

# Auf einen Blick

1	Neues in Java 7 .....	41
2	Threads und nebenläufige Programmierung .....	65
3	Datenstrukturen und Algorithmen .....	175
4	Raum und Zeit .....	305
5	Dateien, Verzeichnisse und Dateizugriffe .....	357
6	Datenströme .....	427
7	Die eXtensible Markup Language (XML) .....	531
8	Dateiformate .....	617
9	Grafische Oberflächen mit Swing .....	645
10	Grafikprogrammierung .....	865
11	Netzwerkprogrammierung .....	955
12	Verteilte Programmierung mit RMI .....	1035
13	RESTful und SOAP Web-Services .....	1057
14	JavaServer Pages und Servlets .....	1079
15	Applets .....	1123
16	Datenbankmanagement mit JDBC .....	1139
17	Technologien für die Infrastruktur .....	1207
18	Reflection und Annotationen .....	1227
19	Dynamische Übersetzung und Skriptsprachen .....	1289
20	Logging und Monitoring .....	1315
21	Java Native Interface (JNI) .....	1337
22	Sicherheitskonzepte .....	1355
23	Dienstprogramme für die Java-Umgebung .....	1381

# Inhalt

Vorwort .....	35
---------------	----

## 1 Neues in Java 7

---

<b>1.1 Sprachänderungen .....</b>	<b>41</b>
1.1.1 Zahlen im Binärsystem schreiben .....	41
1.1.2 Unterstriche bei Ganzzahlen .....	41
1.1.3 switch (String) .....	42
1.1.4 Zusammenfassen gleicher catch-Blöcke mit dem multi-catch .....	43
1.1.5 Präzisiertes rethrows .....	47
1.1.6 Automatisches Ressourcen-Management (try mit Ressourcen) .....	51
1.1.7 try mit Ressourcen .....	51
1.1.8 Ausnahmen vom close() bleiben bestehen .....	53
1.1.9 Die Schnittstelle AutoCloseable .....	53
1.1.10 Mehrere Ressourcen nutzen .....	55
1.1.11 Unterdrückte Ausnahmen * .....	56
<b>1.2 JVM-Änderungen .....</b>	<b>59</b>
1.2.1 invokedynamic .....	60
<b>1.3 Neue Pakete und Klassen .....</b>	<b>63</b>
1.3.1 Weitere Änderungen .....	63

## 2 Threads und nebenläufige Programmierung

---

<b>2.1 Threads erzeugen .....</b>	<b>65</b>
2.1.1 Threads über die Schnittstelle Runnable implementieren .....	65
2.1.2 Thread mit Runnable starten .....	66
2.1.3 Die Klasse Thread erweitern .....	68
<b>2.2 Thread-Eigenschaften und -Zustände .....</b>	<b>71</b>
2.2.1 Der Name eines Threads .....	71
2.2.2 Wer bin ich? .....	72
2.2.3 Die Zustände eines Threads * .....	72

2.2.4	Schläfer gesucht .....	73
2.2.5	Mit yield() auf Rechenzeit verzichten .....	75
2.2.6	Der Thread als Dämon .....	76
2.2.7	Das Ende eines Threads .....	78
2.2.8	Einen Thread höflich mit Interrupt beenden .....	79
2.2.9	UncaughtExceptionHandler für unbehandelte Ausnahmen .....	81
2.2.10	Der stop() von außen und die Rettung mit ThreadDeath * .....	82
2.2.11	Ein Rendezvous mit join() * .....	84
2.2.12	Arbeit niederlegen und wieder aufnehmen * .....	87
2.2.13	Priorität * .....	87
<b>2.3</b>	<b>Der Ausführer (Executor) kommt .....</b>	<b>88</b>
2.3.1	Die Schnittstelle Executor .....	89
2.3.2	Die Thread-Pools .....	91
2.3.3	Threads mit Rückgabe über Callable .....	92
2.3.4	Mehrere Callable abarbeiten .....	96
2.3.5	ScheduledExecutorService für wiederholende Ausgaben und Zeitsteuerungen nutzen .....	97
<b>2.4</b>	<b>Synchronisation über kritische Abschnitte .....</b>	<b>98</b>
2.4.1	Gemeinsam genutzte Daten .....	98
2.4.2	Probleme beim gemeinsamen Zugriff und kritische Abschnitte .....	98
2.4.3	Punkte parallel initialisieren .....	100
2.4.4	i++ sieht atomar aus, ist es aber nicht * .....	102
2.4.5	Kritische Abschnitte schützen .....	103
2.4.6	Kritische Abschnitte mit ReentrantLock schützen .....	106
2.4.7	Synchronisieren mit synchronized .....	113
2.4.8	Synchronized-Methoden der Klasse StringBuffer * .....	114
2.4.9	Mit synchronized synchronisierte Blöcke .....	115
2.4.10	Dann machen wir doch gleich alles synchronisiert! .....	117
2.4.11	Lock-Freigabe im Fall von Exceptions .....	118
2.4.12	Deadlocks .....	119
2.4.13	Mit synchronized nachträglich synchronisieren * .....	121
2.4.14	Monitore sind reentrant – gut für die Geschwindigkeit * .....	122
2.4.15	Synchronisierte Methodenaufrufe zusammenfassen * .....	123
<b>2.5</b>	<b>Synchronisation über Warten und Benachrichtigen .....</b>	<b>124</b>
2.5.1	Die Schnittstelle Condition .....	125
2.5.2	It's Disco-Time * .....	129

2.5.3	Warten mit wait() und Aufwecken mit notify() *	135
2.5.4	Falls der Lock fehlt: IllegalMonitorStateException *	136
<b>2.6</b>	<b>Datensynchronisation durch besondere Concurrency-Klassen *</b>	<b>138</b>
2.6.1	Semaphor	138
2.6.2	Barrier und Austausch	143
2.6.3	Stop and go mit Exchanger	145
<b>2.7</b>	<b>Atomare Operationen und frische Werte mit volatile *</b>	<b>145</b>
2.7.1	Der Modifizierer volatile bei Objekt-/Klassenvariablen	146
2.7.2	Das Paket java.util.concurrent.atomic	148
<b>2.8</b>	<b>Teile und herrsche mit Fork und Join *</b>	<b>149</b>
2.8.1	Algorithmendesign per »teile und herrsche«	149
2.8.2	Paralleles Lösen von D&C-Algorithmen	151
2.8.3	Fork und Join	152
<b>2.9</b>	<b>Mit dem Thread verbundene Variablen *</b>	<b>156</b>
2.9.1	ThreadLocal	156
2.9.2	InheritableThreadLocal	159
2.9.3	ThreadLocalRandom als Zufallszahlengenerator	160
2.9.4	ThreadLocal bei der Performance-Optimierung	161
<b>2.10</b>	<b>Threads in einer Thread-Gruppe *</b>	<b>162</b>
2.10.1	Aktive Threads in der Umgebung	162
2.10.2	Etwas über die aktuelle Thread-Gruppe herausfinden	163
2.10.3	Threads in einer Thread-Gruppe anlegen	166
2.10.4	Methoden von Thread und ThreadGroup im Vergleich	169
<b>2.11</b>	<b>Zeitgesteuerte Abläufe</b>	<b>170</b>
2.11.1	Die Typen Timer und TimerTask	171
2.11.2	Job-Scheduler Quartz	172
<b>2.12</b>	<b>Einen Abbruch der virtuellen Maschine erkennen</b>	<b>173</b>
<b>2.13</b>	<b>Zum Weiterlesen</b>	<b>174</b>

## 3 Datenstrukturen und Algorithmen

---

<b>3.1</b>	<b>Datenstrukturen und die Collection-API</b>	<b>175</b>
3.1.1	Designprinzip mit Schnittstellen, abstrakten und konkreten Klassen	176
3.1.2	Die Basis-Schnittstellen Collection und Map	176
3.1.3	Die Utility-Klassen Collections und Arrays	177

