

Christian Ullenboom ,192'.

Java ist auch eine Insel

Programmieren mit der Java Platform, Standard Edition 6

• HOCHSCHULE
LIECHTENSTEIN
Bibliothek

Galileo Press

Vorwort.....39

1 Java ist auch eine Sprache.....

1.1 Der erste Kontakt.....55

1.2 Historischer Hintergrund.....55

1.3 Eigenschaften von Java.....57

1.3.1 Bytecode und die virtuelle Maschine.....57

⇒y ... 1.3.2 Objektorientierung in Java.....58

1.3.3 Java-Security-Modell.....59

1.3.4 Zeiger und Referenzen.....59

1.3.5 Bring den Müll raus, Garbage-Collector!.....60

01* . . . 1.3.6 Ausnahmebehandlung.....61

1.3.7 Kein Präprozessor für Textersetzungen.....61

1.3.8 Keine überladenen Operatoren.....62

1.3.9 Java als Sprache, Laufzeitumgebung und Bibliothek.....62

Hf _ 1.3.10 Wofür sich Java nicht eignet.....63

1.3.11 Java im Vergleich zu anderen Sprachen.....64

1.3.12 Java ist Open Source.....64

£g. 1.4 Die Rolle von Java im Web.....65

1.4.1 Vollwertige Applikationen statt Applets.....65

1.5 Die Java Platform Standard Edition (Java SE).....66

1.5.1 JDK undJRE.....66

1.5.2 Java-Versionen.....66

1.5.3 Java für die Kleinen.....68

1.5.4 Java für die Großen.....68

1.5.5 Installationsanleitung für Java SE.....68

1.6 Das erste Programm compilieren und testen.....70

1.6.1 Ein Quadratzahlen-Programm.....70

1.6.2 Der Compilerlauf.....71

1.6.3 Die Laufzeitumgebung.....72

1.6.4 Häufige Compiler- und Interpreterprobleme.....72

1.7 Entwicklungsumgebungen im Allgemeinen.....73

1.7.1 Die Entwicklungsumgebung Eclipse.....73

1.7.2 NetBeans von Sun.....74

1.7.3 Ein Wort zu Microsoft, Java und zu J++.....74

1.8 Eclipse im Speziellen.....75

1.8.1 Eclipse starten.....75

1.8.2 Das erste Projekt anlegen.....76

1.8.3 Eine Klasse hinzufügen.....78

1.8.4	Übersetzen und Ausführen.....	79
1.8.5	JDK statt JRE.....	80
1.8.6	Start eines Programms ohne Speicheraufforderung.....	80
1.8.7	Projekt einfügen oder Workspace für die Aufgaben wechseln	81
1.8.8	Plugins für Eclipse.....	82
1.8.9	Eclipse Web Tools Platform (WTP).....	82
1.9	Zum Weiterlesen.....	83

2 Sprachbeschreibung.....

2.1	Elemente der Programmiersprache Java.....	85
2.1.1	Textkodierung durch Unicode-Zeichen.....	85
2.1.2	Literale.....	88
2.1.3	Bezeichner.....	88
2.1.4	Reservierte Schlüsselwörter.....	89
2.1.5	Token.....	90
2.1.6	Kommentare.....	91
2.2	Anweisungen formen Programme.....	92
2.2.1	Anweisungen.....	92
2.2.2	Eine Klasse bildet den Rahmen.....	93
2.2.3	Die Reise beginnt am mainO.....	94
2.2.4	Programme übersetzen und starten.....	95
2.2.5	Funktionsaufrufe als Ausdrücke und Anweisungen.....	95
2.2.6	printO, printInO und printfö für Bildschirmausgaben.....	95
2.2.7	Modifizierer.....	98
2.2.8	Anweisungen und Blöcke.....	99
2.3	Datentypen.....	100
2.3.1	Primitive Datentypen im Überblick.....	100
2.3.2	Wahrheitswerte.....	102
2.3.3	Variablendeklarationen.....	102
2.3.4	Ganzzahlige Datentypen.....	106
2.3.5	Die Fließkommazahlen float und double.....	108
2.3.6	Alphanumerische Zeichen.....	109
2.4	Ausdrücke, Operanden und Operatoren.....	110
2.4.1	Zuweisungsoperator.....	111
2.4.2	Arithmetische Operatoren.....	112
2.4.3	Unäres Minus und Plus.....	115
2.4.4	Zuweisung mit Operation.....	115
2.4.5	Präfix- oder Postfix-Inkrement und -Dekrement.....	116
2.4.6	Die relationalen Operatoren und die Gleichheitsoperatoren.....	118
2.4.7	Logische Operatoren Und, Oder, Xor, Nicht.....	119
2.4.8	Rang der Operatoren in der Auswertungsreihenfolge.....	120
2.4.9	Die Typanpassung (das Casting).....	122

..*	2.4.10	Überladenes Plus für Strings.....	127
	2.4.11	Was C(++)-Programmierer vermissen könnten.....	128
2.5		Bedingte Anweisungen oder Fallunterscheidungen.....	128
	2.5.1	Die if-Anweisung.....	129
f8F	2.5.2	Die Alternative mit einer if/else-Anweisung wählen.....	130
	2.5.3	Die switch-Anweisung bietet die Alternative.....	133
2.6		Schleifen.....	136
	2.6.1	Die while-Schleife.....	136
	2.6.2	Schleifenbedingungen und Vergleiche mit ==	137
	2.6.3	Die do-while-Schleife.....	138
	2.6.4	Die for-Schleife.....	140
	2.6.5	Ausbruch planen mit break und if/iedereinstieg mit continue	143
	2.6.6	break und continue mit Sprungmarken.....	146
2.7		Methoden einer Klasse.....	147
	2.7.1	Bestandteil einer Funktion.....	148
	2.7.2	Beschreibungen in der Java-API.....	148
	2.7.3	Aufruf einer Methode.....	150
	2.7.4	Methoden ohne Parameter.....	150
	2.7.5	Statische Funktionen (Klassenmethoden).....	151
	2.7.6	Parameter, Argument und Wertübergabe.....	152
	2.7.7	Methoden vorzeitig mit return beenden.....	153
	2.7.8	Nicht erreichbarer Quellcode bei Funktionen.....	154
	2.7.9	Rückgabewerte.....	154
	2.7.10	Methoden überladen.....	158
	2.7.11	Vorgegebener Wert für nicht aufgeführte Argumente.....	160
	2.7.12	Finale lokale Variablen.....	160
	2.7.13	Rekursive Funktionen.....	162
	2.7.14	Die Türme von Hanoi.....	164
2.8		Weitere Operatoren.....	166
	2.8.1	Bits und Bytes.....	166
	2.8.2	Operationen auf Bit-Ebene.....	167
	2.8.3	Die Verschiebeoperatoren.....	169
	2.8.4	Ein Bit setzen, löschen, umdrehen und testen.....	171
	2.8.5	Bit-Funktionen der Integer- und Long-Klasse.....	171
	2.8.6	Der Bedingungsoperator.....	172
2.9		Einfache Benutzereingaben.....	174
2.10		Zum Weiterlesen.....	175
3 Klassen und Objekte.....			177
3.1		Objektorientierte Programmierung.....	177
	3.1.1	Warum überhaupt OOP?.....	177
	3.1.2	Wiederverwertbarkeit.....	178

Inhalt

3.2	Eigenschaften einer Klasse.....	179
3.2.1	Die Klasse Point.....	179
3.3	Die UML (Unified Modeling Language).....	180
...	3.3.1 Hintergrund und Geschichte zur UML.....	180
	3.3.2 Wichtige Diagrammtypen der UML.....	181
3.4	Neue Objekte erzeugen.....	181
3.4.1	Anlegen eines Exemplars einer Klasse mit dem new-Operator....	182
3.4.2	Deklarieren von Referenzvariablen.....	183
3.4.3	Zugriff auf Variablen und Methoden mit dem ».«.....	183
3.4.4	Konstruktoren nutzen.....	186
3.4.5	Die API-Dokumentation.....	186
3.5	Import und Pakete.....	187
3.6	Mit Referenzen arbeiten.....	188
3.6.1	Die null-Referenz.....	188
..	3.6.2 Zuweisungen bei Referenzen.....	190
	3.6.3 Funktionen mit nicht-primitiven Parametern.....	191
3.7	Identität und Gleichheit.....	193
3.7.1	Identität von Objekten.....	193
3.7.2	Gleichheit und die Methode equalsO.....	193
3.8	Wrapper-Klassen und Autoboxing.....	195
3.8.1	Die Basisklasse Number für numerische Wrapper-Objekte.....	197
...	3.8.2 Die Klasse Integer.....	198
„	3.8.3 Unterschiedliche Ausgabeformate.....	199
	3.8.4 Autoboxing: Boxing und Unboxing.....	200
	3.8.5 Die Boolean-Klasse.....	202
	3.8.6 Die Klassen Double und Float für Fließkommazahlen.....	203
3.9	Arrays.....	204
3.9.1	Deklaration von Arrays.....	205
.	3.9.2 Arrays mit Inhalt.....	205
	3.9.3 Die Länge eines Arrays über das Attribut length.....	206
	3.9.4 Zugriff auf die Elemente über den Index.....	206
	3.9.5 Array-Objekte erzeugen.....	207
	3.9.6 Fehler bei Arrays.....	209
	3.9.7 Vorinitialisierte Arrays.....	210
	3.9.8 Die erweiterte for-Schleife.....	211
	3.9.9 Arrays mit nicht-primitiven Elementen.....	212
	3.9.10 Mehrdimensionale Arrays.....	213
	3.9.11 Die Wahrheit über die Array-Initialisierung.....	215
	3.9.12 Mehrere Rückgabewerte.....	216
	3.9.13 Methode mit variabler Argumentanzahl (Vararg).....	217
	3.9.14 Klonen kann sich lohnen -Arrays vermehren.....	218
	3.9.15 Feldinhalte kopieren.....	219
	3.9.16 Die Klasse Arrays zum Vergleichen, Füllen und Suchen.....	220

