

Christian Ullenboom

Java

ist auch eine Insel



Inhalt

Vorwort **35**

Vorwort 3 - Update.....47

Vorwort Version 4.....48

1 Schon wieder eine neue Sprache? **49**

1.1 Der erste Kontakt.....51

1.2 Historischer Hintergrund.....51

1.3 Eigenschaften von Java.....53

1.3.1 Bytecode und die virtuelle Maschine.....53

1.3.2 Kein Präprozessor.....55

1.3.3 Keine überladenen Operatoren.....55

1.3.4 Zeiger und Referenzen.....56

1.3.5 Bring den Müll raus, Garbage-Collector.....57

1.3.6 Ausnahmenbehandlung.....57

1.3.7 Objektorientierung in Java.....58

1.3.8 Java-Security-Model.....58

1.3.9 Wofür Java nicht geeignet ist.....59

1.4 Java im Vergleich zu anderen Sprachen.....59

1.4.1 Java und JavaScript.....60

1.4.2 Normierungsversuche.....60

1.4.3 Die Rolle von Java im Web.....61

1.4.4 Vollwertige Applikationen statt Applets.....61

1.6 Entwicklungs- und Laufzeitumgebungen.....61

1.6.1 Aller Anfang mit dem Java SDK.....62

1.6.2 Die Entwicklungsumgebung von Sun: Sun ONE Studio und NetBeans.....62

1.6.3 IBM und Eclipse.....62

1.6.4 JBuildervon Borland.....63

1.6.5 Together.....64

1.6.6 Ein Wort zu Microsoft, Java und zu J++.....64

1.6.7 Direkt ausführbare Programme.....65

1.7 Installationsanleitung für das Java 2.....65

1.7.1 Das Java2 SDK beziehen.....66

1.7.2 JavaSDK unter Windows installieren.....66

1.7.3 Compiler und Interpreter nutzen.....67

1.7.4 Installation der Java Laufzeitumgebung (1.4) unter Linux.....68

1.8 Das erste Programm compilieren und testen.....69

1.8.1 Häufige Compiler-und Interpreterprobleme.....71

1.9	Eclipse.....	71
1.9.1	Eclipse starten.....	72
1.9.2	Das erste Projekt anlegen.....	73
1.9.3	Eine Klasse hinzufügen.....	74
1.9.4	Übersetzen und Ausführen.....	75
1.9.5	J2SDK statt JRE.....	75
1.9.6	Start eines Programmes ohne Speicheraufforderung.....	76
1.9.7	Java 5.0 in Eclipse.....	76
1.9.8	Projekt einfügen oder Workspace für die Aufgaben wechseln.....	77
1.9.9	Plugins für Eclipse.....	78

2 Sprachbeschreibung 79

2.1	Anweisungen und Programme.....	81
2.2	Elemente der Programmiersprache Java.....	83
2.2.1	Textkodierung durch Unicode-Zeichen.....	83
2.2.2	Unicode-Tabellen unter Windows.....	84
2.2.3	Literale.....	85
2.2.4	Bezeichner.....	85
2.2.5	Reservierte Schlüsselwörter.....	87
2.2.6	Token.....	88
2.2.7	Semantik.....	88
2.2.8	Kommentare.....	91
2.2.9	Funktionsaufrufe als Anweisungen.....	92
2.2.10	Die leere Anweisung.....	95
2.2.11	Der Block.....	95
2.3	Datentypen.....	95
2.3.1	Primitive Datentypen.....	96
2.3.2	Wahrheitswerte.....	98
2.3.3	Variablendeklarationen.....	98
2.3.4	Ganzzahlige Datentypen.....	100
2.3.5	Die Fließkommazahlen.....	102
2.3.6	Alphanumerische Zeichen.....	103
2.3.7	Die Typanpassung (das Casting).....	104
2.3.8	Lokale Variablen, Blöcke und Sichtbarkeit.....	108
2.3.9	Initialisierung von lokalen Variablen.....	110
2.4	Ausdrücke, Operanden und Operatoren.....	110
2.4.1	Zuweisungsoperator.....	111
2.4.2	Verbundoperatoren.....	112
2.4.3	Präfix- oder Postfix-Inkrement und -Dekrement.....	113
2.4.4	Unäres Minus und Plus.....	114
2.4.5	Arithmetische Operatoren.....	115
2.4.6	Die relationalen Operatoren.....	118
2.4.7	Logische Operatoren.....	119
2.4.8	Reihenfolge und Rang der Operatoren in der Auswertungsreihenfolge. . .	119
2.4.9	Überladenes Plus für Strings.....	122
2.4.10	Was C(++)-Programmierer vermissen könnten.....	123

2.5	Bedingte Anweisungen oder Fallunterscheidungen.....	123
2.5.1	Die if-Anweisung.....	124
2.5.2	Die Alternative wählen mit einer if/else-Anweisung.....	125
2.5.3	Die switch-Anweisung bietet die Alternative.....	128
2.6	Schleifen.....	131
2.6.1	Die while-Schleife.....	131
2.6.2	Schleifenbedingungen und Vergleiche mit ==.....	132
2.6.3	Die do/while-Schleife.....	133
2.6.4	Die for-Schleife.....	135
2.6.5	Ausbruch planen mit break und Wiedereinstieg mit continue.....	139
2.6.6	break und continue mit Sprungmarken.....	142
2.7	Methoden einer Klasse.....	142
2.7.1	Bestandteil einer Funktion.....	143
2.7.2	Aufruf.....	145
2.7.3	Methoden ohne Parameter.....	145
2.7.4	Statische Funktionen (Klassenmethoden).....	146
2.7.5	Parameter, Argument und Wertübergabe.....	147
2.7.6	Methoden vorzeitig mit return beenden.....	148
2.7.7	Nicht erreichbarer Quellcode bei Funktionen.....	149
2.7.8	Rückgabewerte.....	149
2.7.9	Methoden überladen.....	153
2.7.10	Vorgegebener Wert für nicht aufgeführte Argumente.....	155
2.7.11	Finale lokale Variablen.....	155
2.7.12	Rekursive Funktionen.....	157
2.7.13	Die Ackermann-Funktion.....	159
2.7.14	Die Türme von Hanoi.....	162
2.8	Dokumentationskommentare.....	163
2.8.1	Ein Dokumentationskommentar setzen.....	164
2.8.2	Mit javadoc eine Dokumentation erstellen.....	164
2.8.3	Generierte Dateien.....	166
2.8.4	Weitere Dokumentationskommentare.....	167
2.8.5	Schalter für das Programm javadoc.....	168
2.8.6	JavaDoc und Doclets.....	169
2.8.7	Bitoperationen.....	169
2.8.8	Vorzeichenlose Bytes in ein Integer und Char konvertieren.....	170
2.8.9	Variablen mit Xor vertauschen.....	171
2.8.10	Die Verschiebeoperatoren.....	172
2.8.11	Setzen, Löschen, Umdrehen und Testen von Bits.....	175
2.8.12	Der Bedingungsoperator.....	176
2.9	Einfache Benutzereingaben.....	178

3 Klassen und Objekte 181

3.1	Objektorientierte Programmierung.....	183
3.1.1	Warum überhaupt OOP?.....	183
3.1.2	Modularität und Wiederverwertbarkeit.....	184
3.2	Klassen benutzen.....	184
3.2.1	Die Klasse Point.....	185

3.2.2	Etwas über die UML.....	186
3.2.3	Anlegen eines Exemplars einer Klasse mit new.....	188
3.2.4	Zugriff auf Variablen und Methoden mit dem ».«.....	189
3.2.5	Konstruktoren.....	191
3.3	Import und Pakete.....	192
3.4	Die API-Dokumentation.....	193
3.5	Mit Referenzen arbeiten.....	194
3.5.1	Die null-Referenz.....	194
3.5.2	Zuweisungen bei Referenzen.....	195
3.5.3	Funktionen mit nichtprimitiven Parametern.....	196
3.5.4	Gleichheit von Objekten und die Methode equalsO.....	197
3.6	Arrays.....	200
3.6.1	Deklaration von Arrays.....	200
3.6.2	Arrays mit Inhalt.....	201
3.6.3	Die Länge eines Arrays über das Attribut length.....	202
3.6.4	Zugriff auf die Elemente.....	202
3.6.5	Array-Objekte erzeugen.....	203
3.6.6	Fehler bei Arrays.....	204
3.6.7	Arrays mit nicht-primitiven Elementen.....	205
3.6.8	Initialisierte Array-Objekte.....	206
3.6.9	Die erweiterte for-Schleife.....	206
3.6.10	Mehrdimensionale Arrays.....	208
3.6.11	Die Wahrheit über die Array-Initialisierung.....	210
3.6.12	Mehrere Rückgabewerte.....	211
3.6.13	Argument per Referenz übergeben.....	211
3.6.14	Arrays klonen.....	212
3.6.15	Feldinhalte kopieren.....	212
3.6.16	Die Klasse Arrays zum Vergleichen, Füllen, Suchen.....	213
3.6.17	Methode mit variabler Argumentanzahl (vararg).....	214
3.7	Der Einstiegspunkt für das Laufzeitsystem main().....	215
3.7.1	Kommandozeilen-Parameter ausgeben.....	216
3.7.2	Der Rückgabewert von mainO und System.exit().....	216
3.7.3	Parser der Kommandozeilenargumente Apache CLI.....	217
3.8	Eigene Pakete schnüren.....	218
3.8.1	Die package-Anweisung.....	219
3.8.2	Importieren von Klassen mit import.....	219
3.8.3	Paketnamen.....	220
3.8.4	Hierarchische Strukturen und das Default-Package.....	220
3.8.5	Klassen mit gleiche Namen in unterschiedlichen Paketen.....	221
3.8.6	Statische Imports.....	221
3.8.7	Eine Verzeichnisstruktur für eigene Projekte.....	222