3.1.4	Das erste Programm mit Container-Klassen .....	177
3.1.5	Die Schnittstelle Collection und Kernkonzepte .....	178
3.1.6	Schnittstellen, die Collection erweitern, und Map .....	182
3.1.7	Konkrete Container-Klassen .....	185
3.1.8	Generische Datentypen in der Collection-API .....	187
3.1.9	Die Schnittstelle Iterable und das erweiterte for .....	187
<b>3.2</b>	<b>Listen</b> .....	<b>188</b>
3.2.1	Erstes Listen-Beispiel .....	189
3.2.2	Auswahlkriterium ArrayList oder LinkedList .....	190
3.2.3	Die Schnittstelle List .....	191
3.2.4	ArrayList .....	197
3.2.5	LinkedList .....	200
3.2.6	Der Feld-Adapter Arrays.asList() .....	201
3.2.7	ListIterator * .....	202
3.2.8	toArray() von Collection verstehen – die Gefahr einer Falle erkennen .....	204
3.2.9	Primitive Elemente in Datenstrukturen verwalten .....	207
<b>3.3</b>	<b>Mengen (Sets)</b> .....	<b>208</b>
3.3.1	Ein erstes Mengen-Beispiel .....	209
3.3.2	Methoden der Schnittstelle Set .....	212
3.3.3	HashSet .....	213
3.3.4	TreeSet – die sortierte Menge .....	214
3.3.5	Die Schnittstellen NavigableSet und SortedSet .....	216
3.3.6	LinkedHashSet .....	218
<b>3.4</b>	<b>Stack (Kellerspeicher, Stapel)</b> .....	<b>218</b>
3.4.1	Die Methoden von Stack .....	219
<b>3.5</b>	<b>Queues (Schlangen) und Deques</b> .....	<b>220</b>
3.5.1	Queue-Klassen .....	221
3.5.2	Deque-Klassen .....	222
3.5.3	Blockierende Queues und Prioritätswarteschlangen .....	223
3.5.4	PriorityQueue .....	223
<b>3.6</b>	<b>Assoziative Speicher</b> .....	<b>228</b>
3.6.1	Die Klassen HashMap und TreeMap .....	228
3.6.2	Einfügen und Abfragen der Datenstruktur .....	231
3.6.3	Über die Bedeutung von equals() und hashCode() .....	234
3.6.4	Eigene Objekte hashen .....	235
3.6.5	IdentityHashMap .....	237
3.6.6	Das Problem von veränderten Elementen .....	237

3.6.7	Aufzählungen und Ansichten des Assoziativspeichers .....	237
3.6.8	Die Arbeitsweise einer Hash-Tabelle * .....	241
<b>3.7</b>	<b>Die Properties-Klasse .....</b>	<b>243</b>
3.7.1	Properties setzen und lesen .....	243
3.7.2	Properties verketteten .....	244
3.7.3	Hierarchische Eigenschaften .....	245
3.7.4	Eigenschaften auf der Konsole ausgeben * .....	245
3.7.5	Properties laden und speichern .....	246
3.7.6	Klassenbeziehungen: Properties und Hashtable * .....	248
<b>3.8</b>	<b>Mit einem Iterator durch die Daten wandern .....</b>	<b>249</b>
3.8.1	Die Schnittstelle Iterator .....	249
3.8.2	Der Iterator kann (eventuell auch) löschen .....	251
3.8.3	<i>Einen Zufallszahleniterator schreiben .....</i>	<i>252</i>
3.8.4	Iteratoren von Sammlungen, das erweiterte for und Iterable .....	254
3.8.5	Fail-Fast-Iterator und die ConcurrentModificationException .....	258
3.8.6	Die Schnittstelle Enumerator * .....	260
<b>3.9</b>	<b>Algorithmen in Collections .....</b>	<b>261</b>
3.9.1	Die Bedeutung von Ordnung mit Comparator und Comparable .....	263
3.9.2	Sortieren .....	264
3.9.3	Den größten und kleinsten Wert einer Collection finden .....	267
3.9.4	Nicht-änderbare Datenstrukturen, immutable oder nur-lesen? .....	270
3.9.5	Null Object Pattern und leere Sammlungen/Iteratoren zurückgeben .....	274
3.9.6	Mit der Halbierungssuche nach Elementen fahnden .....	278
3.9.7	Ersetzen, Kopieren, Füllen, Umdrehen, Rotieren * .....	279
3.9.8	Listen durchwürfeln * .....	281
3.9.9	Häufigkeit eines Elements * .....	282
3.9.10	Singletons * .....	282
3.9.11	nCopies() * .....	283
<b>3.10</b>	<b>Spezielle thread-sichere Datenstrukturen .....</b>	<b>284</b>
3.10.1	Wait-free-Algorithmen .....	284
3.10.2	Nebenläufiger Assoziativspeicher und die Schnittstelle ConcurrentMap .....	284
3.10.3	ConcurrentLinkedQueue .....	285
3.10.4	CopyOnWriteArrayList und CopyOnWriteArraySet .....	285
3.10.5	Wrapper zur Synchronisation .....	286
3.10.6	Blockierende Warteschlangen .....	287

3.10.7	ArrayBlockingQueue und LinkedBlockingQueue .....	287
3.10.8	PriorityBlockingQueue .....	290
<b>3.11</b>	<b>Google Guava (aka Google Collections Library) .....</b>	<b>294</b>
3.11.1	Beispiel Multi-Set und Multi-Map .....	294
3.11.2	Datenstrukturen aus Guava .....	295
3.11.3	Utility-Klassen von Guava .....	297
3.11.4	Prädikate .....	298
3.11.5	Transformationen .....	299
<b>3.12</b>	<b>Die Klasse BitSet für Bitmengen * .....</b>	<b>299</b>
3.12.1	Ein BitSet anlegen, füllen und erfragen .....	300
3.12.2	Mengenorientierte Operationen .....	301
3.12.3	Methodenübersicht .....	302
3.12.4	Primzahlen in einem BitSet verwalten .....	304
<b>3.13</b>	<b>Zum Weiterlesen .....</b>	<b>304</b>

## 4 Raum und Zeit

---

<b>4.1</b>	<b>Weltzeit * .....</b>	<b>305</b>
<b>4.2</b>	<b>Wichtige Datum-Klassen im Überblick .....</b>	<b>306</b>
4.2.1	Der 1.1.1970 .....	307
4.2.2	System.currentTimeMillis() .....	307
4.2.3	Einfache Zeitumrechnungen durch TimeUnit .....	307
<b>4.3</b>	<b>Sprachen der Länder .....</b>	<b>308</b>
4.3.1	Sprachen und Regionen über Locale-Objekte .....	309
<b>4.4</b>	<b>Internationalisierung und Lokalisierung .....</b>	<b>312</b>
4.4.1	ResourceBundle-Objekte und Ressource-Dateien .....	313
4.4.2	Ressource-Dateien zur Lokalisierung .....	314
4.4.3	Die Klasse ResourceBundle .....	314
4.4.4	Ladestrategie für ResourceBundle-Objekte .....	316
<b>4.5</b>	<b>Die Klasse Date .....</b>	<b>317</b>
4.5.1	Objekte erzeugen und Methoden nutzen .....	317
4.5.2	Date-Objekte sind nicht immutable .....	319
<b>4.6</b>	<b>Calendar und GregorianCalendar .....</b>	<b>320</b>
4.6.1	Die abstrakte Klasse Calendar .....	320
4.6.2	Der gregorianische Kalender .....	322

4.6.3	Calendar nach Date und Millisekunden fragen .....	325
4.6.4	Ostertage * .....	326
4.6.5	Abfragen und Setzen von Datumselementen über Feldbezeichner .....	328
4.6.6	Wie viele Tage hat der Monat, oder wie viele Monate hat ein Jahr? * .....	333
4.6.7	Wann beginnt die Woche und wann die erste Woche im Jahr? * .....	335
<b>4.7</b>	<b>Zeitzone in Java *</b> .....	<b>336</b>
4.7.1	Zeitzone durch die Klasse TimeZone repräsentieren .....	337
4.7.2	SimpleTimeZone .....	337
4.7.3	Methoden von TimeZone .....	339
<b>4.8</b>	<b>Zeitdauer und der XML-Datentyp Duration *</b> .....	<b>340</b>
4.8.1	DatatypeFactory als Fabrik .....	341
4.8.2	Die Duration-Klasse und ihre Methoden .....	342
<b>4.9</b>	<b>Formatieren und Parsen von Datumsangaben</b> .....	<b>345</b>
4.9.1	Ausgaben mit printf() .....	345
4.9.2	Mit DateFormat und SimpleDateFormat formatieren .....	346
4.9.3	Parsen von Datumswerten .....	352
<b>4.10</b>	<b>Zum Weiterlesen</b> .....	<b>355</b>

## 5 Dateien, Verzeichnisse und Dateizugriffe

---

<b>5.1</b>	<b>Datei und Verzeichnis</b> .....	<b>357</b>
5.1.1	Dateien und Verzeichnisse mit der Klasse File .....	358
5.1.2	Verzeichnis oder Datei? Existiert es? .....	361
5.1.3	Verzeichnis- und Dateieigenschaften/-attribute .....	362
5.1.4	Umbenennen und Verzeichnisse anlegen .....	365
5.1.5	Verzeichnisse auflisten und Dateien filtern .....	365
5.1.6	Dateien berühren, neue Dateien anlegen, temporäre Dateien .....	369
5.1.7	Dateien und Verzeichnisse löschen .....	370
5.1.8	Verzeichnisse nach Dateien iterativ durchsuchen * .....	372
5.1.9	Wurzelverzeichnis, Laufwerksnamen, Plattenspeicher * .....	373
5.1.10	URL-, URI- und Path-Objekte aus einem File-Objekt ableiten * .....	376
5.1.11	Mit Locking Dateien sperren * .....	376
5.1.12	Sicherheitsprüfung * .....	377
5.1.13	Zugriff auf SMB-Server mit jCIFS * .....	378



