Auf einen Blick

	Vorwort	35
1	Schon wieder eine neue Sprache?	49
2	Sprachbeschreibung	81
3	Klassen und Objekte	177
4	Der Umgang mit Zeichenketten	219
5	Mathematisches	267
6	Eigene Klassen schreiben	289
7	Exceptions	421
8	Die Funktionsbibliothek	445
9	Threads und nebenläufige Programmierung	491
10	Raum und Zeit	569
11	Datenstrukturen und Algorithmen	603
12	Datenströme und Dateien	671
13	Die eXtensible Markup Language (XML)	793
14	Grafikprogrammierung mit dem AWT	839
15	Komponenten, Container und Ereignisse	937
16	Das Netz	1081
17	Servlets und Java Server Pages	1151
18	Verteilte Programmierung mit RMI und SOAP	1213
19	Applets, Midlets und Sound	1235
20	Datenbankmanagement mit JDBC	1255
21	Reflection	1299
22	Komponenten durch Bohnen	1347
23	Java Management Extensions (JMX)	1371
24	Java Native Interface (JNI)	1381
25	Sicherheitskonzepte	1395
26	Dienstprogramme für die Java-Umgebung	1417
A	Die Begleit-CD	1417
	Index	1437

Inhalt

Vorwort 35

Vorwort zur 4. Auflage 46
Vorwort zur 5. Auflage 46

1 Schon wieder eine neue Sprache? 49

1.1 Der erste Kontakt 51
1.2 Historischer Hintergrund 51
1.3 Eigenschaften von Java 52
1.3.1 Bytecode und die virtuelle Maschine 53
1.3.2 Kein Präprozessor für Textersetzungen 54
1.3.3 Keine überladenen Operatoren 54
1.3.4 Zeiger und Referenzen 55
1.3.5 Bring den Müll raus, Garbage-Collector! 56
1.3.6 Ausnahmenbehandlung 56
1.3.7 Objektorientierung in Java 57
1.3.8 Java-Security-Modell 57
1.3.9 Wofür sich Java nicht eignet 57
1.4 Java im Vergleich zu anderen Sprachen 58
1.4.1 Java und JavaScript 59
1.4.2 Normierungsversuche 59
1.4.3 Die Rolle von Java im Web 59
1.4.4 Vollwertige Applikationen statt Applets 60
1.5 Die Java Platform Standard Edition (Java SE) 60
1.5.1 JDK und JRE 60
1.5.2 Java-Versionen 61
1.5.3 Java für die Kleinen 62
1.5.4 Java für die Großen 62
1.5.5 Direkt ausführbare Programme, native Compiler 62
1.6 Entwicklungsumgebungen 63
1.6.1 Sun ONE Studio und NetBeans von Sun 63
1.6.2 IBM und Eclipse 63
1.6.3 Together 64
1.6.4 Ein Wort zu Microsoft, Java und zu J++ 65
1.7 Installationsanleitung für Java 2 65
1.7.1 Das Java SE beziehen 66
1.7.2 Java SE unter Windows installieren 66
1.7.3 Compiler und Interpreter nutzen 67
1.7.4 Installation der Laufzeitumgebung (1.4) unter Linux 68

1.8 **Das erste Programm compilieren und testen** ... 69
1.8.1 Häufige Compiler- und Interpreterprobleme ... 70
1.9 **Eclipse** ... 71
1.9.1 Eclipse starten ... 71
1.9.2 Das erste Projekt anlegen ... 72
1.9.3 Eine Klasse hinzufügen ... 75
1.9.4 Übersetzen und Ausführen ... 76
1.9.5 JDK statt JRE ... 77
1.9.6 Start eines Programms ohne Speicheraufforderung ... 77
1.9.7 Projekt einfügen oder Workspace für die Aufgaben wechseln ... 78
1.9.8 Allgemeine Plugins für Eclipse ... 79
1.9.9 Eclipse Web Tools Platform (WTP) ... 79

2 Sprachbeschreibung **81**

2.1 **Anweisungen und Programme** ... 83
2.2 **Elemente der Programmiersprache Java** ... 85
2.2.1 Textkodierung durch Unicode-Zeichen ... 85
2.2.2 Literale ... 87
2.2.3 Bezeichner ... 87
2.2.4 Reservierte Schlüsselwörter ... 89
2.2.5 Token ... 90
2.2.6 Semantik ... 90
2.2.7 Kommentare ... 93
2.2.8 Funktionsaufrufe als Anweisungen ... 94
2.2.9 Die leere Anweisung ... 97
2.2.10 Der Block ... 97
2.3 **Datentypen** ... 97
2.3.1 Primitive Datentypen ... 98
2.3.2 Wahrheitswerte ... 100
2.3.3 Variablendeklarationen ... 100
2.3.4 Ganzzahlige Datentypen ... 102
2.3.5 Die Fließkommazahlen ... 104
2.3.6 Alphanumerische Zeichen ... 105
2.3.7 Die Typanpassung (das Casting) ... 106
2.3.8 Lokale Variablen, Blöcke und Sichtbarkeit ... 111
2.3.9 Initialisierung von lokalen Variablen ... 112
2.4 **Ausdrücke, Operanden und Operatoren** ... 113
2.4.1 Zuweisungsoperator ... 114
2.4.2 Verbundoperatoren ... 115
2.4.3 Präfix- oder Postfix-Inkrement und -Dekrement ... 116
2.4.4 Unäres Minus und Plus ... 118
2.4.5 Arithmetische Operatoren ... 118
2.4.6 Die relationalen Operatoren ... 121
2.4.7 Logische Operatoren ... 121
2.4.8 Rang der Operatoren in der Auswertungsreihenfolge ... 122
2.4.9 Überladenes Plus für Strings ... 124
2.4.10 Was C(++)-Programmierer vermissen könnten ... 126

2.5 **Bedingte Anweisungen oder Fallunterscheidungen** ... 126
2.5.1 Die if-Anweisung ... 126
2.5.2 Die Alternative wählen mit einer if/else-Anweisung ... 128
2.5.3 Die switch-Anweisung bietet die Alternative ... 131
2.6 **Schleifen** ... 133
2.6.1 Die while-Schleife ... 133
2.6.2 Schleifenbedingungen und Vergleiche mit == ... 134
2.6.3 Die do/while-Schleife ... 136
2.6.4 Die for-Schleife ... 137
2.6.5 Ausbruch planen mit break und Wiedereinstieg mit continue ... 142
2.6.6 break und continue mit Sprungmarken ... 144
2.7 **Methoden einer Klasse** ... 145
2.7.1 Bestandteil einer Funktion ... 146
2.7.2 Aufruf einer Methode ... 147
2.7.3 Methoden ohne Parameter ... 148
2.7.4 Statische Funktionen (Klassenmethoden) ... 149
2.7.5 Parameter, Argument und Wertübergabe ... 149
2.7.6 Methoden vorzeitig mit return beenden ... 151
2.7.7 Nicht erreichbarer Quellcode bei Funktionen ... 151
2.7.8 Rückgabewerte ... 152
2.7.9 Methoden überladen ... 156
2.7.10 Vorgegebener Wert für nicht aufgeführte Argumente ... 158
2.7.11 Finale lokale Variablen ... 158
2.7.12 Rekursive Funktionen ... 160
2.7.13 Die Ackermann-Funktion ... 162
2.7.14 Die Türme von Hanoi ... 165
2.8 **Weitere Operatoren** ... 166
2.8.1 Bitoperationen ... 166
2.8.2 Vorzeichenlose Zahlen in ein Integer und Char konvertieren ... 167
2.8.3 Variablen mit Xor vertauschen ... 168
2.8.4 Die Verschiebeoperatoren ... 169
2.8.5 Ein Bit setzen, löschen, umdrehen und testen ... 172
2.8.6 Der Bedingungsoperator ... 172
2.9 **Einfache Benutzereingaben** ... 174

3 Klassen und Objekte 177

3.1 **Objektorientierte Programmierung** ... 179
3.1.1 Warum überhaupt OOP? ... 179
3.1.2 Wiederverwertbarkeit ... 180
3.2 **Klassen benutzen** ... 180
3.2.1 Die Klasse Point ... 181
3.3 **Die UML (Unified Modeling Language)** ... 181
3.3.1 Hintergrund und Geschichte zur UML ... 182
3.3.2 Wichtige Diagrammtypen der UML ... 182
3.4 **Anlegen und Nutzen eines Punktes** ... 183
3.4.1 Definieren von Referenz-Variablen ... 183

3.4.2 Anlegen eines Exemplars einer Klasse mit dem new-Operator ... 184
3.4.3 Zugriff auf Variablen und Methoden mit dem ».« ... 184
3.4.4 Konstruktoren nutzen ... 187

3.5 Import und Pakete ... 188

3.6 Die API-Dokumentation ... 188

3.7 Mit Referenzen arbeiten ... 189

3.7.1 Die null-Referenz ... 189
3.7.2 Zuweisungen bei Referenzen ... 191
3.7.3 Funktionen mit nichtprimitiven Parametern ... 192
3.7.4 Gleichheit von Objekten und die Methode equals() ... 193

3.8 Arrays ... 195

3.8.1 Deklaration von Arrays ... 196
3.8.2 Arrays mit Inhalt ... 196
3.8.3 Die Länge eines Arrays über das Attribut length ... 197
3.8.4 Zugriff auf die Elemente über den Index ... 198
3.8.5 Array-Objekte erzeugen ... 198
3.8.6 Fehler bei Arrays ... 200
3.8.7 Arrays mit nicht-primitiven Elementen ... 201
3.8.8 Vorinitialisierte Arrays ... 202
3.8.9 Die erweiterte for-Schleife ... 202
3.8.10 Mehrdimensionale Arrays ... 204
3.8.11 Die Wahrheit über die Array-Initialisierung ... 206
3.8.12 Mehrere Rückgabewerte ... 207
3.8.13 Argument per Referenz übergeben ... 207
3.8.14 Arrays klonen ... 208
3.8.15 Feldinhalte kopieren ... 208
3.8.16 Die Klasse Arrays zum Vergleichen, Füllen und Suchen ... 209
3.8.17 Methode mit variabler Argumentanzahl (vararg) ... 211

