

# Inhalt

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

## **1 Neues in Java 9** 43

---

<b>1.1 Klassenlader (Class Loader) und Modul-/Klassenpfad .....</b>	<b>43</b>
1.1.1 Klassenladen auf Abruf .....	43
1.1.2 Klassenlader bei der Arbeit zusehen .....	44
1.1.3 JMOD-Dateien und JAR-Dateien .....	45
1.1.4 Woher die kleinen Klassen kommen: die Suchorte und spezielle Klassenlader .....	46
1.1.5 Setzen des Modulpfades .....	47
<b>1.2 Module entwickeln und einbinden .....</b>	<b>49</b>
1.2.1 Wer sieht wen .....	49
1.2.2 Plattform-Module und JMOD-Beispiel .....	50
1.2.3 Verbotene Plattformeigenschaften nutzen, --add-exports .....	51
1.2.4 Plattformmodule einbinden, --add-modules und --add-opens .....	53
1.2.5 Projektabhängigkeiten in Eclipse .....	55
1.2.6 Benannte Module und module-info.java .....	57
1.2.7 Automatische Module .....	61
1.2.8 Unbenanntes Modul .....	62
1.2.9 Lesbarkeit und Zugreifbarkeit .....	62
1.2.10 Modul-Migration .....	63
<b>1.3 Sprachänderungen in Java 9 .....</b>	<b>64</b>
<b>1.4 Bibliotheksänderungen in Java 9 .....</b>	<b>65</b>
<b>1.5 Änderungen in den Werkzeugen von Java 9 .....</b>	<b>65</b>
<b>1.6 JDK 9-HotSpot-JVM-Änderungen .....</b>	<b>66</b>
<b>1.7 Zum Weiterlesen .....</b>	<b>66</b>

## **2 Die Klassenbibliothek** 67

---

<b>2.1 Die Java-Klassenphilosophie .....</b>	<b>67</b>
2.1.1 Modul, Paket, Typ .....	67
2.1.2 Übersicht über die Pakete der Standardbibliothek .....	70

<b>2.2</b>	<b>Die Utility-Klassen System und Properties</b> .....	74
2.2.1	Systemeigenschaften der Java-Umgebung .....	76
2.2.2	Zeilenumbruchzeichen, line.separator .....	77
2.2.3	Eigene Properties von der Konsole aus setzen * .....	78
2.2.4	Umgebungsvariablen des Betriebssystems * .....	80
2.2.5	Einfache Zeitmessung und Profiling * .....	81
<b>2.3</b>	<b>Versionskennungen auslesen, aufbauen, parsen und vergleichen</b> .....	84
<b>2.4</b>	<b>Einfache Benutzereingaben</b> .....	86
2.4.1	Grafischer Eingabedialog über JOptionPane .....	86
2.4.2	Geschützte Passwort-Eingaben mit der Klasse Console * .....	87
<b>2.5</b>	<b>Benutzereinstellungen *</b> .....	88
2.5.1	Benutzereinstellungen mit der Preferences-API .....	89
2.5.2	Einträge einfügen, auslesen und löschen .....	90
2.5.3	Auslesen der Daten und Schreiben in einem anderen Format .....	92
2.5.4	Auf Ereignisse horchen .....	92
2.5.5	Zugriff auf die gesamte Windows-Registry .....	94
<b>2.6</b>	<b>Maven: Build-Management und Abhängigkeiten auflösen</b> .....	95
2.6.1	Beispielprojekt in Eclipse mit Maven .....	95
2.6.2	Properties hinzunehmen .....	96
2.6.3	Dependency hinzunehmen .....	96
2.6.4	Lokales- und Remote-Repository .....	98
2.6.5	Lebenszyklus, Phasen und Maven-Plugins .....	98
2.6.6	Archetypes .....	98
<b>2.7</b>	<b>Zum Weiterlesen</b> .....	99

## **3 Fortgeschrittene String-Verarbeitung** 101

---

<b>3.1</b>	<b>Erweiterte Zeicheneigenschaften</b> .....	101
3.1.1	isXXX(...) -Methoden .....	101
3.1.2	Unicode-Blöcke .....	102
3.1.3	Unicode-Skripte .....	103
<b>3.2</b>	<b>Reguläre Ausdrücke</b> .....	104
3.2.1	Pattern.matches(...) bzw. String#matches(...) .....	104
3.2.2	Die Klassen Pattern und Matcher .....	108
3.2.3	Finden und nicht matchen .....	113
3.2.4	Gruppen .....	114
3.2.5	Gierige und nicht gierige Operatoren * .....	115

3.2.6	Mit MatchResult alle Ergebnisse einsammeln *	116
3.2.7	Suchen und Ersetzen mit Mustern	118
3.2.8	Hangman Version 2	121
<b>3.3</b>	<b>Zerlegen von Zeichenketten</b>	122
3.3.1	Zerlegen von Zeichensequenzen über String oder Pattern	122
3.3.2	Mehr vom Scanner	123
3.3.3	Die Klasse StringTokenizer *	129
3.3.4	BreakIterator als Zeichen-, Wort-, Zeilen- und Satztrenner *	130
3.3.5	StreamTokenizer *	133
<b>3.4</b>	<b>Zeichenkodierungen, XML/HTML-Entities, Base64 *</b>	137
3.4.1	Unicode und 8-Bit-Abbildungen/Unicode Encoding	137
3.4.2	Kodierungen über die Klasse String vornehmen	137
3.4.3	Das Paket java.nio.charset und der Typ Charset	138
3.4.4	Konvertieren mit OutputStreamWriter/InputStreamReader-Klassen	139
3.4.5	XML/HTML-Entities ausmaskieren	140
3.4.6	Base64-Kodierung	141
<b>3.5</b>	<b>Ausgaben formatieren</b>	142
3.5.1	Die Formatter-Klasse *	142
3.5.2	Formatieren mit Masken *	145
3.5.3	Format-Klassen	146
3.5.4	Zahlen, Prozente und Währungen mit NumberFormat und DecimalFormat formatieren *	149
3.5.5	MessageFormat und Pluralbildung mit ChoiceFormat	152
<b>3.6</b>	<b>Sprachabhängiges Vergleichen und Normalisierung *</b>	154
3.6.1	Die Klasse Collator	154
3.6.2	Effiziente interne Speicherung für die Sortierung	157
3.6.3	Normalisierung	158
<b>3.7</b>	<b>Phonetische Vergleiche *</b>	159
<b>3.8</b>	<b>Zum Weiterlesen</b>	160

## **4 Datenstrukturen und Algorithmen** 161

---

<b>4.1</b>	<b>Datenstrukturen und die Collection-API</b>	161
4.1.1	Designprinzip mit Schnittstellen, abstrakten und konkreten Klassen	162
4.1.2	Die Basisschnittstellen Collection und Map	162
4.1.3	Die Utility-Klassen Collections und Arrays	163
4.1.4	Das erste Programm mit Container-Klassen	163

4.1.5	Die Schnittstelle Collection und Kernkonzepte .....	164
4.1.6	Schnittstellen, die Collection erweitern, und Map .....	168
4.1.7	Konkrete Container-Klassen .....	170
4.1.8	Generische Datentypen in der Collection-API .....	172
4.1.9	Die Schnittstelle Iterable und das erweiterte for .....	173
<b>4.2</b>	<b>Listen</b> .....	<b>173</b>
4.2.1	Erstes Listen-Beispiel .....	174
4.2.2	Auswahlkriterium ArrayList oder LinkedList .....	175
4.2.3	Die Schnittstelle List .....	175
4.2.4	ArrayList .....	181
4.2.5	LinkedList .....	183
4.2.6	Der Array-Adapter Arrays.asList(...) .....	184
4.2.7	ListIterator * .....	186
4.2.8	toArray(...) von Collection verstehen – die Gefahr einer Falle erkennen .....	187
4.2.9	Primitive Elemente in Datenstrukturen verwalten .....	190
<b>4.3</b>	<b>Mengen (Sets)</b> .....	<b>191</b>
4.3.1	Ein erstes Mengen-Beispiel .....	191
4.3.2	Methoden der Schnittstelle Set .....	193
4.3.3	HashSet .....	195
4.3.4	TreeSet – die sortierte Menge .....	195
4.3.5	Die Schnittstellen NavigableSet und SortedSet .....	197
4.3.6	LinkedHashSet .....	200
<b>4.4</b>	<b>Queues (Schlangen) und Deques</b> .....	<b>201</b>
4.4.1	Queue-Klassen .....	201
4.4.2	Deque-Klassen .....	203
4.4.3	Blockierende Queues und Prioritätswarteschlangen .....	204
4.4.4	PriorityQueue .....	204
<b>4.5</b>	<b>Stack (Kellerspeicher, Stapel)</b> .....	<b>209</b>
4.5.1	Die Methoden von java.util.Stack .....	210
<b>4.6</b>	<b>Assoziative Speicher</b> .....	<b>211</b>
4.6.1	Die Klassen HashMap und TreeMap .....	212
4.6.2	Einfügen und Abfragen des Assoziativspeichers .....	214
4.6.3	Über die Bedeutung von equals(...) und hashCode() bei Elementen .....	222
4.6.4	Eigene Objekte hashen .....	222
4.6.5	LinkedHashMap und LRU-Implementierungen .....	224
4.6.6	IdentityHashMap .....	225
4.6.7	Das Problem veränderter Elemente .....	225
4.6.8	Aufzählungen und Ansichten des Assoziativspeichers .....	226
4.6.9	Die Arbeitsweise einer Hash-Tabelle * .....	229
4.6.10	Die Properties-Klasse .....	232