<b>5.2</b>	<b>Dateien mit wahlfreiem Zugriff</b> .....	378
5.2.1	Ein RandomAccessFile zum Lesen und Schreiben öffnen .....	379
5.2.2	Aus dem RandomAccessFile lesen .....	380
5.2.3	Schreiben mit RandomAccessFile .....	383
5.2.4	Die Länge des RandomAccessFile .....	384
5.2.5	Hin und her in der Datei .....	384
<b>5.3</b>	<b>Dateisysteme unter NIO.2</b> .....	385
5.3.1	FileSystem und Path .....	386
5.3.2	Die Utility-Klasse Files .....	392
5.3.3	Dateien kopieren und verschieben .....	395
5.3.4	Dateiattribute * .....	397
5.3.5	Neue Dateien, Verzeichnisse, symbolische Verknüpfungen anlegen und löschen .....	406
5.3.6	MIME-Typen testen * .....	408
5.3.7	Verzeichnislistings (DirectoryStream) und Filter * .....	410
5.3.8	Rekursive Abläufe des Verzeichnisbaums (FileVisitor) * .....	412
5.3.9	Dateisysteme und Dateisystemattribute * .....	416
5.3.10	Verzeichnisse im Dateisystem überwachen * .....	420
<b>5.4</b>	<b>Wahlfreier Zugriff mit SeekableByteChannel und ByteBuffer *</b> .....	422
5.4.1	SeekableByteChannel .....	422
5.4.2	ByteBuffer .....	423
5.4.3	Beispiel mit Path + SeekableByteChannel + ByteBuffer .....	423
5.4.4	FileChannel .....	424
<b>5.5</b>	<b>Zum Weiterlesen</b> .....	426

## 6 Datenströme

---

<b>6.1</b>	<b>Stream-Klassen und Reader/Writer am Beispiel von Dateien</b> .....	427
6.1.1	Mit dem FileWriter Texte in Dateien schreiben .....	428
6.1.2	Zeichen mit der Klasse FileReader lesen .....	430
6.1.3	Kopieren mit FileOutputStream und FileInputStream .....	431
6.1.4	Das FileDescriptor-Objekt * .....	435
6.1.5	Datenströme über Files mit NIO.2 beziehen .....	436
<b>6.2</b>	<b>Basisklassen für die Ein-/Ausgabe</b> .....	438
6.2.1	Die abstrakten Basisklassen .....	438
6.2.2	Übersicht über Ein-/Ausgabeklassen .....	439

6.2.3	Die abstrakte Basisklasse OutputStream .....	441
6.2.4	Die Schnittstellen Closeable, AutoCloseable und Flushable .....	443
6.2.5	Ein Datenschlucker * .....	444
6.2.6	Die abstrakte Basisklasse InputStream .....	445
6.2.7	Ressourcen aus dem Klassenpfad und aus Jar-Archiven laden .....	446
6.2.8	Ströme mit SequenceInputStream zusammensetzen * .....	447
6.2.9	Die abstrakte Basisklasse Writer .....	449
6.2.10	Die Schnittstelle Appendable * .....	451
6.2.11	Die abstrakte Basisklasse Reader .....	451
<b>6.3</b>	<b>Formatierte Textausgaben .....</b>	<b>454</b>
6.3.1	Die Klassen PrintWriter und PrintStream .....	455
6.3.2	System.out, System.err und System.in .....	460
<b>6.4</b>	<b>Schreiben und Lesen aus Strings und Byte-Feldern .....</b>	<b>462</b>
6.4.1	Mit dem StringWriter ein String-Objekt füllen .....	463
6.4.2	CharArrayWriter .....	464
6.4.3	StringReader und CharArrayReader .....	465
6.4.4	Mit ByteArrayOutputStream in ein Byte-Feld schreiben .....	467
6.4.5	Mit ByteArrayInputStream aus einem Byte-Feld lesen .....	467
<b>6.5</b>	<b>Datenströme filtern und verketteten .....</b>	<b>468</b>
6.5.1	Streams als Filter verketteten (verschachteln) .....	469
6.5.2	Gepufferte Ausgaben mit BufferedWriter und BufferedOutputStream ....	469
6.5.3	Gepufferte Eingaben mit BufferedReader/BufferedInputStream .....	471
6.5.4	LineNumberReader zählt automatisch Zeilen mit * .....	474
6.5.5	Daten mit der Klasse PushbackReader zurücklegen * .....	474
6.5.6	DataOutputStream/DataInputStream * .....	478
6.5.7	Basisklassen für Filter * .....	478
6.5.8	Die Basisklasse FilterWriter * .....	479
6.5.9	Ein LowerCaseWriter * .....	480
6.5.10	Eingaben mit der Klasse FilterReader filtern * .....	482
6.5.11	Anwendungen für FilterReader und FilterWriter * .....	483
<b>6.6</b>	<b>Vermittler zwischen Byte-Streams und Unicode-Strömen .....</b>	<b>491</b>
6.6.1	Datenkonvertierung durch den OutputStreamWriter .....	491
6.6.2	Automatische Konvertierungen mit dem InputStreamReader .....	492
<b>6.7</b>	<b>Kommunikation zwischen Threads mit Pipes * .....</b>	<b>493</b>
6.7.1	PipedOutputStream und PipedInputStream .....	494
6.7.2	PipedWriter und PipedReader .....	496

<b>6.8</b>	<b>Prüfsummen</b> .....	498
6.8.1	Die Schnittstelle Checksum .....	498
6.8.2	Die Klasse CRC32 .....	499
6.8.3	Die Adler32-Klasse .....	501
<b>6.9</b>	<b>Persistente Objekte und Serialisierung</b> .....	502
6.9.1	Objekte mit der Standard-Serialisierung speichern und lesen .....	503
6.9.2	Zwei einfache Anwendungen der Serialisierung * .....	506
6.9.3	Die Schnittstelle Serializable .....	507
6.9.4	Nicht serialisierbare Attribute aussparen .....	509
6.9.5	Das Abspeichern selbst in die Hand nehmen .....	511
6.9.6	Tiefe Objektkopien * .....	515
6.9.7	Versionenverwaltung und die SUID .....	517
6.9.8	Wie die ArrayList serialisiert * .....	519
6.9.9	Probleme mit der Serialisierung .....	520
<b>6.10</b>	<b>Alternative Datenaustauschformate</b> .....	522
6.10.1	Serialisieren in XML-Dateien .....	522
6.10.2	XML-Serialisierung von JavaBeans mit JavaBeans Persistence * .....	522
6.10.3	Die Open-Source-Bibliothek XStream * .....	525
6.10.4	Binäre Serialisierung mit Google Protocol Buffers * .....	526
<b>6.11</b>	<b>Tokenizer *</b> .....	526
6.11.1	StreamTokenizer .....	526
<b>6.12</b>	<b>Zum Weiterlesen</b> .....	529

## **7 Die eXtensible Markup Language (XML)**

---

<b>7.1</b>	<b>Auszeichnungssprachen</b> .....	531
7.1.1	Die Standard Generalized Markup Language (SGML) .....	532
7.1.2	Extensible Markup Language (XML) .....	532
<b>7.2</b>	<b>Eigenschaften von XML-Dokumenten</b> .....	533
7.2.1	Elemente und Attribute .....	533
7.2.2	Beschreibungssprache für den Aufbau von XML-Dokumenten .....	535
7.2.3	Schema – eine Alternative zu DTD .....	539
7.2.4	Namensraum (Namespace) .....	542
7.2.5	XML-Applikationen * .....	543

<b>7.3</b>	<b>Die Java-APIs für XML</b> .....	544
7.3.1	Das Document Object Model (DOM) .....	545
7.3.2	Simple API for XML Parsing (SAX) .....	545
7.3.3	Pull-API StAX .....	545
7.3.4	Java Document Object Model (JDOM) .....	545
7.3.5	JAXP als Java-Schnittstelle zu XML .....	546
7.3.6	DOM-Bäume einlesen mit JAXP * .....	547
<b>7.4</b>	<b>Java Architecture for XML Binding (JAXB)</b> .....	548
7.4.1	Bean für JAXB aufbauen .....	548
7.4.2	JAXBContext und die Marshaller .....	549
7.4.3	Ganze Objektgraphen schreiben und lesen .....	550
7.4.4	Validierung .....	553
7.4.5	Weitere JAXB-Annotationen * .....	557
7.4.6	Beans aus XML-Schema-Datei generieren .....	565
<b>7.5</b>	<b>Serielle Verarbeitung mit StAX</b> .....	570
7.5.1	Unterschiede der Verarbeitungsmodelle .....	571
7.5.2	XML-Dateien mit dem Cursor-Verfahren lesen .....	572
7.5.3	XML-Dateien mit dem Iterator-Verfahren verarbeiten * .....	576
7.5.4	Mit Filtern arbeiten * .....	578
7.5.5	XML-Dokumente schreiben .....	579
<b>7.6</b>	<b>Serielle Verarbeitung von XML mit SAX *</b> .....	582
7.6.1	Schnittstellen von SAX .....	583
7.6.2	SAX-Parser erzeugen .....	584
7.6.3	Operationen der Schnittstelle ContentHandler .....	584
7.6.4	ErrorHandler und EntityResolver .....	587
<b>7.7</b>	<b>XML-Dateien mit JDOM verarbeiten</b> .....	588
7.7.1	JDOM beziehen .....	588
7.7.2	Paketübersicht * .....	588
7.7.3	Die Document-Klasse .....	590
7.7.4	Eingaben aus der Datei lesen .....	591
7.7.5	Das Dokument im XML-Format ausgeben .....	592
7.7.6	Der Dokumenttyp * .....	593
7.7.7	Elemente .....	594
7.7.8	Zugriff auf Elementinhalte .....	597
7.7.9	Liste mit Unterelementen erzeugen * .....	599
7.7.10	Neue Elemente einfügen und ändern .....	600