3.9 Der Einstiegspunkt für das Laufzeitsystem main() ... 212

3.9.1 Kommandozeilen-Parameter ausgeben ... 212
3.9.2 Der Rückgabewert von main() und System.exit() ... 213
3.9.3 Parser der Kommandozeilenargumente Apache CLI ... 213

3.10 Eigene Pakete schnüren ... 215

3.10.1 Die package-Anweisung ... 215
3.10.2 Importieren von Klassen mit import ... 215
3.10.3 Paketnamen ... 216
3.10.4 Hierarchische Strukturen und das Default-Package ... 216
3.10.5 Klassen mit gleichen Namen in unterschiedlichen Paketen ... 217
3.10.6 Statisches Import ... 217
3.10.7 Eine Verzeichnisstruktur für eigene Projekte ... 218

4 Der Umgang mit Zeichenketten 219

4.1 Strings und deren Anwendung ... 221

4.1.1 String-Literale als String-Objekte für konstante Zeichenketten ... 222
4.1.2 String-Objekte neu anlegen ... 222
4.1.3 String-Länge ... 224

4.1.4 Gut, dass wir verglichen haben ... 225
4.1.5 String-Teile extrahieren ... 226
4.1.6 Suchen und ersetzen ... 229
4.1.7 Veränderte Strings liefern ... 232
4.1.8 Unterschiedliche Typen in Zeichenketten konvertieren ... 234

4.2 Veränderbare Zeichenketten mit StringBuffer/StringBuilder ... 234

4.2.1 Anlegen von StringBuffer/StringBuilder-Objekten ... 235
4.2.2 Die Länge eines StringBuffer/Builder-Objekts lesen und setzen ... 236
4.2.3 Daten anhängen ... 236
4.2.4 Zeichen(folgen) setzen, erfragen, löschen und umdrehen ... 237

4.3 Vergleiche von Zeichenketten ... 238

4.3.1 equals() in String und StringBuffer/StringBuilder ... 238
4.3.2 equals() und hashCode() bei StringBuffer/StringBuilder ... 239
4.3.3 Sprachabhängiges Vergleichen mit der Collator-Klasse ... 239
4.3.4 Effiziente interne Speicherung für die Sortierung ... 242

4.4 Reguläre Ausdrücke ... 243

4.4.1 Die Klassen Pattern und Matcher ... 243
4.4.2 Mit MatchResult alle Ergebnisse einsammeln ... 244
4.4.3 Das alternative Paket org.apache.regexp ... 245

4.5 Zerlegen von Zeichenketten ... 245

4.5.1 Splitten von Zeichenketten mit split() aus Pattern ... 245
4.5.2 split() in String ... 246
4.5.3 Die Klasse Scanner ... 246
4.5.4 StringTokenizer ... 250
4.5.5 Der BreakIterator als Wort- und Satztrenner ... 252

4.6 Zeichenkodierungen ... 254

4.6.1 Kodierungen für unterschiedliche Codepages ... 254
4.6.2 Andere Klassen zur Konvertierung und das Paket java.nio.charset ... 255
4.6.3 Base64-Kodierung ... 256

4.7 Formatieren von Ausgaben ... 257

4.7.1 Zahlen, Prozente und Währungen mit NumberFormat formatieren ... 259
4.7.2 Dezimalzahlformatierung mit DecimalFormat ... 259
4.7.3 Formatieren mit format() aus String ... 261
4.7.4 Ausgaben formatieren mit MessageFormat ... 263

5 Mathematisches 267

5.1 Arithmetik in Java ... 269

5.1.1 Java-Sondertypen im Beispiel ... 269

5.2 Die Funktionen der Math-Klasse ... 270

5.2.1 Attribute ... 270
5.2.2 Winkelfunktionen – trigonometrische Funktionen und Arcus-Funktionen ... 270
5.2.3 Runden von Werten ... 271
5.2.4 Wurzel und Exponentialfunktionen ... 273
5.2.5 Der Logarithmus ... 273

5.2.6 Rest der ganzzahligen Division ... 274
5.2.7 Absolutwerte und Maximum/Minimum ... 274
5.2.8 Zufallszahlen ... 275
5.2.9 Der nächste Bitte ... 275
5.3 Mathe bitte strikt ... 275
5.3.1 Strikt Fließkomma mit strictfp ... 276
5.3.2 Die Klassen Math und StrictMath ... 276
5.4 Die Random-Klasse ... 276
5.5 Große Zahlen ... 278
5.5.1 Die Klasse BigInteger ... 278
5.5.2 Funktionen von BigInteger ... 280
5.5.3 Ganz lange Fakultäten ... 282
5.5.4 Große Fließkommazahlen mit BigDecimal ... 283
5.5.5 Komfortabel die Rechengenauigkeit setzen mit MathContext ... 284
5.6 Rechnen mit Einheiten: Java Units Specification ... 286
5.7 Weitere Klassen für mathematische Probleme ... 287

6 Eigene Klassen schreiben 289

6.1 Eigene Klassen definieren ... 291
6.1.1 Methodenaufrufe und Nebeneffekte ... 294
6.1.2 Argumentübergabe mit Referenzen ... 294
6.1.3 Die this-Referenz ... 295
6.1.4 Überdeckte Objektvariablen nutzen ... 296
6.2 Assoziationen zwischen Objekten ... 297
6.2.1 Gegenseitige Abhängigkeiten von Klassen ... 298
6.3 Privatsphäre und Sichtbarkeit ... 298
6.3.1 Wieso nicht freie Methoden und Variablen für alle? ... 300
6.3.2 Privat ist nicht ganz privat: Es kommt darauf an, wer's sieht ... 300
6.3.3 Zugriffsmethoden für Attribute definieren ... 301
6.4 Statische Methoden und statische Attribute ... 304
6.4.1 Warum statische Eigenschaften sinnvoll sind ... 305
6.4.2 Statische Eigenschaften mit static ... 305
6.4.3 Statische Eigenschaften über Referenzen nutzen? ... 306
6.4.4 Warum die Groß- und Kleinschreibung wichtig ist ... 306
6.4.5 Statische Eigenschaften und Objekteigenschaften ... 307
6.4.6 Statische Variablen zum Datenaustausch ... 308
6.4.7 Statische Blöcke als Klasseninitialisierer ... 309
6.5 Konstanten und Aufzählungen ... 310
6.5.1 Konstanten über öffentliche statische final-Variablen ... 310
6.5.2 Problem mit finalen Klassenvariablen ... 311
6.5.3 Typsicherere Konstanten ... 312
6.5.4 Aufzählungen mit enum in Java 5 ... 314
6.5.5 enum-Konstanten in switch ... 314

6.6 Objekte anlegen und zerstören ... 315
6.6.1 Konstruktoren schreiben ... 316
6.6.2 Einen anderen Konstruktor der gleichen Klasse aufrufen ... 319
6.6.3 Initialisierung der Objekt- und Klassenvariablen ... 322
6.6.4 Finale Werte im Konstruktor und in statischen Blöcken setzen ... 324
6.6.5 Exemplarinitialisierer (Instanzinitialisierer) ... 325
6.6.6 Zerstörung eines Objekts durch den Müllaufsammler ... 327
6.6.7 Implizit erzeugte String-Objekte ... 328
6.6.8 Private Konstruktoren, Utility-Klassen, Singleton und Fabriken ... 328

6.7 Vererbung ... 331
6.7.1 Vererbung in Java ... 331
6.7.2 Einfach- und Mehrfachvererbung ... 331
6.7.3 Gebäude modelliert ... 332
6.7.4 Konstruktoren in der Vererbung ... 333
6.7.5 Sichtbarkeit protected ... 335
6.7.6 Das Substitutionsprinzip ... 336
6.7.7 Automatische und explizite Typanpassung ... 337
6.7.8 Typen testen mit dem binären Operator instanceof ... 339
6.7.9 Array-Typen und Kovarianz ... 339
6.7.10 Methoden überschreiben ... 340
6.7.11 Mit super eine Methode der Oberklasse aufrufen ... 343
6.7.12 Kovariante Rückgabetypen ... 345
6.7.13 Finale Klassen ... 346
6.7.14 Nicht überschreibbare Funktionen ... 346
6.7.15 Zusammenfassung zur Sichtbarkeit ... 347
6.7.16 Sichtbarkeit in der UML ... 348
6.7.17 Zusammenfassung: Konstruktoren und Methoden ... 348

6.8 Object ist die Mutter aller Oberklassen ... 349
6.8.1 Klassenobjekte ... 349
6.8.2 Objektidentifikation mit toString() ... 350
6.8.3 Objektgleichheit mit equals() und Identität ... 352
6.8.4 Klonen eines Objekts mit clone() ... 355
6.8.5 Hashcodes ... 358
6.8.6 Aufräumen mit finalize() ... 359
6.8.7 Synchronisation ... 361

6.9 Die Oberklasse gibt Funktionalität vor ... 361
6.9.1 Dynamisches Binden als Beispiel für Polymorphie ... 363
6.9.2 Keine Polymorphie bei privaten, statischen und finalen Methoden ... 364
6.9.3 Polymorphie bei Konstruktoraufrufen ... 366