<b>4.7</b>	<b>Immutable Datenstrukturen</b> .....	235
4.7.1	Nichtänderbare Datenstrukturen, immutable oder nur Lesen? .....	236
4.7.2	Null Object Pattern und leere Sammlungen/Iteratoren zurückgeben .....	237
4.7.3	Immutable Datenstrukturen mit einem Element: Singletons .....	239
4.7.4	Collections.unmodifiableXXX(...) .....	240
4.7.5	Statische ofXXX(...)-Methoden zum Aufbau unveränderbarer Set-, List-, Map-Datenstrukturen .....	243
<b>4.8</b>	<b>Mit einem Iterator durch die Daten wandern</b> .....	245
4.8.1	Iterator-Schnittstelle .....	245
4.8.2	Der Iterator kann (eventuell auch) löschen .....	246
4.8.3	Operationen auf allen Elementen durchführen .....	247
4.8.4	Einen Zufallszahlen-Iterator schreiben .....	247
4.8.5	Iteratoren von Sammlungen, das erweiterte for und Iterable .....	248
4.8.6	Fail-Fast-Iterator und die ConcurrentModificationException .....	252
4.8.7	Die Schnittstelle Enumeration * .....	253
<b>4.9</b>	<b>Algorithmen in Collections</b> .....	255
4.9.1	Die Bedeutung von Ordnung mit Comparator und Comparable .....	255
4.9.2	Sortieren .....	256
4.9.3	Den größten und kleinsten Wert einer Collection finden .....	259
4.9.4	Echte typsichere Container .....	262
4.9.5	Mit der Halbierungssuche nach Elementen fahnden .....	263
4.9.6	Ersetzen, Kopieren, Füllen, Umdrehen, Rotieren * .....	264
4.9.7	Listen durchwürfeln * .....	266
4.9.8	Häufigkeit eines Elements * .....	267
4.9.9	nCopies(...) * .....	267
<b>4.10</b>	<b>Datenstrukturen mit Änderungsmeldungen</b> .....	268
4.10.1	Das Paket javafx.collections .....	268
4.10.2	Fabrikmethoden in FXCollections .....	269
4.10.3	Änderungen melden über InvalidationListener .....	271
4.10.4	Änderungen melden über XXXChangeListener .....	271
4.10.5	Change-Klassen .....	272
4.10.6	Weitere Hilfsmethoden einer ObservableList .....	275
4.10.7	Melden von Änderungen an Arrays .....	276
4.10.8	Transformierte FXCollections .....	276
4.10.9	Weitere statische Methoden in FXCollections .....	277
<b>4.11</b>	<b>Stream-API</b> .....	278
4.11.1	Stream erzeugen .....	281
4.11.2	Terminale Operationen .....	284
4.11.3	Intermediäre Operationen .....	295
4.11.4	Streams mit primitiven Werten .....	301

4.11.5	Stream-Beziehungen, AutoCloseable .....	312
4.11.6	Stream-Builder .....	314
4.11.7	Spliterator .....	315
4.11.8	Klasse StreamSupport .....	316
<b>4.12</b>	<b>Spezielle threadsichere Datenstrukturen .....</b>	<b>317</b>
4.12.1	Zu Beginn nur synchronisierte Datenstrukturen in Java 1.0 .....	317
4.12.2	Nichtsynchronisierte Datenstrukturen in der Standard-Collection-API .....	317
4.12.3	Nebenläufiger Assoziativspeicher und die Schnittstelle ConcurrentMap .....	318
4.12.4	ConcurrentLinkedQueue .....	318
4.12.5	CopyOnWriteArrayList und CopyOnWriteArraySet .....	319
4.12.6	Wrapper zur Synchronisation .....	319
4.12.7	Blockierende Warteschlangen .....	320
4.12.8	ArrayBlockingQueue und LinkedBlockingQueue .....	321
4.12.9	PriorityBlockingQueue .....	323
4.12.10	Transfer-Warteschlangen – TransferQueue und LinkedTransferQueue .....	326
<b>4.13</b>	<b>Google Guava (Google Collections Library) .....</b>	<b>327</b>
4.13.1	Beispiel Multi-Set und Multi-Map .....	327
4.13.2	Datenstrukturen aus Guava .....	328
4.13.3	Utility-Klassen von Guava .....	330
4.13.4	Prädikate .....	331
4.13.5	Transformationen .....	332
<b>4.14</b>	<b>Die Klasse BitSet für Bitmengen * .....</b>	<b>332</b>
4.14.1	Ein BitSet anlegen .....	333
4.14.2	BitSet füllen und Zustände erfragen .....	334
4.14.3	Mengenorientierte Operationen .....	335
4.14.4	Weitere Methoden von BitSet .....	336
4.14.5	Primzahlen in einem BitSet verwalten .....	337
<b>4.15</b>	<b>Zum Weiterlesen .....</b>	<b>338</b>

## **5 Threads und nebenläufige Programmierung** 339

---

<b>5.1</b>	<b>Threads erzeugen .....</b>	<b>339</b>
5.1.1	Threads über die Schnittstelle Runnable implementieren .....	339
5.1.2	Thread mit Runnable starten .....	340
5.1.3	Die Klasse Thread erweitern .....	342
<b>5.2</b>	<b>Thread-Eigenschaften und Zustände .....</b>	<b>344</b>
5.2.1	Der Name eines Threads .....	344
5.2.2	Wer bin ich? .....	345

5.2.3	Die Zustände eines Threads *	346
5.2.4	Schläfer gesucht	347
5.2.5	Mit yield() und onSpinWait() auf Rechenzeit verzichten	348
5.2.6	Der Thread als Dämon	349
5.2.7	Freiheit für den Thread – das Ende	351
5.2.8	Einen Thread höflich mit Interrupt beenden	352
5.2.9	UncaughtExceptionHandler für unbehandelte Ausnahmen	354
5.2.10	Der stop() von außen und die Rettung mit ThreadDeath *	355
5.2.11	Ein Rendezvous mit join(...) *	356
5.2.12	Arbeit niederlegen und wieder aufnehmen *	358
5.2.13	Priorität *	359
<b>5.3</b>	<b>Der Ausführer (Executor) kommt</b>	<b>360</b>
5.3.1	Die Schnittstelle Executor	361
5.3.2	Glücklich in der Gruppe – die Thread-Pools	362
5.3.3	Threads mit Rückgabe über Callable	364
5.3.4	Mehrere Callable-Objekte abarbeiten	368
5.3.5	ScheduledExecutorService für wiederholende Ausgaben und Zeitsteuerungen nutzen	369
5.3.6	Asynchrones Programmieren mit CompletableFuture (CompletionStage) ...	369
<b>5.4</b>	<b>Synchronisation über kritische Abschnitte</b>	<b>372</b>
5.4.1	Gemeinsam genutzte Daten	372
5.4.2	Probleme beim gemeinsamen Zugriff und kritische Abschnitte	372
5.4.3	Punkte nebenläufig initialisieren	374
5.4.4	i++ sieht atomar aus, ist es aber nicht *	376
5.4.5	Kritische Abschnitte schützen	377
5.4.6	Kritische Abschnitte mit ReentrantLock schützen	379
5.4.7	Synchronisieren mit synchronized	386
5.4.8	Synchronized-Methoden der Klasse StringBuffer *	387
5.4.9	Mit synchronized synchronisierte Blöcke	388
5.4.10	Dann machen wir doch gleich alles synchronisiert!	389
5.4.11	Lock-Freigabe im Fall von Exceptions	390
5.4.12	Deadlocks	391
5.4.13	Mit synchronized nachträglich synchronisieren *	393
5.4.14	Monitore sind reentrant – gut für die Geschwindigkeit *	394
5.4.15	Synchronisierte Methodenaufrufe zusammenfassen *	395
<b>5.5</b>	<b>Synchronisation über Warten und Benachrichtigen</b>	<b>396</b>
5.5.1	Die Schnittstelle Condition	397
5.5.2	It's Disco-Time *	400
5.5.3	Warten mit wait(...) und Aufwecken mit notify()/notifyAll() *	405
5.5.4	Falls der Lock fehlt – IllegalMonitorStateException *	406