7.7.11	Attributinhalt lesen und ändern .....	603
7.7.12	XPath .....	606
<b>7.8</b>	<b>Transformationen mit XSLT *</b> .....	<b>610</b>
7.8.1	Templates und XPath als Kernelemente von XSLT .....	610
7.8.2	Umwandlung von XML-Dateien mit JDOM und JAXP .....	613
<b>7.9</b>	<b>XML-Schema-Validierung *</b> .....	<b>614</b>
7.9.1	SchemaFactory und Schema .....	614
7.9.2	Validator .....	614
7.9.3	Validierung unterschiedlicher Datenquellen durchführen .....	615
<b>7.10</b>	<b>Zum Weiterlesen</b> .....	<b>615</b>

## 8 Dateiformate

---

<b>8.1</b>	<b>Einfache Dateiformate für strukturierte Daten</b> .....	<b>618</b>
8.1.1	Property-Dateien .....	618
8.1.2	CSV-Dateien .....	618
8.1.3	JSON-Serialisierung mit Jackson .....	619
<b>8.2</b>	<b>Dokumentenformate</b> .....	<b>620</b>
8.2.1	(X)HTML .....	620
8.2.2	PDF-Dokumente .....	622
8.2.3	Microsoft Office-Dokumente .....	622
8.2.4	OASIS Open Document Format .....	624
<b>8.3</b>	<b>Datenkompression *</b> .....	<b>624</b>
8.3.1	Java-Unterstützung beim Komprimieren .....	625
8.3.2	Daten packen und entpacken .....	626
8.3.3	Datenströme komprimieren .....	627
8.3.4	Zip-Archive .....	632
8.3.5	Jar-Archive .....	640
<b>8.4</b>	<b>Bild-Formate</b> .....	<b>640</b>
<b>8.5</b>	<b>Audio-Dateien</b> .....	<b>640</b>
8.5.1	Die Arbeit mit AudioClip .....	640
8.5.2	MIDI-Dateien abspielen .....	641

## 9 Grafische Oberflächen mit Swing

---

<b>9.1</b>	<b>Fenster zur Welt</b> .....	645
9.1.1	Swing-Fenster mit <code>javax.swing.JFrame</code> darstellen .....	645
9.1.2	Fenster schließbar machen – <code>setDefaultCloseOperation()</code> .....	647
9.1.3	Sichtbarkeit des Fensters .....	648
9.1.4	Größe und Position des Fensters verändern .....	648
9.1.5	Fenster- und Dialog-Dekoration, Transparenz * .....	649
9.1.6	Die Klasse <code>Toolkit</code> * .....	650
9.1.7	Dynamisches Layout während einer Größenänderung * .....	651
9.1.8	Zum Vergleich: AWT-Fenster darstellen * .....	651
<b>9.2</b>	<b>Beschriftungen (<code>JLabel</code>)</b> .....	653
9.2.1	Mehrzeiliger Text, HTML in der Darstellung .....	656
<b>9.3</b>	<b>Icon und <code>ImageIcon</code> für Bilder auf Swing-Komponenten</b> .....	657
9.3.1	Die Klasse <code>ImageIcon</code> .....	657
9.3.2	Die Schnittstelle <code>Icon</code> und eigene Icons zeichnen * .....	659
<b>9.4</b>	<b>Es tut sich was – Ereignisse beim AWT</b> .....	661
9.4.1	Die Ereignisquellen und Horcher ( <code>Listener</code> ) von Swing .....	662
9.4.2	<code>Listener</code> implementieren .....	663
9.4.3	<code>Listener</code> bei dem Ereignisauslöser anmelden/abmelden .....	666
9.4.4	Adapterklassen nutzen .....	668
9.4.5	Innere Mitgliedsklassen und innere anonyme Klassen .....	670
9.4.6	Aufrufen der <code>Listener</code> im AWT-Event-Thread .....	672
9.4.7	Ereignisse, etwas genauer betrachtet * .....	673
<b>9.5</b>	<b>Schaltflächen</b> .....	676
9.5.1	Normale Schaltflächen ( <code>JButton</code> ) .....	676
9.5.2	Der aufmerksame <code>ActionListener</code> .....	678
9.5.3	Schaltflächen-Ereignisse vom Typ <code>ActionEvent</code> .....	679
9.5.4	Basisklasse <code>AbstractButton</code> .....	680
9.5.5	Wechselknopf ( <code>JToggleButton</code> ) .....	682
<b>9.6</b>	<b>Textkomponenten</b> .....	683
9.6.1	Text in einer Eingabezeile .....	684
9.6.2	Die Oberklasse der Text-Komponenten ( <code>JTextComponent</code> ) .....	685
9.6.3	Geschützte Eingaben ( <code>JPasswordField</code> ) .....	686
9.6.4	Validierende Eingabefelder ( <code>JFormattedTextField</code> ) .....	687
9.6.5	Einfache mehrzeilige Textfelder ( <code>JTextArea</code> ) .....	688
9.6.6	Editor-Klasse ( <code>JEditorPane</code> ) * .....	692

<b>9.7 Swing Action *</b> .....	694
<b>9.8 JComponent und Component als Basis aller Komponenten</b> .....	697
9.8.1 Hinzufügen von Komponenten .....	697
9.8.2 Tooltips (Kurzhinweise) .....	698
9.8.3 Rahmen (Border) * .....	699
9.8.4 Fokus und Navigation * .....	701
9.8.5 Ereignisse jeder Komponente * .....	702
9.8.6 Die Größe und Position einer Komponente * .....	706
9.8.7 Komponenten-Ereignisse * .....	707
9.8.8 UI-Delegate – der wahre Zeichner * .....	708
9.8.9 Undurchsichtige (opake) Komponente * .....	711
9.8.10 Properties und Listener für Änderungen * .....	711
<b>9.9 Container</b> .....	711
9.9.1 Standardcontainer (JPanel) .....	712
9.9.2 Bereich mit automatischen Rollbalken (JScrollPane) .....	713
9.9.3 Reiter (JTabbedPane) .....	714
9.9.4 Teilungskomponente (JSplitPane) .....	715
<b>9.10 Alles Auslegungssache: die Layoutmanager</b> .....	716
9.10.1 Übersicht über Layoutmanager .....	717
9.10.2 Zuweisen eines Layoutmanagers .....	717
9.10.3 Im Fluss mit FlowLayout .....	718
9.10.4 BorderLayout .....	720
9.10.5 Mit BorderLayout in alle Himmelsrichtungen .....	721
9.10.6 Rasteranordnung mit GridLayout .....	724
9.10.7 Der GridBagLayoutmanager * .....	725
9.10.8 Null-Layout * .....	731
9.10.9 Weitere Layoutmanager .....	732
<b>9.11 Rollbalken und Schieberegler</b> .....	733
9.11.1 Schieberegler (JSlider) .....	733
9.11.2 Rollbalken (JScrollBar) * .....	734
<b>9.12 Kontrollfelder, Optionsfelder, Kontrollfeldgruppen</b> .....	739
9.12.1 Kontrollfelder (JCheckBox) .....	740
9.12.2 ItemSelectable, ItemListener und das ItemEvent .....	741
9.12.3 Sich gegenseitig ausschließende Optionen (JRadioButton) .....	743
<b>9.13 Fortschritte bei Operationen überwachen *</b> .....	745
9.13.1 Fortschrittsbalken (JProgressBar) .....	745
9.13.2 Dialog mit Fortschrittsanzeige (ProgressMonitor) .....	748

