

Elisabeth Jung

Java 6
Das Übungsbuch
Band II



mitp

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie. Detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-8266-5956-0

1. Auflage 2008

Alle Rechte, auch die der Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Der Verlag übernimmt keine Gewähr für die Funktion einzelner Programme oder von Teilen derselben. Insbesondere übernimmt er keinerlei Haftung für eventuelle aus dem Gebrauch resultierende Folgeschäden.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Printed in Germany

© Copyright 2008 by REDLINE GMBH, Heidelberg

www.mitp.de

Lektorat: Sabine Schulz

Fach- und Sprachkorrektorat: Hubert Partl

Sprachkorrektorat: Sabine Janatschek

Satz: III-satz, Husby, www.drei-satz.de

Inhaltsverzeichnis

	Einleitung	19
I	Dateien und Streams	25
1.1	Dateien und Dateiverzeichnisse.....	25
	☆☆ Aufgabe 1.1: Die Konstruktoren und Methoden der Klasse File.....	26
	☆ Aufgabe 1.2: Auflisten von Einträgen aus Dateiverzeichnissen.....	27
	☆☆ Aufgabe 1.3: Die Klassen FileFilter und FilenameFilter.....	27
1.2	Die Definition und Klassifikation von Streams.....	28
	☆☆ Aufgabe 1.4: Operationen mit primitiven Datentypen.....	30
1.3	Die gemeinsamen Methoden zum Schreiben und Lesen der Oberklassen OutputStream und Writer bzw. InputStream und Reader.....	34
1.4	Die Klassen FileInputStream und FileReader bzw. FileOutputStream und FileWriter und der Zugriff auf Dateien.....	35
	☆ Aufgabe 1.5: Die Klassen FileOutputStream und FileInputStream.....	35
	☆ Aufgabe 1.6: Die Klassen FileWriter und FileReader.....	36
1.5	Der Zugriff auf Daten aus dem Arbeitsspeicher mithilfe der Klassen ByteArrayOutputStream und CharArrayWriter bzw. ByteArrayInputStream und CharArrayReader.....	36
	☆☆ Aufgabe 1.7: Die Klassen ByteArrayOutputStream und ByteArrayInputStream.....	37
	☆☆ Aufgabe 1.8: Die Klassen CharArrayWriter und CharArrayReader.....	38
1.6	Die Filterklassen FilterInputStream und FilterReader bzw. FilterOutputStream und FilterWriter.....	38
	☆☆☆ Aufgabe 1.9: Das Ketten von FilterOutputStreams mit FileOutputStreams.....	39

	☆☆	Aufgabe 1.10: Das Ketten von FilterOutputStreams mit der System.out-Instanz.	40
	☆☆	Aufgabe 1.11: Das Ketten von FilterWriter-Streams mit der System.out-Instanz.	41
	☆☆	Aufgabe 1.12: Das Ketten von FilterInputStreams mit FileInputStreams	42
	☆☆	Aufgabe 1.13: Das Ketten von FilterReader-Streams mit FileReader-Streams.	43
1.7		Das Puffern von Daten und die Klassen BufferedOutputStream und BufferedWriter bzw. BufferedInputStream und BufferedReader	43
	☆☆	Aufgabe 1.14: Das Ketten von BufferedInputStreams und BufferedOutputStreams mit FileInputStreams.	44
	☆☆	Aufgabe 1.15: Das Ketten von BufferedReader-Streams und BufferedWriter-Streams mit FileReader-Streams	45
	☆☆	Aufgabe 1.16: Die readLine()-Methode der Klasse BufferedReader und das Selektieren von Textzeilen	45
1.8		Die Schnittstellen DataOutput und DataInput und deren Methoden zum Schreiben und Lesen von primitiven Datentypen. ...	46
	☆☆	Aufgabe 1.17: Das Schreiben und Lesen von primitiven Datentypen	47
	☆☆	Aufgabe 1.18: Das Filtern von primitiven Datentypen.	48
1.9		Formatierte Ein-/Ausgaben	49
1.10		Standard Ein-/Ausgabe-Kanäle.	52
	☆☆	Aufgabe 1.19: Die Klasse Formatter und das Formatieren von Ausgaben.	53
	☆☆	Aufgabe 1.20: Die Klasse Scanner und das Suchen, Ersetzen und Zerlegen von Zeichenketten. ...	54
	☆	Aufgabe 1.21: Die System.out-Instanz der Klasse PrintStream	55
	☆	Aufgabe 1.22: Andere PrintStream-Instanzen.	56
	☆	Aufgabe 1.23: Die printf()-Methode	56
	☆	Aufgabe 1.24: Die Klasse PrintWriter.	56