3.10	Der Einstiegspunkt für das Laufzeitsystem mainO.....	227
3.10.1	Kommandozeilen-Argumente verarbeiten.....	227
3.10.2	Der Rückgabewert von mainO und System.exitO.....	228
3.11	Eigene Pakete schnüren.....	228
3.11.1	Die package-Anweisung.....	229
3.11.2	Importieren von Klassen mit import.....	229
3.11.3	Hierarchische Strukturen und das Default-Package.....	229
3.11.4	Paketnamen.....	230
3.11.5	Klassen mit gleichen Namen in unterschiedlichen Paketen.....	231
3.11.6	Statisches Import.....	231
3.11.7	Eine Verzeichnisstruktur für eigene Projekte.....	232
3.12	Zum Weiterlesen.....-*	232
4.1	Einzelne Zeichen mit der Character-Klasse behandeln.....	233
4.2	Strings und deren Anwendung.....	234
4.2.1	String-Literale als String-Objekte für konstante Zeichenketten ...	236
4.2.2	String-Länge und Test auf Leerstring.....	238
4.2.3	Nach enthaltenen Zeichen und Zeichenfolgen suchen.....	238
4.2.4	Gut, dass wir verglichen haben.....	240
4.2.5	String-Teile extrahieren.....	242
4.2.6	Strings anhängen, Groß-/Kleinschreibung und Leerraum.....	244
4.2.7	Suchen und ersetzen.....	246
4.2.8	String-Objekte mit Konstruktoren neu anlegen.....	248
4.3	Konvertieren zwischen Primitiven und Strings.....	250
4.3.1	Unterschiedliche Typen in Zeichenketten konvertieren.....	250
4.3.2	String in primitives Element konvertieren.....	251
4.4	Veränderbare Zeichenketten mit StringBuffer/StringBuilder.....	252
4.4.1	Anlegen von StringBuffer/StringBuilder-Objekten.....	252
4.4.2	Die Länge eines StringBufferZ-Builder-Objekts.....	253
4.4.3	Daten anhängen.....	254
4.4.4	Zeichenfolgen) setzen, erfragen, löschen und umdrehen.....	255
4.4.5	Vergleichen von String/StringBuffer/StringBuilder.....	256
4.4.6	hashCodeO bei StringBuffer/StringBuilder.....	257
4.5	Sprachabhängiges Vergleichen und Normalisierung.....	257
4.5.1	Die Klasse Collator.....	258
4.5.2	Effiziente interne Speicherung für die Sortierung.....	260
4.5.3	Normalisierung.....	262
4.6	Reguläre Ausdrücke.....	262
4.6.1	Die Klassen Pattern und Matcher.....	262
4.6.2	Mit MatchResult alle Ergebnisse einsammeln.....	264

4.7	Zerlegen von Zeichenketten.....	266
4.7.1	Splitten von Zeichenketten mit splitO.....	266
4.7.2	splitO in Pattern.....	267
4.7.3	Die Klasse Scanner.....	268
4.7.4	StringTokenizer.....	273
4.7.5	Breaklterator als Zeichen-, Wort-, Zeilen- und Satztrenner.	275
4.8	Zeichenkodierungen und Base64.....	278
4.8.1	Über die Klasse String Kodierungen vornehmen.....	278
4.8.2	Konvertieren mit OutputStreamWriter-Klassen.....	279
4.8.3	Das Paket java.nio.charset.....	279
4.8.4	Base64-Kodierung.....	280
4.9	Formatieren von Ausgaben.....	281
4.9.1	Formatieren mit formatO aus String.....	281
4.9.2	Die Format-Klassen im Überblick.....	284
4.9.3	Zahlen, Prozente und Währungen mit NumberFormat und DecimalFormat formatieren.....	285
4.9.4	Ausgaben mit MessageFormat formatieren.....	287
4.10	Zum Weiterlesen.....	289

Mathematisches.....291

5.1	Repräsentation <i>ganzer</i> Zahlen - das Zweierkomplement.....	291
5.2	Fließkommaarithmetik in Java.....	291
5.2.1	Mantisse und Exponent.....	292
5.2.2	Spezialwerte Unendlich, Null, NaN.....	293
5.3	Wertebereich eines Typs und Überlaufkontrolle.....	295
5.3.1	Behandlung des Überlaufs.....	296
5.4	Die Eigenschaften der Klasse Math.....	297
5.4.1	Attribute.....	297
5.4.2	Absolutwerte und Maximum/Minimum.....	298
5.4.3	Winkelfunktionen.....	299
5.4.4	Runden von Werten.....	300
5.4.5	Wurzel und Exponentialfunktionen.....	302
5.4.6	Der Logarithmus.....	303
5.4.7	Rest der ganzzahligen Division.....	303
5.4.8	Zufallszahlen.....	304
5.5	Mathe bitte strikt.....	305
5.5.1	Strikt Fließkomma mit strictfp.....	305
5.5.2	Die Klassen Math und StrictMath.....	305
5.6	Die Random-Klasse.....	306
5.6.1	Objekte aufbauen und der Seed.....	306
5.6.2	Zufallszahlen erzeugen.....	307
5.6.3	Pseudo-Zufallszahlen in der Normalverteilung.....	307

- 5.7 Große Zahlen.....308
 - 5.7.1 Die Klasse BigInteger.....308
 - 5.7.2 Funktionen von BigInteger..... \wedge ,... \wedge ,|A,t,... 310
 - 5.7.3 Ganz lange Fakultäten.....312
 - 5.7.4 Große Fließkommazahlen mit BigDecimal.....313
 - 5.7.5 Mit MathContext komfortabel die Rechengenauigkeit setzen 315
- 5.8 Zum Weiterlesen.....317

6 Eigene Klassen schreiben.....319

- 6.1 Eigene Klassen mit Eigenschaften deklarieren.....319
 - 6.1.1 Attribute deklarieren.....319
 - 6.1.2 Methoden deklarieren.....320
 - 6.1.3 Die this-Referenz.....322
- 6.2 Privatsphäre und Sichtbarkeit.....324
 - 6.2.1 Für die Öffentlichkeit: public.....324
 - 6.2.2 Paketsichtbar.....324
 - 6.2.3 Kein Public Viewing - Passwörter sind privat.....325
 - 6.2.4 Wieso nicht freie Methoden und Variablen für alle?.....327
 - 6.2.5 Privat ist nicht ganz privat: Es kommt darauf an, wer's sieht 327
 - 6.2.6 Zugriffsmethoden für Attribute deklarieren.....328
 - 6.2.7 Setter und Getter nach der JavaBeans-Spezifikation.....329
- 6.3 Statische Methoden und statische Attribute.....330
 - 6.3.1 Warum statische Eigenschaften sinnvoll sind.....331
 - 6.3.2 Statische Eigenschaften mitstatic.....331
 - 6.3.3 Statische Eigenschaften über Referenzen nutzen?.....332
 - 6.3.4 Warum die Groß- und Kleinschreibung wichtig ist.....333
 - 6.3.5 Statische Eigenschaften und Objekteigenschaften.....333
 - 6.3.6 Statische Variablen zum Datenaustausch.....334
 - 6.3.7 Statische Blöcke als Klasseninitialisierer.....335
- 6.4 Konstanten und Aufzählungen.....336
 - 6.4.1 Konstanten über öffentliche statische finale Variablen.....337
 - 6.4.2 Eincompilierte Belegungen der Klassenvariablen.....337
 - 6.4.3 Typ(un)sicherere Aufzählungen.....338
 - 6.4.4 Aufzählungen mit enum.....339
- 6.5 Objekte anlegen und zerstören.....342
 - 6.5.1 Konstruktoren schreiben.....342
 - 6.5.2 Der Standard-Konstruktor.....343
 - 6.5.3 Parametrisierte und überladene Konstruktoren.....344
 - 6.5.4 Konstruktor nimmt ein Objekt vom eigenen Typ an
(Copy-Konstruktor).....346
 - 6.5.5 Einen anderen Konstruktor der gleichen Klasse aufrufen.....347
 - 6.5.6 Initialisierung der Objekt- und Klassenvariablen.....350

'''•'	6.5.7	Finale Werte im Konstruktor und in statischen Blöcken setzen ...	352
	6.5.8	Exemplarinitialisierer (Instanzinitialisierer).....	353
•	6.5.9	Ihr fehlt uns nicht-der Garbage-Collector.....	355
-	6.5.10	Implizit erzeugte String-Objekte.....	356
	6.5.11	Private Konstruktoren, Utility-Klassen, Singleton, Fabriken.....	357
6.6		Assoziationen zwischen Objekten.....	359
	6.6.1	Unidirektionale 1:1-Beziehung.....	360
	6.6.2	Bidirektionale 1:1-Beziehungen.....	360
	6.6.3	Unidirektionale 1:n-Beziehung.....	361
6.7		Vererbung.....	363
	6.7.1	Vererbung in Java.....*».....	363
	6.7.2	Spielobjekte modelliert.....	364
	6.7.3	Einfach- und Mehrfachvererbung.....	365
	6.7.4	Sichtbarkeit protected.....	366
	6.7.5	Konstruktoren in der Vererbung und super.....	367
	6.7.6	Automatische und explizite Typanpassung.....	371
	6.7.7	Das Substitutionsprinzip.....	373
	6.7.8	Typen mit dem binären Operator instanceof testen.....	375
	6.7.9	Methoden überschreiben.....	376
	6.7.10	Mit super an die Eltern.....	378
	6.7.11	Kovariante Rückgabetypen.....	380
	6.7.12	Array-Typen und Kovarianz.....	381
	6.7.13	Zusammenfassung zur Sichtbarkeit.....	382
6.8		Dynamisches Binden.....	383
	6.8.1	Unpolymorph bei privaten, statischen und finalen Methoden	386
	6.8.2	Polymorphie bei Konstruktoraufrufen.....	387
	6.8.3	Finale Klassen.....	389
	6.8.4	Nicht überschreib bare (finale) Methoden.....	390
6.9		Abstrakte Klassen und abstrakte Methoden.....	391
	6.9.1	Abstrakte Klassen.....	391
	6.9.2	Abstrakte Methoden.....	392
6.10		Schnittstellen.....	395
	6.10.1	Deklarieren von Schnittstellen.....	395
	6.10.2	Implementieren von Schnittstellen.....	396
	6.10.3	Markierungsschnittstellen.....	398
	6.10.4	Ein Polymorphie-Beispiel mit Schnittstellen ...-.....	398
	6.10.5	Die Mehrfachvererbung bei Schnittstellen.....	399
	6.10.6	Keine Kollisionsgefahr bei Mehrfachvererbung.....	402
	6.10.7	Erweitern von Interfaces-Subinterfaces.....	403
	6.10.8	Vererbte Konstanten bei Schnittstellen.....	404
	6.10.9	Schnittstellenmethoden, die nicht implementiert werden müssen.....	406
	6.10.10	Abstrakte Klassen und Schnittstellen im Vergleich.....	406

