

Inhaltsverzeichnis

1	Einführung in Java	17
	<i>von Jens Gallenbacher</i>	
1.1	Warum Java?	17
1.1.1	Programme im Web?	18
1.1.2	Plattformunabhängigkeit	19
1.1.3	Compiler versus Interpreter	20
1.1.4	Die Lösung des Dilemmas	21
1.2	Standardisierung von Java	23
1.2.1	Versionen	24
1.2.2	Neuerungen der Version 1.1 gegenüber 1.0	25
1.2.3	Unterschiede zwischen 1.1 und 1.2 (Plattform 2)	27
1.2.4	Neuerungen der Version 1.3	28
1.2.5	Neuerungen der Version 1.4	28
1.2.6	Welche Version eignet sich für das WWW?	29
1.3	Objektorientiertes programmieren	29
1.3.1	Die Grundbegriffe	30
1.4	Das erste Java-Programm mit dem JBuilder	38
1.4.1	Der Quelltext	39
1.4.2	Die erste Java-Webseite	41
1.4.3	Wie ein Applet funktioniert	44
1.4.4	Zusammenspiel mit dem Browser	46
1.4.5	Bibliotheken und Import	47
1.5	JBuilder und die IDE	48
1.5.1	Parameter und Variablen	59
1.5.2	Objektinstanziierungen, Variablen Deklarationen und Zuweisungen	61
1.5.3	Schleifen	68
1.5.4	Statische Methoden	70
1.6	Der allgemeine Aufbau eines Java-Applets	72
1.6.1	Eine Vorlage für Applets	72
1.6.2	Stop and go...	75
1.6.3	Interaktionen	76
1.6.4	Wir veranschaulichen die Funktionsweise des Applets	77
1.6.5	Die Anzeige aktualisieren	80
1.6.6	Ab Java 1.1	81
1.7	Arbeiten mit der Dokumentation	85
1.7.1	Ein Quiz	85
1.7.2	Welche Methode macht was?	86
1.7.3	Die Implementierung	88
1.7.4	java.util.Random	89
1.7.5	java.awt.Color	90
1.7.6	switch ... case	90



1.8	Nebenwirkungen...	95
1.8.1	Date	96
1.8.2	Der Thread	97
1.8.3	implements	99
1.8.4	Pausen	100
1.8.5	Exceptions	101
1.8.6	Die Müllsammlung	102
1.8.7	Eine Laufschrift	102
1.8.8	Ab Java 1.1	104
1.9	Grundlagen des Abstract Window Toolkit	105
1.9.1	Eingabelemente	105
1.9.2	Knöpfe und Labels	108
1.9.3	Der Layout Manager	109
1.9.4	Panel	110
1.9.5	...uuuuund action	111
1.9.6	Interaktionselemente und Layout Manager	111
1.9.7	ab Java 1.1	111
1.9.8	Java 2	114
1.10	Java meets HTML	115
1.10.1	AppletContext	115
1.10.2	Eine automatische HTML-Show	116
1.10.3	Magischer Knopf	119
1.11	Applet versus Applikation	122
1.11.1	Java-Applets: Mit Sicherheit?	123
1.11.2	Zugriff auf den lokalen Rechner	124
1.11.3	Netzverbindungen	124
1.11.4	Fenster	124
1.11.5	Sicherheitsloch Java ab 1.1?	124
1.11.6	Was bieten uns Applets also?	126
2	Parallele Programmierung mit Java-Threads	127
	<i>von Dr. Manfred Simon</i>	
2.1	Realisierung von Java-Threads	132
2.1.1	Ableiten von der Klasse »Thread«	132
2.1.2	Implementieren des Interfaces »Runnable«	134
2.1.3	Zentrale Methoden der Thread-Klasse beziehungsweise des Interfaces »Runnable«	135
2.2	Thread-Prioritäten und Scheduling	136
2.2.1	Synchronisation und atomare Bereiche	137
2.2.2	Implementierung mit Semaphoren und Monitoren	139
2.3	Gruppierung von Threads zu Thread-Gruppen	141
2.4	Gefahren der Thread-Programmierung	142
3	Collections	145
	<i>von Anja Austermann</i>	
3.1	Java-Implementationen der Mengenklassen	147
3.2	Arbeitersparnis durch Collections	151
3.3	Synchronisation von Collections	152
3.4	Das Interface »Iterator«	152
3.5	Suchen, Sortieren und andere Algorithmen	153