4 Der Umgang mit Zeichenketten

223

4.1	Strings und deren Anwendung.....	225
4.1.1	String-Objekte für konstante Zeichenketten.....	225
4.1.2	String-Länge.....	227

4.1.3	Cut, dass wir verglichen haben.....	227
4.1.4	String-Teile extrahieren.....	229
4.1.5	Suchen und Ersetzen.....	231
4.1.6	Veränderte Strings liefern.....	234
4.1.7	Unterschiedliche Typen in Zeichenketten konvertieren.....	236
4.2	Veränderbare Zeichenketten mit StringBuffer/StringBuilder.....	237
4.2.1	Anlegen von StringBuffer-Objekten.....	238
4.2.2	Die Länge eines StringBuffer-Objekts lesen und setzen.....	239
4.2.3	Daten anhängen.....	239
4.2.4	Zeichen(folgen) setzen, erfragen, löschen und umdrehen.....	240
4.3	Vergleiche von Zeichenketten.....	241
4.3.1	equalsO in String und StringBuffer.....	241
4.3.2	Sollte es ein equalsO und hash() bei StringBuffer geben?.....	242
4.3.3	Sprachabhängiges Vergleichen mit der Collator-Klasse.....	242
4.3.4	Effiziente interne Speicherung für die Sortierung.....	244
4.4	Zeichenkodierungen.....	246
4.4.1	Standard-Encodings.....	246
4.4.2	Base64-Kodierung.....	247
4.5	Reguläre Ausdrücke.....	248
4.6	Zerlegen von Zeichenketten.....	249
4.6.1	StringTokenizer.....	249
4.6.2	Splitten von Zeichenketten mit splitO aus Pattern.....	251
4.6.3	splitO in String.....	252
4.6.4	Die neue Klasse Scanner.....	253
4.6.5	Der Breakliterator als Wort- und Satztrenner.....	254
4.7	Formatieren von Ausgaben.....	256
4.7.1	Prozente, Zahlen und Währungen ausgeben mit NumberFormat.....	257
4.7.2	Ausgaben formatieren mit MessageFormat.....	258
4.7.3	Dezimalzahlformatierung mit Decimal Format.....	260
4.7.4	Formatieren mit formatO.....	262

5 Mathematisches 265

5.1	Arithmetik in Java.....	267
5.1.1	Java-Sondertypen im Beispiel.....	267
5.1.2	Soll eine Division durch Null zur Übersetzungszeit erkannt werden?.....	268
5.2	Die Funktionen der Math-Klasse.....	269
5.2.1	Attribute.....	269
5.2.2	Winkelfunktionen (trigonometrische Funktionen und Arcus-Funktionen) .	269
5.2.3	Runden von Werten.....	270
5.2.4	Wurzel und Exponentialfunktionen.....	271
5.2.5	Der Logarithmus.....	272
5.2.6	Rest der ganzzahligen Division.....	272
5.2.7	Absolutwerte und Maximum, Minimum.....	273
5.2.8	Zufallszahlen.....	273

5.3	Mathe bitte strikt.....	274
5.3.1	Strikt Fließkomma mit strictfp	274
5.3.2	Die Klassen Math und StrictMath.....	274
5.4	Die Random-Klasse	275
5.5	Große Zahlen	276
5.5.1	Die Klasse BigInteger.....	277
5.5.2	Funktionen von BigInteger.....	279
5.5.3	Ganz lange Fakultäten.....	281
5.5.4	BigDecimal.....	282
5.6	Rechnen mit Einheiten: Java Units Specification	282

6 Eigene Klassen schreiben 285

6.1	Eigene Klassen definieren	287
6.1.1	Methodenaufrufe und Nebeneffekte.....	290
6.1.2	Argumentübergabe mit Referenzen.....	290
6.1.3	Die this-Referenz.....	291
6.1.4	Überdeckte Objektvariablen nutzen.....	292
6.2	Assoziationen zwischen Objekten	293
6.3	Privatsphäre und Sichtbarkeit	294
6.3.1	Wieso nicht freie Methoden und Variablen für alle?.....	296
6.3.2	Privat ist nicht ganz privat. Es kommt darauf an, wer's sieht.....	296
6.3.3	Zugriffsmethoden für Attribute definieren.....	297
6.3.4	Zusammenfassung zur Sichtbarkeit.....	300
6.3.5	Sichtbarkeit in der UML.....	300
6.4	Statische Methoden und Attribute	301
6.4.1	Warum statische Eigenschaften sinnvoll sind.....	301
6.4.2	Statische Eigenschaften mit static.....	302
6.4.3	Statische Eigenschaften als Objekteigenschaften nutzen.....	302
6.4.4	Statische Eigenschaften und Objekteigenschaften.....	303
6.4.5	Statische Variablen zum Datenaustausch.....	303
6.4.6	Warum die Groß- und Kleinschreibung wichtig ist.....	305
6.4.7	Statische Blöcke.....	305
6.5	Konstanten und Aufzählungen	306
6.5.1	Konstanten mit dem Schlüsselwort final bei Variablen.....	306
6.5.2	Problem mit finalen Klassenvariablen.....	308
6.5.3	Typsicherere Konstanten.....	308
6.5.4	Aufzählungen und enum in Java 5.....	310
6.5.5	enum-Konstanten in switch.....	310
6.6	Objekte anlegen und zerstören	311
6.6.1	Konstruktoren schreiben.....	311
6.6.2	Einen anderen Konstruktor der gleichen Klasse aufrufen.....	315
6.6.3	Initialisierung der Objekt- und Klassenvariablen.....	318
6.6.4	Finale Werte im Konstruktor setzen.....	320
6.6.5	Exemplarinitialisierer (Instanzinitialisierer).....	320

6.6.6	Zerstörung eines Objekts durch den AAüllaufsammler.....	323
6.6.7	Implizit erzeugte String-Objekte.....	324
6.6.8	Private Konstruktoren, Utility-Klassen, Singleton und Fabriken.....	325
6.6.9	Zusammenfassung: Konstruktoren und Methoden.....	327
6.7	Vererbung.....	327
6.7.1	Vererbung in Java.....	327
6.7.2	Einfach- und AAehrfachvererbung.....	328
6.7.3	Gebäude modelliert.....	328
6.7.4	Konstruktoren in der Vererbung.....	329
6.7.5	Sichtbarkeit.....	332
6.7.6	Das Substitutionsprinzip.....	333
6.7.7	Automatische und explizite Typanpassung.....	334
6.7.8	Array-Typen.....	335
6.7.9	Finale Klassen.....	336
6.7.10	Unterklassen prüfen mit dem Operator instanceof.....	336
6.7.11	Methoden überschreiben.....	337
6.7.12	super: Aufrufen einer Methode aus der Oberklasse.....	338
6.7.13	Nicht überschreibbare Funktionen.....	341
6.8	Die oberste aller Klassen: Object.....	341
6.8.1	Klassenobjekte.....	342
6.8.2	Objektidentifikation mit toStringO.....	342
6.8.3	Objektgleichheit mit equalsO und Identität.....	343
6.8.4	Klonen eines Objekts mit clone().....	346
6.8.5	Hashcodes.....	349
6.8.6	Aufräumen mit finalizeO.....	350
6.8.7	Synchronisation.....	350
6.9	Die Oberklasse gibt Funktionalität vor.....	351
6.9.1	Dynamisches Binden als Beispiel für Polymorphie.....	353
6.9.2	Keine Polymorphie bei privaten, statischen und finalen Methoden.....	354
6.9.3	Polymorphie bei Konstruktoraufrufen.....	356
6.10	Abstrakte Klassen.....	358
6.10.1	Abstrakte Klassen.....	358
6.10.2	Abstrakte Methoden.....	359
6.10.3	Über abstract final.....	363
6.11	Schnittstellen.....	364
6.11.1	Ein Polymorphie-Beispiel mit Schnittstellen.....	366
6.11.2	Die Mehrfachvererbung bei Schnittstellen.....	368
6.11.3	Erweitern von Interfaces - Subinterfaces.....	370
6.11.4	Vererbte Konstanten bei Schnittstellen.....	370
6.11.5	Vordefinierte Methoden einer Schnittstelle.....	372
6.11.6	CharSequence als Beispiel einer Schnittstelle.....	374
6.11.7	Die Schnittstelle Iterable.....	376
6.12	Innere Klassen.....	378
6.12.1	Statische innere Klassen und Schnittstellen.....	378
6.12.2	Mitglieds- oder Elementklassen.....	379
6.12.3	Lokale Klassen.....	382
6.12.4	Anonyme innere Klassen.....	382
6.12.5	Eine Sich-Selbst-Implementierung.....	386
6.12.6	this und Vererbung.....	387

