entwurfsmuster</b>	<b>25</b>
2.1 Was sind Entwurfsmuster? . . . . .	25
2.2 Das Observer-Pattern . . . . .	27
2.2.1 Zugrunde liegende Idee . . . . .	27
2.2.2 Das Objektmodell . . . . .	27
2.2.3 Beispiel-Realisierung . . . . .	28
2.2.3.1 Das Arbeiten mit nur einem Observer . . . . .	28
2.2.3.2 Das Arbeiten mit mehreren Observern . . . . .	31
2.2.4 Variationen des Pattern . . . . .	36
2.2.5 Zusammenfassung . . . . .	37
2.2.6 Übungsaufgaben . . . . .	37
2.3 Das Composite-Pattern . . . . .	38
2.3.1 Zugrunde liegende Idee . . . . .	38
2.3.2 Das Objektmodell . . . . .	40
2.3.3 Beispiel-Realisierung . . . . .	42
2.3.3.1 Summe zweier Funktionen . . . . .	42
2.3.3.2 Produkt zweier Funktionen . . . . .	43
2.3.4 Variationen des Pattern . . . . .	45
2.3.5 Zusammenfassung . . . . .	47
2.3.6 Übungsaufgaben . . . . .	48

<b>3</b>	<b>Einige wichtige Hilfsklassen</b>	<b>49</b>
3.1	Die Klasse <code>StringBuffer</code> . . . . .	49
3.1.1	Arbeiten mit <code>String</code> -Objekten . . . . .	49
3.1.2	Arbeiten mit <code>StringBuffer</code> -Objekten . . . . .	51
3.1.3	Übungsaufgaben . . . . .	54
3.2	Die Wrapper-Klassen (Hüll-Klassen) . . . . .	55
3.2.1	Arbeiten mit „eingepackten“ Daten . . . . .	55
3.2.2	Aufbau der Wrapper-Klassen . . . . .	56
3.2.3	Ein Anwendungsbeispiel . . . . .	59
3.2.4	Übungsaufgaben . . . . .	60
3.3	Die Klassen <code>BigInteger</code> und <code>BigDecimal</code> . . . . .	60
3.3.1	Arbeiten mit langen Ganzzahlen . . . . .	61
3.3.2	Aufbau der Klasse <code>BigInteger</code> . . . . .	63
3.3.3	Übungsaufgaben . . . . .	64
3.3.4	Arbeiten mit langen Gleitkommazahlen . . . . .	65
3.3.5	Aufbau der Klasse <code>BigDecimal</code> . . . . .	68
3.3.6	Viele Stellen von Nullstellen gefällig? . . . . .	70
3.3.7	Übungsaufgaben . . . . .	72
3.4	Die Klasse <code>DecimalFormat</code> . . . . .	73
3.4.1	Standard-Ausgaben in Java . . . . .	73
3.4.2	Arbeiten mit <code>Format</code> -Objekten . . . . .	74
3.4.3	Übungsaufgaben . . . . .	76
3.5	Die Klassen <code>Date</code> und <code>Calendar</code> . . . . .	77
3.5.1	Arbeiten mit „Zeitpunkten“ . . . . .	77
3.5.2	Auf die Plätze, fertig, los! . . . . .	78
3.5.3	Spezielle <code>Calendar</code> -Klassen . . . . .	79
3.5.4	Und noch mal: Zeitmessung . . . . .	81
3.5.5	Übungsaufgaben . . . . .	83
3.6	Die Klassen <code>SimpleDateFormat</code> und <code>DateFormat</code> . . . . .	83
3.6.1	Arbeiten mit <code>Format</code> -Objekten für Datum/Zeit-Angaben . . . . .	83
3.6.2	Übungsaufgaben . . . . .	88
3.7	Die <code>Collection</code> -Klassen . . . . .	88
3.7.1	„Sammlungen“ von Objekten – Der Aufbau des Interface <code>Collection</code> . . . . .	89
3.7.2	„Sammlungen“ durchgehen – Der Aufbau des Interface <code>Iterator</code> . . . . .	90
3.7.3	Mengen . . . . .	91
3.7.3.1	Das Interface <code>Set</code> . . . . .	91
3.7.3.2	Die Klasse <code>HashSet</code> . . . . .	92
3.7.3.3	Das Interface <code>SortedSet</code> . . . . .	93
3.7.3.4	Die Klasse <code>TreeSet</code> . . . . .	94