4	Java Virtual Machine Debug Interface	159
	<i>von Michael Spörl</i>	
4.1	Anforderung und Freigabe von Speicher	162
4.2	Funktionen für Klassen	163
4.2.1	Status einer Klasse	163
4.2.2	Dateiname und Signatur einer Klasse	164
4.2.3	Zugriffsflags einer Klasse	164
4.2.4	Ermittlung von Variablen und Methoden einer Klasse	164
4.2.5	Interfaces einer Klasse	165
4.2.6	Weitere Informationen über Klassen	165
4.3	Funktionen für Variablen	166
4.3.1	Name und Signatur einer Variablen	166
4.3.2	Zugriffsflags einer Variablen	166
4.3.3	Weitere Informationen über Variablen	166
4.4	Funktionen für Methoden	167
4.4.1	Name und Signatur einer Methode	167
4.4.2	Zugriffsflags einer Methode	167
4.4.3	Startadresse der Methode ermitteln	168
4.4.4	Lokale Variablen einer Methode	168
4.4.5	Größe der lokalen Variablen und Methodenargumente	169
4.4.6	Quelltextzeilen ermitteln	169
4.4.7	Bytecode der Methode ermitteln	170
4.4.8	Exceptions	170
4.4.9	Weitere Informationen über Methoden	170
4.5	Breakpoints	171
4.6	Überwachung von Variablen	171
4.6.1	Überwachung des Zugriffs auf Variablen	171
4.6.2	Überwachung der Modifikation von Variablen	172
4.7	Threads	172
4.7.1	Status eines Threads	172
4.7.2	Alle Threads ermitteln	173
4.7.3	Steuerung von Threads	173
4.7.4	Informationen über einen Thread	173
4.8	Events	174
4.8.1	Aktivieren und Deaktivieren von Events	175
4.8.2	Behandlung von Events	176
4.9	Weitere Funktionen	179
4.9.1	Ermittlung aller geladener Klassen	179
4.9.2	JVMDI-Versionsnummer ermitteln	180
4.9.3	Eigenschaften des JVMDI ermitteln	180
5	Das generelle Format von Klassendateien	181
	<i>von Michael Spörl</i>	
5.1	Allgemeines	181
5.1.1	Speicherung von ganzen Zahlen	181
5.1.2	Speicherung von Strings	182
5.1.3	Speicherung von Klassennamen	182
5.1.4	16-Bit-Indizes	183
5.1.5	Deskriptoren	183



5.2	Der Aufbau einer Klassendatei	184
5.3	Der Konstantenpool	187
5.3.1	Die CONSTANT_Utf8_Info-Struktur	188
5.3.2	Die CONSTANT_Integer_Info-Struktur	188
5.3.3	Die CONSTANT_Float_Info-Struktur	189
5.3.4	Die CONSTANT_Long_Info-Struktur	190
5.3.5	Die CONSTANT_Double_Info-Struktur	190
5.3.6	Die CONSTANT_Class_Info-Struktur	191
5.3.7	Die CONSTANT_String_Info-Struktur	191
5.3.8	Die CONSTANT_Fieldref_Info-Struktur	191
5.3.9	Die CONSTANT_Methodref_Info-Struktur	192
5.3.10	Die CONSTANT_InterfaceMethodref_Info-Struktur	192
5.3.11	Die CONSTANT_NameAndType_Info-Struktur	193
5.4	Variablen der Klasse	193
5.5	Methoden der Klasse	194
5.6	Attribute in einer Klassendatei	196
5.6.1	Das ConstantValue-Attribut	196
5.6.2	Das Code-Attribut	197
5.6.3	Das Exceptions-Attribut	198
5.6.4	Das InnerClasses-Attribut	199
5.6.5	Das Synthetic-Attribut	200
5.6.6	Das SourceFile-Attribut	200
5.6.7	Das LineNumberTable-Attribut	201
5.6.8	Das LocalVariableTable-Attribut	202
5.6.9	Das Deprecated-Attribut	203
5.7	Der Bytecode der JVM	203
5.8	Stackbefehle	204
5.8.1	Pushen von Werten	204
5.8.2	Laden und Speichern von lokalen Variablen	206
5.8.3	Weitere Stackbefehle	209
5.9	Befehle zur Typkonvertierung	212
5.10	Arithmetische und logische Befehle	215
5.10.1	Arithmetische Befehle	215
5.10.2	Logische Befehle	218
5.10.3	Der Befehl iinc	220
5.11	Vergleichsbefehle	221
5.12	Kontrollflußbefehle	222
5.12.1	Befehle für bedingte Sprünge nach Integer-Vergleich	223
5.12.2	Befehle für bedingte Sprünge nach Vergleich mit Null	224
5.12.3	Befehle für unbedingte Sprünge	225
5.12.4	Befehle für unbedingte Sprünge mit Tabellen	225
5.13	Befehle für Objekte	228
5.13.1	Erzeugen von Objekten	228
5.13.2	Zugriff auf Klassen- und Instanzvariablen	228
5.13.3	Befehle zur Typüberprüfung	230
5.14	Befehle für Arrays	231
5.14.1	Erzeugen von Arrays	231
5.14.2	Ermitteln der Länge eines Arrays	234
5.14.3	Zugriff auf Elemente eines Arrays	234