	☆☆	Aufgabe 1.25: Die Klasse PrintWriter als Filterklasse einsetzen	57
	☆☆☆	Aufgabe 1.26: Die Klasse StringWriter	57
	☆	Aufgabe 1.27: Die System.in-Instanz der Klasse InputStream	58
	☆	Aufgabe 1.28: Das Umlenken des Standard-Eingabe-Kanals	59
	☆	Aufgabe 1.29: Das Einlesen von Tastatureingaben in ein Array	59
	☆	Aufgabe 1.30: Das Puffern von Tastatureingaben beim Einlesen	60
	☆☆	Aufgabe 1.31: Das Schreiben auf den Fehlerkanal mit der System.err-Instanz	60
	☆☆	Aufgabe 1.32: Einlesen von primitiven Datentypen mithilfe der Klasse Scanner	61
	☆	Aufgabe 1.33: Einlesen von Tastatureingaben mithilfe der Klasse Scanner	62
	☆	Aufgabe 1.34: Die printf()-Methode der Klasse Console	63
	☆	Aufgabe 1.35: Die readLine()-Methode der Klasse Console	63
I.11		ObjectStreams und die Serialisierung von Objekten	63
I.12		Die Versionsverwaltung von Klassen	64
	☆	Aufgabe 1.36: Das Speichern und Einlesen eines Objektes	65
	☆☆	Aufgabe 1.37: Das Speichern und Einlesen von Objekten, deren Klassen Referenzfelder definieren	65
	☆☆	Aufgabe 1.38: Die serialVersionUID einer Klasse	66
I.13		Die Klasse RandomAccessFile	67
	☆	Aufgabe 1.39: Wahlfreier Zugriff auf eine Datei	68
I.14		Lösungen	69
		Lösung I.1	69
		Lösung I.2	71
		Lösung I.3	73
		Lösung I.4	75
		Lösung I.5	77
		Lösung I.6	78

	Lösung I.7	79
	Lösung I.8	81
	Lösung I.9	83
	Lösung I.10	84
	Lösung I.11	86
	Lösung I.12	87
	Lösung I.13	89
	Lösung I.14	90
	Lösung I.15	91
	Lösung I.16	93
	Lösung I.17	94
	Lösung I.18	96
	Lösung I.19	98
	Lösung I.20	99
	Lösung I.21	101
	Lösung I.22	102
	Lösung I.23	103
	Lösung I.24	103
	Lösung I.25	104
	Lösung I.26	106
	Lösung I.27	109
	Lösung I.28	110
	Lösung I.29	110
	Lösung I.30	111
	Lösung I.31	112
	Lösung I.32	114
	Lösung I.33	116
	Lösung I.34	116
	Lösung I.35	117
	Lösung I.36	118
	Lösung I.37	120
	Lösung I.38	122
	Lösung I.39	126
2	Multithreading	129
2.1	Programme, Prozesse und Threads	129
	☆ Aufgabe 2.1: Thread-Definition als Unterklasse der Klasse Thread	132

	☆	Aufgabe 2.2: Thread-Definition mithilfe der Runnable-Schnittstelle	132
	☆☆	Aufgabe 2.3: Thread-Ausführungsintervalle	133
	☆	Aufgabe 2.4: Der main-Thread und von ihm gestartete Arbeits-Threads	133
2.2		Vordergrund- und Hintergrund-Threads und das Setzen von Prioritäten, die Zustände eines Threads, Threads beenden und unterbrechen und die Klasse ThreadGroup	134
	☆	Aufgabe 2.5: Threadprioritäten setzen, den Status von Threads abfragen und Thread-StackTraces ausgeben	137
	☆☆	Aufgabe 2.6: Das Beenden von Threads über einen vom Benutzer definierten Interrupt-Flag	137
	☆☆	Aufgabe 2.7: Das Beenden von Threads über einen vom System definierten Interrupt-Flag	138
	☆	Aufgabe 2.8: Vom System eingerichtete Threadgruppen und ihre Threads	139
	☆	Aufgabe 2.9: Vom Benutzer definierte Threadgruppen	139
2.3		Die Klassen java.util.Timer und java.util.TimerTask	140
	☆☆	Aufgabe 2.10: Timer- und TimerTask-Instanzen	140
2.4		Die Threads von graphischen Benutzeroberflächen.	141
	☆	Aufgabe 2.11: Die Threads von graphischen Benutzeroberflächen anzeigen.	143
	☆	Aufgabe 2.12: Der AWT-EventQueue-Thread und die Ereignisbehandlung	143
	☆	Aufgabe 2.13: Die Methode invokeAndWait()	144
	☆	Aufgabe 2.14: Die Methode invokeLater()	145
	☆	Aufgabe 2.15: Ein Timer, der periodisch ein ActionEvent-Objekt sendet.	145
	☆	Aufgabe 2.16: Das Erstellen einer GUI mit dem Ereignisbehandlungs-Thread (EDT)	146
	☆	Aufgabe 2.17: Das Editieren von Swing-Komponenten und die Methode invokeLater()	146
	☆	Aufgabe 2.18: Das Editieren von Swing-Komponenten und die Methode invokeAndWait()	147