3.7.4	Listen . . . . .	96
3.7.4.1	Das Interface List . . . . .	96
3.7.4.2	Die Klassen ArrayList und LinkedList . . . . .	96
3.7.4.3	Suchen und Sortieren – Die Klassen Collections und Arrays . . . . .	98
3.7.5	Übungsaufgaben . . . . .	101
3.8	Die Klasse StringTokenizer . . . . .	101
3.8.1	Übungsaufgaben . . . . .	104
<b>4</b>	<b>Praxisbeispiele</b>	<b>105</b>
4.1	Rechnen mit rationalen Werten . . . . .	105
4.1.1	Variablen und Konstruktoren . . . . .	106
4.1.2	toString, equals und hashCode . . . . .	108
4.1.3	Die vier Grundrechenarten . . . . .	109
4.2	Wem die Stunde schlägt . . . . .	111
4.2.1	Designphase . . . . .	112
4.2.2	Modell und View . . . . .	114
4.2.3	Controller und Hauptprogramm . . . . .	115
4.2.4	Ausblick . . . . .	116
4.3	Die Türme von Hanoi . . . . .	117
4.3.1	Designphase . . . . .	118
4.3.2	Die Klasse Scheibe . . . . .	118
4.3.3	Die Klasse Stange . . . . .	120
4.3.4	Die Klasse Hanoi, erster Teil . . . . .	122
4.3.5	Der Algorithmus . . . . .	123
<b>II</b>	<b>Grafische Oberflächen in Java</b>	<b>127</b>
<b>5</b>	<b>Aufbau grafischer Oberflächen in Frames – von AWT nach Swing</b>	<b>131</b>
5.1	Grundsätzliches zum Aufbau grafischer Oberflächen . . . . .	131
5.2	Ein einfaches Beispiel mit dem AWT . . . . .	133
5.3	Let's swing now! . . . . .	135
5.4	Etwas „Fill-in“ gefällig? . . . . .	137
5.5	Die AWT- und Swing-Klassenbibliothek im Überblick . . . . .	139
5.6	Übungsaufgaben . . . . .	141
<b>6</b>	<b>Swing-Komponenten</b>	<b>143</b>
6.1	Die abstrakte Klasse Component . . . . .	143
6.2	Die Klasse Container . . . . .	144
6.3	Die abstrakte Klasse JComponent . . . . .	145
6.4	Layout-Manager, Farben und Schriften . . . . .	147
6.4.1	Die Klasse Color . . . . .	147
6.4.2	Die Klasse Font . . . . .	149

6.4.3	Layout-Manager	150
6.4.3.1	Die Klasse <code>FlowLayout</code>	151
6.4.3.2	Die Klasse <code>BorderLayout</code>	153
6.4.3.3	Die Klasse <code>GridLayout</code>	155
6.5	Einige Grundkomponenten	157
6.5.1	Die Klasse <code>JLabel</code>	157
6.5.2	Die abstrakte Klasse <code>AbstractButton</code>	159
6.5.3	Die Klasse <code>JButton</code>	161
6.5.4	Die Klasse <code>JToggleButton</code>	163
6.5.5	Die Klasse <code>JCheckBox</code>	165
6.5.6	Die Klassen <code>JRadioButton</code> und <code>ButtonGroup</code>	166
6.5.7	Die Klasse <code>JComboBox</code>	169
6.5.8	Die Klasse <code>JList</code>	171
6.5.9	Die abstrakte Klasse <code>JTextComponent</code>	174
6.5.10	Die Klassen <code>JTextField</code> und <code>JPasswordField</code>	175
6.5.11	Die Klasse <code>JTextArea</code>	177
6.5.12	Die Klasse <code>JScrollPane</code>	179
6.5.13	Die Klasse <code>JPanel</code>	182
6.6	Spezielle Container, Menüs und Toolbars	184
6.6.1	Die Klasse <code>JFrame</code>	184
6.6.2	Die Klasse <code>JWindow</code>	185
6.6.3	Die Klasse <code>JDialog</code>	185
6.6.4	Die Klasse <code>JMenuBar</code>	189
6.6.5	Die Klasse <code>JToolBar</code>	191
6.7	Übungsaufgaben	194
<b>7</b>	<b>Ereignisverarbeitung</b>	<b>197</b>
7.1	Zwei einfache Beispiele	198
7.1.1	Zufällige Grautöne als Hintergrund	198
7.1.2	Ein interaktiver Bilderrahmen	201
7.2	Programmiervarianten für die Ereignisverarbeitung	205
7.2.1	Innere Klasse als Listener-Klasse	205
7.2.2	Anonyme Klasse als Listener-Klasse	205
7.2.3	Container-Klasse als Listener-Klasse	206
7.2.4	Separate Klasse als Listener-Klasse	208
7.3	Event-Klassen und -Quellen	209
7.4	Listener-Interfaces und Adapter-Klassen	213
7.5	Listener-Registrierung bei den Event-Quellen	219
7.6	Auf die Plätze, fertig, los!	222
7.7	Übungsaufgaben	226