6.12.7	Implementierung einer verketteten Liste.....	388
6.12.8	Funktionszeiger.....	390
6.13	Generische Datentypen.....	392
6.13.1	Einfache Klassenschablonen.....	393
6.13.2	Einfache Methodenschablonen.....	394
6.13.3	Generics und Vererbung.....	394
6.13.4	Einschränken der Typen.....	395
6.13.5	Umsetzen der Generics, Typlöschung und Raw-Types.....	396
6.13.6	Wildcards.....	398
6.14	Die Spezial-Oberklasse Enum.....	399
6.14.1	Methoden auf Enum-Objekten.....	399
6.14.2	enum mit eigenen Konstruktoren und Methoden.....	400
6.15	Gegenseitige Abhängigkeiten von Klassen.....	403
6.16	Veraltete (deprecated) Methoden/Konstruktoren.....	403

7 Exceptions 407

7.1	Problembereiche einzäunen.....	409
7.1.1	Exceptions in Java mit try und catch.....	409
7.1.2	Eine Datei auslesen mit RandomAccessFile.....	410
7.1.3	Ablauf einer Ausnahmesituation.....	411
7.1.4	Wiederholung kritischer Bereiche.....	411
7.1.5	throws im Methodenkopf angeben.....	412
7.1.6	Abschließende Arbeiten mit finally	413
7.1.7	Nicht erreichbare catch-Klauseln.....	415
7.2	Die Klassenhierarchie der Fehler.....	416
7.2.1	Die Exception-Hierarchie.....	416
7.2.2	Oberausnahmen fangen.....	417
7.2.3	Alles geht als Exception durch.....	418
7.2.4	Ausnahmen, die nicht aufgefangen werden müssen: RuntimeException ...	419
7.2.5	Harte Fehler: Error.....	420
7.3	Werfen eigener Exceptions.....	420
7.3.1	Typecast auf ein null-Objekt für eine NullPointerException.....	421
7.3.2	Neue Exception-Klassen definieren.....	422
7.4	Rückgabewerte bei ausgelösten Ausnahmen.....	424
7.5	Stack-Aufruf analysieren.....	425
7.6	Assertions.....	426
7.6.1	Assertions in eigenen Programmen nutzen.....	426
7.6.2	Assertions aktivieren.....	427

8 Die Funktionsbibliothek 429

8.1	Die Java-Klassenphilosophie.....	431
8.1.1	Übersicht über die Pakete der Standardbibliothek.....	431

8.2	Wrapper-Klassen.....	437
8.2.1	Die Character-Klasse.....	438
8.2.2	Die Boolean-Klasse.....	441
8.2.3	Die Basisklasse Number für numerische Wrapper-Objekte.....	442
8.2.4	Die Klasse Integer.....	443
8.2.5	Behandlung von Überlauf.....	445
8.2.6	Unterschiedliche Ausgabeformate.....	446
8.2.7	Autoboxing: Boxing und Unboxing.....	447
8.3	Benutzereinstellungen.....	450
8.3.1	Eine zentrale Registry.....	450
8.3.2	Einträge einfügen, auslesen und löschen.....	451
8.3.3	Auslesen der Daten und schreiben in anderem Format.....	453
8.3.4	Auf Ereignisse horchen.....	453
8.4	Systemeigenschaften der Java-Umgebung.....	453
8.4.1	line.Separator.....	454
8.4.2	Browser-Version abfragen.....	454
8.4.3	Property von der Konsole aus setzen.....	455
8.4.4	Umgebungsvariablen des Betriebssystems.....	456
8.5	Ausführung von externen Programmen.....	456
8.5.1	DOS-Programme aufrufen.....	458
8.5.2	Die Windows-Registry verwenden.....	460
8.5.3	Einen HTML-Browser unter Windows aufrufen.....	461
8.6	Klassenlader (Class Loader).....	462
8.6.1	Woher die kleinen Klassen kommen.....	462
8.6.2Drei	
	Typen von Klassenladern.....	463
8.6.3	Der Java.lang.Class Loader.....	464
8.6.4	Hot Deployment mit dem URL-ClassLoader.....	464
8.6.5	Das jre/lib/endorsed-Verzeichnis.....	465
8.6.6	Wie heißt die Klasse mit der Methode mainO?.....	466
8.7	Zeitmessung und Profiling.....	467

9 Threads und nebenläufige Programmierung 469

9.1	Prozesse und Threads.....	471
9.1.1	Wie parallele Programme die Geschwindigkeit steigern können.....	472
9.2	Threads erzeugen.....	474
9.2.1	Threads über die Schnittstelle Runnable implementieren.....	474
9.2.2	Threads über Runnable starten.....	475
9.2.3	Die Klasse Thread erweitern.....	477
9.2.4	Erweitern von Thread oder Implementieren von Runnable?.....	479
9.3	Die Zustände eines Threads.....	480
9.3.1	Das Ende eines Threads.....	480
9.3.2	Einen Thread höflich mit Interrupt beenden.....	481
9.3.3	Der stopO von außen.....	483
9.3.4	Das ThreadDeath-Objekt.....	484

9.3.5	Auf das Ende warten mitjoin().....	485
9.3.6	Threads schlafen.....	487
9.3.7	Eine Zeituhr.....	488
9.4	Arbeit niederlegen und wieder aufnehmen.....	489
9.5	Priorität.....	489
9.5.1	Threads hoher Priorität und das AWT.....	490
9.5.2	Cranularität und Vorrang.....	490
9.6	Dämonen.....	491
9.7	Kooperative und nichtkooperative Threads.....	492
9.8	Synchronisation über kritische Abschnitte.....	493
9.8.1	Gemeinsam genutzte Daten.....	493
9.8.2	Probleme beim gemeinsamen Zugriff und kritische Abschnitte.....	494
9.8.3	Punkte parallel initialisieren.....	494
9.8.4	i++ sieht atomar aus, ist es aber nicht.....	495
9.8.5	Abschnitte mit synchronized schützen.....	496
9.8.6	Monitore.....	497
9.8.7	Synchronized-Methode am Beispiel der Klasse StringBuffer.....	497
9.8.8	Synchronisierte Blöcke.....	498
9.8.9	Vor- und Nachteile von synchronisierten Blöcken und Methoden.....	499
9.8.10	Nachträglich synchronisieren.....	500
9.8.11	Monitore sind reentrant, gut für die Geschwindigkeit.....	500
9.8.12	Deadlocks.....	501
9.8.13	Erkennen von Deadlocks.....	503
9.9	Synchronisation über Warten und Benachrichtigen.....	503
9.9.1	Warten mit wait() und Aufwecken mit notifyO.....	504
9.9.2	Falls der Lock fehlt: WegaIMonitorStateException.....	506
9.9.3	Mehrere Wartende und notifyAHO.....	507
9.9.4	waitO mit einer Zeitspanne.....	507
9.9.5	Beispiel Erzeuger-Verbraucher-Programm.....	508
9.9.6	Semaphoren.....	511
9.10	Atomares und frische Werte mit volatile.....	512
9.10.1	Das Paket java.util.concurrent.atomic.....	514
9.11	Aktive Threads in der Umgebung.....	514
9.12	Gruppen von Threads in einer Thread-Gruppe.....	514
9.12.1	Etwas über die aktuelle Thread-Gruppe herausfinden.....	515
9.12.2	Threads in einer Thread-Gruppe anlegen.....	518
9.12.3	Methoden von Thread und ThreadGroup im Vergleich.....	520
9.13	Die Klassen Timer und TimerTask.....	522
9.13.1	Job-Scheduler Quartz.....	523
9.14	Einen Abbruch der virtuellen Maschine erkennen.....	523