<b>9.14 Menüs und Symbolleisten</b> .....	748
9.14.1 Die Menüleisten und die Einträge .....	748
9.14.2 Menüeinträge definieren .....	750
9.14.3 Einträge durch Action-Objekte beschreiben .....	751
9.14.4 Mit der Tastatur: Mnemonics und Shortcut .....	752
9.14.5 Der Tastatur-Shortcut (Accelerator) .....	753
9.14.6 Tastenkürzel (Mnemonics) .....	755
9.14.7 Symbolleisten alias Toolbars .....	756
9.14.8 Popup-Menüs .....	759
9.14.9 System-Tray nutzen * .....	764
<b>9.15 Das Model-View-Controller-Konzept</b> .....	765
<b>9.16 Auswahlmenüs, Listen und Spinner</b> .....	767
9.16.1 Listen (JList) .....	767
9.16.2 Auswahlmenü (JComboBox) .....	775
9.16.3 Drehfeld (JSpinner) * .....	787
9.16.4 Datumsauswahl * .....	789
<b>9.17 Tabellen (JTable)</b> .....	789
9.17.1 Ein eigenes Tabellen-Modell .....	791
9.17.2 Basisklasse für eigene Modelle (AbstractTableModel) .....	792
9.17.3 Ein vorgefertigtes Standard-Modell (DefaultTableModel) .....	796
9.17.4 Ein eigener Renderer für Tabellen .....	798
9.17.5 Zell-Editoren .....	801
9.17.6 Größe und Umrandung der Zellen * .....	803
9.17.7 Spalteninformationen * .....	803
9.17.8 Tabellenkopf von Swing-Tabellen * .....	804
9.17.9 Selektionen einer Tabelle * .....	805
9.17.10 Automatisches Sortieren und Filtern mit RowSorter * .....	806
<b>9.18 Bäume (JTree)</b> .....	809
9.18.1 JTree und sein TreeModel und TreeNode .....	809
9.18.2 Selektionen bemerken .....	811
9.18.3 Das TreeModel von JTree * .....	811
<b>9.19 JRootPane und JDesktopPane *</b> .....	815
9.19.1 Wurzelkomponente der Top-Level-Komponenten (JRootPane) .....	815
9.19.2 JDesktopPane und die Kinder von JInternalFrame .....	815
9.19.3 JLayeredPane .....	817
<b>9.20 Dialoge und Window-Objekte</b> .....	818
9.20.1 JWindow und JDialog .....	818



9.20.2	Modal oder nicht-modal? .....	819
9.20.3	Standarddialoge mit JOptionPane .....	820
9.20.4	Der Dateiauswahldialog .....	823
9.20.5	Der Farbauswahldialog JColorChooser * .....	827
<b>9.21</b>	<b>Flexibles Java-Look-and-Feel .....</b>	<b>830</b>
9.21.1	Look and Feel global setzen .....	830
9.21.2	UIManager .....	830
9.21.3	Die Windows-Optik mit JGoodies Looks verbessern * .....	833
<b>9.22</b>	<b>Swing-Komponenten neu erstellen oder verändern * .....</b>	<b>833</b>
9.22.1	Überlagerungen mit dem Swing-Komponenten-Dekorator JLayer .....	834
<b>9.23</b>	<b>Die Zwischenablage (Clipboard) .....</b>	<b>836</b>
9.23.1	Clipboard-Objekte .....	836
9.23.2	Mit Transferable auf den Inhalt zugreifen .....	837
9.23.3	DataFlavor ist das Format der Daten in der Zwischenablage .....	838
9.23.4	Einfügungen in der Zwischenablage erkennen .....	840
9.23.5	Drag & Drop .....	841
<b>9.24</b>	<b>Undo durchführen * .....</b>	<b>842</b>
<b>9.25</b>	<b>AWT, Swing und die Threads .....</b>	<b>844</b>
9.25.1	Ereignisschlange (EventQueue) und AWT-Event-Thread .....	844
9.25.2	Swing ist nicht thread-sicher .....	845
9.25.3	invokeLater() und invokeAndWait() .....	848
9.25.4	SwingWorker .....	851
9.25.5	Eigene Ereignisse in die Queue setzen * .....	853
9.25.6	Auf alle Ereignisse hören * .....	854
<b>9.26</b>	<b>Barrierefreiheit mit der Java Accessibility API .....</b>	<b>854</b>
<b>9.27</b>	<b>Zeitliches Ausführen mit dem javax.swing.Timer .....</b>	<b>855</b>
<b>9.28</b>	<b>Die Zusatzkomponentenbibliothek SwingX .....</b>	<b>856</b>
9.28.1	Im Angebot: Erweiterte und neue Swing-Komponenten .....	857
9.28.2	Überblick über erweiterte Standard-Swing-Klassen .....	857
9.28.3	Neue Swing-Klassen .....	858
9.28.4	Weitere SwingX-Klassen .....	859
9.28.5	SwingX-Installation .....	860
<b>9.29</b>	<b>Alternativen zu programmierten Oberflächen, AWT und Swing * .....</b>	<b>860</b>
9.29.1	Deklarative Beschreibungen der Oberfläche: Swing JavaBuilder, Swixml .....	860
9.29.2	SWT (Standard Widget Toolkit) .....	862
<b>9.30</b>	<b>Zum Weiterlesen .....</b>	<b>864</b>

## 10 Grafikprogrammierung

---

<b>10.1</b>	<b>Grundlegendes zum Zeichnen</b> .....	865
10.1.1	Die paint()-Methode für das AWT-Frame .....	865
10.1.2	Zeichnen von Inhalten auf ein JFrame .....	867
10.1.3	Auffordern zum Neuzeichnen mit repaint() .....	869
10.1.4	Java 2D-API .....	869
<b>10.2</b>	<b>Einfache Zeichenmethoden</b> .....	870
10.2.1	Linien .....	870
10.2.2	Rechtecke .....	871
10.2.3	Ovale und Kreisbögen .....	872
10.2.4	Polygone und Polylines .....	874
<b>10.3</b>	<b>Zeichenketten schreiben und Fonts</b> .....	876
10.3.1	Zeichenfolgen schreiben .....	877
10.3.2	Die Font-Klasse .....	878
10.3.3	Einen neuen Font aus einem gegebenen Font ableiten .....	880
10.3.4	Zeichensätze des Systems ermitteln * .....	880
10.3.5	Neue TrueType-Fonts in Java nutzen .....	881
10.3.6	Font-Metadaten durch FontMetrics * .....	882
<b>10.4</b>	<b>Geometrische Objekte</b> .....	886
10.4.1	Die Schnittstelle Shape .....	888
10.4.2	Kreisförmiges .....	889
10.4.3	Kurviges * .....	890
10.4.4	Area und die konstruktive Flächengeometrie * .....	890
10.4.5	Pfade * .....	891
10.4.6	Punkt in einer Form, Schnitt von Linien, Abstand Punkt/Linie * .....	895
<b>10.5</b>	<b>Das Innere und Äußere einer Form</b> .....	896
10.5.1	Farben und die Paint-Schnittstelle .....	896
10.5.2	Farben mit der Klasse Color .....	897
10.5.3	Die Farben des Systems über SystemColor * .....	903
10.5.4	Composite und Xor * .....	907
10.5.5	Dicke und Art der Linien von Formen bestimmen über Stroke * .....	908
<b>10.6</b>	<b>Bilder</b> .....	913
10.6.1	Eine Übersicht über die Bilder-Bibliotheken .....	914
10.6.2	Bilder mit ImageIO lesen .....	915
10.6.3	Ein Bild zeichnen .....	917
10.6.4	Programm-Icon/Fenster-Icon setzen .....	921

10.6.5	Splash-Screen *	922
10.6.6	Bilder im Speicher erzeugen *	922
10.6.7	Pixel für Pixel auslesen und schreiben *	925
10.6.8	Bilder skalieren *	927
10.6.9	Schreiben mit ImageIO	929
10.6.10	Asynchrones Laden mit getImage() und dem MediaTracker *	934
<b>10.7</b>	<b>Weitere Eigenschaften von Graphics *</b>	<b>935</b>
10.7.1	Eine Kopie von Graphics erstellen	935
10.7.2	Koordinatensystem verschieben	936
10.7.3	Beschnitt (Clipping)	937
10.7.4	Zeichenhinweise durch RenderingHints	941
10.7.5	Transformationen mit einem AffineTransform-Objekt	941
<b>10.8</b>	<b>Drucken *</b>	<b>944</b>
10.8.1	Drucken der Inhalte	944
10.8.2	Bekannte Drucker	946
<b>10.9</b>	<b>Benutzerinteraktionen automatisieren, Robot und Screenshots *</b>	<b>947</b>
10.9.1	Der Roboter	947
10.9.2	Automatisch in die Tasten hauen	948
10.9.3	Automatisierte Mausoperationen	949
10.9.4	Methoden zur Zeitsteuerung	949
10.9.5	Bildschirmabzüge (Screenshots)	950
10.9.6	Funktionsweise und Beschränkungen	952
10.9.7	MouseInfo und PointerInfo	952
<b>10.10</b>	<b>Zum Weiterlesen</b>	<b>953</b>