<b>8 Praxisbeispiele</b>	<b>231</b>
8.1 Wem die Stunde schlägt, Iteration 2 . . . . .	231
8.1.1 Eine Digitalanzeige . . . . .	231
8.1.2 Eine neue Steuerung . . . . .	233
8.1.3 Nicht aus dem Rahmen fallen! . . . . .	235
8.1.4 Zusammenfassung . . . . .	237
8.2 Body-Mass-Index . . . . .	237
8.2.1 Design und Layout . . . . .	237
8.2.2 Events und Anwendungslogik . . . . .	241
8.2.3 Das gesamte Programm im Überblick . . . . .	243
8.3 Wem die Stunde schlägt, Iteration 3 . . . . .	246
8.3.1 Design und Layout . . . . .	246
8.3.2 Wechsel des Look and feel . . . . .	249
<b>9 Einige Ergänzungen zu Swing-Komponenten</b>	<b>253</b>
9.1 Zeichnen in Swing-Komponenten . . . . .	253
9.1.1 Grafische Darstellung von Komponenten . . . . .	253
9.1.2 Das Grafik-Koordinatensystem . . . . .	254
9.1.3 Die abstrakte Klasse Graphics . . . . .	255
9.1.4 Ein einfaches Zeichenprogramm . . . . .	258
9.2 Noch mehr Swing gefällig? . . . . .	261
9.3 Übungsaufgaben . . . . .	262
<b>10 Applets</b>	<b>265</b>
10.1 Erstellen und Ausführen von Applets . . . . .	265
10.1.1 Vom Frame zum Applet am Beispiel . . . . .	265
10.1.2 Applet in HTML-Datei einbetten . . . . .	267
10.1.3 Applet über HTML-Datei ausführen . . . . .	269
10.2 Die Methoden der Klasse JApplet . . . . .	270
10.3 Zwei Beispiele . . . . .	273
10.3.1 Auf die Plätze, fertig, los! . . . . .	273
10.3.2 Punkte verbinden im Applet . . . . .	276
10.4 Details zur HTML-Einbettung . . . . .	277
10.4.1 Der Applet-Tag . . . . .	277
10.4.2 Die Methode showDocument . . . . .	280
10.5 Sicherheitseinschränkungen bei Applets . . . . .	282
10.6 Übungsaufgaben . . . . .	286