<b>5.6</b>	<b>Datensynchronisation durch besondere Concurrency-Klassen *</b> .....	408
5.6.1	Semaphor .....	408
5.6.2	Barrier und Austausch .....	412
5.6.3	Stop-and-go mit Exchanger .....	413
<b>5.7</b>	<b>Atomare Operationen und frische Werte mit volatile *</b> .....	414
5.7.1	Der Modifizierer volatile bei Objekt-/Klassenvariablen .....	414
5.7.2	Das Paket java.util.concurrent.atomic .....	416
<b>5.8</b>	<b>Teile und herrsche mit Fork und Join *</b> .....	418
5.8.1	Algorithmendesign per »teile und herrsche« .....	418
5.8.2	Nebenläufiges Lösen von D&C-Algorithmen .....	420
5.8.3	Fork und Join .....	421
<b>5.9</b>	<b>Mit dem Thread verbundene Variablen *</b> .....	424
5.9.1	ThreadLocal .....	425
5.9.2	InheritableThreadLocal .....	427
5.9.3	ThreadLocalRandom als schneller nebenläufiger Zufallszahlengenerator .....	428
5.9.4	ThreadLocal bei der Performance-Optimierung .....	430
<b>5.10</b>	<b>Threads in einer Thread-Gruppe *</b> .....	430
5.10.1	Aktive Threads in der Umgebung .....	431
5.10.2	Etwas über die aktuelle Thread-Gruppe herausfinden .....	431
5.10.3	Threads in einer Thread-Gruppe anlegen .....	434
5.10.4	Methoden von Thread und ThreadGroup im Vergleich .....	437
<b>5.11</b>	<b>Reaktive Programmierung und die Flow-API *</b> .....	438
<b>5.12</b>	<b>Zeitgesteuerte Abläufe</b> .....	440
5.12.1	Die Typen Timer und TimerTask .....	440
<b>5.13</b>	<b>Ausführen externer Programme, Starten von Prozessen *</b> .....	442
5.13.1	ProcessBuilder und Prozesskontrolle mit Process .....	442
5.13.2	ProcessHandle und Prozess-IDs .....	447
5.13.3	Einen Browser, E-Mail-Client oder Editor aufrufen .....	449
<b>5.14</b>	<b>Den Stack-Trace erfragen *</b> .....	450
5.14.1	StackTraceElement .....	451
5.14.2	printStackTrace(...) .....	452
5.14.3	StackTraceElement vom Thread erfragen .....	452
5.14.4	StackWalker und Stack-Walking API .....	453
<b>5.15</b>	<b>Einen Abbruch der virtuellen Maschine erkennen</b> .....	455
5.15.1	Shutdown-Hook .....	455
5.15.2	Signale .....	456
<b>5.16</b>	<b>Zum Weiterlesen</b> .....	457



<b>6.1 Sprachen der Länder</b> .....	459
6.1.1 Sprachen in Regionen über Locale-Objekte .....	459
<b>6.2 Internationalisierung und Lokalisierung</b> .....	463
6.2.1 ResourceBundle-Objekte und Ressource-Dateien .....	463
6.2.2 Ressource-Dateien zur Lokalisierung .....	464
6.2.3 Die Klasse ResourceBundle .....	465
6.2.4 Ladestrategie für ResourceBundle-Objekte .....	465
6.2.5 Ladeprozess und Format anpassen * .....	467
<b>6.3 Weltzeit *</b> .....	468
<b>6.4 Wichtige Datum-Klassen im Überblick</b> .....	469
6.4.1 Der 1.1.1970 .....	470
6.4.2 System.currentTimeMillis() .....	470
6.4.3 Einfache Zeitumrechnungen durch TimeUnit .....	471
<b>6.5 Die Klasse Date</b> .....	472
6.5.1 Objekte erzeugen und Methoden nutzen .....	472
6.5.2 Date-Objekte sind nicht immutable .....	474
<b>6.6 Date-Time-API</b> .....	474
6.6.1 Menschenzeit und Maschinenzzeit .....	476
6.6.2 Datumsklasse LocalDate .....	479
6.6.3 Ostertage * .....	480
6.6.4 Die Klasse YearMonth .....	481
6.6.5 Die Klasse MonthDay .....	482
6.6.6 Aufzählung DayOfWeek und Month .....	482
6.6.7 Klasse LocalTime .....	483
6.6.8 Klasse LocalDateTime .....	483
6.6.9 Klasse Year .....	484
6.6.10 Zeitzone-Klassen ZoneId und ZoneOffset .....	485
6.6.11 Temporale Klassen mit Zeitzoneinformationen .....	486
6.6.12 Klassen Period und Duration .....	489
6.6.13 Klasse Instant .....	492
6.6.14 Parsen und Formatieren von temporalen Typen .....	493
6.6.15 Das Paket java.time.temporal * .....	496
6.6.16 Rock Around the Clock .....	502
6.6.17 Konvertierungen zwischen der klassischen API und der Date-Time-API .....	503
<b>6.7 Calendar und GregorianCalendar</b> .....	504
6.7.1 Die abstrakte Klasse Calendar .....	505
6.7.2 Calendar nach Date und Millisekunden fragen .....	506

6.7.3	Abfragen und Setzen von Datumselementen über Feldbezeichner .....	506
6.7.4	Kalender-Exemplare bauen über den Calendar.Builder .....	510
6.7.5	Wie viele Tage hat der Monat, oder wie viele Monate hat ein Jahr? * .....	511
6.7.6	Wann beginnt die Woche und wann die erste Woche im Jahr? * .....	512
6.7.7	Der gregorianische Kalender .....	513
<b>6.8</b>	<b>Zeitzone in Java *</b> .....	517
6.8.1	Zeitzone durch die Klasse TimeZone repräsentieren .....	517
6.8.2	SimpleTimeZone .....	518
6.8.3	Methoden von TimeZone .....	519
<b>6.9</b>	<b>Formatieren und Parsen von Date-Objekten</b> .....	520
6.9.1	Mit DateFormat und SimpleDateFormat formatieren .....	521
6.9.2	Parsen von Datumswerten .....	526
<b>6.10</b>	<b>Die Default-Fälle</b> .....	528
<b>6.11</b>	<b>Zum Weiterlesen</b> .....	529

## **7 Dateien, Verzeichnisse und Dateizugriffe** 531

---

<b>7.1</b>	<b>Alte und neue Welt in java.io und java.nio</b> .....	531
7.1.1	java.io-Paket mit File-Klasse .....	531
7.1.2	NIO.2 und java.nio-Paket .....	532
7.1.3	java.io.File oder java.nio.*? .....	532
<b>7.2</b>	<b>Dateisysteme und Pfade</b> .....	533
7.2.1	FileSystem und Path .....	533
7.2.2	Die Utility-Klasse Files .....	539
7.2.3	Dateien kopieren und verschieben .....	541
7.2.4	Dateiattribute * .....	543
7.2.5	Neue Dateien, Verzeichnisse, symbolische Verknüpfungen anlegen und löschen .....	552
7.2.6	MIME-Typen herausfinden * .....	554
7.2.7	Verzeichnislistings (DirectoryStream/Stream) und Filter * .....	556
7.2.8	Rekursives Ablaufen des Verzeichnisbaums * .....	558
7.2.9	Rekursiv nach Dateien/Ordern suchen mit Files.find(...) * .....	562
7.2.10	FileSystem: Abstraktion eines Dateisystems * .....	563
7.2.11	Verzeichnisse im Dateisystem überwachen * .....	566
<b>7.3</b>	<b>Datei- und Verzeichnisoperationen mit der Klasse File</b> .....	569
7.3.1	Dateien und Verzeichnisse mit der Klasse File .....	569
7.3.2	Existiert das Verzeichnis oder die Datei, und ist es ...? .....	573

7.3.3	Verzeichnis- und Dateieigenschaften/-attribute .....	573
7.3.4	Umbenennen und Verzeichnisse anlegen .....	576
7.3.5	Verzeichnisse auflisten und Dateien filtern .....	576
7.3.6	Dateien berühren, neue Dateien anlegen, temporäre Dateien .....	579
7.3.7	Dateien und Verzeichnisse löschen .....	580
7.3.8	Wurzelverzeichnis, Laufwerksnamen, Plattenspeicher * .....	581
7.3.9	URL-, URI- und Path-Objekte aus einem File-Objekt ableiten * .....	583
7.3.10	Mit Locking Dateien sperren * .....	584
7.3.11	Sicherheitsprüfung * .....	585
<b>7.4</b>	<b>Dateien mit wahlfreiem Zugriff .....</b>	<b>585</b>
7.4.1	Ein RandomAccessFile zum Lesen und Schreiben öffnen .....	586
7.4.2	Aus dem RandomAccessFile lesen .....	586
7.4.3	Schreiben mit RandomAccessFile .....	589
7.4.4	Die Länge des RandomAccessFile .....	590
7.4.5	Hin und her in der Datei .....	590
<b>7.5</b>	<b>Wahlfreier Zugriff mit SeekableByteChannel und ByteBuffer * .....</b>	<b>591</b>
7.5.1	SeekableByteChannel .....	591
7.5.2	ByteBuffer .....	592
7.5.3	Beispiel mit Path + SeekableByteChannel + ByteBuffer .....	592
7.5.4	FileChannel .....	593
<b>7.6</b>	<b>Zum Weiterlesen .....</b>	<b>595</b>

## **8 Datenströme** 597

---

<b>8.1</b>	<b>Basisklassen für die Ein-/Ausgabe .....</b>	<b>597</b>
8.1.1	Die vier abstrakten Basisklassen .....	598
8.1.2	Die abstrakte Basisklasse OutputStream .....	598
8.1.3	Ein Datenschlucker * .....	601
8.1.4	Die abstrakte Basisklasse InputStream .....	601
8.1.5	Die abstrakte Basisklasse Writer .....	603
8.1.6	Die Schnittstelle Appendable * .....	605
8.1.7	Die abstrakte Basisklasse Reader .....	605
8.1.8	Die Schnittstellen Closeable, AutoCloseable und Flushable .....	608
<b>8.2</b>	<b>Lesen aus Dateien und Schreiben in Dateien .....</b>	<b>610</b>
8.2.1	Byteorientierte Datenströme über Files beziehen .....	610
8.2.2	Zeichenorientierte Datenströme über Files beziehen .....	611
8.2.3	Funktion von OpenOption bei den Files.newXXX(...)-Methoden .....	613
8.2.4	Ressourcen aus dem Modulpfad und aus JAR-Dateien laden .....	614