## 11 Netzwerkprogrammierung

---

<b>11.1</b>	<b>Grundlegende Begriffe</b>	<b>955</b>
<b>11.2</b>	<b>URI und URL</b>	<b>957</b>
11.2.1	Die Klasse URI	957
11.2.2	Die Klasse URL	958
11.2.3	Informationen über eine URL *	961
11.2.4	Der Zugriff auf die Daten über die Klasse URL	963
<b>11.3</b>	<b>Die Klasse URLConnection *</b>	<b>965</b>
11.3.1	Methoden und Anwendung von URLConnection	965
11.3.2	Protokoll- und Content-Handler	967

11.3.3	Im Detail: vom URL zur URLConnection .....	968
11.3.4	Der Protokoll-Handler für Jar-Dateien .....	969
11.3.5	Basic Authentication und Proxy-Authentifizierung .....	971
<b>11.4</b>	<b>Mit GET und POST Daten übergeben *</b> .....	<b>973</b>
11.4.1	Kodieren der Parameter für Serverprogramme .....	974
11.4.2	Eine Suchmaschine mit GET-Request ansprechen .....	975
11.4.3	POST-Request absenden .....	976
<b>11.5</b>	<b>Host- und IP-Adressen</b> .....	<b>977</b>
11.5.1	Lebt der Rechner? .....	980
11.5.2	IP-Adresse des lokalen Hosts .....	980
11.5.3	Das Netz ist klasse .....	981
11.5.4	NetworkInterface .....	981
<b>11.6</b>	<b>Mit dem Socket zum Server</b> .....	<b>983</b>
11.6.1	Das Netzwerk ist der Computer .....	983
11.6.2	Sockets .....	983
11.6.3	Eine Verbindung zum Server aufbauen .....	984
11.6.4	Server unter Spannung: die Ströme .....	985
11.6.5	Die Verbindung wieder abbauen .....	986
11.6.6	Informationen über den Socket * .....	986
11.6.7	Reine Verbindungsdaten über SocketAddress * .....	988
<b>11.7</b>	<b>Client-Server-Kommunikation</b> .....	<b>989</b>
11.7.1	Warten auf Verbindungen .....	990
11.7.2	Ein Multiplikationsserver .....	991
11.7.3	Blockierendes Lesen .....	995
11.7.4	Von außen erreichbar sein * .....	996
<b>11.8</b>	<b>Apache HttpComponents und Commons Net *</b> .....	<b>997</b>
11.8.1	HttpComponents .....	997
11.8.2	Apache Commons Net .....	998
<b>11.9</b>	<b>Arbeitsweise eines Webservers *</b> .....	<b>998</b>
11.9.1	Das Hypertext Transfer Protocol (HTTP) .....	998
11.9.2	Anfragen an den Server .....	999
11.9.3	Die Antworten vom Server .....	1002
11.9.4	Webserver mit com.sun.net.httpserver.HttpServer .....	1006
<b>11.10</b>	<b>Verbindungen durch einen Proxy-Server *</b> .....	<b>1007</b>
11.10.1	System-Properties .....	1008
11.10.2	Verbindungen durch die Proxy-API .....	1008

<b>11.11 Datagram-Sockets *</b> .....	1010
11.11.1 Die Klasse DatagramSocket .....	1012
11.11.2 Datagramme und die Klasse DatagramPacket .....	1013
11.11.3 Auf ein hereinkommendes Paket warten .....	1014
11.11.4 Ein Paket zum Senden vorbereiten .....	1015
11.11.5 Methoden der Klasse DatagramPacket .....	1016
11.11.6 Das Paket senden .....	1017
<b>11.12 E-Mail *</b> .....	1019
11.12.1 Wie eine E-Mail um die Welt geht .....	1019
11.12.2 Das Simple Mail Transfer Protocol und RFC 822 .....	1019
11.12.3 POP (Post Office Protocol) .....	1020
11.12.4 Die JavaMail API .....	1020
11.12.5 E-Mails mittels POP3 abrufen .....	1023
11.12.6 Multipart-Nachrichten verarbeiten .....	1025
11.12.7 E-Mails versenden .....	1027
11.12.8 Ereignisse und Suchen .....	1030
<b>11.13 Tiefer liegende Netzwerkeigenschaften *</b> .....	1031
11.13.1 Internet Control Message Protocol (ICMP) .....	1031
11.13.2 MAC-Adresse .....	1032
<b>11.14 Zum Weiterlesen</b> .....	1033

## **12 Verteilte Programmierung mit RMI**

---

<b>12.1 Entfernte Objekte und Methoden</b> .....	1035
12.1.1 Stellvertreter helfen bei entfernten Methodenaufrufen .....	1035
12.1.2 Standards für entfernte Objekte .....	1037
<b>12.2 Java Remote Method Invocation</b> .....	1038
12.2.1 Zusammenspiel von Server, Registry und Client .....	1038
12.2.2 Wie die Stellvertreter die Daten übertragen .....	1038
12.2.3 Probleme mit entfernten Methoden .....	1039
12.2.4 Nutzen von RMI bei Middleware-Lösungen .....	1040
12.2.5 Zentrale Klassen und Schnittstellen .....	1041
12.2.6 Entfernte und lokale Objekte im Vergleich .....	1042
<b>12.3 Auf der Serverseite</b> .....	1042
12.3.1 Entfernte Schnittstelle deklarieren .....	1042
12.3.2 Remote-Objekt-Implementierung .....	1043

12.3.3	Stellvertreterobjekte .....	1044
12.3.4	Der Namensdienst (Registry) .....	1045
12.3.5	Remote-Objekt-Implementierung exportieren und beim Namensdienst anmelden .....	1047
12.3.6	Einfaches Logging .....	1049
12.3.7	Aufräumen mit dem DGC * .....	1050
<b>12.4</b>	<b>Auf der Clientseite .....</b>	<b>1051</b>
<b>12.5</b>	<b>Entfernte Objekte übergeben und laden .....</b>	<b>1052</b>
12.5.1	Klassen vom RMI-Klassenlader nachladen .....	1052
<b>12.6</b>	<b>Weitere Eigenschaften von RMI .....</b>	<b>1053</b>
12.6.1	RMI und CORBA .....	1053
12.6.2	RMI über HTTP getunnelt .....	1054
12.6.3	Automatische Remote-Objekt-Aktivierung .....	1054
<b>12.7</b>	<b>Java Message Service (JMS) .....</b>	<b>1055</b>
<b>12.8</b>	<b>Zum Weiterlesen .....</b>	<b>1055</b>

## **13 RESTful und SOAP Web-Services**

---

<b>13.1</b>	<b>Web-Services .....</b>	<b>1057</b>
<b>13.2</b>	<b>RESTful Web-Services .....</b>	<b>1059</b>
13.2.1	Aus Prinzip REST .....	1059
13.2.2	Jersey .....	1060
13.2.3	JAX-RS-Annotationen für den ersten REST-Service .....	1061
13.2.4	Test-Server starten .....	1062
13.2.5	REST-Services konsumieren .....	1063
13.2.6	Content-Handler, Marshaller und verschiedene MIME-Typen .....	1064
13.2.7	REST-Parameter .....	1067
13.2.8	REST-Services mit Parametern über die Jersey-Client-API aufrufen .....	1069
13.2.9	PUT-Anforderungen und das Senden von Daten .....	1070
13.2.10	PUT/POST/DELETE-Sendungen mit der Jersey-Client-API absetzen .....	1071
<b>13.3</b>	<b>Daily Soap und das SOAP-Protokoll .....</b>	<b>1071</b>
13.3.1	Die technische Realisierung .....	1072
13.3.2	Web-Service-APIs und Implementierungen .....	1072
13.3.3	@WebService in Java 6 .....	1073
13.3.4	Einen Web-Service definieren .....	1074

13.3.5	Web-Services veröffentlichen .....	1075
13.3.6	Einen JAX-WS-Client implementieren .....	1076
<b>13.4</b>	<b>Zum Weiterlesen .....</b>	<b>1078</b>