6.11	Geschachtelte (innere) Klassen, Schnittstellen, Aufzählungen.....	407
6.11.1	Statische innere Klassen und Schnittstellen.....	408
6.11.2	Mitglieds- oder Elementklassen.....	409
6.11.3	Lokale Klassen.....	413
6.11.4	Anonyme innere Klassen.....	413
6.11.5	this und Vererbung.....	416
6.12	Generische Datentypen.....	418
6.12.1	Einfache Klassenschablonen.....	419
6.12.2	Einfache Methodenschablonen.....	420
6.12.3	Umsetzen der Generics, Typlöschung und Raw-Types.....	420
6.12.4	Einschränken der Typen.....	422
6.12.5	Generics und Vererbung, Invarianz.....	424
6.12.6	Wildcards.....	425
6.13	Die Spezial-Oberklasse Enum.....	426
6.13.1	Methoden auf Enum-Objekten.....	426
6.13.2	enum mit eigenen Konstruktoren und Methoden.....	428
6.14	Dokumentationskommentare mit JavaDoc.....	430
6.14.1	Einen Dokumentationskommentar setzen.....	431
6.14.2	Mit javadoc eine Dokumentation erstellen.....	433
6.14.3	HTML-Tags in Dokumentationskommentaren.....	433
6.14.4	Generierte Dateien.....	433
6.14.5	Dokumentationskommentare im Überblick.....	434
6.14.6	JavaDoc und Doclets.....	435
6.14.7	Veraltete (deprecated) Klassen, Konstruktoren und Methoden	436

7 Angewandte Objektorientierung.....439

7.1	Schnittstellen in der Anwendung.....	439
7.1.1	CharSequence als Beispiel einer Schnittstelle.....	439
7.1.2	Die Schnittstelle Iterable.....	441
7.1.3	Funktionszeiger.....	443
7.1.4	Implementierung einer verketteten Liste.....	445
7.2	Design-Pattern (Entwurfsmuster).....	446
7.2.1	Design-Pattern.....	447
7.2.2	Das Beobachter-Pattern (Observer/Observable).....	448
7.2.3	Ereignisse über Listener.....	451
7.2.4	Multicast und Unicast.....	454
7.3	JavaBean ..*\$&&&&•.....•.....•.....	455
7.3.1	Properties (Eigenschaften).....	455
7.3.2	Einfache Eigenschaften.....	456
7.3.3	Indizierte Eigenschaften.....	456
7.3.4	Gebundene Eigenschaften.....	457
7.3.5	Veto-Eigenschaften - dagegen!.....	459

Exceptions

8.1	Problembereiche einzäunen.....	465
8.1.1	Exceptions in Java mit try und catch	465
8.1.2	Eine Datei mit RandomAccessFile auslesen.....	466
8.1.3	Ablauf einer Ausnahmesituation.....	468
8.1.4	Wiederholung abgebrochener Bereiche.....	468
8.1.5	throws im Methodenkopf angeben.....	469
8.1.6	Abschlussbehandlung mit finally	472
8.1.7	Nicht erreichbare catch-Klauseln.....	475
8.2	Die Klassenhierarchie der Fehler.....	476
8.2.1	Die Exception-Hierarchie.....	477
8.2.2	Oberausnahmen auffangen.....	477
8.2.3	Alles geht als Exception durch	478
8.2.4	RuntimeException muss nicht aufgefangen werden	479
8.2.5	Harte Fehler: Error	480
8.3	Auslösen eigener Exceptions.....	481
8.3.1	Mit throw Ausnahmen auslösen.....	481
8.3.2	Neue Exception-Klassen deklarieren.....	483
8.3.3	Abfangen und Weiterleiten.....	484
8.3.4	Geschachtelte Ausnahmen.....	485
8.3.5	Rückgabewerte bei ausgelösten Ausnahmen.....	486
8.4	Der Stack-Trace	487
8.4.1	Stack-Trace erfragen.....	487
8.5	Assertions.....	488
8.5.1	Assertions in eigenen Programmen nutzen.....	489
8.5.2	Assertions aktivieren.....	490

9 Die Funktionsbibliothek.....491

9.1	Die Java-Klassenphilosophie.....	491
9.1.1	Übersicht über die Pakete der Standardbibliothek	491
9.2	Object ist die Mutter aller Oberklassen.....	498
9.2.1	Klassenobjekte.....	498
9.2.2	Objektidentifikation mit toStringO.....	498
9.2.3	Objektgleichheit mit equalsO und Identität.....	500
9.2.4	Klonen eines Objekts mitcloneO.....	504
9.2.5	Hashcodes über hashCodeO liefern.....	507
9.2.6	Aufräumen mit finalizeO.....	510
9.2.7	Synchronisation.....	512
9.3	Klassenlader (Class Loader).....	512
9.3.1	Woher die kleinen Klassen kommen	513
9.3.2	Setzen des Klassenpfades.....	514

9.3.3	Die wichtigsten drei Typen von Klassenladern.....	515
9.3.4	Derjava.lang.ClassLoader.....	515
9.3.5	Hot Deployment mit dem URL-ClassLoader.....	517
9.3.6	Dasjre/lib/endorsed-Verzeichnis.....	519
9.3.7	getContextClassLoaderO vom Thread.....	520
9.4	Die Utility-Klasse System und Properties.....	520
9.4.1	System eigen schaffen der Java-Umgebung.....	521
9.4.2	line.separator.....	522
9.4.3	Browser-Version abfragen.....	523
9.4.4	Property von der Konsole aus setzen.....	523
9.4.5	Umgebungsvariablen des Betriebssystems.....	524
9.4.6	Einfache Zeitmessung und Profilüjg.....	525
9.5	Ausführen externer Programme und Skripte.....	526
9.5.1	ProcessBuilder und Prozesskontrolle mit Process.....	526
9.5.2	Einen Browser/E-Mail-Client/Editor aufrufen.....	529
9.5.3	Ausführen von Skripten.....	530
9.6	Benutzereinstellungen.....	532
9.6.1	Benutzereinstellungen in Windows-Registry oder XML-Dokumenten.....	532
9.6.2	Einträge einfügen, austesen und löschen.....	534
9.6.3	Auslesen der Daten und Schreiben in anderem Format.....	536
9.6.4	Auf Ereignisse horchen.....	536
9.6.5	Zugriff auf die gesamte Windows-Registry.....	536
9.7	Musik abspielen.....	537
9.7.1	Die Arbeit mit AudioClip.....	537
9.7.2	Java Sound API.....	538
9.8	Annotationen.....	539
9.8.1	Annotationstypen @Override, @Deprecated, @ Sup press War nings.....	540
9.8.2	Common Annotations.....	541
9.8.3	Annotationen für Web-Services.....	542
9.8.4	Annotationen für XML-Mapping.....	542
9.9	Zum Weiterlesen.....	542
10.1	Nebenläufigkeit.....	543
10.1.1	Threads und Prozesse.....	543
10.1.2	Wie parallele Programme die Geschwindigkeit steigern können.....	544
10.1.3	Was Java für Nebenläufigkeit alles bietet.....	546
10.2	Threads erzeugen.....	546
10.2.1	Threads über die Schnittstelle Runnable implementieren.....	546

10.2.2	Thread mit Runnable starten.....	547
10.2.3	Der Name eines Threads.....	549
10.2.4	Die Klasse Thread erweitern.....	550
10.2.5	Wer bin ich?.....	552
10.3	Die Zustände eines Threads.....	553
10.3.1	Threads schlafen.....	553
10.3.2	Mit yieldO auf Rechenzeit verzichten.....	555
10.3.3	Das Ende eines Threads.....	555
10.3.4	UncaughtExceptionHandler für unbehandelte Ausnahmen.....	556
10.3.5	Einen Thread höflich mit Interrupt beenden.....	557
10.3.6	Der stopO von außen und die Rettung mit ThreadDeath.....	559
10.3.7	Ein Rendezvous mit joinO.....	560
10.3.8	Barrier und Austausch mit Exchanger.....	562
10.3.9	Arbeit niederlegen und wieder aufnehmen.....	563
10.3.10	Priorität.....	563
10.3.11	Der Thread ist ein Dämon.....	564
10.4	Der Ausführer (Executor) kommt.....	566
10.4.1	Die Schnittstelle Executor.....	566
10.4.2	Die Thread-Pools.....	568
10.4.3	Threads mit Rückgabe über Callable.....	569
10.4.4	Mehrere Callable abarbeiten.....	572
10.4.5	Mit ScheduledExecutorService wiederholende Ausgaben und Zeitsteuerungen.....	572
10.5	Synchronisation über kritische Abschnitte.....	573
10.5.1	Gemeinsam genutzte Daten.....	573
10.5.2	Probleme beim gemeinsamen Zugriff und kritische Abschnitte ...	573
10.5.3	Punkte parallel initialisieren.....	574
10.5.4	i++ sieht atomar aus, ist es aber nicht.....	576
10.5.5	Kritische Abschnitte schützen.....	577
10.5.6	Schützen mit ReentrantLock.....	578
10.5.7	Synchronisieren mit synchronized.....	582
10.5.8	Synchronized-Methoden der Klasse StringBuffer.....	583
10.5.9	Mit synchronized synchronisierte Blöcke.....	584
10.5.10	Dann machen wir doch gleich alles synchronisiert!.....	585
10.5.11	Lock-Freigabe im Fall von Exceptions.....	586
10.5.12	Mit synchronized nachträglich synchronisieren.....	587
10.5.13	Monitore sind reentrant - gut für die Geschwindigkeit.....	588
10.5.14	Synchronisierte Methodenaufrufe zusammenfassen.....	589
10.5.15	Deadlocks.....	590
10.6	Synchronisation über Warten und Benachrichtigen.....	592
10.6.1	Die Schnittstelle Condition.....	593
10.6.2	Beispiel: Erzeuger-Verbraucher-Programm.....	596