<b>8.3</b>	<b>Übersicht über Ein-/Ausgabeklassen</b> .....	616
8.3.1	Eingaben lesen .....	616
8.3.2	Ausgaben schreiben .....	617
8.3.3	Klassen gruppiert nach Ressourcen .....	617
8.3.4	Ströme mit SequencelInputStream zusammensetzen * .....	618
<b>8.4</b>	<b>Formatierte Textausgaben</b> .....	620
8.4.1	Die Klassen PrintWriter und PrintStream .....	620
8.4.2	System.out, System.err und System.in .....	625
<b>8.5</b>	<b>Die FileXXX-Stromklassen</b> .....	628
8.5.1	FileOutputStream und FileInputStream .....	629
8.5.2	Das FileDescriptor-Objekt * .....	632
8.5.3	Mit dem FileWriter Texte in Dateien schreiben .....	633
8.5.4	Zeichen mit der Klasse FileReader lesen .....	634
<b>8.6</b>	<b>Schreiben und Lesen aus Strings und Byte-Feldern</b> .....	635
8.6.1	Mit dem StringWriter ein String-Objekt füllen .....	636
8.6.2	CharArrayWriter .....	637
8.6.3	StringReader und CharArrayReader .....	637
8.6.4	Mit ByteArrayOutputStream in ein Byte-Feld schreiben .....	639
8.6.5	Mit ByteArrayInputStream aus einem Byte-Feld lesen .....	640
<b>8.7</b>	<b>Datenströme filtern und verketteten</b> .....	640
8.7.1	Streams als Filter verketteten (verschachteln) .....	641
8.7.2	Gepufferte Ausgaben mit BufferedWriter und BufferedOutputStream .....	642
8.7.3	Gepufferte Eingaben mit BufferedReader/BufferedInputStream .....	644
8.7.4	LineNumberReader zählt automatisch Zeilen mit * .....	645
8.7.5	Daten mit der Klasse PushbackReader zurücklegen * .....	646
8.7.6	DataOutputStream/DataInputStream * .....	649
8.7.7	Basisklassen für Filter * .....	650
8.7.8	Die Basisklasse FilterWriter * .....	651
8.7.9	Eingaben mit der Klasse FilterReader filtern * .....	652
<b>8.8</b>	<b>Vermittler zwischen Byte-Streams und Unicode-Strömen</b> .....	653
8.8.1	Datenkonvertierung durch den OutputStreamWriter .....	653
8.8.2	Automatische Konvertierungen mit dem InputStreamReader .....	655
<b>8.9</b>	<b>Kommunikation zwischen Threads mit Pipes *</b> .....	657
8.9.1	PipedOutputStream und PipedInputStream .....	657
8.9.2	PipedWriter und PipedReader .....	659
<b>8.10</b>	<b>Prüfsummen</b> .....	660
8.10.1	Die Schnittstelle Checksum .....	661
8.10.2	Die Klasse CRC32 .....	662
8.10.3	Die Adler32-Klasse .....	664

<b>8.11 Persistente Objekte und Serialisierung</b> .....	664
8.11.1 Objekte mit der Standardserialisierung speichern und lesen .....	665
8.11.2 Zwei einfache Anwendungen der Serialisierung * .....	668
8.11.3 Die Schnittstelle Serializable .....	669
8.11.4 Nicht serialisierbare Attribute aussparen .....	671
8.11.5 Deserialisierung absichern mit einem ObjectInputFilter * .....	673
8.11.6 Das Abspeichern selbst in die Hand nehmen .....	676
8.11.7 Tiefe Objektkopien * .....	679
8.11.8 Versionenverwaltung und die SUID .....	681
8.11.9 Wie die ArrayList serialisiert * .....	683
8.11.10 Probleme mit der Serialisierung .....	684
<b>8.12 Alternative Datenaustauschformate</b> .....	685
8.12.1 Serialisieren in XML-Dateien .....	685
8.12.2 XML-Serialisierung von JavaBeans mit JavaBeans Persistence * .....	685
8.12.3 Die Open-Source-Bibliothek XStream * .....	687
8.12.4 Binäre Serialisierung mit Google Protocol Buffers * .....	688
<b>8.13 Zum Weiterlesen</b> .....	688

## **9 Dateiformate** 689

---

<b>9.1 Einfache Dateiformate für strukturierte Daten</b> .....	690
9.1.1 Property-Dateien mit java.util.Properties lesen und schreiben .....	690
9.1.2 CSV-Dateien .....	692
<b>9.2 Dokumentenformate</b> .....	694
9.2.1 (X)HTML .....	694
9.2.2 PDF-Dokumente .....	695
9.2.3 Microsoft-Office-Dokumente .....	696
9.2.4 OASIS Open Document Format .....	697
<b>9.3 Datenkompression *</b> .....	697
9.3.1 Java-Unterstützung beim Komprimieren .....	698
9.3.2 Daten packen und entpacken .....	699
9.3.3 Datenströme komprimieren .....	701
9.3.4 ZIP-Archive .....	704
9.3.5 JAR-Archive .....	711
<b>9.4 Bildformate</b> .....	711
<b>9.5 Audiodateien</b> .....	711
9.5.1 Die Arbeit mit AudioSystem und Clip .....	711
9.5.2 AudioClip von JavaFX .....	712

9.5.3	MIDI-Dateien abspielen .....	712
9.5.4	ID-Tags aus MP3-Dateien .....	714
<b>9.6</b>	<b>Zum Weiterlesen .....</b>	<b>715</b>

---

## **10 Die eXtensible Markup Language (XML) und JSON** 717

---

<b>10.1</b>	<b>Auszeichnungssprachen .....</b>	<b>717</b>
10.1.1	Die Standard Generalized Markup Language (SGML) .....	717
10.1.2	Extensible Markup Language (XML) .....	718
<b>10.2</b>	<b>Eigenschaften von XML-Dokumenten .....</b>	<b>718</b>
10.2.1	Elemente und Attribute .....	718
10.2.2	Beschreibungssprache für den Aufbau von XML-Dokumenten .....	721
10.2.3	Schema – die moderne Alternative zu DTD .....	725
10.2.4	Namensraum (Namespace) .....	727
10.2.5	XML-Applikationen * .....	729
<b>10.3</b>	<b>Die Java-APIs für XML .....</b>	<b>729</b>
10.3.1	Das Document Object Model (DOM) .....	730
10.3.2	Simple API for XML Parsing (SAX) .....	730
10.3.3	Pull-API StAX .....	730
10.3.4	Java Document Object Model (JDOM) .....	731
10.3.5	JAXP als Java-Schnittstelle zu XML .....	731
10.3.6	DOM-Bäume einlesen mit JAXP * .....	732
<b>10.4</b>	<b>Java Architecture for XML Binding (JAXB) .....</b>	<b>732</b>
10.4.1	Bean für JAXB aufbauen .....	733
10.4.2	Utility-Klasse JAXB .....	734
10.4.3	Ganze Objektgraphen schreiben und lesen .....	735
10.4.4	JAXBContext und Marshaller/Unmarshaller nutzen .....	736
10.4.5	Validierung .....	739
10.4.6	Weitere JAXB-Annotationen * .....	742
10.4.7	JAXB-Beans aus XML-Schema-Datei generieren .....	749
<b>10.5</b>	<b>Serielle Verarbeitung mit StAX .....</b>	<b>755</b>
10.5.1	Unterschiede der Verarbeitungsmodelle .....	756
10.5.2	XML-Dateien mit dem Cursor-Verfahren lesen .....	757
10.5.3	XML-Dateien mit dem Iterator-Verfahren verarbeiten * .....	761
10.5.4	Mit Filtern arbeiten * .....	763
10.5.5	XML-Dokumente schreiben .....	764
<b>10.6</b>	<b>Serielle Verarbeitung von XML mit SAX * .....</b>	<b>768</b>

10.6.1	Schnittstellen von SAX .....	768
10.6.2	SAX-Parser erzeugen .....	769
10.6.3	Operationen der Schnittstelle ContentHandler .....	769
10.6.4	ErrorHandler und EntityResolver .....	772
<b>10.7</b>	<b>XML-Dateien mit JDOM verarbeiten</b> .....	<b>772</b>
10.7.1	JDOM beziehen .....	773
10.7.2	Paketübersicht * .....	774
10.7.3	Die Document-Klasse .....	775
10.7.4	Eingaben aus der Datei lesen .....	776
10.7.5	Das Dokument im XML-Format ausgeben .....	777
10.7.6	Der Dokumenttyp * .....	778
10.7.7	Elemente .....	779
10.7.8	Zugriff auf Elementinhalte .....	781
10.7.9	Liste mit Unterelementen erfragen * .....	784
10.7.10	Neue Elemente einfügen und ändern .....	784
10.7.11	Attributinhalt lesen und ändern .....	787
10.7.12	XPath .....	790
<b>10.8</b>	<b>Transformationen mit XSLT *</b> .....	<b>794</b>
10.8.1	Templates und XPath als Kernelemente von XSLT .....	794
10.8.2	Umwandlung von XML-Dateien mit JDOM und JAXP .....	797
<b>10.9</b>	<b>XML-Schema-Validierung *</b> .....	<b>797</b>
10.9.1	SchemaFactory und Schema .....	798
10.9.2	Validator .....	798
10.9.3	Validierung unterschiedlicher Datenquellen durchführen .....	799
<b>10.10</b>	<b>JSON-Serialisierung mit Jackson-Datenformat</b> .....	<b>799</b>
10.10.1	JSON im Kontext von JavaScript .....	799
10.10.2	JSON mit JavaScript-Engine von Java .....	800
10.10.3	JSON-Verarbeitung mit der Java API for JSON Processing .....	801
<b>10.11</b>	<b>Zum Weiterlesen</b> .....	<b>803</b>