## **14 JavaServer Pages und Servlets**

---

<b>14.1</b>	<b>Dynamisch generierte Webseiten .....</b>	<b>1079</b>
14.1.1	Was sind Servlets? .....	1080
14.1.2	Was sind JavaServer Pages? .....	1081
<b>14.2</b>	<b>Servlets und JSPs mit Tomcat entwickeln .....</b>	<b>1082</b>
14.2.1	Servlet-Container .....	1082
14.2.2	Entwicklung der Servlet/JSP-Spezifikationen .....	1083
14.2.3	Webserver mit Servlet-Funktionalität .....	1083
14.2.4	Tomcat installieren .....	1084
14.2.5	Ablageort für eigene JSPs .....	1086
14.2.6	Webapplikationen .....	1087
14.2.7	Zuordnung von Webapplikationen zu physikalischen Verzeichnissen ...	1088
14.2.8	Web-Projekt mit Eclipse IDE for Java EE Developers .....	1088
<b>14.3</b>	<b>Statisches und Dynamisches .....</b>	<b>1090</b>
14.3.1	Statischer Template-Code .....	1090
14.3.2	Dynamische Inhalte .....	1090
14.3.3	Kommentare .....	1091
<b>14.4</b>	<b>Die Expression Language (EL) .....</b>	<b>1091</b>
14.4.1	Operatoren der EL .....	1092
14.4.2	Literale .....	1092
14.4.3	Implizite EL-Objekte .....	1093
<b>14.5</b>	<b>Formulardaten .....</b>	<b>1094</b>
14.5.1	Einen Parameter auslesen .....	1094
14.5.2	HTML-Formulare .....	1095
<b>14.6</b>	<b>Auf Beans zurückgreifen .....</b>	<b>1096</b>
14.6.1	Beans in JSPs anlegen .....	1097
14.6.2	Properties einer Bean im EL-Ausdruck erfragen .....	1097
14.6.3	Properties mit <jsp:setProperty> setzen .....	1097
14.6.4	Bean-Klasse zum Testen von E-Mail-Adressen .....	1098
14.6.5	Parameterwerte in Bean übertragen .....	1100

<b>14.7</b>	<b>JSP-Tag-Libraries</b> .....	1101
14.7.1	Standard Tag Library (JSTL) .....	1101
<b>14.8</b>	<b>Skripting-Elemente in JSPs</b> .....	1105
14.8.1	Scriptlets .....	1106
14.8.2	JSP-Ausdrücke .....	1106
14.8.3	JSP-Deklarationen .....	1107
14.8.4	Quoting .....	1107
14.8.5	Entsprechende XML-Tags .....	1108
14.8.6	Implizite Objekte für Scriptlets und JSP-Ausdrücke .....	1108
<b>14.9</b>	<b>Sitzungsverfolgung (Session Tracking)</b> .....	1109
14.9.1	Lösungen für die Sitzungsverfolgung .....	1109
14.9.2	Sitzungen in JSPs .....	1110
14.9.3	Auf Session-Dateien zurückgreifen .....	1111
<b>14.10</b>	<b>Servlets</b> .....	1111
14.10.1	Servlets compilieren .....	1115
14.10.2	Servlet-Mapping .....	1116
14.10.3	Der Lebenszyklus eines Servlets .....	1117
14.10.4	Mehrere Anfragen beim Servlet und die Thread-Sicherheit .....	1118
14.10.5	Servlets und Sessions .....	1118
14.10.6	Weiterleiten und Einbinden von Servlet-Inhalten .....	1119
<b>14.11</b>	<b>Zum Weiterlesen</b> .....	1120

## 15 Applets

---

<b>15.1</b>	<b>Applets in der Wiege von Java</b> .....	1123
15.1.1	Applets heute .....	1123
15.1.2	(J)Applet und Applikationen .....	1124
15.1.3	Das erste Hallo-Applet .....	1125
<b>15.2</b>	<b>Die Applet-API</b> .....	1127
15.2.1	Die Zyklen eines Applets .....	1127
15.2.2	Parameter an das Applet übergeben .....	1127
15.2.3	Wie das Applet den Browser-Inhalt ändern kann * .....	1129
15.2.4	Den Ursprung des Applets erfragen .....	1130
15.2.5	Datenaustausch zwischen Applets * .....	1131
15.2.6	Was ein Applet alles darf * .....	1134



15.2.7	Ist Java im Browser aktiviert? *	1135
15.2.8	Applet unter Firefox (Netscape) oder Microsoft Internet Explorer? *	1135
<b>15.3</b>	<b>Webstart</b>	1137
<b>15.4</b>	<b>Zum Weiterlesen</b>	1137

## **16 Datenbankmanagement mit JDBC**

---

<b>16.1</b>	<b>Relationale Datenbanken</b>	1139
16.1.1	Das relationale Modell	1139
<b>16.2</b>	<b>Datenbanken und Tools</b>	1140
16.2.1	HSQLDB	1141
16.2.2	Weitere Datenbanken *	1142
16.2.3	Eclipse-Plugins zum Durchschauen von Datenbanken	1145
<b>16.3</b>	<b>JDBC und Datenbanktreiber</b>	1148
16.3.1	Treibertypen *	1149
16.3.2	JDBC-Versionen *	1151
<b>16.4</b>	<b>Eine Beispielabfrage</b>	1152
16.4.1	Schritte zur Datenbankabfrage	1152
16.4.2	Ein Client für die HSQLDB-Datenbank	1153
16.4.3	Datenbankbrowser und eine Beispielabfrage unter NetBeans	1155
<b>16.5</b>	<b>Mit Java an eine Datenbank andocken</b>	1159
16.5.1	Der Treiber-Manager *	1160
16.5.2	Den Treiber laden	1160
16.5.3	Eine Aufzählung aller Treiber *	1162
16.5.4	Log-Informationen *	1162
16.5.5	Verbindung zur Datenbank auf- und abbauen	1163
<b>16.6</b>	<b>Datenbankabfragen</b>	1167
16.6.1	Abfragen über das Statement-Objekt	1168
16.6.2	Ergebnisse einer Abfrage in ResultSet	1169
16.6.3	Java und SQL-Datentypen	1170
16.6.4	Date, Time und Timestamp	1173
16.6.5	Unicode in der Spalte korrekt auslesen	1175
16.6.6	Eine SQL-NULL und wasNull() bei ResultSet	1175
16.6.7	Wie viele Zeilen hat ein ResultSet? *	1176

<b>16.7</b>	<b>Elemente einer Datenbank hinzufügen und aktualisieren</b>	1176
16.7.1	Batch-Updates	1177
16.7.2	Die Ausnahmen bei JDBC, SQLException und Unterklassen	1179
<b>16.8</b>	<b>ResultSet und RowSet *</b>	1182
16.8.1	Die Schnittstelle RowSet	1182
16.8.2	Implementierungen von RowSet	1182
16.8.3	Der Typ CachedRowSet	1183
16.8.4	Der Typ WebRowSet	1184
<b>16.9</b>	<b>Vorbereitete Anweisungen (Prepared Statements)</b>	1187
16.9.1	PreparedStatement-Objekte vorbereiten	1187
16.9.2	Werte für die Platzhalter eines PreparedStatement	1189
<b>16.10</b>	<b>Transaktionen</b>	1190
<b>16.11</b>	<b>Metadaten *</b>	1190
16.11.1	Metadaten über die Tabelle	1190
16.11.2	Informationen über die Datenbank	1194
<b>16.12</b>	<b>Vorbereitete Datenbankverbindungen</b>	1195
16.12.1	DataSource	1195
16.12.2	Gepoolte Verbindungen	1198
<b>16.13</b>	<b>JPA-Beispiel mit der NetBeans-IDE</b>	1199
16.13.1	Entity-Beans generieren	1199
16.13.2	Die Quellen im Überblick	1201
16.13.3	Persistence Unit	1203
16.13.4	Ein JPA-Beispielprogramm	1204
<b>16.14</b>	<b>Zum Weiterlesen</b>	1205

## **17 Technologien für die Infrastruktur**

---

<b>17.1</b>	<b>Property-Validierung durch Bean Validation</b>	1207
17.1.1	Technische Abhängigkeiten und POJOs	1218
<b>17.2</b>	<b>Wie eine Implementierung an die richtige Stelle kommt</b>	1220
17.2.1	Arbeiten mit dem ServiceLoader	1221
17.2.2	Die Utility-Klasse Lookup als ServiceLoader-Fassade	1223
17.2.3	Contexts and Dependency Injection (CDI) aus dem JSR-299	1224
<b>17.3</b>	<b>Zum Weiterlesen</b>	1226

## 18 Reflection und Annotationen

---

<b>18.1</b>	<b>Metadaten</b> .....	1227
18.1.1	Metadaten durch JavaDoc-Tags .....	1227
<b>18.2</b>	<b>Metadaten der Klassen mit dem Class-Objekt</b> .....	1228
18.2.1	An ein Class-Objekt kommen .....	1228
18.2.2	Was das Class-Objekt beschreibt * .....	1231
18.2.3	Der Name der Klasse .....	1233
18.2.4	instanceof mit Class-Objekten * .....	1235
18.2.5	Oberklassen finden * .....	1236
18.2.6	Implementierte Interfaces einer Klasse oder eines Interfaces * .....	1237
18.2.7	Modifizierer und die Klasse Modifier * .....	1238
18.2.8	Die Arbeit auf dem Feld * .....	1240
<b>18.3</b>	<b>Attribute, Methoden und Konstruktoren</b> .....	1241
18.3.1	Reflections – Gespür für die Attribute einer Klasse .....	1243
18.3.2	Methoden einer Klasse erfragen .....	1246
18.3.3	Properties einer Bean erfragen .....	1250
18.3.4	Konstruktoren einer Klasse .....	1250
18.3.5	Annotationen .....	1253
<b>18.4</b>	<b>Objekte erzeugen und manipulieren</b> .....	1253
18.4.1	Objekte erzeugen .....	1253
18.4.2	Die Belegung der Variablen erfragen .....	1255
18.4.3	Eine generische eigene toString()-Methode * .....	1257
18.4.4	Variablen setzen .....	1260
18.4.5	Bean-Zustände kopieren * .....	1262
18.4.6	Private Attribute ändern .....	1262
18.4.7	Methoden aufrufen .....	1263
18.4.8	Statische Methoden aufrufen .....	1265
18.4.9	Dynamische Methodenaufrufe bei festen Methoden beschleunigen * ....	1266
<b>18.5</b>	<b>Eigene Annotationstypen *</b> .....	1268
18.5.1	Annotationen zum Laden von Ressourcen .....	1268
18.5.2	Neue Annotationen deklarieren .....	1269
18.5.3	Annotationen mit genau einem Attribut .....	1270
18.5.4	Element-Werte-Paare (Attribute) hinzufügen .....	1271
18.5.5	Annotationsattribute vom Typ einer Aufzählung .....	1272