2.5	Wiederholungsaufgaben	147
	☆☆☆ Aufgabe 2.19: Wiederholungsaufgabe	147
	☆☆ Aufgabe 2.20: Wiederholungsaufgabe	148
	☆☆ Aufgabe 2.21: Wiederholungsaufgabe	149
	☆☆ Aufgabe 2.22: Wiederholungsaufgabe	149
2.6	Die Klasse SwingWorker von Java 6	150
	☆ Aufgabe 2.23: Das Zusammenspiel der Methoden doInBackground(), done() und get() der Klasse SwingWorker	152
	☆ Aufgabe 2.24: Das Editieren von Swing- Komponenten mit den Methoden der Klasse SwingWorker	152
	☆ Aufgabe 2.25: Das Zusammenspiel der Methoden publish() und process() der Klasse SwingWorker	153
	☆☆ Aufgabe 2.26: Wiederholungsaufgabe	154
	☆☆ Aufgabe 2.27: Wiederholungsaufgabe	154
2.7	Synchronized-Blöcke und -Methoden	156
	☆ Aufgabe 2.28: Threadkommunikation über die Werte von Klassenfeldern	158
	☆ Aufgabe 2.29: Threadkommunikation über PipedWriter- und PipedReader-Streams	159
	☆☆ Aufgabe 2.30: Threadkommunikation über Piped- InputStream- und PipedOutputStream-Instanzen	160
	☆ Aufgabe 2.31: Threadkommunikation und Synchronisation über die Methode join() der Thread-Klasse	161
	☆ Aufgabe 2.32: »Klassen-Sperre« mit einem synchronized-Block setzen	161
	☆ Aufgabe 2.33: »Klassen-Sperre« mit einer synchronized-Methode setzen	161
	☆☆ Aufgabe 2.34: Eine beliebige static-Referenz als Argument in einem synchronized-Block angeben	162
	☆☆☆ Aufgabe 2.35: »Objekt-Sperre« mit synchronized-Methoden setzen	163
	☆☆ Aufgabe 2.36: »Objekt-Sperre« mit einem synchronized(this)-Block setzen	164
	☆☆ Aufgabe 2.37: Das Deadlock-Problem	165

	☆☆☆ Aufgabe 2.38: »Klassen-« und »Objekt-Sperren«	165
2.8	Die Methoden wait(), notify() und notifyAll()	167
	☆☆ Aufgabe 2.39: Die Methoden wait() und notify()	168
	☆☆ Aufgabe 2.40: Die Methoden wait() und notifyAll()	170
	☆☆☆ Aufgabe 2.41: Wiederholungsaufgabe	170
2.9	Lösungen	173
	Lösung 2.1	173
	Lösung 2.2	173
	Lösung 2.3	174
	Lösung 2.4	175
	Lösung 2.5	177
	Lösung 2.6	179
	Lösung 2.7	181
	Lösung 2.8	183
	Lösung 2.9	185
	Lösung 2.10	187
	Lösung 2.11	190
	Lösung 2.12	192
	Lösung 2.13	194
	Lösung 2.14	195
	Lösung 2.15	196
	Lösung 2.16	198
	Lösung 2.17	199
	Lösung 2.18	201
	Lösung 2.19	203
	Lösung 2.20	207
	Lösung 2.21	210
	Lösung 2.22	212
	Lösung 2.23	215
	Lösung 2.24	216
	Lösung 2.25	218
	Lösung 2.26	220
	Lösung 2.27	223
	Lösung 2.28	226
	Lösung 2.29	227
	Lösung 2.30	229
	Lösung 2.31	232
	Lösung 2.32	234
	Lösung 2.33	235