## **11 Netzwerkprogrammierung** 805

---

<b>11.1</b>	<b>Grundlegende Begriffe</b> .....	<b>805</b>
<b>11.2</b>	<b>URI und URL</b> .....	<b>807</b>
11.2.1	Die Klasse URI .....	807
11.2.2	Die Klasse URL .....	807
11.2.3	Informationen über eine URL * .....	809
11.2.4	Der Zugriff auf die Daten über die Klasse URL .....	811

<b>11.3</b>	<b>URL-Parameter kodieren</b> .....	812
<b>11.4</b>	<b>Die Klassen URLConnection und HttpURLConnection</b> .....	813
11.4.1	Methoden und Anwendung von URLConnection .....	815
11.4.2	HttpURLConnection .....	816
11.4.3	Webseiten aufrufen, mit GET und POST Daten übergeben * .....	817
11.4.4	POST-Request absenden .....	819
11.4.5	Der Protokoll-Handler für JAR-Dateien .....	820
11.4.6	Basic Authentication und Proxy-Authentifizierung .....	821
<b>11.5</b>	<b>Host- und IP-Adressen</b> .....	822
11.5.1	Lebt der Rechner? .....	824
11.5.2	IP-Adresse des lokalen Hosts .....	825
11.5.3	Das Netz ist klasse * .....	826
11.5.4	NetworkInterface .....	826
<b>11.6</b>	<b>Socket-Verbindungen</b> .....	827
11.6.1	Das Netzwerk ist der Computer .....	827
11.6.2	Sockets .....	828
11.6.3	Eine Verbindung zum Server aufbauen .....	829
11.6.4	Sockets unter Spannung – die Ströme .....	830
11.6.5	Die Verbindung wieder abbauen .....	830
11.6.6	Zeitserver ansprechen .....	831
11.6.7	Informationen über den Socket * .....	832
11.6.8	Reine Verbindungsdaten über SocketAddress * .....	834
11.6.9	Die Serverseite mit ServerSocket vorbereiten .....	835
11.6.10	Ein Multiplikationsserver .....	836
11.6.11	Blockierendes Lesen .....	839
<b>11.7</b>	<b>HTTP Client API in Java 9</b> .....	841
<b>11.8</b>	<b>Neue externe Netzbibliotheken *</b> .....	843
11.8.1	HttpComponents .....	843
11.8.2	Async Http Client .....	844
11.8.3	Apache Commons Net .....	844
<b>11.9</b>	<b>Arbeitsweise eines Webservers *</b> .....	844
11.9.1	Das Hypertext Transfer Protocol (HTTP) .....	845
11.9.2	Anfragen an den Server .....	845
11.9.3	Die Antworten vom Server .....	847
11.9.4	Webserver mit com.sun.net.httpserver.HttpServer .....	851
<b>11.10</b>	<b>Verbindungen durch einen Proxy-Server *</b> .....	853
11.10.1	System-Properties .....	853
11.10.2	Verbindungen durch die Proxy-API .....	854



<b>11.11</b>	<b>Bidirektionale binäre Webkommunikation über WebSockets *</b> .....	855
11.11.1	WebSocket-Standard .....	855
11.11.2	WebSocket-Implementierungen .....	856
<b>11.12</b>	<b>Datagram-Sockets *</b> .....	856
11.12.1	Die Klasse DatagramSocket .....	858
11.12.2	Datagramme und die Klasse DatagramPacket .....	859
11.12.3	Auf ein hereinkommendes Paket warten .....	859
11.12.4	Ein Paket zum Senden vorbereiten .....	860
11.12.5	Methoden der Klasse DatagramPacket .....	861
11.12.6	Das Paket senden .....	861
<b>11.13</b>	<b>Tiefer liegende Netzwerkeigenschaften *</b> .....	862
11.13.1	MAC-Adressen auslesen .....	863
11.13.2	Internet Control Message Protocol (ICMP) .....	863
<b>11.14</b>	<b>Zum Weiterlesen</b> .....	863

---

## **12 RESTful und SOAP-Web-Services** 865

<b>12.1</b>	<b>Web-Services</b> .....	865
<b>12.2</b>	<b>RESTful Web-Services</b> .....	866
12.2.1	Aus Prinzip REST .....	867
12.2.2	Jersey .....	868
<b>12.3</b>	<b>Daily Soap und das SOAP-Protokoll</b> .....	868
12.3.1	Die technische Realisierung .....	869
12.3.2	Web-Service-APIs und Implementierungen .....	869
12.3.3	@WebService .....	870
12.3.4	Web-Service-Modul angeben .....	870
12.3.5	Einen Web-Service definieren .....	870
12.3.6	Web-Services veröffentlichen .....	872
12.3.7	Einen JAX-WS-Client implementieren .....	872
<b>12.4</b>	<b>Zum Weiterlesen</b> .....	874

---

## **13 Verteilte Programmierung mit RMI** 875

<b>13.1</b>	<b>Entfernte Objekte und Methoden</b> .....	875
13.1.1	Stellvertreter helfen bei entfernten Methodenaufrufen .....	875
13.1.2	Standards für entfernte Objekte .....	877

<b>13.2</b>	<b>Java Remote Method Invocation</b> .....	877
13.2.1	Zusammenspiel von Server, Registry und Client .....	877
13.2.2	Wie die Stellvertreter die Daten übertragen .....	878
13.2.3	Probleme mit entfernten Methoden .....	878
13.2.4	Nutzen von RMI bei Middleware-Lösungen .....	880
13.2.5	Zentrale Klassen und Schnittstellen .....	880
13.2.6	Entfernte und lokale Objekte im Vergleich .....	881
<b>13.3</b>	<b>Auf der Serverseite</b> .....	881
13.3.1	Entfernte Schnittstelle deklarieren .....	882
13.3.2	Remote-Objekt-Implementierung .....	882
13.3.3	Stellvertreterobjekte .....	883
13.3.4	Der Namensdienst (Registry) .....	883
13.3.5	Remote-Objekt-Implementierung exportieren und beim Namensdienst anmelden .....	885
13.3.6	Einfaches Logging .....	888
13.3.7	Aufräumen mit dem DGC * .....	889
<b>13.4</b>	<b>Auf der Client-Seite</b> .....	889
<b>13.5</b>	<b>Entfernte Objekte übergeben und laden</b> .....	890
13.5.1	Klassen vom RMI-Klassenlader nachladen .....	891
<b>13.6</b>	<b>Automatische Remote-Objekt-Aktivierung</b> .....	891
<b>13.7</b>	<b>Java Message Service (JMS)</b> .....	892
<b>13.8</b>	<b>Zum Weiterlesen</b> .....	892

---

## **14 Typen, Reflection und Annotationen** 895

<b>14.1</b>	<b>Metadaten</b> .....	895
14.1.1	Metadaten durch Javadoc-Tags .....	895
<b>14.2</b>	<b>Die Klasse Class</b> .....	896
14.2.1	An ein Class-Objekt kommen .....	896
14.2.2	Eine Class ist ein Type .....	899
<b>14.3</b>	<b>Klassenlader</b> .....	899
14.3.1	Die Klasse java.lang.ClassLoader .....	900
14.3.2	Hot Deployment mit dem URL-Classloader * .....	901
<b>14.4</b>	<b>Metadaten der Typen mit dem Class-Objekt</b> .....	905
14.4.1	Der Name des Typs .....	905
14.4.2	Was das Class-Objekt beschreibt * .....	908
14.4.3	instanceof mit Class-Objekten * .....	910

14.4.4	Oberklassen finden *	911
14.4.5	Implementierte Interfaces einer Klasse oder eines Interfaces *	912
14.4.6	Modifizierer und die Klasse Modifier *	912
14.4.7	Die Arbeit auf dem Feld *	914
<b>14.5</b>	<b>Attribute, Methoden und Konstruktoren</b>	<b>915</b>
14.5.1	Reflections – Gespür für die Attribute einer Klasse	917
14.5.2	Schnittstelle Member für Eigenschaften	918
14.5.3	Field-Klasse	919
14.5.4	Methoden einer Klasse erfragen	920
14.5.5	Properties einer Bean erfragen	923
14.5.6	Konstruktoren einer Klasse	924
14.5.7	Annotationen	925
14.5.8	Reflexionen über Module	925
<b>14.6</b>	<b>Objekte erzeugen und manipulieren</b>	<b>927</b>
14.6.1	Objekte erzeugen	927
14.6.2	Die Belegung der Variablen erfragen	928
14.6.3	Eine generische eigene toString()-Methode *	930
14.6.4	Variablen setzen	932
14.6.5	Bean-Zustände kopieren *	934
14.6.6	Private Attribute auslesen/ändern und der Typ AccessibleObject	934
14.6.7	Methoden aufrufen	936
14.6.8	Statische Methoden aufrufen	937
14.6.9	Dynamische Methodenaufrufe bei festen Methoden beschleunigen *	938
14.6.10	java.lang.reflect.Parameter	940
<b>14.7</b>	<b>Schneller aufrufen mit MethodType und MethodHandle</b>	<b>941</b>
<b>14.8</b>	<b>Eigene Annotationstypen *</b>	<b>942</b>
14.8.1	Annotationen zum Laden von Ressourcen	943
14.8.2	Neue Annotationen deklarieren	943
14.8.3	Annotationen mit genau einem Attribut	944
14.8.4	Element-Wert-Paare (Attribute) hinzufügen	945
14.8.5	Annotationsattribute vom Typ einer Aufzählung	946
14.8.6	Felder von Annotationsattributen	946
14.8.7	Vorbelegte Attribute	948
14.8.8	Annotieren von Annotationstypen	949
14.8.9	Deklarationen für unsere Ressourcen-Annotationen	954
14.8.10	Annotierte Elemente auslesen	956
14.8.11	Auf die Annotationsattribute zugreifen	958
14.8.12	Komplettbeispiel zum Initialisieren von Ressourcen	959
14.8.13	Mögliche Nachteile von Annotationen	961
<b>14.9</b>	<b>Zum Weiterlesen</b>	<b>962</b>