10 Raum und Zeit

525

10.1	Greenwich Mean Time (GMT).....	527
10.2	Wichtige Datum-Klassen im Überblick.....	528

11.6.6	Aufzählen der Elemente.....	590
11.6.7	Der Gleichheitstest und der Hash-Wert einer Hash-Tabelle.....	591
11.6.8	Klonen.....	592
11.7	Die Properties-Klasse.....	592
11.7.1	Properties setzen und lesen.....	592
11.7.2	Properties verketten.....	593
11.7.3	Eigenschaften ausgeben.....	594
11.7.4	Hierarchische Eigenschaften.....	595
11.7.5	Properties speichern.....	595
11.7.6	Über die Beziehung Properties und Hashtable.....	597
11.8	Mengen (Sets).....	597
11.8.1	HashSet.....	598
11.8.2	TreeSet-die Menge durch Bäume.....	599
11.9	Algorithmen in Collections.....	600
11.9.1	Datenmanipulation: Umdrehen, Füllen, Kopieren.....	601
11.9.2	Vergleichen von Objekten mit Comparator und Comparable.....	602
11.9.3	Größten und kleinsten Wert einer Collection finden.....	603
11.9.4	Sortieren.....	604
11.9.5	nCopies().....	606
11.9.6	Singletons.....	607
11.9.7	Elemente in der Collection suchen.....	607
11.10	Synchronisation der Datenstrukturen.....	609
11.11	Die abstrakten Basisklassen für Container.....	610
11.11.1	Optionale Methoden.....	611
11.12	Die Klasse BitSet für Bitmengen.....	611
11.12.1	Ein BitSet anlegen und füllen.....	611
11.12.2	Mengenorientierte Operationen.....	612
11.12.3	Funktionsübersicht.....	612
11.12.4	Primzahlen in einem BitSet verwalten.....	613
11.13	Ein Design-Pattern durch Beobachten von Änderungen.....	614
11.13.1	Design-Pattern.....	615
11.13.2	Das Beobachter-Pattern (Observer/Observable).....	615

12 Datenströme und Dateien

621

12.1	Datei und Verzeichnis.....	623
12.1.1	Dateien und Verzeichnisse mit der Klasse File.....	624
12.1.2	Dateieigenschaften und -attribute.....	625
12.1.3	Dateien berühren, neue Dateien anlegen.....	627
MA A	Umbenennen und Verzeichnisse anlegen.....	628
12.1.5	Die Wurzel aller Verzeichnisse/Laufwerke.....	628
12.1.6	Verzeichnisse listen und Dateien filtern.....	630
12.1.7	Dateien und Verzeichnisse löschen.....	633
12.1.8	Implementierungsmöglichkeiten für die Klasse File.....	634
MAS	Verzeichnisse nach Dateien rekursiv durchsuchen.....	635
12.1.10	Sicherheitsprüfung.....	637

12.1.11	Namen der Laufwerke.....	637
12.1.12	Locking.....	637
12.2	Dateien mit wahlfreiem Zugriff.....	638
12.2.1	Ein RandomAccessFile öffnen.....	638
12.2.2	Aus dem RandomAccessFile lesen.....	639
12.2.3	Schreiben.....	640
12.2.4	Die Länge des RandomAccessFile.....	640
12.2.5	Hin und her in der Datei.....	641
12.3	Stream-Klassen und Reader/Writer.....	642
12.3.1	Die abstrakten Basisklassen.....	642
12.3.2	Übersicht über Ein-/Ausgabeklassen.....	642
12.4	Binäre Ein-/Ausgabe-Klassen InputStream und OutputStream.....	644
12.4.1	Die Klasse OutputStream.....	644
12.4.2	Ein Datenschlucken.....	646
12.4.3	Anwendung der Klasse FileOutputStream.....	646
12.4.4	Die Eingabeklasse InputStream.....	647
12.4.5	Anwenden der Klasse FileInputStream.....	648
12.4.6	Kopieren von Dateien.....	650
12.5	Daten filtern durch FilterInputStream und FilterOutputStream.....	652
12.5.1	Der besondere Filter PrintStream.....	653
12.5.2	Die Schnittstelle Appendable.....	654
12.5.3	System.in und System.out.....	655
12.6	Besondere OutputStream- und InputStream-Klassen.....	660
12.6.1	Bytes in den Strom schreiben mit ByteArrayOutputStream.....	660
12.6.2	Bytes in den Strom schreiben mit ByteArrayInputStream.....	661
12.6.3	Ströme zusammensetzen mit SequenceInputStream.....	661
12.7	Ressourcen wie Grafiken aus dem Klassenpfad und aus Jar-Archiven laden.....	664
12.8	Die Unterklassen von Writer.....	664
12.8.1	Die abstrakte Klasse Writer.....	665
12.8.2	Datenkonvertierung durch den OutputStreamWriter.....	667
12.8.3	In Dateien schreiben mit der Klasse FileWriter.....	668
12.8.4	StringWriter und CharArrayWriter.....	669
12.8.5	Writerals Filter verketteten.....	671
12.8.6	Gepufferte Ausgabe durch BufferedWriter.....	672
12.8.7	Ausgabemöglichkeiten durch PrintWriter erweitern.....	675
12.8.8	Daten mit FilterWriter filtern.....	676
12.9	Die Klassen um Reader.....	682
12.9.1	Die abstrakte Basisklasse Reader.....	682
12.9.2	Automatische Konvertierungen mit dem InputStreamReader.....	684
12.9.3	Dateien lesen mit der Klasse FileReader.....	685
12.9.4	StringReader und CharArrayReader.....	686
12.10	Die Filter für Zeichenströme.....	688
12.10.1	Gepufferte Eingaben mit der Klasse BufferedReader.....	688
12.10.2	LineNumberReader zählt automatisch Zeilen mit.....	689
12.10.3	Eingaben filtern mit der Klasse FilterReader.....	691
12.10.4	Daten mit der Klasse PushbackReader zurücklegen.....	693

12.11	Kommunikation zwischen Threads mit Pipes.....	696
12.11.1	PipedOutputStream und PipedInputStream.....	696
12.11.2	PipedWriter und PipedReader.....	698
12.12	Datenkompression.....	700
12.12.1	Die Java-Unterstützung beim Komprimieren und Zusammenpacken,	701
12.12.2	Datenströme komprimieren.....	701
12.12.3	Zip-Archive.....	704
12.12.4	Jar-Archive.....	712
12.13	Prüfsummen.....	712
12.13.1	Die Schnittstelle Checksum.....	713
12.13.2	Die Klasse CRC32.....	713
12.13.3	Die Adler32-Klasse.....	715
12.14	Persistente Objekte und Serialisierung.....	716
12.14.1	Objekte speichern mit der Standard-Serialisierung.....	716
12.14.2	Objekte über die Standard-Serialisierung lesen.....	719
12.14.3	Die Schnittstelle Serializable.....	721
12.14.4	Nicht serialisierbare Attribute mit transient aussparen.....	722
12.14.5	Das Abspeichern selbst in die Hand nehmen.....	723
12.14.6	Tiefe Objektkopien.....	726
12.14.7	Versionenverwaltung und die SUID.....	727
12.14.8	Wie die ArrayList serialisiert.....	730
12.14.9	Serialisieren in XML-Dateien.....	730
12.14.10	XML-API von Sun.....	731
12.15	Zugriff auf SMB-Server.....	732
12.15.1	jCIFS.....	733
12.16	Tokenizer.....	733
12.16.1	StreamTokenizer.....	733
12.16.2	CSV (Comma Separated Values)-Dateien verarbeiten.....	737
12.17	Die Logging-API.....	737

13 Die extensible Markup Language (XML) 741

13.1	Auszeichnungssprachen.....	743
13.1.1	Die Standard Generalized Markup Language (SGML).....	743
13.1.2	Extensible Markup Language (XML).....	744
13.2	Eigenschaften von XML-Dokumenten.....	744
13.2.1	Elemente und Attribute.....	744
13.2.2	Beschreibungssprache für den Aufbau von XML-Dokumenten.....	746
13.2.3	Schema-eine Alternative zu DTD.....	749
13.2.4	Namensraum (Namespace).....	752
13.2.5	XML-Applikationen.....	753
13.3	Die Java-APIs für XML.....	753
13.3.1	Das Document Object Model (DOM).....	753
13.3.2	Simple API for XML Parsing (SAX).....	754
13.3.3	Java Document Object Model (JDOM).....	754