10.6.3	Warten mit wait() und Aufwecken mit notify().	600
10.6.4	Falls der Lock fehlt: IllegalMonitorStateException.	601
10.6.5	Semaphor.	603
10.7	Atomare Operationen und frische Werte mit volatile.	606
10.7.1	Der Modifizierer volatile bei Objekt-/Klassenvariablen.	606
10.7.2	Das Paket java.util.concurrent.atomic.	607
10.8	Mit dem Thread verbundene Variablen.	608
10.8.1	ThreadLocal.	609
10.8.2	InheritableThreadLocal.	610
10.9	Gruppen von Threads in einer Thread-Gruppe.	612
10.9.1	Aktive Threads in der Umgebung.	612
10.9.2	Etwas über die aktuelle Thread-Gruppe herausfinden.	613
10.9.3	Threads in einer Thread-Gruppe anlegen.	615
10.9.4	Methoden von Thread und ThreadGroup im Vergleich.	618
10.10	Zeitgesteuerte Abläufe.	619
10.10.1	Die Klassen Timer und TimerTask.	619
10.10.2	Job-Scheduler Quartz.	621
10.11	Einen Abbruch der virtuellen Maschine erkennen.	621
10.12	Zum Weiterlesen.	622
11.1	Weltzeit.	623
11.2	Wichtige Datum-Klassen im Überblick.	624
11.3	Sprachen der Länder.	624
11.3.1	Sprachen und Regionen über Locale-Objekte.	625
11.4	Internationalisierung und Lokalisierung.	628
11.4.1	ResourceBundle-Objekte und Ressource-Dateien.	629
11.4.2	Ressource-Dateien zur Lokalisierung.	629
11.4.3	Die Klasse ResourceBundle.	630
11.4.4	Ladestrategie für ResourceBundle-Objekte.	631
11.5	Zeitzonen.	632
11.5.1	Zeitzonen durch die Klasse TimeZone repräsentieren.	632
11.6	Die Klasse Date.	634
11.6.1	Objekte erzeugen und Methoden nutzen.	634
11.7	Calendar und GregorianCalendar.	635
11.7.1	Die abstrakte Klasse Calendar.	636
11.7.2	Der gregorianische Kalender.	636
11.7.3	Ostertage.	639
11.7.4	Abfragen und Setzen von Datumselementen.	641
11.8	Formatieren und Parsen von Datumsangaben.	647
11.8.1	Ausgaben mit printf().	647
11.8.2	Mit DateFormat und SimpleDateFormat formatieren.	647

11.8.3	Parsen von Datumswerten.....	652
11.9	Zum Weiterlesen.....	654
12.1	Datenstrukturen und die Collection-API.....	655
12.1.1	Designprinzip mit Schnittstellen, abstrakten Klassen, konkreten Klassen.....	656
12.1.2	Die Basis-Schnittstellen Collection und Map.....	657
12.1.3	Das erste Programm mit Container-Klassen.....	657
12.1.4	Die Schnittstelle Collection.....	658
12.1.5	Schnittstellen, die Collection erweitern, und Map.....	660
12.1.6	Konkrete Container-Klassen.....	662
12.1.7	Welche Klasse nehmen?.....	663
12.1.8	Generische Datentypen in der Collection-API.....	663
12.1.9	Die Schnittstelle Iterable und das erweiterte for.....	664
12.2	Mit einem Iterator durch die Daten wandern.....	665
12.2.1	Die Schnittstellen Enumeration und Iterator.....	665
12.2.2	Iteratoren von Sammlungen und das erweiterte for.....	667
12.2.3	Fail-Fast Iterator und die ConcurrentModificationException.....	669
12.3	Listen.....	669
12.3.1	ArrayList oder LinkedList? Speicherung im Feld oder in einer verketteten Liste.....	670
12.3.2	Die Schnittstelle List.....	670
12.3.3	ArrayList.....	677
12.3.4	LinkedList.....	680
12.3.5	Der Feld-Adapter Arrays.asListÖ.....	681
12.3.6	toArrayO von Collection verstehen - die Gefahr einer Falle erkennen.....	682
12.3.7	Primitive Elemente in den Collection-Datenstrukturen.....	684
12.4	Vergleichen von Objekten.....	685
12.4.1	Die Schnittstellen Comparator und Comparable.....	685
12.4.2	Algorithmen mit Such- und Sortiermöglichkeiten.....	687
12.4.3	Den größten und kleinsten Wert einer Collection finden.....	688
12.4.4	Sortieren.....	689
12.5	Mengen (Sets).....	692
12.5.1	HashSet.....	694
12.5.2	TreeSet - die Menge durch Bäume.....	695
12.5.3	LinkedHashSet.....	698
12.6	Stack (Kellerspeicher, Stapel).....	698
12.6.1	Die Methoden von Stack.....	699
12.6.2	Ein Stack ist ein Vector-aha!.....	699

12.7	Queues (Schlangen) und Deques.....	700
12.7.1	Die Schnittstelle Queue.....	700
12.7.2	Blockierende Queues und Prioritätswarteschlangen.....	701
12.7.3	Deque-Klassen.....	702
12.8	Assoziative Speicher.....	702
12.8.1	Die Klassen HashMap und TreeMap.....	702
12.8.2	Einfügen und Abfragen der Datenstruktur.....	704
12.8.3	Die Bedeutung von equalsO, hashCodeO und IdentityHashMap.....	707
12.8.4	Elemente im Assoziativspeicher müssen unveränderbar bleiben.....	709
12.8.5	Aufzählungen und Sichten auf dgja Assoziativspeicher.....	709
12.8.6	Der Gleichheitstest, Hash-Wert und Klon einer Hash-Tabelle. . . .	712
12.8.7	Die Arbeitsweise einer Hash-Tabelle.....	712
12.8.8	Multi-Maps.....	715
12.9	Die Properties-Klasse.....	715
12.9.1	Properties setzen und lesen.....	715
12.9.2	Properties verketten.....	715
12.9.3	Eigenschaften ausgeben.....	717
12.9.4	Hierarchische Eigenschaften.....	717
12.9.5	Properties speichern.....	717
12.9.6	Klassenbeziehungen: Properties und Hashtable.....	719
12.10	Algorithmen in Collections.....	719
12.10.1	Listenoperationen: Ersetzen, Kopieren, Füllen, Umdrehen, Rotieren, Durchmischen.....	720
12.10.2	Mit der Halbierungssuche nach Elementen fahnden.....	721
12.10.3	Nicht-änderbare Datenstrukturen.....	723
12.10.4	Häufigkeit eines Elements.....	723
12.10.5	nCopiesO.....	723
12.10.6	Singletons.....	724
12.11	Synchronisation der Datenstrukturen.....	725
12.11.1	Lock-Free-Algorithmen aus java.util.concurrent.....	725
12.11.2	Wrapper zur Synchronisation.....	726
12.11.3	CopyOnWriteArrayList und CopyOnWriteArraySet.....	727
12.12	Die Klasse BitSet für Bitmengen.....	727
12.12.1	Ein BitSet anlegen, füllen und erfragen.....	727
12.12.2	Mengenorientierte Operationen.....	728
12.12.3	Funktionsübersicht.....	729
12.12.4	Primzahlen in einem BitSet verwalten.....	730

13	Dateien und Datenströme.....	731
13.1	Datei und Verzeichnis.....	732
13.1.1	Dateien und Verzeichnisse mit der Klasse File.....	732
13.1.2	Verzeichnis oder Datei? Existiert es?.....	734
13.1.3	Verzeichnis- und Dateieigenschaften/-attribute.....	735
13.1.4	Wurzelverzeichnis, Laufwerksnamen, Plattenspeicher.....	737
13.1.5	Umbenennen und Verzeichnisse anlegen.....	740
13.1.6	Verzeichnisse listen und Dateien filtern.....	740
13.1.7	Dateien berühren, neue Dateien anlegen, temporäre Dateien	743
13.1.8	Dateien und Verzeichnisse löschen.....	745
13.1.9	Verzeichnissenach Dateien iterativ durchsuchen.....	746
13.1.10	URL- und URI-Objekte aus einem File-Objekt ableiten.....	747
13.1.11	Mit Locking Dateien sperren.....	747
13.1.12	Sicherheitsprüfung.....	748
13.1.13	Mime-Typen mit dem JavaBeans Activation Framework (JAF)	748
13.1.14	Zugriff auf SMB-Server mit jCIFS.....	749
13.2	Dateien mit wahlfreiem Zugriff.....	750
13.2.1	Ein RandomAccessFile zum Lesen und Schreiben öffnen.....	751
13.2.2	Aus dem RandomAccessFile lesen.....	751
13.2.3	Schreiben mit RandomAccessFile.....	754
13.2.4	Die Länge des RandomAccessFile.....	754
13.2.5	Hin und her in der Datei.....	754
13.2.6	Wahlfreier Zugriff und Pufferung mit Unified I/O.....	755
13.3	Stream-Klassen und Reader/Writer am Beispiel von Dateien.....	755
13.3.1	Mit dem FileWriter Texte in Dateien schreiben.....	756
13.3.2	Zeichen mit der Klasse FileReader lesen.....	758
13.3.3	Kopieren mit FileOutputStream und FileInputStream.....	759
13.3.4	Das FileDescriptor-Objekt.....	762
13.4	Basisklassen für die Ein-/Ausgabe.....	762
13.4.1	Die abstrakten Basisklassen.....	762
13.4.2	Übersicht über Ein-ZAusgabeklassen.....	763
13.4.3	Die abstrakte Basisklasse OutputStream.....	764
13.4.4	Die Schnittstellen Closeable und Flushable.....	766
13.4.5	Ein Datenschluckler.....	766
13.4.6	Die abstrakte Basisklasse InputStream.....	767
13.4.7	Ressourcen wie Grafiken aus dem Klassenpfad und aus Jar-Archiven laden.....	768
13.4.8	Ströme mit SequenceInputStream zusammensetzen.....	769
13.4.9	Die abstrakte Basisklasse Writer.....	770
13.4.10	Die Schnittstelle Appendable.....	772
13.4.11	Die abstrakte Basisklasse Reader.....	773