## 15 Logging und Monitoring 963

---

<b>15.1</b>	<b>Logging mit Java</b> .....	963
15.1.1	Logging-APIs .....	963
15.1.2	Logging mit java.util.logging .....	964
15.1.3	Die Simple Logging Facade (SLF4J) .....	970
15.1.4	Logging mit log4j 2 * .....	972
15.1.5	Aktuelle Entwicklungen der Java-Logging-APIs .....	975
15.1.6	System-Logging * .....	975
<b>15.2</b>	<b>Systemzustände überwachen</b> .....	976
<b>15.3</b>	<b>MBean-Typen, MBean-Server und weitere Begriffe</b> .....	976
15.3.1	MXBeans des Systems .....	977
<b>15.4</b>	<b>Geschwätzige Programme und JConsole</b> .....	979
15.4.1	JConsole .....	979
<b>15.5</b>	<b>Der MBeanServer</b> .....	981
<b>15.6</b>	<b>Eine eigene Standard-MBean</b> .....	982
15.6.1	Management-Schnittstelle .....	982
15.6.2	Implementierung der Managed Ressource .....	982
15.6.3	Anmeldung beim Server .....	983
15.6.4	Eine eigene Bean in JConsole einbringen .....	983
<b>15.7</b>	<b>Zum Weiterlesen</b> .....	986

## 16 Datenbankmanagement mit JDBC 987

---

<b>16.1</b>	<b>Relationale Datenbanken und Datenbankmanagementsysteme</b> .....	987
16.1.1	Das relationale Modell .....	987
16.1.2	Datenbanken und Tools .....	988
16.1.3	HSQLDB .....	988
16.1.4	Weitere Datenbanken * .....	990
16.1.5	Eclipse Data Tools Platform (DTP) zum Durchschauen von Datenbanken ....	991
<b>16.2</b>	<b>JDBC und Datenbanktreiber</b> .....	993
16.2.1	JDBC-Versionen * .....	994
<b>16.3</b>	<b>Eine Beispielabfrage</b> .....	995
16.3.1	Schritte zur Datenbankabfrage .....	995
16.3.2	Ein Client für die HSQLDB-Datenbank .....	995

<b>16.4</b>	<b>Mit Java an eine Datenbank andocken</b> .....	996
16.4.1	Der Treiber-Manager * .....	997
16.4.2	Den Treiber laden .....	997
16.4.3	Eine Aufzählung aller Treiber * .....	999
16.4.4	Log-Informationen * .....	999
16.4.5	Verbindung zur Datenbank auf- und abbauen .....	1000
<b>16.5</b>	<b>Datenbankabfragen</b> .....	1003
16.5.1	Abfragen über das Statement-Objekt .....	1003
16.5.2	Ergebnisse einer Abfrage in ResultSet .....	1005
16.5.3	Java und SQL-Datentypen .....	1007
16.5.4	Date, Time und Timestamp .....	1009
16.5.5	Unicode in der Spalte korrekt auslesen .....	1011
16.5.6	Eine SQL-NULL und wasNull() bei ResultSet .....	1012
16.5.7	Wie viele Zeilen hat ein ResultSet? * .....	1012
<b>16.6</b>	<b>Elemente einer Datenbank ändern</b> .....	1013
16.6.1	Einzelne INSERT-, UPDATE- oder DELETE-Anweisungen senden .....	1013
16.6.2	Aktualisierbares ResultSet .....	1014
16.6.3	Batch-Updates .....	1015
<b>16.7</b>	<b>Die Ausnahmen bei JDBC, SQLException und Unterklassen</b> .....	1016
16.7.1	JDBC-Fehlerbasisklasse SQLException .....	1016
16.7.2	SQLWarning .....	1017
<b>16.8</b>	<b>ResultSet und RowSet *</b> .....	1019
16.8.1	Die Schnittstelle RowSet .....	1019
16.8.2	Implementierungen von RowSet .....	1019
16.8.3	Der Typ CachedRowSet .....	1020
16.8.4	Der Typ WebRowSet .....	1021
<b>16.9</b>	<b>Vorbereitete Anweisungen (Prepared Statements)</b> .....	1023
16.9.1	PreparedStatement-Objekte vorbereiten .....	1024
16.9.2	Werte für die PreparedStatement-Platzhalter .....	1025
<b>16.10</b>	<b>Transaktionen</b> .....	1026
<b>16.11</b>	<b>Vorbereitete Datenbankverbindungen</b> .....	1027
16.11.1	DataSource .....	1027
16.11.2	Gepoolte Datenbankverbindungen .....	1030
<b>16.12</b>	<b>Zum Weiterlesen</b> .....	1031

## 17 Grafische Oberflächen mit Swing 1033

---

<b>17.1</b>	<b>AWT, JavaFoundation Classes und Swing</b> .....	1033
17.1.1	Das Abstract Window Toolkit (AWT) .....	1033
17.1.2	Java Foundation Classes (JFC) .....	1034
17.1.3	Was Swing von AWT-Komponenten unterscheidet .....	1037
<b>17.2</b>	<b>Mit NetBeans zur ersten Swing-Oberfläche</b> .....	1038
17.2.1	Projekt anlegen .....	1038
17.2.2	Eine GUI-Klasse hinzufügen .....	1039
17.2.3	Programm starten .....	1041
17.2.4	Grafische Oberfläche aufbauen .....	1042
17.2.5	Swing-Komponenten-Klassen .....	1044
17.2.6	Funktionalität geben .....	1046
<b>17.3</b>	<b>Aller Swing-Anfang – Fenster zur Welt</b> .....	1049
17.3.1	Eine Uhr, bei der die Zeit nie vergeht .....	1050
17.3.2	Swing-Fenster mit javax.swing.JFrame darstellen .....	1051
17.3.3	Mit add(...) auf den Container .....	1051
17.3.4	Fenster schließbar machen – setDefaultCloseOperation(int) .....	1051
17.3.5	Sichtbarkeit des Fensters .....	1052
17.3.6	Größe und Position des Fensters verändern .....	1053
17.3.7	Fenster- und Dialogdekoration, Transparenz * .....	1054
17.3.8	Die Klasse Toolkit * .....	1055
17.3.9	Zum Vergleich: AWT-Fenster darstellen * .....	1055
<b>17.4</b>	<b>Beschriftungen (JLabel)</b> .....	1057
17.4.1	Mehrzeiliger Text, HTML in der Darstellung .....	1059
<b>17.5</b>	<b>Icon und ImagemIcon für Bilder auf Swing-Komponenten</b> .....	1060
17.5.1	Die Klasse ImagemIcon .....	1060
<b>17.6</b>	<b>Es tut sich was – Ereignisse beim AWT</b> .....	1062
17.6.1	Die Ereignisquellen und Horcher (Listener) von Swing .....	1063
17.6.2	Listener implementieren .....	1064
17.6.3	Listener bei dem Ereignisauslöser anmelden/abmelden .....	1066
17.6.4	Adapterklassen nutzen .....	1068
17.6.5	Listener-Code in inneren Klassen und Lambda-Ausdrücken .....	1070
17.6.6	Aufrufen der Listener im AWT-Event-Thread .....	1072
17.6.7	Ereignisse, etwas genauer betrachtet * .....	1072
<b>17.7</b>	<b>Schaltflächen</b> .....	1076
17.7.1	Normale Schaltflächen (JButton) .....	1076
17.7.2	Der aufmerksame ActionListener .....	1078