6.10 Abstrakte Klassen und abstrakte Methoden ... 368
6.10.1 Abstrakte Klassen ... 368
6.10.2 Abstrakte Methoden ... 369

6.11 Schnittstellen ... 372
6.11.1 Ein Polymorphie-Beispiel mit Schnittstellen ... 375
6.11.2 Die Mehrfachvererbung bei Schnittstellen ... 377
6.11.3 Erweitern von Interfaces – Subinterfaces ... 379
6.11.4 Vererbte Konstanten bei Schnittstellen ... 379
6.11.5 Vordefinierte Methoden einer Schnittstelle ... 381
6.11.6 Abstrakte Klassen und Schnittstellen im Vergleich ... 383

6.11.7 CharSequence als Beispiel einer Schnittstelle ... 383
6.11.8 Die Schnittstelle Iterable ... 385

6.12 Innere Klassen ... 388

6.12.1 Statische innere Klassen und Schnittstellen ... 388
6.12.2 Mitglieds- oder Elementklassen ... 389
6.12.3 Lokale Klassen ... 392
6.12.4 Anonyme innere Klassen ... 392
6.12.5 this und Vererbung ... 396
6.12.6 Implementierung einer verketteten Liste ... 397
6.12.7 Funktionszeiger ... 399

6.13 Generische Datentypen ... 401

6.13.1 Einfache Klassenschablonen ... 402
6.13.2 Einfache Methodenschablonen ... 403
6.13.3 Umsetzen der Generics, Typlöschung und Raw-Types ... 404
6.13.4 Einschränken der Typen ... 405
6.13.5 Generics und Vererbung, Invarianz ... 406
6.13.6 Wildcards ... 407

6.14 Die Spezial-Oberklasse Enum ... 408

6.14.1 Methoden auf Enum-Objekten ... 408
6.14.2 enum mit eigenen Konstruktoren und Methoden ... 410

6.15 Dokumentationskommentare mit javaDoc ... 412

6.15.1 Einen Dokumentationskommentar setzen ... 413
6.15.2 Mit javadoc eine Dokumentation erstellen ... 414
6.15.3 HTML-Tags in Dokumentationskommentaren ... 414
6.15.4 Generierte Dateien ... 415
6.15.5 Weitere Dokumentationskommentare ... 415
6.15.6 javaDoc und Doclets ... 416
6.15.7 Veraltete (deprecated) Klassen, Konstruktoren und Methoden ... 417

7 Exceptions ... 421

7.1 Problembereiche einzäunen ... 423

7.1.1 Exceptions in Java mit try und catch ... 423
7.1.2 Eine Datei auslesen mit RandomAccessFile ... 424
7.1.3 Ablauf einer Ausnahmesituation ... 425
7.1.4 Wiederholung kritischer Bereiche ... 425
7.1.5 throws im Methodenkopf angeben ... 426
7.1.6 Abschließende Arbeiten mit finally ... 427
7.1.7 Nicht erreichbare catch-Klauseln ... 430

7.2 Die Klassenhierarchie der Fehler ... 431

7.2.1 Die Exception-Hierarchie ... 431
7.2.2 Oberausnahmen fangen ... 432
7.2.3 Alles geht als Exception durch ... 433
7.2.4 RuntimeException muss nicht aufgefangen werden ... 434
7.2.5 Harte Fehler: Error ... 435

7.3 Werfen eigener Exceptions ... 435

7.3.1 Mit throw Ausnahmen auslösen ... 435

7.3.2 Neue Exception-Klassen definieren ... 437
7.3.3 Geschachtelte Ausnahmen ... 438
7.4 Rückgabewerte bei ausgelösten Ausnahmen ... 439
7.5 Der Stack Trace ... 440
7.5.1 Stack Trace aus Throwable ... 440
7.5.2 Stack Trace aus Thread ... 442
7.6 Assertions ... 442
7.6.1 Assertions in Java ... 442
7.6.2 Assertions in eigenen Programmen nutzen ... 442
7.6.3 Assertions aktivieren ... 443

8 Die Funktionsbibliothek 445

8.1 Die Java-Klassenphilosophie ... 447
8.1.1 Übersicht über die Pakete der Standardbibliothek ... 447
8.2 Wrapper-Klassen ... 453
8.2.1 Die Basisklasse Number für numerische Wrapper-Objekte ... 454
8.2.2 Die Klasse Integer ... 455
8.2.3 Wertebereich eines Typs und Überlaufkontrolle ... 458
8.2.4 Unterschiedliche Ausgabeformate ... 459
8.2.5 Autoboxing: Boxing und Unboxing ... 460
8.2.6 Die Character-Klasse ... 463
8.2.7 Die Boolean-Klasse ... 465
8.3 Benutzereinstellungen ... 466
8.3.1 Eine zentrale Registry ... 466
8.3.2 Einträge einfügen, auslesen und löschen ... 467
8.3.3 Auslesen der Daten und Schreiben in anderem Format ... 469
8.3.4 Auf Ereignisse horchen ... 469
8.4 Die Utility-Klasse System ... 469
8.4.1 Systemeigenschaften der Java-Umgebung ... 469
8.4.2 line.separator ... 470
8.4.3 Browser-Version abfragen ... 470
8.4.4 Property von der Konsole aus setzen ... 471
8.4.5 Umgebungsvariablen des Betriebssystems ... 472
8.4.6 Einfache Zeitmessung und Profiling ... 473
8.5 Ausführen externer Programme ... 475
8.5.1 Arbeiten mit dem ProcessBuilder ... 475
8.5.2 Die Rückgabe Process übernimmt die Prozesskontrolle ... 476
8.5.3 DOS-Programme aufrufen ... 477
8.5.4 Umgebungsvariablen und Startverzeichnis ... 477
8.5.5 Auf das Ende warten ... 478
8.5.6 Die Windows-Registry verwenden ... 478
8.5.7 Einen HTML-Browser unter Windows aufrufen ... 480
8.6 Klassenlader (Class Loader) ... 480
8.6.1 Woher die kleinen Klassen kommen ... 480
8.6.2 Die wichtigsten drei Typen von Klassenladern ... 482

8.6.3 Der java.lang.ClassLoader ... 483
8.6.4 Hot Deployment mit dem URL-ClassLoader ... 484
8.6.5 Das jre/lib/endorsed-Verzeichnis ... 486
8.6.6 getContextClassLoader() vom Thread ... 486
8.6.7 Wie heißt die Klasse mit der Methode main()? ... 487

8.7 Annotationen ... 488

8.7.1 Die eingebauten Annotations-Typen aus java.lang ... 488
8.7.2 @Deprecated ... 489
8.7.3 Annotationen mit zusätzlichen Informationen ... 489
8.7.4 @SuppressWarnings ... 489

9 Threads und nebenläufige Programmierung 491

9.1 Prozesse und Threads ... 493

9.1.1 Wie parallele Programme die Geschwindigkeit steigern können ... 494

9.2 Threads erzeugen ... 496

9.2.1 Threads über die Schnittstelle Runnable implementieren ... 496
9.2.2 Thread mit Runnable starten ... 497
9.2.3 Der Name eines Threads ... 499
9.2.4 Die Klasse Thread erweitern ... 499
9.2.5 Wer bin ich? ... 501

9.3 Der Ausführer (Executor) kommt ... 502

9.3.1 Die Schnittstelle Executor ... 502
9.3.2 Die Thread-Pools ... 503
9.3.3 Threads mit Rückgabe über Callable ... 504
9.3.4 Mehrere Callable abarbeiten ... 506
9.3.5 Mit ScheduledExecutorService wiederholende Ausgaben und Zeitsteuerungen ... 506

9.4 Die Zustände eines Threads ... 507

9.4.1 Threads schlafen ... 508
9.4.2 Das Ende eines Threads ... 510
9.4.3 UncaughtExceptionHandler für unbehandelte Ausnahmen ... 510
9.4.4 Einen Thread höflich mit Interrupt beenden ... 511
9.4.5 Der stop() von außen und die Rettung mit ThreadDeath ... 513
9.4.6 Ein Rendezvous mit join() und Barrier sowie Austausch mit Exchanger ... 514
9.4.7 Mit yield() auf Rechenzeit verzichten ... 517
9.4.8 Arbeit niederlegen und wieder aufnehmen ... 517
9.4.9 Priorität ... 518
9.4.10 Der Thread ist ein Dämon ... 519

9.5 Synchronisation über kritische Abschnitte ... 521

9.5.1 Gemeinsam genutzte Daten ... 521
9.5.2 Probleme beim gemeinsamen Zugriff und kritische Abschnitte ... 521
9.5.3 Punkte parallel initialisieren ... 522
9.5.4 i++ sieht atomar aus, ist es aber nicht ... 523
9.5.5 Kritische Abschnitte schützen ... 524
9.5.6 Schützen mit ReentrantLock ... 525
9.5.7 Synchronisieren mit synchronized ... 529
9.5.8 Synchronized-Methoden der Klasse StringBuffer ... 531

9.5.9 Mit synchronized synchronisierte Blöcke ... 531
9.5.10 Look-Freigabe im Fall von Exceptions ... 532
9.5.11 Mit synchronized nachträglich synchronisieren ... 533
9.5.12 Monitore sind reentrant, gut für die Geschwindigkeit ... 534
9.5.13 Synchronisierte Methodenaufrufe zusammenfassen ... 535
9.5.14 Deadlocks ... 536
9.5.15 Erkennen von Deadlocks ... 537

9.6 Synchronisation über Warten und Benachrichtigen ... 538

9.6.1 Die Schnittstelle Condition ... 539
9.6.2 Beispiel Erzeuger-Verbraucher-Programm ... 542
9.6.3 Warten mit wait() und Aufwecken mit notify() ... 546
9.6.4 Falls der Lock fehlt: IllegalMonitorStateException ... 547
9.6.5 Semaphoren ... 548