13.5	Formatierte Textausgaben.....	774
13.5.1	Die Klassen PrintWriter und PrintStream.....	775
13.5.2	System.out, System.err und System.in.....	779
13.5.3	Geschützte Passwort-Eingaben mit der Klasse Console.....	781
13.6	Schreiben und Lesen aus Strings und Byte-Feldern.....	781
13.6.1	Mit dem StringWriter ein String-Objekt füllen.....	782
13.6.2	CharArrayWriter.....	783
13.6.3	StringReader und CharArrayReader.....	784
13.6.4	Mit ByteArrayOutputStream in ein Byte-Feld schreiben.....	785
13.6.5	Mit ByteArrayInputStream aus einem Byte-Feld lesen.....	786
13.7	Datenströme filtern und verketten.....	786
13.7.1	Streamsals Filter verketten.....	787
13.7.2	Gepufferte Ausgaben mit BufferedWriter und BufferedOutputStream.....	787
13.7.3	Gepufferte Eingaben mit BufferedReader und BufferedReader.....	789
13.7.4	LineNumberReader zählt automatisch Zeilen mit.....	791
13.7.5	Daten mit der Klasse PushbackReader zurücklegen.....	792
13.7.6	DataOutputStream/DataInputStream.....	795
13.7.7	Basisklassen für Filter.....	795
13.7.8	Die Basisklasse FilterWriter.....	795
13.7.9	Ein LowerCaseWriter.....	796
13.7.10	Eingaben mit der Klasse FilterReader filtern.....	798
13.8	Vermittler zwischen Byte-Streams und Unicode-Strömen.....	799
13.8.1	Datenkonvertierung durch den OutputStreamWriter.....	799
13.8.2	Automatische Konvertierungen mit dem InputStreamReader.....	800
13.9	Kommunikation zwischen Threads mit Pipes.....	801
13.9.1	PipedOutputStream und PipedInputStream.....	801
13.9.2	PipedWriter und PipedReader.....	803
13.10	Datenkompression.....	805
13.10.1	Java-Unterstützung beim Komprimieren und Zusammenpacken.....	806
13.10.2	Datenströme komprimieren.....	806
13.10.3	Zip-Archive.....	810
13.10.4	Jar-Archive.....	816
13.11	Prüfsummen.....	816
13.11.1	Die Schnittstelle Checksum.....	817
13.11.2	Die Klasse CRC32.....	818
13.11.3	Die Adler32-Klasse.....	819
13.12	Persistente Objekte und Serialisierung.....	820
13.12.1	Objekte mit der Standard-Serialisierung speichern und lesen.....	821
13.12.2	Zwei einfache Anwendungen der Serialisierung.....	823
13.12.3	Die Schnittstelle Serializable.....	824

	13.12.4	Nicht serialisierbare Attribute aussparen.....	826
	13.12.5	Das Abspeichern selbst in die Hand nehmen.....	827
	13.12.6	Tiefe Objektkopien.....	831
	13.12.7	Versionenverwaltung und die SUID.....	832
	13.12.8	Wie die ArrayList serialisiert.....	834
	13.12.9	Probleme mit der Serialisierung.....	835
Sfi	13.12.10	Serialisieren in XML-Dateien..... li&VÄ.....	835
	13.12.11	JavaBeans Persistence.....	836
	13.12.12	XStream.....	838
13.13		Tokenizer.....	838
	13.13.1	StreamTokenizer.....	838
	13.13.2	CSV-(CommaSeparated Values-)Dateien verarbeiten.....	841
13.14		Zum Weiterlesen.....	842
14.1		Auszeichnungssprachen.....	843
	14.1.1	Die Standard Generalized Markup Language (SGML).....	843
	14.1.2	Extensible Markup Language (XML).....	844
14.2		Eigenschaften von XML-Dokumenten.....	844
	14.2.1	Elemente und Attribute.....	844
	14.2.2	Beschreibungssprache für den Aufbau von XML-Dokumenten ...	846
	14.2.3	Schema - eine Alternative zu DTD.....	849
	14.2.4	Namensraum (Namespace).....	852
	14.2.5	XML-Applikationen.....	853
14.3		Die Java-APIs für XML.....	853
	14.3.1	Das Document Object Model (DOM).....	854
	14.3.2	Simple API for XML Parsing (SAX).....	854
	14.3.3	Pull-APIStAX.....	854
	14.3.4	Java Document Object Model (JDOM).....	855
	14.3.5	JAXP als Java-Schnittstelle zu XML.....	855
	14.3.6	DOM-Bäume einlesen mit JAXP.....	856
14.4		Serielle Verarbeitung mit StAX.....	856
	14.4.1	Unterschiede der Verarbeitungsmodelle.....	857
	14.4.2	XML-Dateien mit dem Cursor-Verfahren lesen.....	858
	14.4.3	XML-Dateien mit dem Iterator-Verfahren verarbeiten.....	860
	14.4.4	Mit Filtern arbeiten.....	861
	14.4.5	XML-Dokumente schreiben.....	862
14.5		Serielle Verarbeitung von XML mit SAX.....	865
	14.5.1	Schnittstellen von SAX.....	865
	14.5.2	SAX-Parser erzeugen.....	866
	14.5.3	Die wichtigsten Methoden der Schnittstelle ContentHandler. . .	866
	14.5.4	ErrorHandler und EntityResolver.....	868

•*	14.6	XML-Dateien mit JDOM verarbeiten.....	869
	14.6.1	JDOM beziehen.....	869
	14.6.2	Paketüberstcht.....	870
	14.6.3	Die Document-Klasse.....	871
	14.6.4	Eingaben aus der Datei lesen.....	872
	14.6.5	Das Dokument im XML-Format ausgeben.....	873
	14.6.6	Der Dokumenttyp.....	873
	14.6.7	Elemente.....	874
	14.6.8	Zugriff auf Elementinhalte.....	876
	14.6.9	Liste mit Unterelementen erzeugen.....	879
	14.6.10	Neue Elemente einfügen und ändern.....	879
	14.6.11	Attributinhalt lesen und ändern*.....	882
	14.6.12	XPath.....	884
	14.7	Transformationen mit XSLT.....	887
	14.7.1	Templates und XPath als Kernelemente von XSLT.....	887
	14.7.2	Umwandlung von XML-Dateien mit JDOM und JAXP.....	889
	14.8	Java Architecture for XML Binding (JAXB).....	890
	14.8.1	Beans für JAXB aufbauen.....	890
	14.8.2	JAXBContext und die Marshaller/Unmarshaller.....	892
	14.9	HTML-Dokumente einlesen.....	893
	14.10	Zum Weiterlesen.....»#ki3totoau>#mt&4m»>>&*.....	894
15 Grafische Oberflächen mit Swing.....			895
	15.1	Das Abstract Window Toolkit und Swing.....	895
	15.1.1	Abstract Window Toolkit (AWT).....	895
	15.1.2	Java Foundation Classes.....	896
	15.1.3	Was Swing von AWT unterscheidet.....	899
	15.1.4	Die Klasse Toolkit.....	899
	15.2	Fenster unter grafischen Oberflächen.....	900
	15.2.1	Swing-Fenster darstellen.....	900
	15.2.2	AWT-Fenster darstellen.....	902
	15.2.3	Sichtbarkeit des Fensters.....	903
	15.2.4	Größe und Position des Fensters verändern.....	903
	15.2.5	Unterklassen der Fenster-Klassen bilden.....	904
	15.2.6	Fenster- und Dialog-Dekoration.....	905
	15.2.7	Dynamisches Layout während einer Größenänderung.....	905
	15.3	Beschriftungen dabei).....	906
	15.3.1	MehrzeiligerText, HTML in der Darstellung.....	909
	15.4	Icon und ImageIcon für Bilder auf Swing-Komponenten.....	909
	15.4.1	Die Schnittstelle Icon.....	911
	15.5	Es tut sich was- Ereignisse beim AWT.....	913
	15.5.1	Die Klasse AWTEvent.....	913

15.5.2	Events auf verschiedenen Ebenen.....	914
15.5.3	Swings Ereignisquellen und Horcher (Listener).....	916
15.5.4	Listener implementieren.....	916
15.5.5	Listener bei dem Ereignisauslöser anmelden/abmelden.....	919
15.5.6	Aufrufen der Listener im AWT-Event-Thread.....	919
15.5.7	Adapterklassen nutzen.....	920
15.5.8	Innere Mitgliedsklassen und innere anonyme Klassen.....	922
15.6	Schaltflächen.....	923
15.6.1	Normale Schaltflächen (JButton).....	923
15.6.2	Der aufmerksame Action Listener.....	925
15.6.3	Basisklasse Abstract Button.....	927
15.6.4	Wechselknopf (JToggleButton)*!*.....	929
15.7	Swing Action.....	929
15.7.1	javax.swing.Action.....	930
15.7.2	Eigenschaften der Action-Objekte.....	930
15.8	JComponent und Component als Basis aller Komponenten.....	931
15.8.1	Tooltips.....	931
15.8.2	Rahmen (Border).....	932
15.8.3	Fokus und Navigation.....	935
15.8.4	Ereignisse jeder Komponente.....	936
15.8.5	Die Größe und Position einer Komponente.....	938
15.8.6	Komponenten-Ereignisse.....	939
15.8.7	Hinzufügen von Komponenten.....	940
15.8.8	UI-Delegate- der wahre Zeichner.....	940
15.8.9	Undurchsichtige (opak) Komponente.....	943
15.8.10	Properties und Listener für Änderungen.....	943
15.9	Container.....	944
15.9.1	Standard Container (JPanel).....	944
15.9.2	Bereich mit automatischen Rollbalken (JScrollPane).....	945
15.9.3	Reiter (JTabbedPane).....	946
15.9.4	Teilung-Komponente (JSplitPane).....	947
15.10	Alles Auslegungssache: die Layoutmanager.....	947
15.10.1	Übersicht über Layoutmanager.....	947
15.10.2	Zuweisen eines Layoutmanagers.....	948
15.10.3	Im Fluss mit FlowLayout.....	949
15.10.4	Mit BorderLayout in allen Himmelsrichtungen.....	951
15.10.5	Rasteranordnung mit GridLayout.....	954
15.10.6	Der GridBagLayout-Manager.....	955
15.10.7	Null-Layout.....	960
15.10.8	BoxLayout.....	961
15.10.9	Weitere Layoutmanager.....	962