17.7.3	Schaltflächen-Ereignisse vom Typ <code>ActionEvent</code> .....	1079
17.7.4	Basisklasse <code>AbstractButton</code> .....	1079
17.7.5	Wechselknopf ( <code>JToggleButton</code> ) .....	1081
<b>17.8</b>	<b>Textkomponenten</b> .....	<b>1081</b>
17.8.1	Text in einer Eingabezeile .....	1082
17.8.2	Die Oberklasse der Textkomponenten ( <code>JTextComponent</code> ) .....	1083
17.8.3	Geschützte Eingaben ( <code>JPasswordField</code> ) .....	1085
17.8.4	Validierende Eingabefelder ( <code>JFormattedTextField</code> ) .....	1086
17.8.5	Einfache mehrzeilige Textfelder ( <code>JTextArea</code> ) .....	1087
17.8.6	Editor-Klasse ( <code>JEditorPane</code> ) * .....	1090
<b>17.9</b>	<b>Swing Action</b> * .....	<b>1093</b>
<b>17.10</b>	<b>JComponent und Component als Basis aller Komponenten</b> .....	<b>1095</b>
17.10.1	Hinzufügen von Komponenten .....	1095
17.10.2	Tooltips (Kurzhinweise) .....	1096
17.10.3	Rahmen ( <code>Border</code> ) * .....	1096
17.10.4	Fokus und Navigation * .....	1099
17.10.5	Ereignisse jeder Komponente * .....	1100
17.10.6	Die Größe und Position einer Komponente * .....	1103
17.10.7	Komponenten-Ereignisse * .....	1104
17.10.8	UI-Delegate – der wahre Zeichner * .....	1105
17.10.9	Undurchsichtige (opaque) Komponente * .....	1108
17.10.10	Properties und Listener für Änderungen * .....	1108
<b>17.11</b>	<b>Container</b> .....	<b>1108</b>
17.11.1	Standard-Container ( <code>JPanel</code> ) .....	1109
17.11.2	Bereich mit automatischen Rollbalken ( <code>JScrollPane</code> ) .....	1109
17.11.3	Reiter ( <code>JTabbedPane</code> ) .....	1110
17.11.4	Teilungskomponente ( <code>JSplitPane</code> ) .....	1111
<b>17.12</b>	<b>Alles Auslegungssache – die Layoutmanager</b> .....	<b>1112</b>
17.12.1	Übersicht über Layoutmanager .....	1112
17.12.2	Zuweisen eines Layoutmanagers .....	1113
17.12.3	Im Fluss mit <code>FlowLayout</code> .....	1114
17.12.4	<code>BoxLayout</code> .....	1116
17.12.5	Mit <code>BorderLayout</code> in alle Himmelsrichtungen .....	1116
17.12.6	Rasteranordnung mit <code>GridLayout</code> .....	1119
17.12.7	Der <code>GridBagLayoutmanager</code> * .....	1121
17.12.8	<code>Null-Layout</code> * .....	1126
<b>17.13</b>	<b>Rollbalken und Schieberegler</b> .....	<b>1126</b>
17.13.1	Schieberegler ( <code>JSlider</code> ) .....	1126
17.13.2	Rollbalken ( <code>JScrollBar</code> ) * .....	1128

<b>17.14</b>	<b>Kontrollfelder, Optionsfelder, Kontrollfeldgruppen</b> .....	1131
17.14.1	Kontrollfelder (JCheckBox) .....	1132
17.14.2	ItemSelectable, ItemListener und das ItemEvent .....	1134
17.14.3	Sich gegenseitig ausschließende Optionen (JRadioButton) .....	1136
<b>17.15</b>	<b>Fortschritte bei Operationen überwachen *</b> .....	1137
17.15.1	Fortschrittsbalken (JProgressBar) .....	1137
17.15.2	Dialog mit Fortschrittsanzeige (ProgressMonitor) .....	1139
<b>17.16</b>	<b>Menüs und Symbolleisten</b> .....	1139
17.16.1	Die Menüleisten und die Einträge .....	1139
17.16.2	Menüeinträge definieren .....	1141
17.16.3	Einträge durch Action-Objekte beschreiben .....	1142
17.16.4	Mit der Tastatur – Mnemonics und Shortcut .....	1143
17.16.5	Der Tastatur-Shortcut (Accelerator) .....	1144
17.16.6	Tastenkürzel (Mnemonics) .....	1146
17.16.7	Symbolleisten alias Toolbars .....	1146
17.16.8	Popup-Menüs .....	1150
17.16.9	System-Tray nutzen * .....	1154
<b>17.17</b>	<b>Das Model-View-Controller-Konzept</b> .....	1155
<b>17.18</b>	<b>Auswahlmenüs, Listen und Spinner</b> .....	1157
17.18.1	Listen (JList) .....	1157
17.18.2	Auswahlmenü (JComboBox) .....	1164
17.18.3	Drehfeld (JSpinner) * .....	1170
17.18.4	Datumsauswahl .....	1171
<b>17.19</b>	<b>Tabellen (JTable)</b> .....	1172
17.19.1	Ein eigenes Tabellen-Model .....	1173
17.19.2	Basisklasse für eigene Modelle (AbstractTableModel) .....	1174
17.19.3	Ein vorgefertigtes Standardmodell (DefaultTableModel) .....	1178
17.19.4	Ein eigener Renderer für Tabellen .....	1179
17.19.5	Zell-Editoren .....	1182
17.19.6	Automatisches Sortieren und Filtern mit RowSorter * .....	1183
<b>17.20</b>	<b>Bäume (JTree)</b> .....	1187
17.20.1	JTree und sein TreeModel und TreeNode .....	1187
17.20.2	Selektionen bemerken .....	1189
17.20.3	Das TreeModel von JTree * .....	1189
<b>17.21</b>	<b>Dialoge und Window-Objekte</b> .....	1192
17.21.1	JWindow und JDialog .....	1192
17.21.2	Modal oder nichtmodal? .....	1193
17.21.3	Standarddialoge mit JOptionPane .....	1193
17.21.4	Der Dateiauswahldialog .....	1196



<b>17.22</b>	<b>Flexibles Java-Look-and-Feel</b> .....	1201
	17.22.1 UIManager .....	1201
<b>17.23</b>	<b>Swing-Komponenten neu erstellen oder verändern *</b> .....	1203
	17.23.1 Überlagerungen mit dem Swing-Komponenten-Dekorator JLayer .....	1204
<b>17.24</b>	<b>Die Zwischenablage (Clipboard)</b> .....	1206
	17.24.1 Clipboard-Objekte .....	1206
	17.24.2 Mit Transferable auf den Inhalt zugreifen .....	1206
	17.24.3 DataFlavor ist das Format der Daten in der Zwischenablage .....	1207
	17.24.4 Einfügungen in der Zwischenablage erkennen .....	1209
	17.24.5 Drag & Drop .....	1210
<b>17.25</b>	<b>AWT, Swing und die Threads</b> .....	1211
	17.25.1 Ereignisschlange (EventQueue) und AWT-Event-Thread .....	1211
	17.25.2 Swing ist nicht threadsicher .....	1212
	17.25.3 invokeLater(...) und invokeAndWait(...) .....	1214
	17.25.4 SwingWorker .....	1216
	17.25.5 Eigene Ereignisse in die Queue setzen * .....	1218
	17.25.6 Auf alle Ereignisse hören * .....	1219
<b>17.26</b>	<b>Barrierefreiheit mit der Java Accessibility API</b> .....	1219
<b>17.27</b>	<b>Zeitliches Ausführen mit dem javax.swing.Timer</b> .....	1220
<b>17.28</b>	<b>Zum Weiterlesen</b> .....	1221

## **18 Grafikprogrammierung** 1223

---

<b>18.1</b>	<b>Grundlegendes zum Zeichnen</b> .....	1223
	18.1.1 Die paint(Graphics)-Methode für das AWT-Frame .....	1223
	18.1.2 Die ereignisorientierte Programmierung ändert Fensterinhalte .....	1225
	18.1.3 Zeichnen von Inhalten auf ein JFrame .....	1226
	18.1.4 Auffordern zum Neuzeichnen mit repaint(...) .....	1228
	18.1.5 Java 2D-API .....	1228
<b>18.2</b>	<b>Einfache Zeichenmethoden</b> .....	1229
	18.2.1 Linien .....	1229
	18.2.2 Rechtecke .....	1230
	18.2.3 Ovale und Kreisbögen .....	1231
	18.2.4 Polygone und Polylines .....	1232
<b>18.3</b>	<b>Zeichenketten schreiben und Fonts</b> .....	1233
	18.3.1 Zeichenfolgen schreiben .....	1233

18.3.2	Die Font-Klasse .....	1234
18.3.3	Font-Metadaten durch FontMetrics * .....	1235
<b>18.4</b>	<b>Geometrische Objekte .....</b>	<b>1239</b>
18.4.1	Die Schnittstelle Shape .....	1240
18.4.2	Pfade * .....	1242
<b>18.5</b>	<b>Das Innere und Äußere einer Form .....</b>	<b>1242</b>
18.5.1	Farben und die Paint-Schnittstelle .....	1243
18.5.2	Farben mit der Klasse Color .....	1243
18.5.3	Composite und XOR * .....	1244
18.5.4	Dicke und Art der Linien von Formen bestimmen über Stroke * .....	1244
<b>18.6</b>	<b>Bilder .....</b>	<b>1249</b>
18.6.1	Eine Übersicht über die Bilder-Bibliotheken .....	1249
18.6.2	Bilder mit ImageIO lesen .....	1250
18.6.3	Ein Bild zeichnen .....	1251
18.6.4	Splash-Screen * .....	1254
18.6.5	Bilder skalieren * .....	1255
18.6.6	Schreiben mit ImageIO .....	1257
18.6.7	Asynchrones Laden mit getImage(...) und dem MediaTracker * .....	1261
<b>18.7</b>	<b>Weitere Eigenschaften von Graphics * .....</b>	<b>1263</b>
18.7.1	Eine Kopie von Graphics erstellen .....	1263
18.7.2	Koordinatensystem verschieben .....	1264
18.7.3	Beschnitt (Clipping) .....	1264
18.7.4	Zeichenhinweise durch RenderingHints .....	1268
18.7.5	Transformationen mit einem AffineTransform-Objekt .....	1269
<b>18.8</b>	<b>Drucken * .....</b>	<b>1271</b>
18.8.1	Drucken der Inhalte .....	1271
18.8.2	Bekannte Drucker .....	1273
<b>18.9</b>	<b>Benutzerinteraktionen automatisieren, Robot und Screenshots * .....</b>	<b>1274</b>
18.9.1	Der Roboter .....	1274
18.9.2	Automatisch in die Tasten hauen .....	1275
18.9.3	Automatisierte Mausoperationen .....	1275
18.9.4	Methoden zur Zeitsteuerung .....	1276
18.9.5	Bildschirmabzüge (Screenshots) .....	1276
18.9.6	Funktionsweise und Beschränkungen .....	1278
18.9.7	MouseInfo und PointerInfo .....	1278
<b>18.10</b>	<b>Zum Weiterlesen .....</b>	<b>1280</b>