9.7 Atomares und frische Werte mit volatile ... 551

9.7.1 Der Modifizierer volatile bei Objekt-/Klassenvariablen ... 552
9.7.2 Das Paket java.util.concurrent.atomic ... 553

9.8 Mit dem Thread verbundene Variablen ... 554

9.8.1 ThreadLocal ... 554
9.8.2 InheritableThreadLocal ... 556

9.9 Gruppen von Threads in einer Thread-Gruppe ... 557

9.9.1 Aktive Threads in der Umgebung ... 557
9.9.2 Etwas über die aktuelle Thread-Gruppe herausfinden ... 558
9.9.3 Threads in einer Thread-Gruppe anlegen ... 560
9.9.4 Methoden von Thread und ThreadGroup im Vergleich ... 563

9.10 Die Klassen Timer und TimerTask ... 564

9.10.1 Job-Scheduler Quartz ... 565

9.11 Einen Abbruch der virtuellen Maschine erkennen ... 565

10 Raum und Zeit 569

10.1 Greenwich Mean Time (GMT) ... 571

10.2 Wichtige Datum-Klassen im Überblick ... 572

10.3 Sprachen der Länder ... 572

10.3.1 Sprachen und Regionen über Locale-Objekte ... 573

10.4 Übersetzung durch ResourceBundle-Objekte ... 576

10.5 Zeitzonen ... 578

10.5.1 Zeitzonen durch die Klasse TimeZone repräsentieren ... 580

10.6 Die Klasse Date ... 581

10.6.1 Objekte erzeugen und Methoden nutzen ... 581

10.7 Calendar und GregorianCalendar ... 583

10.7.1 Die abstrakte Klasse Calendar ... 583
10.7.2 Der gregorianische Kalender ... 584
10.7.3 Ostertage ... 586
10.7.4 Abfragen und Setzen von Datumselementen ... 588

10.8 Formatieren der Datumsangaben ... 593
10.8.1 Mit DateFormat und SimpleDateFormat formatieren ... 593
10.8.2 Parsen von Datumswerten ... 600
10.8.3 Parsen und Formatieren ab bestimmten Positionen ... 602

11 Datenstrukturen und Algorithmen 603

11.1 Datenstrukturen und die Collection-API ... 605
11.1.1 Die Schnittstelle Collection ... 605
11.1.2 Das erste Programm mit Container-Klassen ... 607
11.1.3 Die Schnittstelle Iterable und das erweiterte for ... 608
11.1.4 Generische Datentypen in der Collection-API ... 609
11.1.5 Generischer Typ bei Iterable und konkreter Typ beim erweiterten for ... 610
11.1.6 Schnittstellen, die Collection erweitern und Map ... 610
11.1.7 Konkrete Container-Klassen ... 612
11.2 Mit einem Iterator durch die Daten wandern ... 612
11.2.1 Die Schnittstellen Enumeration und Iterator ... 612
11.2.2 Der typisierte Iterator ... 614
11.3 Listen ... 616
11.3.1 Die Schnittstelle List ... 617
11.3.2 Beispiel mit List-Methoden ... 619
11.3.3 ArrayList ... 621
11.3.4 Arrays.asList() und die »echten« Listen ... 623
11.3.5 toArray() von Collection verstehen – die Gefahr einer Falle erkennen ... 623
11.3.6 Die interne Arbeitsweise von ArrayList und Vector ... 626
11.3.7 LinkedList ... 627
11.4 Stack (Kellerspeicher, Stapel) ... 627
11.4.1 Die Methoden von Stack ... 628
11.4.2 Ein Stack ist ein Vector – aha! ... 628
11.5 Queues (Schlangen) ... 629
11.5.1 Blockierende Queues und Prioritätswarteschlangen ... 630
11.6 Assoziative Speicher HashMap und TreeMap ... 631
11.6.1 Ein Objekt der Klasse HashMap erzeugen ... 631
11.6.2 Einfügen und Abfragen der Datenstruktur ... 631
11.6.3 Wichtige Eigenschaften von Assoziativspeichern ... 634
11.6.4 Elemente im Assoziativspeicher müssen unveränderbar bleiben ... 634
11.6.5 Aufzählen der Elemente ... 635
11.6.6 Der Gleichheitstest, Hash-Wert und Klon einer Hash-Tabelle ... 636
11.6.7 Die Arbeitsweise einer Hash-Tabelle ... 637
11.7 Die Properties-Klasse ... 639
11.7.1 Properties setzen und lesen ... 639
11.7.2 Properties verketten ... 639
11.7.3 Eigenschaften ausgeben ... 641
11.7.4 Hierarchische Eigenschaften ... 641
11.7.5 Properties speichern ... 641
11.7.6 Über die Beziehung Properties und Hashtable ... 643

11.8 Mengen (Sets) ... 643
11.8.1 HashSet ... 645
11.8.2 TreeSet – die Menge durch Bäume ... 646
11.8.3 LinkedHashSet ... 647
11.9 Algorithmen in Collections ... 648
11.9.1 Datenmanipulation: Umdrehen, Füllen, Kopieren ... 649
11.9.2 Vergleichen von Objekten mit Comparator und Comparable ... 649
11.9.3 Den größten und kleinsten Wert einer Collection finden ... 652
11.9.4 Sortieren ... 653
11.9.5 Elemente in der Collection suchen ... 656
11.9.6 Nicht-änderbare Datenstrukturen ... 657
11.9.7 nCopies() ... 657
11.9.8 Singletons ... 658
11.10 Synchronisation der Datenstrukturen ... 658
11.10.1 Lock-Free-Algorithmen aus java.util.concurrent ... 658
11.10.2 Wrapper zur Synchronisation ... 659
11.10.3 CopyOnWriteArrayList und CopyOnWriteArraySet ... 660
11.11 Die abstrakten Basisklassen für Container ... 660
11.11.1 Optionale Methoden ... 661
11.12 Die Klasse BitSet für Bitmengen ... 661
11.12.1 Ein BitSet anlegen und füllen ... 661
11.12.2 Mengenorientierte Operationen ... 662
11.12.3 Funktionsübersicht ... 663
11.12.4 Primzahlen in einem BitSet verwalten ... 664
11.13 Ein Design-Pattern durch Beobachten von Änderungen ... 665
11.13.1 Design-Pattern ... 665
11.13.2 Das Beobachter-Pattern (Observer/Observable) ... 666

12 Datenströme und Dateien 671

12.1 Datei und Verzeichnis ... 673
12.1.1 Dateien und Verzeichnisse mit der Klasse File ... 674
12.1.2 Dateieigenschaften und -attribute ... 674
12.1.3 Änderungsdatum einer Datei, Nur-Lese-Rechte setzen ... 676
12.1.4 Dateien berühren, neue Dateien anlegen, temporäre Dateien ... 677
12.1.5 Umbenennen und Verzeichnisse anlegen ... 678
12.1.6 Die Wurzel aller Verzeichnisse/Laufwerke ... 678
12.1.7 Verzeichnisse listen und Dateien filtern ... 680
12.1.8 Dateien und Verzeichnisse löschen ... 682
12.1.9 Verzeichnisse nach Dateien rekursiv durchsuchen ... 684
12.1.10 Namen der Laufwerke ... 685
12.1.11 URL- und URI-Objekte aus einem File-Objekt ableiten ... 686
12.1.12 Locking ... 686
12.1.13 Sicherheitsprüfung ... 687
12.1.14 Implementierungsmöglichkeiten für die Klasse File ... 687
12.1.15 Mime-Typen ... 688

12.2 Dateien mit wahlfreiem Zugriff 689
12.2.1 Ein RandomAccessFile zum Lesen und Schreiben öffnen 689
12.2.2 Aus dem RandomAccessFile lesen 689
12.2.3 Schreiben 691
12.2.4 Die Länge des RandomAccessFile 691
12.2.5 Hin und her in der Datei 691
12.2.6 Wahlfreier Zugriff und Pufferung mit Unified I/O 693
12.3 Stream-Klassen und Reader/Writer 693
12.3.1 Die abstrakten Basisklassen 693
12.3.2 Übersicht über Ein-/Ausgabeklassen 694
12.4 Binäre Ein-/Ausgabe-Klassen InputStream/OutputStream 695
12.4.1 Die abstrakte Basisklasse OutputStream 695
12.4.2 Die Schnittstellen Closeable und Flushable 697
12.4.3 Ein Datenschlucker 697
12.4.4 Anwendung der Klasse FileOutputStream 698
12.4.5 Die abstrakte Eingabeklasse InputStream 699
12.4.6 Ressourcen wie Grafiken aus dem Klassenpfad und aus Jar-Archiven laden 700
12.4.7 Anwenden der Klasse FileInputStream 700
12.4.8 Kopieren von Dateien 701
12.4.9 Das FileDescriptor-Objekt 703
12.5 PrintStream und Konsolenausgaben 703
12.5.1 Die Klasse PrintStream 704
12.5.2 Die Schnittstelle Appendable 705
12.5.3 System.in und System.out 706
12.5.4 Ströme umlenken 708
12.5.5 Den Bildschirm löschen und Textausgaben optisch aufwerten 710
12.6 Daten filtern durch FilterInputStream und FilterOutputStream 711
12.6.1 DataOutputStream/DataInputStream 712
12.7 Besondere OutputStream- und InputStream-Klassen 712
12.7.1 Mit ByteArrayOutputStream in ein Byte-Feld schreiben 713
12.7.2 Mit ByteArrayInputStream aus einem Byte-Feld lesen 714
12.7.3 Ströme zusammensetzen mit SequenceInputStream 714
12.8 Die Unterklassen von Writer 716
12.8.1 Die abstrakte Basisklasse Writer 717
12.8.2 Ausgabemöglichkeiten durch PrintWriter erweitern 719
12.8.3 Datenkonvertierung durch den OutputStreamWriter 721
12.8.4 In Dateien schreiben mit der Klasse FileWriter 722
12.8.5 StringWriter und CharArrayWriter 723
12.8.6 Writer als Filter verketten 725
12.8.7 Gepufferte Ausgabe durch BufferedWriter 726
12.8.8 Daten mit FilterWriter filtern 728
12.9 Die Klassen um Reader 733
12.9.1 Die abstrakte Basisklasse Reader 734
12.9.2 Automatische Konvertierungen mit dem InputStreamReader 736
12.9.3 Dateien lesen mit der Klasse FileReader 737
12.9.4 StringReader und CharArrayReader 738