13.4	XML-Dateien mit JDOM verarbeiten.....	754
13.4.1	JDOM beziehen.....	755
13.4.2	Paketübersicht.....	755
13.4.3	Die Document-Klasse.....	757
13.4.4	Eingaben aus der Datei lesen.....	757
13.4.5	Das Dokument als XML-Datei ausgeben.....	758
13.4.6	Der Dokumenttyp.....	758
13.4.7	Elemente.....	759
13.4.8	Zugriff auf Elementinhalte.....	761
13.4.9	Liste mit Unterelementen erzeugen.....	763
13.4.10	Neue Elemente einfügen und ändern.....	764
13.4.11	Attributinhalte lesen und ändern.....	767
13.4.12	Kurzeinführung in XPath.....	769
13.5	JAXP als Java-Schnittstelle zu XML.....	771
13.5.1	Einführung in XSLT.....	772
13.5.2	Umwandlung von XML-Dateien mit JDOM und JAXP.....	774
13.6	Serielle Verarbeitung von XML mit SAX.....	775
13.6.1	Schnittstellen von SAX.....	775
13.6.2	SAX-Parser erzeugen.....	776
13.6.3	Wichtigsten Methoden der Schnittstelle ContentHandler.....	776
13.7	HTML-Dokumenten mit JTiny und cyberneko einlesen.....	778

14 Grafikprogrammierung mit dem AWT 779

14.1	Das Abstract-Window-Toolkit.....	781
14.1.1	Java Foundation Classes.....	781
14.2	Das Toolkit.....	782
14.3	Fenster unter grafischen Oberflächen.....	783
14.3.1	AWT-Fenster darstellen.....	783
14.3.2	Swing-Fenster darstellen.....	784
14.3.3	Sichtbarkeit des Fensters.....	785
14.3.4	Größe und Position des Fensters verändern.....	785
14.3.5	Hauptprogramm von Frame/JFrame ableiten.....	786
14.3.6	Fenster- und Dialog-Dekoration.....	787
14.4	Grundlegendes zum Zeichnen.....	788
14.4.1	Die paintO-Methode für das AWT-Frame.....	788
14.4.2	Zeichen von Inhalten mit JFrame.....	789
14.4.3	Auffordern zum Neuzeichnen mit repaintO.....	790
14.4.4	Fensterinhalte ändern und die ereignisorientierte Programmierung.....	791
14.5	Einfache Zeichenfunktionen.....	794
14.5.1	Linien.....	795
14.5.2	Rechtecke.....	795
14.5.3	Ovale und Kreisbögen.....	796
14.5.4	Polygone und Polylines.....	797

14.6	Zeichenketten schreiben.....	800
14.6.1	Einen neuen Zeichensatz bestimmen.....	801
14.6.2	Ableiten eines neuen Fonts aus einem gegebenen Font.....	803
14.6.3	Zeichensätze des Systems ermitteln.....	803
14.6.4	Die Klasse FontMetrics.....	805
14.6.5	TrueType Fonts.....	807
14.7	Clipping-Operationen.....	809
14.8	Farben.....	812
14.8.1	Zufällige Farbblocke zeichnen.....	813
14.8.2	Farbanteile zurückgeben.....	814
14.8.3	Vordefinierte Farben.....	815
14.8.4	Farben aus Hexadezimalzahlen erzeugen.....	815
14.8.5	Einen helleren oder dunkleren Farbton wählen.....	817
14.8.6	Farbmodelle HSB und RGB.....	818
14.8.7	Die Farben des Systems.....	819
14.9	Bilder anzeigen und Grafiken verwalten.....	822
14.9.1	Bilder laden: eine Übersicht.....	822
14.9.2	Bilder über Toolkit laden.....	823
14.9.3	Das Image zeichnen.....	825
14.9.4	Grafiken zentrieren.....	827
14.9.5	Bilder im Speicher erzeugen.....	828
14.9.6	Kein Flackern durch Double-Buffering.....	829
14.9.7	Bilder skalieren.....	831
14.9.8	VolatileImage.....	833
14.9.9	Programm-Icon/Fenster-Icon setzen.....	833
14.9.10	Bilder lesen mit ImageIO.....	834
14.9.11	Schreiben mit ImageIO.....	835
14.9.12	Kann ImageIO ein Format behandeln?.....	835
14.9.13	Komprimieren mit ImageIO.....	836
14.9.14	Bilder im GIF-Format speichern.....	837
14.9.15	Gif speichern mit dem ACME-Paket.....	839
14.9.16	JPEG-Dateien mit dem Sun-Paket schreiben.....	840
14.9.17	Java Image Management Interface (JIMI).....	842
14.10	Java2D-API.....	844
14.10.1	Grafische Objekte zeichnen.....	844
14.10.2	Geometrische Objekte durch Shape gekennzeichnet.....	846
14.10.3	Eigenschaften geometrischer Objekte.....	849
14.10.4	Transformationen mit einem AffineTransform-Objekt.....	856
14.11	Von Produzenten, Konsumenten und Beobachtern.....	858
14.11.1	Producer und Consumer für Bilder.....	858
14.11.2	Beispiel für die Übermittlung von Daten.....	859
14.11.3	Bilder selbst erstellen.....	862
14.11.4	Die Bildinformationen wieder auslesen.....	865
14.12	Filter.....	868
14.12.1	Grundlegende Eigenschaft von Filtern.....	868
14.12.2	Konkrete Filterklassen.....	868
14.12.3	Mit CroplmageFilter Teile ausschneiden.....	870
14.12.4	Transparenz.....	871

14.13	Drucken.....	871
14.13.1	Drucken mit dem einfachen Ansatz.....	871
14.13.2	Ein PrintJob.....	872
14.13.3	Drucken der Inhalte.....	874
14.13.4	Komponenten drucken.....	874
14.13.5	Den Drucker am Parallelport ansprechen.....	875
14.13.6	Bekannte Drucker.....	875
14.14	Graphic Layers Framework.....	876
14.15	Grafikverarbeitung ohne grafische Oberfläche.....	876
14.15.1	Xvfb-Server.....	876
14.15.2	Pure Java AWT Toolkit (PJA).....	877

15 Komponenten, Container und Ereignisse 879

15.1	Es tut sich was - Ereignisse beim AWT.....	881
15.1.1	Was ist ein Ereignis?.....	881
15.1.2	Die Klasse AWTEvent.....	881
15.1.3	Events auf verschiedenen Ebenen.....	882
15.1.4	Ereignisquellen,-senken und Horcher (Listener).....	884
15.1.5	Listener implementieren.....	884
15.1.6	Listener bei Ereignisauslöser anmelden/abmelden.....	885
15.1.7	Aufrufen der Listener.....	886
15.2	Varianten, das Fenster zu schließen.....	886
15.2.1	Eine Klasse implementiert die Schnittstelle WindowListener.....	887
15.2.2	Adapterklassen nutzen.....	889
15.2.3	Innere Mitgliedsklassen und innere anonyme Klassen.....	890
15.2.4	Generic Listener.....	891
15.3	Komponenten im AWT und in Swing.....	892
15.3.1	Peer-Klassen und Lightweight-Komponenten.....	893
15.3.2	Die Basis aller Komponenten: Component und JComponent.....	893
15.3.3	Proportionales Vergrößern eines Fensters.....	897
15.3.4	Dynamisches Layout während einer Größenänderung.....	898
15.3.5	Hinzufügen von Komponenten.....	899
15.4	Das Swing-Fenster JFrame.....	900
15.4.1	Kinder auf einem Swing-Fenster.....	900
15.4.2	Schließen eines Swing-Fensters.....	900
15.4.3	JWindow und JDialog.....	901
15.5	Informationstext über die Klasse JLabel.....	902
15.5.1	Mehrzelliger Text, HTML in der Darstellung.....	904
15.6	Die Klasse ImageIcon.....	905
15.6.1	Die Schnittstelle Icon.....	907
15.6.2	Was Icon und Image verbindet.....	908
15.7	Eine Schaltfläche (JButton).....	909
15.7.1	Der aufmerksame Action Listener.....	911
15.7.2	Generic Listener für Schaltflächen-Ereignisse verwenden.....	912