15.11	Rollbalken und Schieberegler.....	962
15.11.1	Schieberegler (JSlider).....	962
15.11.2	Rollbalken (JScrollBar).....	964
15.12	Kontrollfelder, Optionsfelder, Kontrollfeldgruppen.....	968
15.12.1	Kontrollfelder (JCheckBox).....	968
15.12.2	ItemSelectable, ItemListener und das ItemEvent.....	971
15.12.3	Sich gegenseitig ausschließende Optionen (JRadioButton).....	972
15.13	Fortschritte bei Operationen überwachen.....	974
15.13.1	Fortschrittsbalken (JProgressBar).....	974
15.13.2	Dialog mit Fortschrittsanzeige (ProgressMonitor).....	976
15.14	Menüs und Symbolleisten.....	976
15.14.1	Die Menüleisten und die Eintrag»».....	977
15.14.2	Menüeinträge definieren.....	978
15.14.3	Einträge durch Action-Objekte beschreiben.....	980
15.14.4	Mit der Tastatur: Mnemonics und Shortcut.....	981
15.14.5	DerTastatur-Shortcut(Accelerator).....	981
15.14.6	Tastenkürzel (Mnemonics).....	983
15.14.7	Symbolleisten alias Toolbars.....	983
15.14.8	Popup-Menüs.....	986
15.14.9	System-Tray nutzen.....	989
15.15	Das Model-View-Controller-Konzept.....	990
15.16	Auswahlmenüs, Listen und Spinner.....	992
15.16.1	Auswahlmenü (JComboBox).....	992
15.16.2	Zuordnung einer Taste mit einem Eintrag.....	996
15.16.3	Datumsauswahl.....	997
15.16.4	Listen (JList).....	997
15.16.5	Drehfeld (JSpinner).....	1002
15.17	Texteingabefelder.....	1004
15.17.1	Text in einer Eingabezeile.....	1004
15.17.2	Die Oberklasse der Text-Komponenten (JTextComponent).....	1005
15.17.3	Geschützte Eingaben (JPasswordField).....	1006
15.17.4	Validierende Eingabefelder (JFormattedTextField).....	1006
15.17.5	Einfache mehrzeilige Textfelder (JTextArea).....	1008
15.17.6	Editor-Klasse (JEditorPane).....	1010
15.18	Tabellen (JTable).....	1013
15.18.1	Ein eigenes Tabellen-Model.....	1014
15.18.2	Basisklasse für eigene Modelle (AbstractTableModel).....	1015
15.18.3	Vorgefertigtes Standard-Modell (DefaultTableModel).....	1018
15.18.4	Ein eigener Renderer für Tabellen.....	1019
15.18.5	Zeil-Editoren.....	1023
15.18.6	Große und Umrandung der Zellen.....	1024
15.18.7	Spalteninformationen.....	1024

Inhalt

...	15.18.8	Tabellenkopf von Swing-Tabellen.....	1025
„	15.18.9	Selektionen einer Tabelle.....	1025
...	15.18.10	Automatisches Sortieren und Filtern mit RowSorter.....	1026
	15.18.11	Ein professionelles Tabellenlayout mit JGrid.....	1027
15.19		Bäume (JTree).....	1028
	15.19.1	JTree und sein TreeModel und TreeNode.....	1028
..	15.19.2	Selektionen bemerken.....	1029
	15.19.3	Das TreeModel von JTree.....	1030
15.20		JRootPane, JLayeredPane und JDesktopPane.....	1032
	15.20.1	Wurzelkomponente der Top-Level-Komponenten (JRootPane) ...	1032
	15.20.2	JLayeredPane.....	1033
	15.20.3	JDesktopPane und die Kinder JInternalFrame.....	103E
15.21		Dialoge und Window-Objekte.....	1035
	15.21.1	JWindow und JDtalog.....	1035
	15.21.2	Modal oder nicht-modal.....	1036
	15.21.3	Standarddialoge mit JOptionPane.....	1036
	15.21.4	Der Farbauswahldialog JColorChooser.....	1038
	15.21.5	Der Dateiauswahldialog.....	1040
15.22		Flexibles Java-Look & Feel.....	1044
	15.22.1	L & F global setzen.....	1044
	15.22.2	UIManager.....	1044
	15.22.3	Verbessern des Aussehens unter Windows mit JGoodies Looks ...	1046
15.23		Die Zwischenablage (Clipboard).....	1046
	15.23.1	Clipboard-Objekte.....	1046
	15.23.2	Auf den Inhalt zugreifen mit Transferable.....	1047
	15.23.3	DataFlavor ist das Format der Daten in der Zwischenablage.	1048
	15.23.4	Einfügungen in der Zwischenablage erkennen. ... rrvv^~.^	1050
	15.23.5	Drag & Drop.....	1050
15.24		Undo durchführen.....	1051
15.25		AWT, Swing und die Threads.....	1053
	15.25.1	Ereignisschlange (EventQueue) und AWT-Event-Thread	1053
	15.25.2	Swing ist nicht Thread-sicher.....	1054
	15.25.3	Swing-Elemente mit invokeLaterO und invokeAndWaitö bedienen.....	1055
	15.25.4	SwingWorker.....	1056
	15.25.5	Eigene Ereignisse in die Queue setzen.....	1058
	15.25.6	Auf alle Ereignisse hören.....	1059
15.26		Barrierefreiheit mit der Java Accessibility API.....	1059
15.27		Benutzerinteraktionen automatisieren.....	1060
	15.27.1	Automatisch in die Tasten hauen.....	1061
	15.27.2	Mausoperationen.....	1062
	15.27.3	Methoden zur Zeitsteuerung..... ^%?*; **ii&^.....>-.j&^.&g.....	1062

15.27.4	Screenshots.....	1063
15.27.5	MouseInfo und PointerInfo.....	1063
15.28	Zeitliches Ausführen mit dem javax.swing.Timer.....	1064
15.29	Alternativen zu AWT und Swing.....	1064
15.29.1	XML-Beschreibungen der Oberfläche: Swixml, XUL/Luxor.....	1065
15.29.2	SWT (Standard Widget Toolkit).....	1065
15.30	Zum Weiterlesen.....	1067

16 Grafikprogrammierung.....1069

16.1	Grundlegendes zum Zeichnen.....	1069
16.1.1	Die paintO-Methode für das AWT-Frame.....	1069
16.1.2	Zeichnen von Inhalten mit JFrame**.....	1071
16.1.3	Auffordern zum Neuzeichnen mit repaintO.....	1072
16.1.4	Grundbegriffe: Koordinaten, Punkte, Pixel.....	1073
16.1.5	Die ereignisorientierte Programmierung ändert Fensterinhalte ...	1073
16.1.6	Java2D-API.....	1074
16.2	Einfache Zeichenfunktionen.....	1075
16.2.1	<i>Linien</i>	1075
16.2.2	Rechtecke.....	1076
16.2.3	Ovale und Kreisbögen.....	1077
16.2.4	Polygone und Polylines.....	1077
16.3	Zeichenketten schreiben und Fonts.....	1080
16.3.1	Zeichenfolgen schreiben.....	1080
16.3.2	Die Font-Klasse.....	1081
16.3.3	Einen neuen Font aus einem gegebenen Font ableiten.....	1082
16.3.4	Zeichensätze des Systems ermitteln.....	1083
16.3.5	Neue TrueType-Fonts in Java nutzen.....	1084
16.3.6	Font-Metadaten durch FontMetrics.....	1085
16.4	Geometrische Objekte.....	1088
16.4.1	Die Schnittstelle Shape.....	1089
16.4.2	Kreisförmiges.....	1090
16.4.3	Kurviges.....	1091
16.4.4	Area und die konstruktive Flächengeometrie.....	1091
16.4.5	Pfade.....	1091
16.4.6	Punkt in Form, Schnitt von Linien, Abstand Punkt/Linie und Weiteres.....	1094
16.5	Das Innere und Äußere einer Form.....*	1095
16.5.1	Farben und die Paint-Schnittstelle.....	1095
16.5.2	Farben mit der Klasse Color.....	1096
16.5.3	Die Farben des Systems über SystemColor.....	1101
16.5.4	Composite und Xor.....	1104
16.5.5	Dicke und Art der Linien von Formen bestimmen über Stroke ...	1105

16.6	Bilder....."	1109
16.6.1	Eine Übersicht über die Bilder-Bibliotheken.....	1110
16.6.2	Bilder mit ImageIO lesen.....	1111
16.6.3	Ein Bild zeichnen.....	1113
16.6.4	Programm-Icon/Fenster-Icon setzen.....	1116
16.6.5	Splash-Screen.....	1117
16.6.6	Bilder im Speicher erzeugen.....	1117
16.6.7	Pixel für Pixel auslesen und schreiben.....	1119
16.6.8	Bilder skalieren.....	1121
16.6.9	Schreiben mit ImageIO.....	1123
16.6.10	Asynchrones Laden mit getImageIO und dem MediaTracker.....	1126
16.6.11	Selbst definierte Cursor....."?"	1127
16.6.12	VolatileImage.....	1129
16.7	Weitere Eigenschaften von Graphics.....	1129
16.7.1	Eine Kopie von Graphics erstellen.....	1129
16.7.2	Koordinatensystem verschieben.....	1130
16.7.3	Beschnitt (Clipping).....	1130
16.8	Zeichenhinweise durch RenderingHints.....	1134
16.9	Transformationen mit einem AffineTransform-Objekt.....	1134
16.10	Drucken.....	1136
16.10.1	Drucken der Inhalte.....	1137
16.10.2	Bekannte Drucker.....	1138
16.11	Grafik Verarbeitung ohne grafische Oberfläche.....	1139
16.11.1	Xvfb-Server.....	1140
16.11.2	Pure Java AWT Toolkit (PJA).....	1140
16.12	Zum Weiterlesen.....	1141
17.1	Grundlegende Begriffe.....	1143
17.1.1	Internet-Standards und RFC.....	1144
17.2	URI und URL.....	1144
17.2.1	URI.....	1144
17.2.2	Die Klasse URL.....	1144
17.2.3	Informationen über eine URL.....	1147
17.2.4	Der Zugriff auf die Daten über die Klasse URL.....	1148
17.2.5	Verbindungen durch einen Proxy-Server.....L	1150
17.3	Die Klasse URLConnection.....	1151
17.3.1	Methoden und Anwendung von URLConnection.....	1151
17.3.2	Protokoll- und Content-Handler.....	1153
17.3.3	Im Detail: vom URL zur URLConnection.....	1154
17.3.4	Der Protokoll-Handler für Jar-Dateien.....	1155