12.10 Die Filter für Zeichenströme ... 739
12.10.1 Gepufferte Eingaben mit der Klasse BufferedReader ... 739
12.10.2 LineNumberReader zählt automatisch Zeilen mit ... 741
12.10.3 Eingaben filtern mit der Klasse FilterReader ... 743
12.10.4 Daten mit der Klasse PushbackReader zurücklegen ... 745
12.11 Kommunikation zwischen Threads mit Pipes ... 748
12.11.1 PipedOutputStream und PipedInputStream ... 748
12.11.2 PipedWriter und PipedReader ... 750
12.12 Datenkompression ... 752
12.12.1 Java-Unterstützung beim Komprimieren und Zusammenpacken ... 753
12.12.2 Datenströme komprimieren ... 753
12.12.3 Zip-Archive ... 756
12.12.4 Jar-Archive ... 763
12.13 Prüfsummen ... 763
12.13.1 Die Schnittstelle Checksum ... 763
12.13.2 Die Klasse CRC32 ... 764
12.13.3 Die Adler32-Klasse ... 766
12.14 Persistente Objekte und Serialisierung ... 766
12.14.1 Objekte speichern mit der Standard-Serialisierung ... 767
12.14.2 Objekte über die Standard-Serialisierung lesen ... 770
12.14.3 Die Schnittstelle Serializable ... 772
12.14.4 Nicht serialisierbare Attribute mit transient aussparen ... 773
12.14.5 Das Abspeichern selbst in die Hand nehmen ... 774
12.14.6 Tiefe Objektkopien ... 777
12.14.7 Versionenverwaltung und die SUID ... 779
12.14.8 Wie die ArrayList serialisiert ... 781
12.14.9 Probleme mit der Serialisierung ... 782
12.14.10 Serialisieren in XML-Dateien ... 782
12.14.11 JavaBeans Persistence ... 782
12.14.12 XStream ... 784
12.15 Zugriff auf SMB-Server ... 785
12.15.1 jCIFS ... 785
12.16 Tokenizer ... 786
12.16.1 StreamTokenizer ... 786
12.16.2 CSV (Comma Separated Values)-Dateien verarbeiten ... 789
12.17 Die Logging-API ... 790

13 Die eXtensible Markup Language (XML) 793

13.1 Auszeichnungssprachen ... 795
13.1.1 Die Standard Generalized Markup Language (SGML) ... 795
13.1.2 Extensible Markup Language (XML) ... 796
13.2 Eigenschaften von XML-Dokumenten ... 796
13.2.1 Elemente und Attribute ... 796
13.2.2 Beschreibungssprache für den Aufbau von XML-Dokumenten ... 798
13.2.3 Schema – eine Alternative zu DTD ... 801

13.2.4 Namensraum (Namespace) ... 803
13.2.5 XML-Applikationen ... 805
13.3 Die Java-APIs für XML ... 805
13.3.1 Das Document Object Model (DOM) ... 805
13.3.2 Simple API for XML Parsing (SAX) ... 806
13.3.3 Java Document Object Model (JDOM) ... 806
13.3.4 Pull-API StAX ... 806
13.4 Serielle Verarbeitung mit StAX ... 807
13.4.1 Unterschiede der Verarbeitungsmodelle ... 807
13.4.2 XML-Dateien mit dem Cursor-Verfahren lesen ... 807
13.4.3 XML-Dateien mit dem Iterator-Verfahren verarbeiten ... 809
13.4.4 Mit Filtern arbeiten ... 810
13.4.5 XML-Dokumente schreiben ... 811
13.5 Serielle Verarbeitung von XML mit SAX ... 813
13.5.1 Schnittstellen von SAX ... 813
13.5.2 SAX-Parser erzeugen ... 813
13.5.3 Die wichtigsten Methoden der Schnittstelle ContentHandler ... 814
13.6 XML-Dateien mit JDOM verarbeiten ... 815
13.6.1 JDOM beziehen ... 816
13.6.2 Paketübersicht ... 816
13.6.3 Die Document-Klasse ... 817
13.6.4 Eingaben aus der Datei lesen ... 818
13.6.5 Das Dokument im XML-Format ausgeben ... 819
13.6.6 Der Dokumenttyp ... 819
13.6.7 Elemente ... 820
13.6.8 Zugriff auf Elementinhalte ... 822
13.6.9 Liste mit Unterelementen erzeugen ... 824
13.6.10 Neue Elemente einfügen und ändern ... 825
13.6.11 Attributinhalte lesen und ändern ... 827
13.6.12 XPath ... 830
13.7 JAXP als Java-Schnittstelle zu XML ... 832
13.7.1 Einführung in XSLT ... 833
13.7.2 Umwandlung von XML-Dateien mit JDOM und JAXP ... 835
13.8 HTML-Dokumente einlesen ... 836

14 Grafikprogrammierung mit dem AWT **839**

14.1 Das Abstract-Window-Toolkit ... 841
14.1.1 Java Foundation Classes ... 841
14.2 Das Toolkit ... 842
14.3 Fenster unter grafischen Oberflächen ... 843
14.3.1 AWT-Fenster darstellen ... 843
14.3.2 Swing-Fenster darstellen ... 844
14.3.3 Sichtbarkeit des Fensters ... 845
14.3.4 Größe und Position des Fensters verändern ... 846
14.3.5 Hauptprogramm von Frame/JFrame ableiten ... 847
14.3.6 Fenster- und Dialog-Dekoration ... 848

14.4 Grundlegendes zum Zeichnen ... 848
14.4.1 Die paint()-Methode für das AWT-Frame ... 848
14.4.2 Zeichen von Inhalten mit JFrame ... 850
14.4.3 Auffordern zum Neuzeichnen mit repaint() ... 851
14.4.4 Die ereignisorientierte Programmierung ändert Fensterinhalte ... 852
14.5 Einfache Zeichenfunktionen ... 854
14.5.1 Grundbegriffe: Koordinaten, Punkte, Pixel ... 854
14.5.2 Linien ... 854
14.5.3 Rechtecke ... 855
14.5.4 Ovale und Kreisbögen ... 856
14.5.5 Polygone und Polylines ... 856
14.6 Zeichenketten schreiben ... 860
14.6.1 Einen neuen Zeichensatz bestimmen ... 860
14.6.2 Ableiten eines neuen Fonts aus einem gegebenen Font ... 862
14.6.3 Zeichensätze des Systems ermitteln ... 863
14.6.4 Die Klasse FontMetrics ... 864
14.6.5 True Type Fonts ... 867
14.7 Clipping-Operationen ... 868
14.8 Farben ... 872
14.8.1 Zufällige Farbblöcke zeichnen ... 873
14.8.2 Farbanteile zurückgeben ... 875
14.8.3 Vordefinierte Farben ... 875
14.8.4 Farben aus Hexadezimalzahlen erzeugen ... 876
14.8.5 Einen helleren oder dunkleren Farbton wählen ... 877
14.8.6 Farbmodelle HSB und RGB ... 878
14.8.7 Die Farben des Systems ... 879
14.9 Bilder anzeigen und Grafiken verwalten ... 882
14.9.1 Bilder laden: eine Übersicht ... 883
14.9.2 Bilder über Toolkit laden für Applikationen ... 884
14.9.3 Bilder in Applets ... 884
14.9.4 Asynchrones Laden mit getImage() und dem MediaTracker ... 885
14.9.5 Programm-Icon/Fenster-Icon setzen ... 886
14.9.6 Das Image zeichnen ... 886
14.9.7 Grafiken zentrieren ... 889
14.9.8 Bilder im Speicher erzeugen ... 889
14.9.9 Kein Flackern durch Double-Buffering ... 890
14.9.10 Bilder skalieren ... 892
14.9.11 VolatileImage ... 894
14.9.12 Bilder lesen mit ImageIO ... 894
14.9.13 Schreiben mit ImageIO ... 895
14.9.14 Kann ImageIO ein Format behandeln? ... 895
14.9.15 Komprimieren mit ImageIO ... 896
14.9.16 Bilder im GIF-Format speichern ... 897
14.9.17 Gif speichern mit dem ACME-Paket ... 899
14.9.18 JPEG-Dateien mit dem Sun-Paket schreiben ... 899
14.9.19 Java Image Management Interface (JIMI) ... 902
14.10 Java 2D-API ... 902
14.10.1 Grafische Objekte zeichnen ... 903
14.10.2 Geometrische Objekte durch Shape gekennzeichnet ... 904

14.10.3 Zeichenhinweise durch RenderingHints ... 906
14.10.4 Windungs-Regel ... 907
14.10.5 Dicke und Art der Linien bestimmen ... 909
14.10.6 Transformationen mit einem AffineTransform-Objekt ... 912

14.11 Produzenten, Konsumenten und Beobachter ... 914

14.11.1 Producer und Consumer für Bilder ... 914
14.11.2 Beispiel für die Übermittlung von Daten ... 915
14.11.3 Bilder selbst erstellen ... 918
14.11.4 Die Bildinformationen wieder auslesen ... 921

14.12 Filter ... 924

14.12.1 Grundlegende Eigenschaft von Filtern ... 924
14.12.2 Konkrete Filterklassen ... 925
14.12.3 Mit CropImageFilter Teile ausschneiden ... 926
14.12.4 Transparenz ... 927

14.13 Drucken ... 928

14.13.1 Drucken mit dem einfachen Ansatz ... 928
14.13.2 Ein PrintJob ... 929
14.13.3 Drucken der Inhalte ... 930
14.13.4 Komponenten drucken ... 932
14.13.5 Den Drucker am Parallelport ansprechen ... 933
14.13.6 Bekannte Drucker ... 933