18.5.6	Felder von Annotationsattributen .....	1273
18.5.7	Vorbelegte Attribute .....	1274
18.5.8	Annotieren von Annotationstypen .....	1276
18.5.9	Deklarationen für unsere Ressourcen-Annotationen .....	1280
18.5.10	Annotierte Elemente auslesen .....	1282
18.5.11	Auf die Annotationsattribute zugreifen .....	1283
18.5.12	Komplettbeispiel zum Initialisieren von Ressourcen .....	1284
18.5.13	Mögliche Nachteile von Annotationen .....	1287
<b>18.6</b>	<b>Zum Weiterlesen .....</b>	<b>1288</b>

## **19 Dynamische Übersetzung und Skriptsprachen**

---

<b>19.1</b>	<b>Codegenerierung .....</b>	<b>1290</b>
19.1.1	Generierung von Java-Quellcode .....	1291
19.1.2	Codetransformationen .....	1294
19.1.3	Erstellen von Java-Bytecode .....	1294
<b>19.2</b>	<b>Programme mit der Compiler API übersetzen .....</b>	<b>1295</b>
19.2.1	Java Compiler API .....	1295
19.2.2	Fehler-Diagnose .....	1298
19.2.3	Eine im String angegebene Compilationseinheit übersetzen .....	1300
19.2.4	Wenn Quelle und Ziel der Speicher sind .....	1302
<b>19.3</b>	<b>Ausführen von Skripten .....</b>	<b>1306</b>
19.3.1	JavaScript-Programme ausführen .....	1306
19.3.2	Groovy .....	1308
<b>19.4</b>	<b>Zum Weiterlesen .....</b>	<b>1314</b>

## **20 Logging und Monitoring**

---

<b>20.1</b>	<b>Logging mit Java .....</b>	<b>1315</b>
20.1.1	Logging-APIs .....	1315
20.1.2	Logging mit java.util.logging .....	1316
20.1.3	Logging mit log4j * .....	1320
20.1.4	Die Simple Logging Facade .....	1323
<b>20.2</b>	<b>Systemzustände überwachen .....</b>	<b>1324</b>

<b>20.3</b>	<b>MBean-Typen, MBean-Server und weitere Begriffe</b> .....	1325
20.3.1	MXBeans des Systems .....	1325
<b>20.4</b>	<b>Geschwätzige Programme und JConsole</b> .....	1328
20.4.1	JConsole .....	1328
<b>20.5</b>	<b>Der MBeanServer</b> .....	1330
<b>20.6</b>	<b>Eine eigene Standard-MBean</b> .....	1331
20.6.1	Management-Schnittstelle .....	1332
20.6.2	Implementierung der Managed-Ressource .....	1332
20.6.3	Anmeldung beim Server .....	1333
20.6.4	Eine eigene Bean in JConsole einbringen .....	1333
<b>20.7</b>	<b>Zum Weiterlesen</b> .....	1336

## **21 Java Native Interface (JNI)**

---

<b>21.1</b>	<b>Java Native Interface und Invocation-API</b> .....	1337
<b>21.2</b>	<b>Eine C-Funktion in ein Java-Programm einbinden</b> .....	1338
21.2.1	Den Java-Code schreiben .....	1338
<b>21.3</b>	<b>Dynamische Bibliotheken erzeugen</b> .....	1340
21.3.1	Die Header-Datei erzeugen .....	1340
21.3.2	Implementierung der Funktion in C .....	1342
21.3.3	Die C-Programme übersetzen und die dynamische Bibliothek überzeugen .....	1343
<b>21.4</b>	<b>Nativ die Stringlänge ermitteln</b> .....	1345
<b>21.5</b>	<b>Erweiterte JNI-Eigenschaften</b> .....	1347
21.5.1	Klassendefinitionen .....	1347
21.5.2	Zugriff auf Attribute .....	1348
21.5.3	Methoden aufrufen .....	1350
21.5.4	Threads und Synchronisation .....	1352
<b>21.6</b>	<b>Einfache Anbindung von existierenden Bibliotheken</b> .....	1352
21.6.1	Generieren von JNI-Wrappern aus C++-Klassen und C-Headern .....	1352
21.6.2	COM-Schnittstellen anzapfen .....	1353
<b>21.7</b>	<b>Invocation-API</b> .....	1353
<b>21.8</b>	<b>Zum Weiterlesen</b> .....	1354

## 22 Sicherheitskonzepte

---

<b>22.1</b>	<b>Zentrale Elemente der Java-Sicherheit</b> .....	1355
22.1.1	Security-API der Java SE .....	1355
22.1.2	Cryptographic Service Providers .....	1356
<b>22.2</b>	<b>Der Sandkasten (Sandbox)</b> .....	1357
<b>22.3</b>	<b>Sicherheitsmanager (Security Manager)</b> .....	1358
22.3.1	Der Sicherheitsmanager bei Applets .....	1359
22.3.2	Sicherheitsmanager aktivieren .....	1361
22.3.3	Rechte durch Policy-Dateien vergeben .....	1363
22.3.4	Erstellen von Rechedateien mit dem grafischen Policy-Tool .....	1365
22.3.5	Kritik an den Policies .....	1366
<b>22.4</b>	<b>Signierung</b> .....	1368
22.4.1	Warum signieren? .....	1368
22.4.2	Digitale Ausweise und die Zertifizierungsstelle .....	1368
22.4.3	Mit keytool Schlüssel erzeugen .....	1369
22.4.4	Signieren mit jarsigner .....	1370
<b>22.5</b>	<b>Digitale Unterschriften *</b> .....	1371
22.5.1	Die MDx-Reihe .....	1371
22.5.2	Secure Hash Algorithm (SHA) .....	1372
22.5.3	Mit der Security-API einen Fingerabdruck berechnen .....	1372
22.5.4	Die Klasse MessageDigest .....	1373
<b>22.6</b>	<b>Verschlüsseln von Daten(-strömen) *</b> .....	1375
22.6.1	Den Schlüssel, bitte .....	1375
22.6.2	Verschlüsseln mit Cipher .....	1377
22.6.3	Verschlüsseln von Datenströmen .....	1377
<b>22.7</b>	<b>Zum Weiterlesen</b> .....	1379

## 23 Dienstprogramme für die Java-Umgebung

---

<b>23.1</b>	<b>Programme des JDK</b> .....	1381
<b>23.2</b>	<b>Monitoringprogramme vom JDK</b> .....	1381
23.2.1	jps .....	1381
23.2.2	jstat .....	1382
23.2.3	jmap .....	1382

23.2.4	jstack .....	1383
23.2.5	VisualVM .....	1385
<b>23.3</b>	<b>Programmieren mit der Tools-API .....</b>	<b>1390</b>
23.3.1	Eigene Doclets .....	1390
<b>23.4</b>	<b>Ant .....</b>	<b>1393</b>
23.4.1	Bezug und Installation von Ant .....	1393
23.4.2	Das Build-Skript build.xml .....	1394
23.4.3	Build den Build .....	1394
23.4.4	Properties .....	1395
23.4.5	Externe und vordefinierte Properties .....	1396
23.4.6	Weitere Ant-Tasks .....	1397
<b>23.5</b>	<b>Disassembler, Decompiler und Obfuscator .....</b>	<b>1398</b>
23.5.1	Der Disassembler javap .....	1399
23.5.2	Decompiler .....	1404
23.5.3	Obfuscatoren .....	1406
<b>23.6</b>	<b>Weitere Dienstprogramme .....</b>	<b>1408</b>
23.6.1	Sourcecode Beautifier .....	1408
23.6.2	Java-Programme als Systemdienst ausführen .....	1409
<b>23.7</b>	<b>Zum Weiterlesen .....</b>	<b>1410</b>
<b>Index</b>	<b>.....</b>	<b>1411</b>