<b>11 Praxisbeispiele</b>	<b>289</b>
11.1 Wem die Stunde schlägt, Iteration 4 . . . . .	289
11.1.1 Erste Schritte . . . . .	289
11.1.2 Von Kreisen und Winkeln . . . . .	293
11.1.3 Die Methode setzeBreite . . . . .	295
11.1.4 Die Methode zeichneLinie . . . . .	296
11.1.5 Zusammenfassung . . . . .	297
11.2 Wem die Stunde schlägt, Iteration 5 . . . . .	298
11.2.1 Vorbereitungen . . . . .	299
11.2.2 Layout in der Klasse SetzeDarstellung . . . . .	300
11.2.3 Vom Layout zur Anwendungslogik . . . . .	302
11.3 Wem die Stunde schlägt, Iteration 6 . . . . .	303
11.3.1 Schritt 1: Auf den Schirm . . . . .	303
11.3.2 Schritt 2: Eine Frage der Einstellung . . . . .	304
11.3.3 Schritt 3: Alles hübsch verpackt . . . . .	307
<b>III Threads, Datenströme und Netzwerk-Anwendungen</b>	<b>311</b>
<b>12 Parallele Programmierung mit Threads</b>	<b>313</b>
12.1 Ein einfaches Beispiel . . . . .	313
12.2 Threads in Java . . . . .	315
12.2.1 Die Klasse Thread . . . . .	316
12.2.2 Das Interface Runnable . . . . .	320
12.2.3 Threads vorzeitig beenden . . . . .	322
12.3 Wissenswertes über Threads . . . . .	324
12.3.1 Lebenszyklus eines Threads . . . . .	324
12.3.2 Thread-Scheduling . . . . .	326
12.3.3 Dämon-Threads und Thread-Gruppen . . . . .	326
12.4 Thread-Synchronisation und -Kommunikation . . . . .	327
12.4.1 Das Leser/Schreiber-Problem . . . . .	328
12.4.2 Das Erzeuger/Verbraucher-Problem . . . . .	332
12.5 Threads in Frames und Applets . . . . .	339
12.5.1 Auf die Plätze, fertig, los! . . . . .	339
12.5.2 Spielereien . . . . .	343
12.6 Übungsaufgaben . . . . .	346

<b>13 Ein- und Ausgabe über Streams</b>	<b>349</b>
13.1 Grundsätzliches zu Streams in Java	350
13.2 Dateien und Verzeichnisse – Die Klasse File	350
13.3 Ein- und Ausgabe über Character-Streams	353
13.3.1 Einfache Reader- und Writer-Klassen	354
13.3.2 Gepufferte Reader- und Writer-Klassen	357
13.3.3 Die Klasse StreamTokenizer	359
13.3.4 Die Klasse PrintWriter	361
13.3.5 Was machen eigentlich die IOTools?	363
13.4 Ein- und Ausgabe über Byte-Streams	364
13.4.1 Einige InputStream- und OutputStream-Klassen	365
13.4.2 Die Serialisierung und Deserialisierung von Objekten	366
13.4.3 Die Klasse PrintStream	369
13.5 Einige abschließende Bemerkungen	369
13.6 Übungsaufgaben	370
<b>14 Client/Server-Programmierung in Netzwerken</b>	<b>373</b>
14.1 Wissenswertes über Netzwerk-Kommunikation	374
14.1.1 Protokolle	374
14.1.2 IP-Adressen	376
14.1.3 Ports und Sockets	377
14.2 Client/Server-Programmierung	378
14.2.1 Die Klassen ServerSocket und Socket	379
14.2.2 Ein einfacher Server	381
14.2.3 Ein einfacher Client	384
14.2.4 Ein Server für mehrere Clients	386
14.2.5 Ein Mehrzweck-Client	389
14.3 Wissenswertes über URLs	391
14.3.1 Client/Server-Kommunikation über URLs	392
14.3.2 Netzwerkverbindungen in Applets	393
14.4 Übungsaufgaben	394
<b>15 Praxisbeispiele</b>	<b>399</b>
15.1 Wem die Stunde schlägt, Iteration 7	399
15.2 Wem die Stunde schlägt, Iteration 8	401
15.2.1 Hätten wir nur <i>einen</i> Socket, ...	401
15.2.2 Die Klasse Zeitserver	403
15.2.3 Ein Testprogramm	404
15.3 Wem die Stunde schlägt, Iteration 9	407
15.3.1 Wenn's am schönsten ist, ...	407
15.3.2 Einige Vorbereitungen	407
15.3.3 Uhrenvergleich	409
15.3.4 Der Einstellungs-Dialog	411
15.3.5 Zusammenfassung	414

<b>IV Ausblick und Anhang</b>	<b>415</b>
<b>16 Blick über den Tellerrand</b>	<b>417</b>
16.1 Der Vorhang fällt	417
16.2 A fool with a tool ...	418
16.3 Alles umsonst?	419
16.4 Und fachlich?	420
16.5 Zu guter Letzt ...	422
<b>A Der Umgang mit der API-Spezifikation</b>	<b>423</b>
A.1 Der Aufbau der API-Spezifikation	423
A.2 Der praktische Einsatz der API-Spezifikation	424
<b>B Glossar</b>	<b>429</b>
<b>Literaturverzeichnis</b>	<b>435</b>
<b>Stichwortverzeichnis</b>	<b>437</b>