14.14 Grafikverarbeitung ohne grafische Oberfläche ... 933

14.14.1 Xvfb-Server ... 934
14.14.2 Pure Java AWT Toolkit (PJA) ... 934

15 Komponenten, Container und Ereignisse 937

15.1 Es tut sich was – Ereignisse beim AWT ... 939

15.1.1 Die Klasse AWTEvent ... 939
15.1.2 Events auf verschiedenen Ebenen ... 940
15.1.3 Ereignisquellen und Horcher (Listener) ... 942
15.1.4 Listener implementieren ... 942
15.1.5 Listener bei Ereignisauslöser anmelden/abmelden ... 943
15.1.6 Aufrufen der Listener ... 944

15.2 Varianten, das Fenster zu schließen ... 944

15.2.1 Eine Klasse implementiert die Schnittstelle WindowListener ... 945
15.2.2 Adapterklassen nutzen ... 947
15.2.3 Innere Mitgliedsklassen und innere anonyme Klassen ... 948
15.2.4 Generic Listener ... 949

15.3 Komponenten im AWT und in Swing ... 949

15.3.1 Peer-Klassen und Lightweight-Komponenten ... 950
15.3.2 Die Basis aller Komponenten: Component und JComponent ... 951
15.3.3 Proportionales Vergrößern eines Fensters ... 955
15.3.4 Dynamisches Layout während einer Größenänderung ... 956
15.3.5 Hinzufügen von Komponenten ... 957

15.4 Swing-Fenster JFrame, JDialog, JWindow 958
15.4.1 Kinder auf einem Swing-Fenster 958
15.4.2 Schließen eines Swing-Fensters 958
15.4.3 JWindow und JDialog 959
15.5 Informationstext über die Klasse JLabel 960
15.5.1 Mehrzeiliger Text, HTML in der Darstellung 962
15.6 Die Klasse ImageIcon 963
15.6.1 Die Schnittstelle Icon 965
15.6.2 Was Icon und Image verbindet 966
15.7 Eine Schaltfläche (JButton) 967
15.7.1 Der aufmerksame ActionListener 969
15.7.2 Generic Listener für Schaltflächen-Ereignisse verwenden 970
15.7.3 AbstractButton 973
15.7.4 JToggleButton 975
15.8 Tooltips 975
15.9 Der Container JPanel 976
15.10 Alles Auslegungssache: die Layoutmanager 977
15.10.1 FlowLayout 978
15.10.2 Mit BorderLayout in allen Himmelsrichtungen 980
15.10.3 GridLayout 982
15.10.4 Der GridBagLayout-Manager 984
15.10.5 Null-Layout 989
15.10.6 Weitere Layoutmanager 989
15.11 Horizontale und vertikale Rollbalken 990
15.11.1 Der AdjustmentListener, der auf Änderungen hört 993
15.12 JSlider 995
15.13 Ein Auswahlmenü – Choice, JComboBox 996
15.13.1 ItemListener 999
15.13.2 Zuordnung einer Taste mit einem Eintrag 1000
15.13.3 DateComboBox 1001
15.14 Eines aus vielen – Kontrollfelder (JCheckBox) 1001
15.14.1 Ereignisse über ItemListener 1003
15.15 Kontrollfeldgruppen, Optionsfelder, JRadioButton 1003
15.16 Der Fortschrittsbalken JProgressBar 1005
15.17 Rahmen (Borders) 1007
15.18 Symbolleisten alias Toolbars 1009
15.19 Menüs 1011
15.19.1 Die Menüleisten und die Einträge 1011
15.19.2 Menüeinträge definieren 1013
15.19.3 Mnemonics und Shortcuts (Accelerator) 1014
15.19.4 Popup-Menüs 1016
15.20 Das Konzept des Model-View-Controllers 1019
15.21 List-Boxen 1021
15.22 JSpinner 1024

15.23 Texteingabefelder ... 1025
15.23.1 Text in einer Eingabezeile ... 1026
15.23.2 Die Oberklasse der JText-Komponenten: JTextComponent ... 1026
15.23.3 JPasswordField ... 1027
15.23.4 Validierende Eingabefelder ... 1028
15.23.5 Mehrzeilige Textfelder ... 1029
15.23.6 Die Editor-Klasse JEditorPane ... 1031
15.24 Bäume mit JTree-Objekten ... 1034
15.24.1 Selektionen bemerken ... 1035
15.25 Tabellen mit JTable ... 1036
15.25.1 Ein eigenes Tabellen-Model ... 1038
15.25.2 AbstractTableModel ... 1038
15.25.3 DefaultTableModel ... 1042
15.25.4 Ein eigener Renderer für Tabellen ... 1043
15.25.5 Zell-Editoren ... 1046
15.25.6 Größe und Umrandung der Zellen ... 1047
15.25.7 Spalteninformationen ... 1048
15.25.8 Tabellenkopf von Swing-Tabellen ... 1048
15.25.9 Selektionen einer Tabelle ... 1049
15.25.10 Ein professionelles Tabellenlayout mit JGrid ... 1050
15.26 JRootPane, JLayeredPane und JDesktopPane ... 1050
15.26.1 JRootPane und JLayeredPane ... 1050
15.26.2 JDesktopPane und die Kinder JInternalFrame ... 1051
15.27 Dialoge ... 1053
15.27.1 Der Farbauswahldialog JColorChooser ... 1054
15.27.2 Der Dateiauswahldialog ... 1056
15.28 Flexibles Java-Look&Feel ... 1058
15.28.1 UIManager ... 1059
15.28.2 Ändern des Aussehens Laufzeit ... 1060
15.28.3 Verbessern des Aussehens unter Windows mit WinLAF ... 1061
15.29 Swing-Beschriftungen einer anderen Sprache geben ... 1061
15.30 Die Zwischenablage (Clipboard) ... 1062
15.31 Undo durchführen ... 1066
15.32 Ereignisverarbeitung auf unterster Ebene ... 1067
15.32.1 Eigene Ereignisse in die Queue setzen ... 1068
15.32.2 Auf alle Ereignisse hören ... 1069
15.33 AWT, Swing und die Threads ... 1069
15.33.1 Swing ist nicht Thread-sicher ... 1070
15.33.2 Swing-Elemente bedienen mit invokeLater(), invokeAndWait() ... 1072
15.34 Selbst definierte Cursor ... 1073
15.34.1 Flackern des Mauszeigers bei Animationen vermeiden ... 1074
15.35 Benutzerinteraktionen automatisieren ... 1075
15.35.1 Automatisch in die Tasten hauen ... 1076
15.35.2 Mausoperationen ... 1077
15.35.3 Methoden zur Zeitsteuerung ... 1077
15.35.4 Screenshots ... 1077

15.35.5 Funktionsweise und Beschränkungen ... 1078
15.35.6 MouseInfo und PointerInfo ... 1078
15.36 Zeitliches Ausführen mit dem javax.swing.Timer ... 1079
15.37 Alternativen zu AWT und Swing ... 1079
15.37.1 XML-Beschreibungen der Oberfläche: Swixml, XUL/Luxor ... 1079
15.37.2 SWT ... 1080

16 Das Netz **1081**

16.1 Grundlegende Begriffe ... 1083
16.1.1 Internet-Standards und RFC ... 1083
16.2 URI und URL ... 1084
16.2.1 URI ... 1085
16.2.2 Die Klasse URL ... 1085
16.2.3 Informationen über eine URL ... 1088
16.2.4 Der Zugriff auf die Daten über die Klasse URL ... 1089
16.2.5 Verbindungen durch einen Proxy-Server ... 1091
16.3 Die Klasse URLConnection ... 1092
16.3.1 Methoden und Anwendung von URLConnection ... 1092
16.3.2 Protokoll- und Content-Handler ... 1095
16.3.3 Bilder-Handler ... 1095
16.3.4 Zusammenfassung Content- und Protokoll-Handler ... 1095
16.3.5 Im Detail: vom URL zu URLConnection ... 1096
16.3.6 Der Protokoll-Handler für Jar-Dateien ... 1097
16.3.7 Autorisierte URL-Verbindungen und Proxy-Authentifizierung mit Basic Authentication ... 1098
16.4 Das Common Gateway Interface ... 1100
16.4.1 Parameter für ein CGI-Programm ... 1100
16.4.2 Kodieren der Parameter für CGI-Programme ... 1101
16.4.3 Eine Suchmaschine ansprechen ... 1103
16.5 Host- und IP-Adressen ... 1104
16.5.1 Lebt der Rechner? ... 1105
16.5.2 Das Netz ist Klasse 1105
16.5.3 IP-Adresse des lokalen Hosts ... 1106
16.6 NetworkInterface ... 1107
16.7 Mit dem Socket zum Server ... 1108
16.7.1 Das Netzwerk ist der Computer ... 1108
16.7.2 Sockets ... 1109
16.7.3 Standarddienste unter Windows nachinstallieren ... 1110
16.7.4 Eine Verbindung zum Server aufbauen ... 1111
16.7.5 Server unter Spannung: die Ströme ... 1112
16.7.6 Die Verbindung wieder abbauen ... 1113
16.7.7 Ein kleines Echo – lebt der Rechner noch? ... 1113
16.7.8 Blockierendes Lesen ... 1114
16.7.9 Informationen über den Socket ... 1115
16.7.10 Mit telnet an den Ports horchen ... 1117
16.7.11 Reine Verbindungsdaten über SocketAddress ... 1118