15.7.3	AbstractButton.....	915
15.7.4	JToggleButton.....	917
15.8	Tooltips.....	917
15.9	Der Container JPanel.....	918
15.10	Alles Auslegungssache: die Layoutmanager.....	918
15.10.1	FlowLayout.....	920
15.10.2	BorderLayout.....	922
15.10.3	GridLayout.....	924
15.10.4	Der GridBagLayout-Manager.....	926
15.10.5	Null-Layout.....	931
15.10.6	Weitere Layoutmanager.....	931
15.11	Horizontale und vertikale Schieberegler.....	932
15.11.1	Der AdjustmentListener, der auf Änderungen hört.....	935
15.12	JSlider.....	936
15.13	Ein Auswahlménú - Choice, JComboBox.....	937
15.13.1	ItemListener.....	940
15.13.2	Zuordnung einer Taste mit einem Eintrag.....	941
15.13.3	DateComboBox.....	942
15.14	Eines aus vielen - Kontrollfelder (JCheckBox).....	943
15.14.1	Ereignisse über ItemListener.....	944
15.15	Kontrollfeldgruppen, Optionsfelder und JRadioButton.....	945
15.16	Der Fortschrittsbalken JProgressBar.....	947
15.17	Rahmen (Borders).....	949
15.18	Symbolleisten alias Toolbars.....	951
15.19	Menüs.....	953
15.19.1	Die Menüleisten und die Einträge.....	953
15.19.2	Menüeinträge definieren.....	954
15.19.3	Mnemonics und Shortcuts (Accelerator).....	955
15.19.4	Beispiel für ein Programm mit Menüleisten.....	957
15.19.5	Popup-Menüs.....	962
15.20	Das Konzept des AModel-View-Controllers.....	965
15.21	List-Boxen.....	967
15.22	JSpinner.....	969
15.23	Texteingabefelder.....	971
15.23.1	Text in einer Eingabezeile.....	971
15.23.2	Die Oberklasse der JText-Komponenten: JTextComponent.....	972
15.23.3	JPasswordField.....	973
15.23.4	Validierende Eingabefelder.....	974
15.23.5	Mehrzellige Textfelder.....	975
15.23.6	Die Editor-Klasse JEditorPane.....	978
15.24	Bäume mit JTree-Objekten.....	980
15.24.1	Selektionen bemerken.....	981
15.25	Tabellen mit JTable.....	982
15.25.1	Ein eigenes Tabellen-Model.....	984
15.25.2	AbstractTableModel.....	984

15.25.3	DefaultTableModel.....	988
15.25.4	Ein eigener Renderer für Tabellen.....	989
15.25.5	Zeil-Editoren.....	992
15.25.6	Größe und Umrandung der Zellen.....	993
15.25.7	Spalteninformationen.....	994
15.25.8	Tabellenkopf von Swing-Tabellen.....	994
15.25.9	Selektionen einer Tabelle.....	995
15.25.10	Ein professionelles Tabellenlayout mit JGrid.....	996
15.26	JRootPane, JLayeredPane und JDesktopPane.....	996
15.26.1	JRootPane und JLayeredPane.....	996
15.26.2	JDesktopPane und die Kinder JInternalFrame.....	997
15.26.3	Der Farbauswahldialog JColorChooser.....	1000
15.26.4	Der Dateiauswahldialog.....	1001
15.27	Flexibles Java-Look&Feel.....	1004
15.28	Swing-Beschriftungen einer anderen Sprache geben.....	1006
15.29	Die Zwischenablage (Clipboard).....	1007
15.30	Undo durchführen.....	1010
15.31	Ereignisverarbeitung auf unterster Ebene.....	1012
15.32	AWT, Swing und die Threads.....	1013
15.32.1	Warum Swing nicht Thread-sicher ist.....	1014
15.32.2	Swing-Elemente bedienen mit invokeLaterO und invokeAndWaitO.....	1017
15.33	Selbst definierte Cursor.....	1018
15.33.1	Flackern des Mauszeigers bei Animationen vermeiden.....	1019
15.34	Mausrad-Unterstützung.....	1020
15.35	Benutzerinteraktionen automatisieren.....	1020
15.35.1	Automatisch in die Tasten hauen.....	1021
15.35.2	Mausoperationen.....	1022
15.35.3	Methoden zur Zeitsteuerung.....	1022
15.35.4	Screenshots.....	1023
15.35.5	Funktionsweise und Beschränkungen.....	1023
15.35.6	Zeitliches Ausführen mit dem javax.swing.Timer.....	1023
15.35.7	MouseInfo und PointerInfo.....	1024
15.36	Alternativen zu AWT und Swing.....	1024
15.36.1	XML-Beschreibungen der Oberfläche: Swixml, XUL/Luxor.....	1024
15.36.2	SWT.....	1025

16 Netzwerkprogrammierung

1027

16.1	Grundlegende Begriffe.....	1029
16.1.1	Internet-Standards und RFC.....	1029
16.2	URL-Verbindungen und URL-Objekte.....	1030
16.2.1	Die Klasse URL.....	1031
16.2.2	Informationen über eine URL.....	1034
16.2.3	Der Zugriff auf die Daten über die Klasse URL.....	1035
16.2.4	Verbindungen durch einen Proxy-Server.....	1037

16.3	Die Klasse URLConnection.....	1038
16.3.1	Methoden und Anwendung von URLConnection.....	1038
16.3.2	Protokoll- und Content-Handler.....	1040
16.3.3	Bilder-Handler.....	1041
16.3.4	Zusammenfassung Content-und Protokoll-Handler.....	1041
16.3.5	Im Detail: vom URL zu URLConnection.....	1042
16.3.6	Der Protokoll-Handler für Jar-Dateien.....	1043
16.3.7	Autorisierte URL-Verbindungen mit Basic Authentication.....	1044
16.4	Das Common Gateway Interface.....	1045
16.4.1	Parameter für ein CGI-Programm.....	1046
16.4.2	Kodieren der Parameter für CGI-Programme.....	1047
16.4.3	Eine Suchmaschine ansprechen.....	1049
16.5	Host- und IP-Adressen.....	1050
16.5.1	Lebt der Rechner?.....	1051
16.5.2	Das Netz ist Klasse.....	1052
16.5.3	IP-Adresse des lokalen Hosts.....	1052
16.5.4	Die Methode getAllByNameO.....	1053
16.6	Networkinterface.....	1053
16.7	Mit dem Socket zum Server.....	1055
16.7.1	Das Netzwerk ist der Computer.....	1055
16.7.2	Standarddienste unter Windows nachinstallieren.....	1057
16.7.3	Eine Verbindung zum Server aufbauen.....	1057
16.7.4	Server unter Spannung: die Ströme.....	1058
16.7.5	Die Verbindung wieder abbauen.....	1059
16.7.6	Ein kleines Echo - lebt der Rechner noch?.....	1059
16.7.7	Blockierendes Lesen.....	1060
16.7.8	Informationen über den Socket.....	1062
16.7.9	MittelnNet an den Ports horchen.....	1063
16.8	Client/Server-Kommunikation.....	1064
16.8.1	Warten auf Verbindungen.....	1065
16.8.2	Ein Multiplikations-Server.....	1066
16.8.3	Von außen erreichbar sein.....	1067
16.9	SLL-VerbindungenmitJSSE.....	1068
16.10	Apache Jakarta Commons HttpClient und Net.....	1069
16.10.1	Jakarta Commons HttpClient.....	1069
16.10.2	Jakarta Commons Net.....	1070
16.11	E-Mail.....	1070
16.11.1	Wie eine E-Mail um die Welt geht.....	1071
16.11.2	Das Simple Mail Transfer Protocol und RFC 822.....	1071
16.11.3	POP (Post Office Protocol).....	1073
16.11.4	E-Mails versenden mit der JavaMail API von SUN.....	1073
16.11.5	MimeMultipart-Nachrichten schicken.....	1074
16.11.6	E-Mails mittels POP3 abrufen.....	1075
16.11.7	Ereignisse und Suchen.....	1077
16.12	Arbeitsweise eines Web-Servers.....	1078
16.12.1	Das Hypertext Transfer Protocol (HTTP).....	1078
16.12.2	Anfragen an den Server.....	1079
16.12.3	Die Antworten vom Server.....	1082