	Lösung 2.34	236
	Lösung 2.35	238
	Lösung 2.36	240
	Lösung 2.37	243
	Lösung 2.38	246
	Lösung 2.39	251
	Lösung 2.40	254
	Lösung 2.41	256
3	Einführung in Java-Applets	265
3.1	Applets erzeugen und starten	265
	☆ Aufgabe 3.1: Die Methoden init(), start(), stop() und destroy() der Applet-Klassen	267
	☆ Aufgabe 3.2: Eine Graphik im Applet zeichnen	267
	☆ Aufgabe 3.3: Parameterübergabe an Applets mithilfe des <param>-Tags der HTML-Datei	267
	☆ Aufgabe 3.4: Die Methode getParameterInfo() der Klasse Applet	268
	☆ Aufgabe 3.5: AWT-Komponenten zu einem Applet hinzufügen	268
	☆ Aufgabe 3.6: Swing-Komponenten zu einem Applet hinzufügen	269
	☆ Aufgabe 3.7: Mit einem Graphics2D-Objekt auf ein Applet zeichnen	270
	☆☆ Aufgabe 3.8: Die Klassen GeneralPath und AffineTransform	270
3.2	Threads und Eventbehandlung	271
	☆ Aufgabe 3.9: Die Threads von Applets anzeigen	271
	☆ Aufgabe 3.10: Das Erstellen einer GUI mit dem Ereignisbehandlungs-Thread (EDT)	272
	☆☆ Aufgabe 3.11: Die Eventbehandlung in Applets und die Methoden paint() und update() von AWT-Komponenten	272
3.3	Lösungen	274
	Lösung 3.1	274
	Lösung 3.2	275
	Lösung 3.3	276
	Lösung 3.4	277
	Lösung 3.5	279

	Lösung 3.6.....	280
	Lösung 3.7.....	282
	Lösung 3.8.....	284
	Lösung 3.9.....	286
	Lösung 3.10.....	288
	Lösung 3.11.....	289
4	Multimedia für Applets und Applikationen	295
4.1	Die Einbindung von Images und Sounds	295
	☆ Aufgabe 4.1: Bild- und Sounddateien im Applet laden.....	301
	☆ Aufgabe 4.2: Die Methoden getCodeBase() und getDocumentBase() der Klasse Applet.....	302
	☆ Aufgabe 4.3: Pakete in Applets.....	302
	☆ Aufgabe 4.4: Sounds in einer Applikation abspielen... ..	303
	☆☆ Aufgabe 4.5: Die Klassen BufferedImage und TexturePaint	303
	☆☆☆ Aufgabe 4.6: Bild- und Sounddateien in einem Applet laden – eine Übersicht	304
	☆☆☆ Aufgabe 4.7: Bild- und Sounddateien in einem Fenster laden – eine Übersicht	307
	☆☆ Aufgabe 4.8: Bild- und Sounddateien für Applets aus einem JAR-Archiv laden	307
	☆☆ Aufgabe 4.9: Bild- und Sounddateien für Frames aus einem JAR-Archiv laden.....	308
4.2	Das Überwachen des Ladevorgangs.....	309
	☆ Aufgabe 4.10: Die Methoden prepareImage() und checkImage() der Klasse Toolkit.....	310
	☆ Aufgabe 4.11: Die Klasse MediaTracker.....	310
	☆☆ Aufgabe 4.12: Das Laden von Images mit dem MediaTracker überwachen	311
	☆☆ Aufgabe 4.13: Die Methode imageUpdate() von Component überschreiben	312
	☆☆ Aufgabe 4.14: Das Laden von Images mit dem ImageObserver überwachen.....	312
	☆☆ Aufgabe 4.15: Den ImageObserver mittels einer anonymen Klasse implementieren.....	313