<b>19</b>	<b>JavaFX</b>	1281
<hr/>		
<b>19.1</b>	<b>Das erste Programm mit JavaFX</b>	1281
<b>19.2</b>	<b>Zentrale Typen in JavaFX</b>	1284
19.2.1	Szenegraph-Knoten und Container-Typen	1284
19.2.2	Datenstrukturen	1285
<b>19.3</b>	<b>JavaFX-Komponenten und Layout-Container-Klassen</b>	1286
19.3.1	Überblick über die Komponenten	1286
19.3.2	Listener/Handler zur Ereignisbeobachtung	1287
19.3.3	Panels mit speziellen Layouts	1288
<b>19.4</b>	<b>Webbrowser</b>	1290
<b>19.5</b>	<b>Geometrische Objekte</b>	1291
19.5.1	Linien und Rechtecke	1292
19.5.2	Kreise, Ellipsen, Kreisförmiges	1294
19.5.3	Es werde kurvig – quadratische und kubische Splines	1295
19.5.4	Pfade *	1297
19.5.5	Polygone und Polylines	1300
19.5.6	Beschriftungen, Texte, Fonts	1300
19.5.7	Die Oberklasse Shape	1302
<b>19.6</b>	<b>Füllart von Formen</b>	1304
19.6.1	Farben mit der Klasse Color	1304
<b>19.7</b>	<b>Grafiken</b>	1308
19.7.1	Klasse Image	1308
19.7.2	ImageView	1309
19.7.3	Programm-Icon/Fenster-Icon setzen	1310
19.7.4	Zugriff auf die Pixel und neue Pixel-Bilder *	1310
<b>19.8</b>	<b>Deklarative Oberflächen mit FXML</b>	1313
<b>19.9</b>	<b>Diagramme (Charts)</b>	1315
19.9.1	Kuchendiagramm	1315
19.9.2	Balkendiagramm	1317
<b>19.10</b>	<b>Animationen</b>	1319
19.10.1	FadeTransition	1320
19.10.2	ScaleTransition	1320
19.10.3	Transitionen parallel oder sequenziell durchführen	1321
<b>19.11</b>	<b>Medien abspielen</b>	1322
<b>19.12</b>	<b>Java 3D</b>	1323
<b>19.13</b>	<b>Das Geometry-Paket *</b>	1323

<b>19.14</b>	<b>JavaFX-Scene in Swing-Applikationen einbetten</b> .....	1324
<b>19.15</b>	<b>Zum Weiterlesen</b> .....	1326

---

## **20 Sicherheitskonzepte** 1327

---

<b>20.1</b>	<b>Zentrale Elemente der Java-Sicherheit</b> .....	1327
20.1.1	Sichere Java Virtual Machine .....	1327
20.1.2	Der Sandkasten (Sandbox) .....	1327
20.1.3	Security-API der Java SE .....	1328
20.1.4	Cryptographic Service Providers .....	1329
<b>20.2</b>	<b>Sicherheitsmanager (Security-Manager)</b> .....	1330
20.2.1	Der Sicherheitsmanager bei Applets .....	1330
20.2.2	Sicherheitsmanager aktivieren .....	1331
20.2.3	Rechte durch Policy-Dateien vergeben .....	1334
20.2.4	Erstellen von Rechtedateien mit dem grafischen Policy-Tool .....	1336
20.2.5	Kritik an den Policies .....	1337
<b>20.3</b>	<b>Signierung</b> .....	1338
20.3.1	Warum signieren? .....	1339
20.3.2	Digitale Ausweise und die Zertifizierungsstelle .....	1339
20.3.3	Mit keytool Schlüssel erzeugen .....	1339
20.3.4	Signieren mit jarsigner .....	1340
<b>20.4</b>	<b>Kryptografische Hashfunktion</b> .....	1341
20.4.1	Die MDx-Reihe .....	1342
20.4.2	Secure Hash Algorithm (SHA) .....	1342
20.4.3	Mit der Security-API einen Fingerabdruck berechnen .....	1343
20.4.4	Die Klasse MessageDigest .....	1343
<b>20.5</b>	<b>Verschlüsseln von Daten(-strömen) *</b> .....	1346
20.5.1	Den Schlüssel, bitte .....	1346
20.5.2	Verschlüsseln mit Cipher .....	1347
20.5.3	Verschlüsseln von Datenströmen .....	1348
<b>20.6</b>	<b>Zum Weiterlesen</b> .....	1349

---

## **21 Dynamische Übersetzung, Skriptsprachen, JShell** 1351

---

<b>21.1</b>	<b>Codegenerierung</b> .....	1352
21.1.1	Generierung von Java-Quellcode .....	1353

21.1.2	Codetransformationen .....	1355
21.1.3	Erstellen von Java-Bytecode .....	1355
<b>21.2</b>	<b>Programme mit der Java Compiler API übersetzen .....</b>	<b>1356</b>
21.2.1	Java Compiler API .....	1356
21.2.2	Fehlerdiagnose .....	1359
21.2.3	Eine im String angegebene Kompilationseinheit übersetzen .....	1361
21.2.4	Wenn Quelle und Ziel der Speicher sind .....	1362
<b>21.3</b>	<b>Ausführen von Skripten .....</b>	<b>1366</b>
21.3.1	Java-Programme mit JavaScript schreiben .....	1366
21.3.2	Kommandozeilenprogramme jrunscript und jjs .....	1367
21.3.3	javax.script-API .....	1367
21.3.4	JavaScript-Programme ausführen .....	1368
21.3.5	Alternative Sprachen für die JVM .....	1369
21.3.6	Von den Schwierigkeiten, dynamische Programmiersprachen auf die JVM zu bringen * .....	1371
<b>21.4</b>	<b>JShell, die interaktive REPL-Shell .....</b>	<b>1375</b>
21.4.1	JShell API .....	1380
<b>21.5</b>	<b>Zum Weiterlesen .....</b>	<b>1381</b>

## **22 Java Native Interface (JNI) 1383**

---

<b>22.1</b>	<b>Java Native Interface und Invocation-API .....</b>	<b>1383</b>
<b>22.2</b>	<b>Eine C-Funktion in ein Java-Programm einbinden .....</b>	<b>1384</b>
22.2.1	Den Java-Code schreiben .....	1384
<b>22.3</b>	<b>Dynamische Bibliotheken erzeugen .....</b>	<b>1386</b>
22.3.1	Die Header-Datei erzeugen .....	1386
22.3.2	Implementierung der Funktion in C .....	1388
22.3.3	Die C-Programme übersetzen und die dynamische Bibliothek erzeugen .....	1389
22.3.4	Abhängige Bibliotheken entdecken .....	1390
22.3.5	Nativ die String-Länge ermitteln .....	1390
<b>22.4</b>	<b>Erweiterte JNI-Eigenschaften .....</b>	<b>1392</b>
22.4.1	Klassendefinitionen .....	1392
22.4.2	Zugriff auf Attribute .....	1392
22.4.3	Methoden aufrufen .....	1395
22.4.4	Threads und Synchronisation .....	1396
22.4.5	@Native Markierungen * .....	1397

<b>22.5 Einfache Anbindung von existierenden Bibliotheken</b> .....	1397
22.5.1 JNA (Java Native Access) .....	1397
22.5.2 BridJ .....	1398
22.5.3 Generieren von JNI-Wrappern aus C++-Klassen und C-Headern .....	1399
22.5.4 COM-Schnittstellen anzapfen .....	1399
<b>22.6 Invocation-API</b> .....	1399
<b>22.7 Zum Weiterlesen</b> .....	1400

## **23 Dienstprogramme für die Java-Umgebung** 1401

---

<b>23.1 Programme des JDK</b> .....	1401
<b>23.2 Monitoringprogramme vom JDK</b> .....	1401
23.2.1 jps .....	1401
23.2.2 jstat .....	1402
23.2.3 jmap .....	1402
23.2.4 jstack .....	1403
23.2.5 jcmd .....	1405
23.2.6 jhsdb .....	1407
23.2.7 VisualVM .....	1407
<b>23.3 Ant</b> .....	1411
23.3.1 Bezug und Installation von Ant .....	1412
23.3.2 Das Build-Skript build.xml .....	1412
23.3.3 Build den Build .....	1413
23.3.4 Properties .....	1413
23.3.5 Externe und vordefinierte Properties .....	1415
23.3.6 Weitere Ant-Tasks .....	1416
<b>23.4 Disassembler, Decompiler und Obfuscator</b> .....	1417
23.4.1 Der Disassembler javap* .....	1417
23.4.2 Decompiler .....	1422
23.4.3 Obfuscatoren .....	1424
<b>23.5 Weitere Dienstprogramme</b> .....	1426
23.5.1 Java-Programme als Systemdienst ausführen .....	1426
<b>23.6 Zum Weiterlesen</b> .....	1426
 Index .....	 1427