17.10 JSP-Direktiven.....	1124
17.10.1 page-Direktiven im Überblick.....	1125
17.10.2 include-Direktive.....	1126
17.10.3 Mit JSPs Bilder generieren.....	1127
17.11 Aktionen.....	1128
17.11.1 Aktion include.....	1128
17.11.2 Aktion forward.....	1128
17.11.3 Aktion plugin.....	1128
17.12 Beans.....	1129
17.12.1 Beans in JSP-Seiten anlegen, Attribute setzen und erfragen.....	1130
17.12.2 Der schnelle Zugriff auf Parameter.....	1131
17.13 Kleine Kekse: die Klasse Cookies.....	1131
17.13.1 Cookies erzeugen und setzen.....	1132
17.13.2 Cookies vom Servlet einlesen.....	1133
17.13.3 Kleine Helfer für Cookies.....	1134
17.13.4 Cookie-Status ändern.....	1135
17.13.5 Langlebige Cookies.....	1136
17.13.6 Ein Warenkorbsystem.....	1136
17.14 Sitzungsverfolgung (Session Tracking).....	1138
17.14.1 Das mit einer Sitzung verbundene Objekt HttpSession.....	1139
17.14.2 Werte mit einer Sitzung assoziieren und auslesen.....	1139
17.14.3 URL-Rewriting.....	1140
17.14.4 Zusätzliche Informationen.....	1140
17.15 Tag-Libraries.....	1143
17.15.1 Standard Tag Library (JSTL).....	1143
17.15.2 Jakarta Taglibs Project.....	1144
17.16 Servlets.....	1145
17.16.1 Servlets compilieren.....	1145
17.16.2 Wohin mit den Servlets: das classes-Verzeichnis.....	1146
17.16.3 Servlets mit dem Sysdeo-Plugin unter Eclipse.....	1146
17.16.4 Servlet-Mapping.....	1146
17.17 Der Lebenszyklus eines Servlets.....	1148
17.17.1 Initialisierung in init().....	1149
17.17.2 Abfragen bei service().....	1151
17.17.3 Mehrere Anfragen beim Servlet und die Thread-Sicherheit.....	1152
17.17.4 Das Ende eines Servlets.....	1152
17.18 Das HttpServletResponse-Objekt.....	1153
17.18.1 Wir generieren eine Web-Seite.....	1153
17.18.2 Binärdaten senden.....	1154
17.18.3 Komprimierte Daten mit Content-Encoding.....	1155
17.18.4 Noch mehr über Header, die der Server setzt.....	1157
17.19 Objekte und Dateien per POST verschicken.....	1158
17.19.1 Datei-Upload.....	1158
17.20 Servlets und Sessions.....	1159
17.21 Weiterleiten und Einbinden von Servlet-Inhalten.....	1160
17.22 Inter-Servlet-Kommunikation.....	1161
17.22.1 Daten zwischen Servlets teilen.....	1161

17.23	Internationalisierung.....	1162
17.23.1	Die Länderkennung des Anfragers auslesen.....	1162
17.23.2	Länderkennung für die Ausgabe setzen.....	1163
17.23.3	Westeuropäische Texte senden.....	1163
17.24	Tomcat: Spezielles.....	1164
17.24.1	Tomcat als Service unter Windows NT ausführen.....	1164
17.24.2	Interessante Links zum Thema Servlets/JSP.....	1164

18 Verteilte Programmierung mit RMI und SOAP 1167

18.1	Entfernte Methoden.....	1169
18.1.1	Wie entfernte Methoden arbeiten.....	1169
18.1.2	Stellvertreter (Proxy).....	1169
18.1.3	RMI.....	1170
18.1.4	Wie die Stellvertreter die Daten übertragen.....	1170
18.1.5	Probleme mit entfernten Methoden.....	1171
18.2	Nutzen von RMI bei Middleware-Lösungen.....	1172
18.3	Die Lösung für Java ist RMI.....	1173
18.3.1	Entfernte Objekte programmieren.....	1173
18.3.2	Entfernte und lokale Objekte im Vergleich.....	1174
18.4	Definition einer entfernten Schnittstelle.....	1174
18.5	Das entfernte Objekt.....	1175
18.5.1	Der Bauplan für entfernte Objekte.....	1175
18.5.2	Der Konstruktor.....	1176
18.5.3	Implementierung der entfernten Methoden.....	1177
18.6	Stellvertreterobjekte erzeugen.....	1178
18.6.1	Das Dienstprogramm rmic.....	1178
18.7	Der Namensdienst (Registry).....	1179
18.7.1	Der Port.....	1180
18.8	Der Server.....	1180
18.8.1	Entfernte Objekte beim Namensdienst anmelden.....	1180
18.8.2	Automatisches Anmelden bei Bedarf.....	1181
18.9	Einen Client programmieren.....	1182
18.9.1	Einfaches Logging.....	1183
18.10	Aufräumen mit dem DCC.....	1183
18.11	Entfernte Objekte übergeben und laden.....	1184
18.11.1	Klassen vom RMI-Klassenlader nachladen.....	1184
18.11.2	Sicherheitsmanager.....	1185
18.12	Registry wird vom Server gestartet.....	1186
18.13	RMI über die Firewall.....	1187
18.13.1	RMI über HTTP getunnelt.....	1187
18.14	RMI und CORBA.....	1188

18.15	UnicastRemoteObject, RemoteServer und RemoteObject.....	1188
18.16	Daily Soap.....	1189
18.16.1	SOAP-Implementierung der Apache-Gruppe.....	1191
18.16.2	Einen Client mit der Apache-Bibliothek implementieren.....	1191
18.16.3	Der Seifen-Server.....	1193
18.17	Java-API für XML Messaging (JAXM).....	1193
18.18	Java Message Service (JMS).....	1194
18.18.1	OpenJMS.....	1195
18.18.2	Beispiel mit Konsument und Produzent im Publish-Subscribe-Modell . . .	1195

19 Applets, AIdlets und Sound 1197

19.1	Applets und Applikationen - wer darf was?.....	1199
19.2	Das erste Hallo-Applet.....	1199
19.3	Die Zyklen eines Applets.....	1200
19.4	Parameter an das Applet übergeben.....	1200
19.4.1	Wie das Applet den Browser-Inhalt ändern kann.....	1201
19.4.2	Den Ursprung des Applets erfragen.....	1202
19.4.3	Was ein Applet alles darf.....	1203
19.5	Fehler in Applets finden.....	1204
19.6	Browserabhängiges Verhalten.....	1204
19.6.1	Java im Browser aktiviert?.....	1204
19.6.2	Läuft das Applet unter Netscape oder Microsoft Explorer?.....	1204
19.6.3	Datenaustausch zwischen Applets und Java-Skripten.....	1206
19.7	Datenaustausch zwischen Applets.....	1206
19.8	Musik in einem Applet und in Applikationen.....	1209
19.8.1	Fest verdrahtete Musikdatei in einem Applet.....	1210
19.8.2	WAV- und MIDI-Dateien abspielen.....	1211
19.9	Webstart.....	1212
19.10	Java2 Micro Edition.....	1212
19.10.1	Konfigurationen.....	1213
19.10.2	Profile.....	1214

20 Datenbankmanagement mit JDBC 1217

20.1	Das relationale Modell.....	1219
20.2	JDBC: der Zugriff auf Datenbanken über Java.....	1220
20.3	Die Rolle von SQL.....	1220
20.3.1	Ein Rundgang durch SQL-Anfragen.....	1221
20.3.2	Datenabfrage mit der Data Query Language (DQL).....	1223
20.3.3	Tabellen anlegen mit der Data Definition Language (DDL).....	1225

20.4	Datenbanktreiber für den Zugriff.....	1225
20.4.1	Treibertypen.....	1225
20.5	Datenbanken und ihre Treiber.....	1226
20.5.1	Die freie Datenbank MySQL.....	1226
20.5.2	JDBC-Treiber für MySQL: MySQL Connector/J.....	1229
20.5.3	Die Datenbank Microsoft Access.....	1229
20.5.4	Ein Typ-4-Treiber für den Microsoft SQL Server 2000.....	1229
20.5.5	DieJDBC-ODBC-Bridge.....	1230
20.5.6	ODBC einrichten und Access damit verwenden.....	1230
20.5.7	Oracle9i Enterprise Edition.....	1231
20.5.8	JDBC-Treiber für mobile Endgeräte.....	1231
20.6	Eine Beispielabfrage.....	1232
20.7	Mit Javaan eine Datenbank andocken.....	1233
20.7.1	Der Treibermanager.....	1233
20.7.2	Eine Aufzählung aller Treiber.....	1234
20.7.3	Log-Informationen.....	1234
20.7.4	Den Treiber laden.....	1235
20.7.5	Verbindung zur Datenbank.....	1236
20.8	Datenbankabfragen.....	1239
20.8.1	Abfragen über das Statement-Objekt.....	1239
20.8.2	Ergebnisse einer Abfrage in ResultSet.....	1240
20.8.3	Java und SQL-Datentypen.....	1241
20.8.4	Unicode in der Spalte korrekt auslesen.....	1245
20.8.5	wasNullO bei ResultSet.....	1245
20.8.6	Wie viele Zeilen hat ein ResultSet?.....	1245
20.9	Die Ausnahmen bei JDBC.....	1246
20.10	Transaktionen.....	1246
20.11	Elemente einer Datenbank hinzufügen und aktualisieren.....	1247
20.11.1	Batch-Updates.....	1248
20.12	Vorbereitete Anweisungen (Prepared Statements).....	1249
20.12.1	PreparedStatement-Objekte vorbereiten.....	1249
20.12.2	Werte für die Platzhalter eines PreparedStatement.....	1250
20.13	Die LOBs (Large Objects).....	1251
20.13.1	Einen BLOB besorgen.....	1251
20.14	Die SQL3-Datentypen ARRAY, STRUCT und REF.....	1252
20.15	Metadaten.....	1252
20.15.1	Metadaten über die Tabelle.....	1253
20.15.2	Informationen über die Datenbank.....	1256
20.16	DataSource.....	1258
20.16.1	Die Schnittstelle DataSource.....	1258
20.17	Java Data Objects (JDO).....	1259
20.18	XML-Datenbanken.....	1260
20.18.1	Apache Xindice.....	1260
20.18.2	eXist und Weitere.....	1261