	☆☆	Aufgabe 4.16: Ein benutzerdefinierter ImageObserver und die Methode setImageObserver() der Klasse ImageIcon	313
	☆☆	Aufgabe 4.17: Der Standard-ImageObserver und die Methode setImageObserver() der Klasse ImageIcon	314
4.3		Die RootPane-Container von Applets.	315
	☆☆☆	Aufgabe 4.18: Die RootPane-Container für JApplet-Instanzen ermitteln und setzen	315
	☆☆☆	Aufgabe 4.19: Wiederholungsaufgabe	316
4.4		Lösungen	317
		Lösung 4.1	317
		Lösung 4.2	320
		Lösung 4.3	321
		Lösung 4.4	322
		Lösung 4.5	324
		Lösung 4.6	326
		Lösung 4.7	332
		Lösung 4.8	338
		Lösung 4.9	340
		Lösung 4.10	342
		Lösung 4.11	344
		Lösung 4.12	346
		Lösung 4.13	348
		Lösung 4.14	349
		Lösung 4.15	352
		Lösung 4.16	355
		Lösung 4.17	358
		Lösung 4.18	362
		Lösung 4.19	364
5		Threads und Animation	367
5.1		Applets und die Definition von Arbeits-Threads.	367
	☆☆	Aufgabe 5.1: Ein Applet, welches die Runnable-Schnittstelle implementiert.	367
	☆☆	Aufgabe 5.2: Das Beenden von Threads in der stop()-Methode eines Applets.	368
	☆☆☆	Aufgabe 5.3: SwingWorker- und Timer-Tasks für Applets	369
	☆☆☆	Aufgabe 5.4: Wiederholungsaufgabe	371

	☆☆☆	Aufgabe 5.5: Laden von mehreren Sounddateien parallel zum Abspielen von Sounds	371
	☆☆☆	Aufgabe 5.6: Wiederholungsaufgabe	372
	☆☆☆	Aufgabe 5.7: Laden von mehreren Imagedateien parallel zum Zeichnen von Images	373
	☆☆☆	Aufgabe 5.8: Wiederholungsaufgabe	374
5.2		Synchronisation und Kommunikation von Threads in Applets	375
	☆☆☆	Aufgabe 5.9: Mehrere Threads, die über eine Monitor-Klasse synchronisiert werden, zum Laden von Sounddateien starten	375
	☆☆	Aufgabe 5.10: Eine einfache Animation	376
	☆☆	Aufgabe 5.11: Wiederholungsaufgabe	377
	☆☆	Aufgabe 5.12: Wiederholungsaufgabe	378
	☆☆☆	Aufgabe 5.13: Animation mit Timer und Image-Komponenten	379
	☆☆☆	Aufgabe 5.14: Wiederholungsaufgabe	380
	☆☆	Aufgabe 5.15: Animation mit Timer und ImageIcon-Komponenten.	381
	☆☆	Aufgabe 5.16: Wiederholungsaufgabe	381
	☆☆☆	Aufgabe 5.17: Animation mit mehreren Threads	381
	☆☆	Aufgabe 5.18: Applet mit »Klassen-Sperre« für Threads	382
	☆☆	Aufgabe 5.19: Wiederholungsaufgabe	384
	☆☆☆	Aufgabe 5.20: Applet mit »Objekt-Sperre« für Threads	384
	☆☆	Aufgabe 5.21: Wiederholungsaufgabe	386
	☆☆	Aufgabe 5.22: Applet mit SwingWorkerTasks.	387
	☆☆☆	Aufgabe 5.23: Threadkommunikation mit Warten auf das Threadende.	388
	☆☆☆	Aufgabe 5.24: Threadkommunikation mit Rückruf nach Ausführung der Aufträge	389
5.3		Lösungen	390
		Lösung 5.1	390
		Lösung 5.2	392
		Lösung 5.3	395
		Lösung 5.4	402
		Lösung 5.5	407
		Lösung 5.6	410
		Lösung 5.7	412

	Lösung 5.8	415
	Lösung 5.9	419
	Lösung 5.10	422
	Lösung 5.11	425
	Lösung 5.12	428
	Lösung 5.13	431
	Lösung 5.14	435
	Lösung 5.15	438
	Lösung 5.16	441
	Lösung 5.17	443
	Lösung 5.18	447
	Lösung 5.19	450
	Lösung 5.20	453
	Lösung 5.21	457
	Lösung 5.22	459
	Lösung 5.23	461
	Lösung 5.24	466
6	Java-Applets und Webbrowser	471
6.1	Zugriffsrechte und Sicherheitsbestimmungen	471
6.2	Kommunikation zwischen Applets sowie zwischen Applets und Webbrowser	475
	☆☆☆ Aufgabe 6.1: Der target-Parameter der Methode showDocument()	476
	☆☆☆ Aufgabe 6.2: Zugriffsrechte für Applets vergeben	477
	☆☆☆ Aufgabe 6.3: Wiederholungsaufgabe	478
	☆☆ Aufgabe 6.4: Kommunikation zwischen Applet und Browser	479
	☆☆ Aufgabe 6.5: Benutzung von Image-Maps in der Kommunikation zwischen Applet und Browser	480
	☆☆ Aufgabe 6.6: Kommunikation zwischen Applets in derselben Webseite	481
	☆☆ Aufgabe 6.7: Benutzung von Image-Maps in der Kommunikation zwischen Applets	482
	☆☆ Aufgabe 6.8: Redirects im Webbrowser	482
	☆☆☆ Aufgabe 6.9: Die Threads von Applets synchronisieren	483
6.3	Lösungen	485
	Lösung 6.1	485