16.8 Client/Server-Kommunikation ... 1119
16.8.1 Warten auf Verbindungen ... 1120
16.8.2 Ein Multiplikations-Server ... 1120
16.8.3 Von außen erreichbar sein ... 1123
16.9 SSL-Verbindungen mit JSSE ... 1124
16.10 Apache Jakarta Commons HttpClient und Net ... 1125
16.10.1 Jakarta Commons HttpClient ... 1125
16.10.2 Jakarta Commons Net ... 1126
16.11 E-Mail ... 1126
16.11.1 Wie eine E-Mail um die Welt geht ... 1126
16.11.2 Das Simple Mail Transfer Protocol und RFC 822 ... 1127
16.11.3 POP (Post Office Protocol) ... 1128
16.11.4 E-Mails versenden mit der JavaMail API von SUN ... 1129
16.11.5 MimeMultipart-Nachrichten schicken ... 1130
16.11.6 E-Mails mittels POP3 abrufen ... 1131
16.11.7 Ereignisse und Suchen ... 1133
16.12 Arbeitsweise eines Web-Servers ... 1133
16.12.1 Das Hypertext Transfer Protocol (HTTP) ... 1134
16.12.2 Anfragen an den Server ... 1134
16.12.3 Die Antworten vom Server ... 1137
16.13 Datagram-Sockets ... 1140
16.13.1 Die Klasse DatagramSocket ... 1142
16.13.2 Datagramme und die Klasse DatagramPacket ... 1143
16.13.3 Auf ein hereinkommendes Paket warten ... 1144
16.13.4 Ein Paket zum Senden vorbereiten ... 1145
16.13.5 Methoden der Klasse DatagramPacket ... 1146
16.13.6 Das Paket senden ... 1147
16.14 Tiefer liegende Netzwerkeigenschaften ... 1148
16.14.1 Internet Control Message Protocol (ICMP) ... 1148
16.14.2 MAC-Adresse ... 1148
16.15 Multicast-Kommunikation ... 1149

17 Servlets und Java Server Pages 1151

17.1 Dynamische Web-Seiten und Servlets ... 1153
17.1.1 Was sind Servlets? ... 1153
17.1.2 Was sind Java Server Pages? ... 1154
17.1.3 Vorteil von JSP/Servlets gegenüber CGI-Programmen ... 1155
17.2 Vom Client zum Server und wieder zurück ... 1156
17.2.1 Der bittende Client ... 1156
17.2.2 Welche Form von Antwort erzeugt ein Web-Server? ... 1158
17.2.3 Wer oder was ist MIME? ... 1158
17.3 Servlets und JSPs entwickeln und testen ... 1159
17.3.1 Servlet-Container ... 1159
17.3.2 Web-Server mit Servlet-Funktionalität ... 1160

17.4 Java Server Pages in Tomcat und Eclipse 1161
17.4.1 Download und Installation 1161
17.4.2 Ablageort für eigene JSP-Seiten 1162
17.4.3 Web-Applikationen 1163
17.4.4 Zuordnung von Web-Applikationen zu physikalischen Verzeichnissen 1163
17.5 Skript-Elemente 1164
17.5.1 Scriptlets 1164
17.5.2 Ausdrücke 1165
17.5.3 Deklarationen 1165
17.5.4 Kommentare 1166
17.5.5 Quoting 1166
17.5.6 Entsprechende XML-Tags 1166
17.6 Implizite Objekte 1167
17.7 Was der Browser mit auf den Weg gibt – HttpServletRequest 1168
17.7.1 Verarbeiten der Header 1168
17.7.2 Hilfsfunktion im Umgang mit Headern 1169
17.7.3 Übersicht der Browser-Header 1169
17.8 Formulardaten 1170
17.9 Das HttpServletResponse-Objekt 1172
17.9.1 Automatisches Neuladen 1172
17.9.2 Seiten umlenken 1173
17.10 JSP-Direktiven 1174
17.10.1 page-Direktiven im Überblick 1174
17.10.2 include-Direktive 1176
17.10.3 Mit JSPs Bilder generieren 1177
17.11 Aktionen 1178
17.11.1 Aktion include 1178
17.11.2 Aktion forward 1178
17.11.3 Aktion plugin 1178
17.12 Beans 1179
17.12.1 Beans in JSP-Seiten anlegen, Attribute setzen und erfragen 1180
17.12.2 Der schnelle Zugriff auf Parameter 1181
17.13 Kleine Kekse: die Klasse Cookies 1181
17.13.1 Cookies erzeugen und setzen 1182
17.13.2 Cookies vom Servlet einlesen 1183
17.13.3 Kleine Helfer für Cookies 1184
17.13.4 Cookie-Status ändern 1184
17.13.5 Langlebige Cookies 1186
17.14 Sitzungsverfolgung (Session Tracking) 1186
17.14.1 Das mit einer Sitzung verbundene Objekt HttpSession 1187
17.14.2 Werte mit einer Sitzung assoziieren und auslesen 1187
17.14.3 URL-Rewriting 1188
17.14.4 Zusätzliche Informationen 1189
17.15 Tag-Libraries 1191
17.15.1 Standard Tag Library (JSTL) 1191
17.15.2 Jakarta Taglibs Project 1192

17.16 Servlets ... 1193
17.16.1 Servlets compilieren ... 1194
17.16.2 Die Servlets in das classes-Verzeichnis ... 1194
17.16.3 Servlet-Mapping ... 1194
17.17 Der Lebenszyklus eines Servlets ... 1196
17.17.1 Initialisierung in init() ... 1197
17.17.2 Abfragen bei service() ... 1199
17.17.3 Mehrere Anfragen beim Servlet und die Thread-Sicherheit ... 1200
17.17.4 Das Ende eines Servlets ... 1201
17.18 Das HttpServletResponse-Objekt ... 1201
17.18.1 Wir generieren eine Web-Seite ... 1201
17.18.2 Binärdaten senden ... 1202
17.18.3 Noch mehr über Header, die der Server setzt ... 1203
17.19 Objekte und Dateien per POST verschicken ... 1204
17.19.1 Datei-Upload ... 1205
17.20 Servlets und Sessions ... 1206
17.21 Weiterleiten und Einbinden von Servlet-Inhalten ... 1207
17.22 Inter-Servlet-Kommunikation ... 1208
17.22.1 Daten zwischen Servlets teilen ... 1208
17.23 Internationalisierung ... 1209
17.23.1 Die Länderkennung des Anfragers auslesen ... 1209
17.23.2 Länderkennung für die Ausgabe setzen ... 1209
17.23.3 Westeuropäische Texte senden ... 1209
17.24 Tomcat: Spezielles ... 1211
17.24.1 Tomcat als Service unter Windows NT ausführen ... 1211
17.24.2 Interessante Links zum Thema Servlets/JSP ... 1211

18 Verteilte Programmierung mit RMI und SOAP 1213

18.1 Entfernte Methoden ... 1215
18.1.1 Wie entfernte Methoden arbeiten ... 1215
18.1.2 Stellvertreter (Proxy) ... 1215
18.1.3 RMI ... 1216
18.1.4 Wie die Stellvertreter die Daten übertragen ... 1216
18.1.5 Probleme mit entfernten Methoden ... 1217
18.2 Nutzen von RMI bei Middleware-Lösungen ... 1218
18.3 Die Lösung für Java ist RMI ... 1219
18.3.1 Entfernte Objekte programmieren ... 1219
18.3.2 Entfernte und lokale Objekte im Vergleich ... 1220
18.4 Definition einer entfernten Schnittstelle ... 1220
18.5 Implementierung der Remote-Schnittstelle ... 1221
18.6 Stellvertreterobjekte erzeugen ... 1221
18.7 Der Namensdienst (Registry) ... 1222
18.7.1 Der Port ... 1222

18.8 Der Server 1222
18.8.1 Entfernte Objekte beim Namensdienst anmelden 1222
18.8.2 Automatisches Anmelden bei Bedarf 1224
18.9 Einen Client programmieren 1224
18.9.1 Einfaches Logging 1225
18.10 Aufräumen mit dem DGC 1226
18.11 Entfernte Objekte übergeben und laden 1226
18.11.1 Klassen vom RMI-Klassenlader nachladen 1227
18.12 Registry wird vom Server gestartet 1227
18.13 RMI über die Firewall 1228
18.13.1 RMI über HTTP getunnelt 1228
18.14 RMI und CORBA 1229
18.15 Daily Soap 1229
18.15.1 SOAP-Implementierungen 1230
18.15.2 Einen Client mit der Apache-Bibliothek implementieren 1231
18.15.3 Der Seifen-Server 1232
18.16 Java Message Service (JMS) 1233

19 Applets, Midlets und Sound 1235

19.1 Applets und Applikationen – wer darf was? 1237
19.2 Das erste Hallo-Applet 1237
19.3 Die Zyklen eines Applets 1238
19.4 Parameter an das Applet übergeben 1239
19.5 Methoden der Applet- und AppletContext-Klasse 1240
19.5.1 Wie das Applet den Browser-Inhalt ändern kann 1240
19.5.2 Den Ursprung des Applets erfragen 1241
19.5.3 Datenaustausch zwischen Applets 1242
19.5.4 Was ein Applet alles darf 1245
19.6 Fehler in Applets finden 1245
19.7 Browserabhängiges Verhalten 1245
19.7.1 Java im Browser aktiviert? 1245
19.7.2 Läuft das Applet unter Netscape oder Microsoft Explorer? 1246
19.7.3 Datenaustausch zwischen Applets und Java-Skripten 1247
19.8 Musik in einem Applet und in Applikationen 1248
19.9 Webstart 1249
19.10 Java 2 Micro Edition 1250
19.10.1 Konfigurationen 1250
19.10.2 Profile 1251