17.3.5	Passwort-geschützte Seiten mit Basic Authentication/ Proxy-Authentifizierung.....	1157
17.4	Mit GET und POST Daten übergeben.....	1159
17.4.1	Kodieren der Parameter für Serverprogramme.....	1159
17.4.2	Eine Suchmaschine ansprechen.....	1160
17.5	Host- und IP-Adressen.....	1161
17.5.1	Lebt der Rechner?.....	1163
17.5.2	Das Netz ist Klasse.....	1164
17.5.3	IP-Adresse des lokalen Hosts.....	1164
17.6	Networkinterface.....	1165
17.7	Mit dem Socket zum Server.....	1166
17.7.1	Das Netzwerk ist der Computer ^.....	1166
17.7.2	Sockets.....	1166
17.7.3	Eine Verbindung zum Server aufbauen.....	1167
17.7.4	Server unter Spannung: die Ströme.....	1168
17.7.5	Die Verbindung wieder abbauen.....	1169
17.7.6	Informationen über den Socket.....	1169
17.7.7	Reine Verbindungsdaten über SocketAddress.....	1171
17.8	Client/Server-Kommunikation.....	1172
17.8.1	Warten auf Verbindungen.....	1173
17.8.2	Ein Multiplikationsserver.....	1174
17.8.3	Blockierendes Lesen.....	1176
17.8.4	Von außen erreichbar sein.....	1177
17.9	Apache Jakarta Commons HttpClient und Net.....	1178
17.9.1	Jakarta Commons HttpClient.....	1178
17.9.2	Jakarta Commons Net.....	1179
17.10	Arbeitsweise eines Webservers.....	1180
17.10.1	Das Hypertext Transfer Protocol (HTTP).....	1180
17.10.2	Anfragen an den Server.....	1180
17.10.3	Die Antworten vom Server.....	1183
17.10.4	Webserver mit com.sun.net.httpserver.HttpServer.....	1186
17.11	Datagram-Sockets.....	1187
17.11.1	Die Klasse DatagramSocket.....	1189
17.11.2	Datagramme und die Klasse DatagramPacket.....	1190
17.11.3	Auf ein hereinkommendes Paket warten.....	1191
17.11.4	Ein Paket zum Senden vorbereiten.....	1192
17.11.5	Methoden der Klasse DatagramPacket.....	1193
17.11.6	Das Paketsenden.....	1193
17.12	E-Mail.....	1194
17.12.1	Wie eine E-Mail um die Welt geht.....	1195
17.12.2	Das Simple Mail Transfer Protocol und RFC 822.....	1195
17.12.3	POP (Post Office Protocol).....	1196

17.12.4	Die JavaMail API!	1196
17.12.5	E-Mails mittels POP3 abrufen	1197
17.12.6	E-Mails versenden	1199
17.12.7	Ereignisse und Suchen	1202
17.13	Tiefer liegende Netzwerkeigenschaften	1203
17.13.1	Internet Control Message Protocol (ICMP)	1203
17.13.2	MAC-Adresse	1203
17.14	Zum Weiterlesen	1204

18 Verteilte Programmierung mit RMI

18.1	Entfernte Objekte und Methoden	1205
18.1.1	Stellvertreter helfen bei entfernten Methodenaufrufen	1205
18.1.2	Standards für entfernte Objekte	1207
18.2	Java Remote Method Invocation	1207
18.2.1	Zusammenspiel von Server, Registry und Client	1207
18.2.2	Wie die Stellvertreter die Daten übertragen	1207
18.2.3	Probleme mit entfernten Methoden	1208
18.2.4	Nutzen von RMI bei Middleware-Lösungen	1210
18.2.5	Zentrale Klassen und Schnittstellen	1210
18.2.6	Entfernte und lokale Objekte im Vergleich	1211
18.3	Auf der Serverseite	1211
18.3.1	Entfernte Schnittstelle deklarieren	1211
18.3.2	Remote-Objekt-Implementierung	1212
18.3.3	Stellvertreterobjekte	1213
18.3.4	Der Namensdienst (Registry)	1213
18.3.5	Remote-Objekt-Implementierung exportieren und beim Namensdienst anmelden	1215
18.3.6	Einfaches Logging	1217
18.3.7	Aufräumen mit dem DGC	1218
18.4	Auf der Clientseite	1218
18.5	Entfernte Objekte übergeben und laden	1219
18.5.1	Klassen vom RMI-Klassenlader nachladen	1220
18.6	Weitere Eigenschaften von RMI	1220
18.6.1	RMI und CORBA	1220
18.6.2	RMI über HTTP getunnelt	1220
18.6.3	Automatische Remote-Objekt-Aktivierung	1221
18.7	Daily Soap	1222
18.7.1	SOAP-Protokoll	1222
18.7.2	Die technische Realisierung	1223
18.7.3	SOAP-Implementierungen	1223
18.7.4	(a>WebService in Java 6	1224

18.7.5	Einen Web-Service definieren.....	1224
18.7.6	Web-Services veröffentlichen.....	1225
18.7.7	Einen JAX-WS-Client implementieren.....	1225
18.8	JavaMessage Service (JMS).....	1227
18.9	Zum Weiterlesen.....	1228

19 JavaServer Pages und Servlets.....1229

19.1	Dynamisch generierte Webseiten.....	1229
19.1.1	Was sind Servlets?.....	1229
19.1.2	Was sind JavaServer Pages?.....	1230
19.2	Servlets und JSPs mit Tomcat entwickeln ^.....	1231
19.2.1	Servlet-Container.....	1231
19.2.2	Entwicklung der Servlet/JSP-Spezifikationen.....	1232
19.2.3	Webserver mit Servlet-Funktionalität.....	1232
19.2.4	Tomcat.....	1232
19.2.5	Ablageort für eigene JSP-Seiten.....	1233
19.2.6	Web-Applikationen.....	1234
19.2.7	Zuordnung von Web-Applikationen zu physikalischen Verzeichnissen.....	1235
19.2.8	Mit dem WTP ein Web-Projekt entwickeln.....	1235
19.3	Statisches und Dynamisches.....	1236
19.3.1	Statischer Template-Code.....	1236
19.3.2	Dynamische Inhalte.....	1237
19.3.3	Kommentare.....	1237
19.4	Die Expression Language (EL).....	1237
19.4.1	Operatoren der EL.....	1238
19.4.2	Literale.....	1238
19.4.3	Implizite EL-Objekte.....	1239
19.5	Formulardaten.....	1239
19.6	Auf Beans zurückgreifen.....	1240
19.6.1	Beans in JSP-Seiten anlegen.....	1240
19.6.2	Properties einer Bean im EL-Ausdruck erfragen.....	1241
19.6.3	Properties mit <jsp:setProperty> setzen.....	1241
19.6.4	Bean-Klasse zum Testen von E-Mail-Adressen.....	1242
19.6.5	Parameterwerte in Bean übertragen.....	1243
19.7	JSP Tag-Libraries.....	1243
19.7.1	Standard Tag Library (JSTL).....	1244
19.7.2	JakartaTaglibs Project.....	1247
19.8	Einbinden und Weiterleiten.....	1248
19.8.1	Einbinden von Inhalten.....	1248
19.8.2	Forward und Redirect.....	1250
19.8.3	Applets einbinden.....	1250

19.9	Skripten von JSPs.....	1251
19.9.1	Scriptlets.....	1251
19.9.2	JSP-Ausdrücke.....	1251
19.9.3	JSP-Deklarationen.....	1252
19.9.4	Quoting.....	1252
19.9.5	Entsprechende XML-Tags.....	1252
19.9.6	Implizite Objekte für Scriptlets und JSP-Ausdrücke.....	1253
19.10	JSP-Direktiven.....	1253
19.10.1	page-Direktiven im Überblick.....	1254
19.10.2	MitJSPs Bilder generieren.....	1255
19.11	Sitzungsverfolgung (Session Tracking).....	1256
19.11.1	Lösungen für Sitzungsverfolgung.....	1257
19.11.2	Auf Session-Dateien zurückgreifen.....	1258
19.12	Servlets.....	1258
19.12.1	Servlets compilieren.....	1259
19.12.2	Servlet-Mapping.....	1260
19.12.3	Der Lebenszyklus eines Servlets.....	1261
19.12.4	Mehrere Anfragen beim Servlet und die Thread-Sicherheit.....	1261
19.12.5	Servlets und Sessions.....	1261
19.12.6	Weiterleiten und Einbinden von Servlet-Inhalten.....	1262
19.13	Internationalisierung.....	1263
19.13.1	Die Länderkennung des Anfragers auslesen.....	1264
19.13.2	Länderkennung für die Ausgabe setzen.....	1264
19.13.3	Westeuropäische Texte senden.....	1264
19.14	Zum Weiterlesen.....	1265
20.1	Applets in der Wiege von Java.....	1267
20.1.1	(J)Applet und Applikationen.....	1267
20.1.2	Das erste Hallo-Applet.....	1267
20.1.3	Die Zyklen eines Applets.....	1269
20.1.4	Parameter an das Applet übergeben.....	1269
20.1.5	Wie das Applet den Browser-Inhalt ändern kann.....	1271
20.1.6	Den Ursprung des Applets erfragen.....	1271
20.1.7	Datenaustausch zwischen Applets.....	1273
20.1.8	Was ein Applet alles darf.....	1275
20.2	Fehler in Applets finden.....	1276
20.2.1	Ist Java im Browser aktiviert?.....	1276
20.2.2	Läuft das Applet unter Netscape oder Microsoft Explorer?.....	1277
20.2.3	Datenaustausch zwischen Applets und Java-Skripten.....	1278
20.3	Webstart.....	1279