	Lösung 6.2.	486
	Lösung 6.3.	489
	Lösung 6.4.	494
	Lösung 6.5.	497
	Lösung 6.6.	500
	Lösung 6.7.	502
	Lösung 6.8.	506
	Lösung 6.9.	509
7	Netzwerkprogrammierung	519
7.1	Einführung in die Netzwerkprogrammierung	519
7.2	URL-Verbindungen	520
	☆☆ Aufgabe 7.1: Die Klasse JEditorPane und ein einfacher Webbrowser	522
	☆ Aufgabe 7.2: Die Methoden der Klasse URL	522
	☆☆ Aufgabe 7.3: Die Methoden der Klassen NetworkInterface, InetAddress und InterfaceAddress	524
	☆ Aufgabe 7.4: Die openStream()-Methode der URL-Klasse	525
	☆ Aufgabe 7.5: Wiederholungsaufgabe	525
	☆ Aufgabe 7.6: Die URLConnection-Klasse und ihre Methoden	526
	☆☆ Aufgabe 7.7: E-Mails über eine URLConnection-Instanz verschicken	527
7.3	Client-Server-Kommunikation über TCP mit Java-Sockets	528
	☆☆☆ Aufgabe 7.8: Kommunikation über TCP mit Java-Sockets.	529
	☆☆☆ Aufgabe 7.9: Wiederholungsaufgabe.	531
7.4	Client-Server-Kommunikation über UDP mit Java-Sockets	532
	☆☆☆ Aufgabe 7.10: Kommunikation über UDP mit Java-Sockets.	533
	☆☆☆ Aufgabe 7.11: Wiederholungsaufgabe	534
7.5	Webbasierte Client-Server-Kommunikation	535
	☆☆ Aufgabe 7.12: Ein einfacher Web-Client, der über TCP-Socket mit einem Webserver kommuniziert	536
	☆☆ Aufgabe 7.13: Ein einfacher Webserver	537

	☆☆	Aufgabe 7.14: Der HttpServer von Java 6	538
	☆☆	Aufgabe 7.15: Wiederholungsaufgabe	540
7.6		Sequentielle und parallele Server	542
	☆☆	Aufgabe 7.16: Ein sequentieller Echo-Server	542
	☆☆☆	Aufgabe 7.17: Ein paralleler Echo-Server mit dynamischer Parallelität.	543
	☆☆☆	Aufgabe 7.18: Ein paralleler Echo-Server mit statischer Parallelität.	544
	☆☆☆	Aufgabe 7.19: Wiederholungsaufgabe	545
	☆☆☆	Aufgabe 7.20: Wiederholungsaufgabe	546
7.7		Updates mit Java 6 zum Thema Netzwerkprogrammierung.	546
	☆☆☆	Aufgabe 7.21: Die Klasse CookieHandler.	547
	☆☆☆	Aufgabe 7.22: Die Klassen CookieManager, CookieStore und CookiePolicy	548
7.8		Lösungen	549
		Lösung 7.1	549
		Lösung 7.2	551
		Lösung 7.3	554
		Lösung 7.4	559
		Lösung 7.5	561
		Lösung 7.6	562
		Lösung 7.7	564
		Lösung 7.8	567
		Lösung 7.9	572
		Lösung 7.10	578
		Lösung 7.11	582
		Lösung 7.12	589
		Lösung 7.13	591
		Lösung 7.14	596
		Lösung 7.15	599
		Lösung 7.16	610
		Lösung 7.17	615
		Lösung 7.18	619
		Lösung 7.19	623
		Lösung 7.20	630
		Lösung 7.21	634
		Lösung 7.22	638
		Stichwortverzeichnis	641