20 Datenbankmanagement mit JDBC 1255

20.1 Das relationale Modell ... 1257
20.2 JDBC: der Zugriff auf Datenbanken über Java ... 1258
20.3 Die Rolle von SQL ... 1258
20.3.1 Ein Rundgang durch SQL-Anfragen ... 1259
20.3.2 Datenabfrage mit der Data Query Language (DQL) ... 1260
20.3.3 Tabellen anlegen mit der Data Definition Language (DDL) ... 1262
20.4 Datenbanktreiber für den Zugriff ... 1263
20.4.1 Treibertypen ... 1263
20.5 Datenbanken und ihre Treiber ... 1264
20.5.1 Derby ... 1264
20.5.2 MySQL ... 1266
20.5.3 Microsoft Access ... 1266
20.5.4 Ein Typ-4-Treiber für den Microsoft SQL Server 2000 ... 1268
20.5.5 Oracle10g ... 1268
20.5.6 Eclipse-Plugins zum Durchschauen von Datenbanken ... 1268
20.6 Eine Beispielabfrage ... 1271
20.7 Mit Java an eine Datenbank andocken ... 1273
20.7.1 Der Treibermanager ... 1273
20.7.2 Den Treiber laden ... 1274
20.7.3 Eine Aufzählung aller Treiber ... 1275
20.7.4 Log-Informationen ... 1276
20.7.5 Verbindung zur Datenbank ... 1277
20.8 Datenbankabfragen ... 1280
20.8.1 Abfragen über das Statement-Objekt ... 1280
20.8.2 Ergebnisse einer Abfrage in ResultSet ... 1280
20.8.3 Java und SQL-Datentypen ... 1281
20.8.4 Unicode in der Spalte korrekt auslesen ... 1285
20.8.5 wasNull() bei ResultSet ... 1285
20.8.6 Wie viele Zeilen hat ein ResultSet? ... 1285
20.9 Die Ausnahmen bei JDBC ... 1286
20.10 Transaktionen ... 1286
20.11 Elemente einer Datenbank hinzufügen und aktualisieren ... 1287
20.11.1 Batch-Updates ... 1288
20.12 Vorbereitete Anweisungen (Prepared Statements) ... 1289
20.12.1 PreparedStatement-Objekte vorbereiten ... 1289
20.12.2 Werte für die Platzhalter eines PreparedStatement ... 1290
20.13 Die LOBs (Large Objects) ... 1291
20.13.1 Einen BLOB besorgen ... 1291
20.14 Die SQL3-Datentypen ARRAY, STRUCT und REF ... 1292
20.15 Metadaten ... 1292
20.15.1 Metadaten über die Tabelle ... 1292
20.15.2 Informationen über die Datenbank ... 1296

20.16 DataSource 1297
20.16.1 Die Schnittstelle DataSource 1297

21 Reflection und Annotationen 1299

21.1 Metadaten 1301
21.1.1 XDoclet 1301
21.2 Mit dem Class-Objekt etwas über Klassen erfahren 1302
21.2.1 An ein Class-Objekt kommen 1302
21.2.2 Was das Class-Objekt beschreibt 1304
21.2.3 Der Name der Klasse 1306
21.2.4 Die Arbeit auf dem Feld 1308
21.2.5 instanceof mit Class-Objekten 1309
21.2.6 Oberklassen finden 1309
21.2.7 Implementierte Interfaces einer Klasse oder eines Inferfaces 1310
21.2.8 Modifizierer und die Klasse Modifier 1311
21.2.9 Die Attribute einer Klasse 1313
21.2.10 Methoden einer Klasse erfragen 1316
21.2.11 Konstruktoren einer Klasse 1319
21.2.12 Annotationen 1321
21.3 Objekte manipulieren 1321
21.3.1 Objekte erzeugen 1321
21.3.2 Die Belegung der Variablen erfragen 1323
21.3.3 Eine generische toString()-Funktion 1325
21.3.4 Variablen setzen 1326
21.3.5 Private Attribute ändern 1328
21.4 Methoden aufrufen 1328
21.4.1 Statische Methoden aufrufen 1330
21.4.2 Dynamische Methodenaufrufe bei festen Methoden beschleunigen 1331
21.5 Informationen und Identifizierung von Paketen 1332
21.5.1 Geladene Pakete 1333
21.6 Annotationen 1334
21.6.1 Neue Annotationen definieren 1335
21.6.2 Annotationen mit genau einem Element 1335
21.6.3 Beliebige Schlüssel-/Werte-Paare 1336
21.6.4 Vorbelegte Elemente 1339
21.6.5 Annotieren von Annotations-Typen 1340
21.6.6 Annotationen zur Laufzeit ausgelesen 1343
21.6.7 Mögliche Nachteile von Annotationen 1345

22 Komponenten durch Bohnen 1347

22.1 Grundlagen der Komponententechnik 1349
22.1.1 Brauchen wir Komponenten überhaupt? 1349
22.1.2 Visuelle und nichtvisuelle Komponenten 1350
22.1.3 Komponenten-Technologien von Microsoft 1350

22.2 Das JavaBeans Development Kit (BDK) ... 1351
22.2.1 Eine Beispielsitzung im BDK ... 1352
22.2.2 Verknüpfungen zwischen Komponenten ... 1353
22.2.3 Beans speichern ... 1354
22.3 Die kleinste Bohne der Welt ... 1354
22.4 Jar-Archive für Komponenten ... 1355
22.5 Worauf JavaBeans basieren ... 1356
22.6 Eigenschaften ... 1357
22.6.1 Einfache Eigenschaften ... 1358
22.6.2 Boolesche Eigenschaften ... 1358
22.6.3 Indizierte Eigenschaften ... 1359
22.7 Ereignisse ... 1360
22.7.1 Multicast und Unicast ... 1360
22.7.2 Namenskonvention ... 1361
22.8 Weitere Eigenschaften ... 1364
22.8.1 Gebundene Eigenschaften ... 1364
22.8.2 Anwendung von PropertyChange bei AWT-Komponenten ... 1366
22.8.3 Veto-Eigenschaften. Dagegen! ... 1366
22.9 Bean-Eigenschaften anpassen ... 1367
22.9.1 Customizer ... 1368
22.10 Property-Editoren ... 1369
22.11 BeanInfo ... 1369
22.12 Beliebte Fehler ... 1370

23 Java Management Extensions (JMX) 1371

23.1 Überwachen von Systemzuständen ... 1373
23.2 MBean-Typen, MBean-Server und weitere Begriffe ... 1374
23.2.1 MBeans des Systems ... 1374
23.3 JConsole ... 1376
23.4 Der MBeanServer ... 1377
23.5 Eine eigene Standard-MBean ... 1379

24 Java Native Interface (JNI) 1381

24.1 Java Native Interface und Invocation-API ... 1383
24.2 Einbinden einer C-Funktion in ein Java-Programm ... 1384
24.2.1 Schreiben des Java-Codes ... 1384
24.2.2 Compilieren des Java-Programms ... 1384
24.2.3 Erzeugen der Header-Datei ... 1385
24.2.4 Implementierung der Methode in C ... 1386

24.2.5 Übersetzen der C-Programme und Erzeugen der dynamischen Bibliothek ... 1387
24.2.6 Setzen der Umgebungsvariablen ... 1388
24.3 Nativ die Stringlänge ermitteln ... 1388
24.4 Erweiterte JNI-Eigenschaften ... 1389
24.4.1 Klassendefinitionen ... 1390
24.4.2 Zugriff auf Attribute ... 1390
24.5 Links und Weiteres ... 1392

25 Sicherheitskonzepte 1395

25.1 Der Sandkasten (Sandbox) ... 1397
25.2 Sicherheitsmanager (Security Manager) ... 1397
25.2.1 Der Sicherheitsmanager bei Applets ... 1399
25.2.2 Sicherheitsmanager aktivieren ... 1401
25.2.3 Wie nutzen die Java-Bibliotheken den Sicherheitsmanager? ... 1402
25.2.4 Rechte vergeben durch Policy-Dateien ... 1402
25.2.5 Erstellen von Rechte-Dateien mit dem grafischen Policy-Tool ... 1404
25.2.6 Kritik an den Policies ... 1405
25.3 Dienstprogramme zur Signierung ... 1406
25.3.1 Mit keytool Schlüssel erzeugen ... 1406
25.3.2 Signieren mit jarsigner ... 1407
25.4 Digitale Unterschriften ... 1407
25.4.1 Die MDx-Reihe ... 1408
25.4.2 Secure Hash Algorithm (SHA) ... 1409
25.4.3 Mit der Security-API einen Fingerabdruck berechnen ... 1409
25.4.4 Die Klasse MessageDigest ... 1409
25.4.5 Unix-Crypt ... 1412
25.5 Verschlüsseln von Daten(-strömen) ... 1412
25.5.1 Den Schlüssel bitte ... 1413
25.5.2 Verschlüsseln mit Cipher ... 1414
25.5.3 Verschlüsseln von Datenströmen ... 1414

26 Dienstprogramme für die Java-Umgebung 1417

26.1 Die Werkzeuge im Überblick ... 1419
26.2 Der Compiler javac ... 1419
26.2.1 Der Java-Interpreter java ... 1420
26.2.2 Der Unterschied zwischen java.exe und javaw.exe ... 1421
26.3 Das Archivformat Jar ... 1422
26.3.1 Das Dienstprogramm Jar benutzen ... 1423
26.3.2 Das Manifest ... 1424
26.3.3 Applikationen in Jar-Archiven starten ... 1425
26.3.4 Applets in Jar-Archiven ... 1426

26.4 Ant ... 1426
26.4.1 Bezug und Installation von Ant ... 1427
26.4.2 Properties ... 1428
26.4.3 Externe und vordefinierte Properties ... 1429
26.4.4 Weitere Leistungen ... 1430
26.5 Einpacken von Java-Programmen in ein exe mit JSmooth ... 1430
26.6 Decompiler ... 1430
26.6.1 Jad, ein schneller Decompiler ... 1431
26.6.2 Decompilieren erschweren ... 1434
26.7 Obfuscate Programm RetroGuard ... 1435
26.8 Sourcecode Beautifier ... 1435

A Die Begleit-CD 1437

Index 1439