21 Reflection und Annotationen 1263

21.1	MetaData.....	1265
21.1.1	XDodet.....	1265
21.1.2	Annotationen.....	1266
21.2	Mit dem Class-Objekt etwas über Klassen erfahren.....	1267
21.2.1	An ein Class-Objekt kommen.....	1267
21.2.2	Was das Class-Objekt beschreibt.....	1269
21.2.3	Der Name der Klasse.....	1270
21.2.4	Instanceof mit Class-Objekten.....	1272
21.2.5	Oberklassen finden.....	1273
21.2.6	Implementierte Interfaces einer Klasse oder eines Interfaces.....	1274
21.2.7	Modifizierer und die Klasse Modifier.....	1275
21.2.8	Die Attribute einer Klasse.....	1276
21.2.9	Methoden einer Klasse erfragen.....	1279
21.2.10	Konstruktoren einer Klasse.....	1282
21.3	Annotationen.....	1283
21.4	Objekte manipulieren.....	1284
21.4.1	Objekte erzeugen.....	1284
21.4.2	Die Belegung der Variablen erfragen.....	1286
21.4.3	Eine generische toStringO-Funktion.....	1288
21.4.4	Variablen setzen.....	1289
21.4.5	Private Attribute ändern.....	1291
21.5	Methoden aufrufen.....	1292
21.5.1	Statische Methoden aufrufen.....	1293
21.5.2	Dynamische Methodenaufrufe bei festen Methoden beschleunigen.....	1293
21.6	Informationen und Identifizierung von Paketen.....	1295
21.6.1	Geladene Pakete.....	1297

22 Komponenten durch Bohnen 1299

22.1	Grundlagen der Komponententechnik.....	1301
22.1.1	Brauchen wir überhaupt Komponenten?.....	1301
22.1.2	Visuelle und nichtvisuelle Komponenten.....	1302
22.1.3	Andere Komponententechnologien oder: Was uns Microsoft brachte. . . .	1302
22.2	Das JavaBeans Development Kit (BDK).....	1303
22.2.1	Eine Beispielsitzung im BDK.....	1304
22.2.2	Verknüpfungen zwischen Komponenten.....	1305
22.2.3	Beans speichern.....	1306
22.3	Die kleinste Bohne der Welt.....	1306
22.4	Jar-Archive für Komponenten.....	1306
22.5	Worauf JavaBeans basieren.....	1308
22.6	Eigenschaften.....	1309
22.6.1	Einfache Eigenschaften.....	1309

22.6.2	Boolesche Eigenschaften.....	1310
22.6.3	Indizierte Eigenschaften.....	1310
22.7	Ereignisse.....	1312
22.7.1	Multicast und Unicast.....	1312
22.7.2	Namenskonvention.....	1312
22.8	Weitere Eigenschaften.....	1315
22.8.1	Gebundene Eigenschaften.....	1315
22.8.2	Anwendung von PropertyChange bei AWT-Komponenten.....	1317
22.8.3	Veto-Eigenschaften. Dagegen!.....	1318
22.9	Bean-Eigenschaften anpassen.....	1319
22.9.1	Customizer.....	1319
22.10	Property-Editoren.....	1320
22.11	BeanInfo.....	1320
22.12	Beliebte Fehler.....	1321

23 Java Native Interface (JNI) 1323

23.1	Java Native Interface und Invocation-API.....	1325
23.2	Die Schritte zur Einbindung einer C-Funktion in ein Java-Programm.....	1326
23.2.1	Schreiben des Java-Codes.....	1326
23.2.2	Compilieren des Java-Codes.....	1326
23.2.3	Erzeugen der Header-Datei.....	1327
23.2.4	Implementierung der Methode in C.....	1328
23.2.5	Übersetzen der C-Programme und Erzeugen der dynamischen Bibliothek	1328
23.2.6	Setzen der Umgebungsvariable.....	1329
23.3	Erweiterung unseres Programms.....	1329
23.4	Erweiterte JNI-Eigenschaften.....	1330
23.4.1	Klassendefinitionen.....	1331
23.4.2	Zugriff auf Attribute.....	1331

24 Sicherheitskonzepte 1335

24.1	Der Sandkasten (Sandbox).....	1337
24.2	Sicherheitsmanager (Security Manager).....	1337
24.2.1	Der Sicherheitsmanager bei Applets.....	1339
24.2.2	Sicherheitsmanager aktivieren.....	1341
24.2.3	Wie nutzen die Java-Bibliotheken den Sicherheitsmanager?.....	1342
24.2.4	Rechte vergeben durch Policy-Dateien.....	1342
24.2.5	Erstellen von Rechte-Dateien mit dem grafischen Policy-Tool.....	1344
24.2.6	Kritik an den Policies.....	1345
24.3	Dienstprogramme zur Signierung.....	1346
24.3.1	Mit keytool Schlüssel erzeugen.....	1346
24.3.2	Signieren mit jarsigner.....	1347

24.4	Digitale Unterschriften.....	1347
24.4.1	Die MDx-Reihe.....	1348
24.4.2	Secure Hash Algorithm (SHA).....	1349
24.4.3	Mit der Security-API einen Fingerabdruck berechnen.....	1349
24.4.4	Die Klasse MessageDigest.....	1349
24.4.5	Unix-Crypt.....	1352
24.5	Verschlüsseln von Daten(-strömen).....	1352
24.5.1	Den Schlüssel bitte.....	1352
24.5.2	Verschlüsseln mit Cipher.....	1354
24.5.3	Verschlüsseln von Datenströmen.....	1354

25 Dienstprogramme für die Java-Umgebung 1357

25.1	Die Werkzeuge im Überblick.....	1359
25.2	Der Compiler javac.....	1359
25.2.1	Der Java-Interpreter Java.....	1360
25.2.2	Der Unterschied zwischen java.exe und javaw.exe.....	1361
25.3	Das Archivformat Jar.....	1362
25.3.1	Das Dienstprogramm Jar benutzen.....	1363
25.3.2	Das Manifest.....	1364
25.3.3	Applikationen in Jar-Archiven starten.....	1365
25.3.4	Applets in Jar-Archiven.....	1366
25.3.5	Der Eintrag Class-Path in einer Manifest-Datei.....	1366
25.4	Nativer Compiler JET.....	1367
25.5	Einpacken von Java-Programmen in ein exe mit JSmooth.....	1367
25.6	Decompiler.....	1367
25.6.1	Jad, ein schneller Decompiler.....	1369
25.6.2	SourceAgain.....	1371
25.6.3	Decompilieren erschweren.....	1371
25.7	Obfuscate Programm RetroGuard.....	1372
25.8	Sourcecode Beautifier.....	1373
25.9	Ant.....	1374
25.9.1	Bezug und Installation von Ant.....	1374
25.9.2	Properties.....	1375
25.9.3	Externe und vordefinierte Properties.....	1376
25.9.4	Weitere Leistungen.....	1377

26 Style-Guide 1379

26.1	Programmierrichtlinien.....	1381
26.2	Allgemeine Richtlinien.....	1381
26.3	Quellcode kommentieren.....	1382
263.1	Strategischer und taktischer Kommentar.....	1382

26.3.2	Bemerkungen über JavaDoc.....	1383
26.3.3	Cotcha-Schlüsselwörter.....	1384
26.4	Bezeichnernamen.....	1385
26.4.1	Ungarische Notation.....	1385
26.4.2	Vorschlag für die Namensgebung.....	1385
26.5	Formatierung.....	1387
26.5.1	Einrücken von Programmcode - die Vergangenheit.....	1387
26.5.2	Verbundene Ausdrücke.....	1388
26.5.3	Kontrollierter Datenfluss.....	1388
26.6	Ausdrücke.....	1389
26.7	Reihenfolge der Eigenschaften in Klassen.....	1390
26.8	Zugriffsrechte und Zugriffsmethoden.....	1391
26.8.1	Accessors/Zugriffsmethoden.....	1391
26.9	Verweise.....	1392

A Die Begleit-CD 1393

Index 1395