21	Midlets und die Java AAE	1281
21.1	Java Platform, Micro Edition (Java ME)	1281
21.2	Konfigurationen	1281
21.2.1	Connected Limited Device Configuration (CLDC)	1281
21.2.2	Connected Device Configuration (CDQ)	1282
21.3	Profile	1282
21.3.1	Mobile Information Device Profile (MIDP)	1282
21.3.2	Weitere Profile	1283
21.4	Wireless Toolkits	1283
21.4.1	Sun Java Wireless Toolkit for CLDC	1283
21.4.2	Eclipse-Plugin	1284
21.5	Die Midlet-API	1286
21.5.1	Paketstruktur Mobile Information Device Profile (2.0)	1286
21.6	Zum Weiterlesen	1287
22	Datenbankmanagement mit JDBC	1289
22.1	Das relationale Modell	1289
22.2	Datenbanken und Tools	1290
22.2.1	HSQLDB	1290
22.2.2	Weitere Datenbanken	1291
22.2.3	Eclipse-Plugins zum Durchschauen von Datenbanken	1293
22.3	JDBC und Datenbanktreiber	1295
22.3.1	Treibertypen	1296
22.3.2	JDBC-Versionen	1297
22.4	Eine Beispielabfrage	1298
22.4.1	Schritte zur Datenbankabfrage	1298
22.4.2	Client für HSQLDB-Datenbank	1299
22.5	Mit Java an eine Datenbank andocken	1300
22.5.1	Der Treiber-Manager	1301
22.5.2	Den Treiber laden	1301
22.5.3	Eine Aufzählung aller Treiber	1302
22.5.4	Log-Informationen	1303
22.5.5	Verbindung zur Datenbank auf- und abbauen	1304
22.5.6	DataSource	1307
22.5.7	Gepoolte Verbindungen	1310
22.6	Datenbankabfragen	1310
22.6.1	Abfragen über das Statement-Objekt	1310
22.6.2	Ergebnisse einer Abfrage in ResultSet	1312
22.6.3	Java und <i>SQL-Datentypen</i>	1314
22.6.4	Unicode in der Spalte korrekt auslesen	1317

Inhalt

22.6.5	Eine SQL-NULL und wasNullO bei ResultSet.....	1317
22.6.6	Wie viele Zeilen hat ein ResultSet?.....	1318
22.7	Die Ausnahmen bei JDBC.....	1318
22.8	Elemente einer Datenbank hinzufügen und aktualisieren.....	1319
22.8.1	Batch-Updates.....	1320
22.9	ResultSets in Bohnen durch RowSet.....	1321
22.9.1	Die Schnittstelle RowSet.....	1321
22.9.2	Implementierungen von RowSet.....	1322
22.9.3	Der Typ CachedRowSet.....	1322
22.9.4	Der Typ WebRowSet.....	1323
22.10	Vorbereitete Anweisungen (Prepared Statements).....	1325
22.10.1	PreparedStatement-Objekte vorbereiten.....	1326
22.10.2	Werte für die Platzhalter eines PreparedStatement.....	1326
22.11	Transaktionen.....	1328
22.12	Metadaten.....	1328
22.12.1	Metadaten über die Tabelle.....	1328
22.12.2	Informationen über die Datenbank.....	1332
22.13	Einführung in SQL.....	1333
22.13.1	Ein Rundgang durch SQL-Anfragen.....	1333
22.13.2	Datenabfrage mit der Data Query Language (DQL).....	1335
22.13.3	Tabellen mit der Data Definition Language (DDL) anlegen.....	1337
22.14	Zum Weiterlesen.....	1337

23 Reflection und Annotationen.....1339

23.1	Metadaten.....	1339
23.1.1	Metadaten durch Java-Doc Tags.....	1339
23.1.2	XDoclet.....	1340
23.2	Metadaten der Klassen mit dem Class-Objekt.....	1340
23.2.1	An ein Class-Objekt kommen.....	1341
23.2.2	Was das Class-Objekt beschreibt.....	1343
23.2.3	Der Name der Klasse.....	1345
23.2.4	instanceof mit Class-Objekten.....	1347
23.2.5	Oberklassen finden.....	1347
23.2.6	Implementierte Interfaces einer Klasse oder eines Interfaces.....	1348
23.2.7	Modifizierer und die Klasse Modifier.....	1349
23.2.8	Die Arbeit auf dem Feld.....	1350
23.3	Attribute, Methoden und Konstruktoren.....	1351
23.3.1	Reflections Gespür für Attribute einer Klasse.....	1352
23.3.2	Methoden einer Klasse erfragen.....	1355
23.3.3	Properties einer Bean erfragen.....	1358
23.3.4	Konstruktoren einer Klasse.....	1359
23.3.5	Annotationen.....	1361

23.4	Objekte erzeugen und manipulieren.....	1361
23.4.1	Objekte erzeugen.....	1361
23.4.2	Die Belegung der Variablen erfragen.....	1363
23.4.3	Eine generische toStringO-Funktion.....	1365
23.4.4	Variablen setzen.....	1366
23.4.5	Private Attribute ändern.....	1368
23.5	Methoden aufrufen.....	1369
23.5.1	Statische Methoden aufrufen.....	1370
23.5.2	Dynamische Methodenaufrufe bei festen Methoden beschleunigen.....	1371
23.6	Informationen und Identifizierung von Paketen.....	1372
23.6.1	Geladene Pakete.....	1373
23.7	Annotationen.....	1373
23.7.1	Neue Annotationen definieren.....	1373
23.7.2	Annotationen mit genau einem Element.....	1373
23.7.3	Beliebige Schlüssel-Werte-Paare.....	1375
23.7.4	Vorbelegte Elemente.....	1378
23.7.5	Annotieren von Annotationstypen.....	1379
23.7.6	Annotationen zur Laufzeit ausgelesen.....	1381
23.7.7	Mögliche Nachteile von Annotationen.....	1383

24 Logging und Monitoring.....1385

24.1	Die Logging-API.....	1385
24.1.1	Einfaches Logging.....	1385
24.1.2	Log-Level.....	1386
24.1.3	Logging in eine Datei.....	1386
24.2	Überwachen von Systemzuständen.....	1387
24.3	MBean-Typen, MBean-Server und weitere Begriffe.....	1388
24.3.1	MBeans des Systems.....	1389
24.4	Geschwätzige Programme und JConsole.....	1391
24.4.1	JConsole.....	1391
24.5	DerMBeanServer.....	1393
24.6	Eine eigene Standard-MBean.....	1394
24.6.1	Management-Schnittstelle.....	1394
24.6.2	Implementierung der managed Ressource.....	1394
24.6.3	Anmeldung beim Server.....	1395
24.6.4	Eigene Bean in JConsole.....	1395
24.6.5	JMX mit RMI-Adaptor.....	1397
24.7	Zum Weiterlesen.....	1399

25	Sicherheitskonzepte.....	1401
25.1	Zentrale Elemente der Java-Sicherheit.....	1401
25.1.1	Security-API der Java SE.....	1401
25.1.2	Cryptographic Service Providers.....	1402
25.2	Der Sandkasten (Sandbox).....	1403
25.3	Sicherheitsmanager (Security Manager).....	1403
25.3.1	Der Sicherheitsmanager bei Applets.....	1405
25.3.2	Sicherheitsmanager aktivieren.....	1406
25.3.3	Wie nutzen die Java-Bibliotheken den Sicherheitsmanager?.....	1407
25.3.4	Rechte durch Policy-Dateien vergeben.....	1408
25.3.5	Erstellen von Rechedateien mit dem grafischen Policy-Tool.....	1410
25.3.6	Kritik an den Policies.....	1410
25.4	Signierung.....	1412
25.4.1	Warum signieren?.....	1412
25.4.2	Digitale Ausweise und die Zertifizierungsstelle.....	1412
25.4.3	Mit keytool Schlüssel erzeugen.....	1413
25.4.4	Signieren mit jarsigner.....	1414
25.5	Digitale Unterschriften.....	1414
25.5.1	Die MDx-Reihe.....	1415
25.5.2	Secure Hash Algorithm (SHA).....	1415
25.5.3	Mit der Security-API einen Fingerabdruck berechnen.....	1416
25.5.4	Die Klasse MessageDigest.....	1416
25.5.5	Unix-Crypt.....	1418
25.6	Verschlüsseln von Daten(-strömen).....	1418
25.6.1	Den Schlüssel bitte.....	1418
25.6.2	Verschlüsseln mit Gpher.....	1420
25.6.3	Verschlüsseln von Datenströmen.....	1420
25.7	Zum Weiterlesen.....	1422
26.1	Java Native Interface und Invocation-API.....	1423
26.2	Einbinden einer C-Funktion in ein Java-Programm.....	1424
26.2.1	Schreiben des Java-Codes.....	1424
26.2.2	Compilieren des Java-Programms.....	1425
26.2.3	Erzeugen der Header-Datei.....	1425
26.2.4	Implementierung der Methode in C.....	1426
26.2.5	Übersetzen der C-Programme und Erzeugen der dynamischen Bibliothek.....	1427
26.2.6	Suchort der dynamischen Bibliothek.....	1429
26.3	Nativ die Stringlänge ermitteln.....	1429

26.4	Erweiterte JNI-Eigenschaften	1430
26.4.1	Klassendefinitionen	1430
26.4.2	Zugriff auf Attribute	1431
26.5	Einfache Anbindung von existierenden Bibliotheken	1433
26.5.1	C++ Klassen ansprechen	1433
26.5.2	COM-Schnittstellen anzapfen	1433
26.6	Zum Weiterlesen	1434
27.1	Die Werkzeuge im Überblick	1435
27.2	Java-Compiler	1435
27.2.1	Bytecode Compiler javac	1435
27.2.2	Native Compiler	1436
27.2.3	Java-Programme in ein natives ausführbares Programm einpacken	1437
27.3	Der Java-Interpreter java	1437
27.3.1	Der Unterschied zwischen java.exe und javaw.exe	1438
27.4	Das Archivformat Jar	1439
27.4.1	Das Dienstprogramm Jar benutzen	1439
27.4.2	Das Manifest	1442
27.4.3	Applikationen in Jar-Archiven starten	1442
27.4.4	Applets in Jar-Archiven	1443
27.5	Monitoringprogramme	1444
27.5.1	jps	1444
27.5.2	jstat	1444
27.5.3	jmap	1444
27.5.4	jstack	1445
27.6	Ant	1445
27.6.1	Bezug und Installation von Ant	1446
27.6.2	Properties	1447
27.6.3	Externe und vordefinierte Properties	1448
27.6.4	Weitere Ant-Tasks	1449
27.7	Decompiler und Obfuscatoren	1450
27.7.1	Der Decompiler Jad	1451
27.7.2	Das Obfuscator-Programm ProGuard	1451
27.8	Weitere Dienstprogramme	1452
27.8.1	Sourcecode Beautifier	1452
27.8.2	Java-Programme als Systemdienst ausführen	1453
27.9	Zum Weiterlesen	1454
A	Die Begleit-DVD	1455
	Index	1457