5.15	Befehle zum Aufruf von Methoden und deren Rückkehr	236
5.15.1	Aufruf von Methoden	236
5.15.2	Befehle für die Methodenrückkehr	239
5.15.3	Aufruf und Rückkehr von Unterprogrammen	240
5.16	Befehle für Exceptions	242
5.17	Befehle zur Synchronisation	242
5.18	Reservierte und nicht kategorisierte Befehle	243
5.19	Die Beispielapplikation	244
6	Java auf dem Web-Server	247
	<i>von Jens Gallenbacher</i>	
6.1	WWW-Formulare	247
6.2	CGI-Programme	250
6.2.1	Übertragung der Daten	250
6.2.2	Ein einfaches CGI-Skript	251
6.2.3	GET versus POST	254
6.2.4	Plattformunabhängigkeit?	255
6.3	Java als CGI-Sprache	255
6.3.1	Das Programm	255
6.3.2	Eine Umfrage	261
6.4	Einbindung von Datenbanken	268
6.5	Java-CGIs abbruchssicher machen	268
6.6	Java-spezifische Server-Programmierung	272
7	Servlets und Java Server Pages – Die Alternative zu CGI	273
	<i>von Anja Austermann</i>	
7.1	Vorteile von Servlets gegenüber CGI-Skripten	273
7.2	Wie werden Servlets programmiert?	274
7.2.1	Kommunikation leichtgemacht – der Java-Chat	281
7.2.2	Cookies – Informative Krümel	287
7.2.3	Sessions – Behalten Sie Ihre Kunden im Auge	293
7.3	Java Server Pages	300
7.3.1	Anpassen der Java Server Page	303
7.3.2	JSP-Aktionen	305
7.3.3	Starten der Java Server Page	305
8	Swing	307
	<i>von Christoph Lange</i>	
8.1	Features	308
8.2	Der erste Einblick	309
8.2.1	Packages	310
8.2.2	Basisklassen	311
8.3	Einführung mit Beispielen	312
8.3.1	Die erste Swing-Anwendung	312
8.3.2	Quelltext	317
8.3.3	Selbst ist der Mann	321
8.3.4	Mit oder ohne Designer?	323



8.3.5	Buntere Anwendungen	326
8.3.6	Tastaturbedienung und Actions	328
8.3.7	Menüs	331
8.3.8	Accessibility	336
8.3.9	plaf	338
8.4	Grundlagen grafischer Benutzeroberflächen	339
8.4.1	Container und Layout-Manager	339
8.4.2	Events	355
8.4.3	Top-Level-Container	376
8.5	Das Projekt: Ein Texteditor	383
8.5.1	Version 1: Menüs mit Action	383
8.5.2	Version 2: Eine Symbolleiste	388
8.5.3	Version 3: Desktops und Dialoge	391
8.5.4	Version 4: Dateien öffnen und speichern	416
8.5.5	Version 5: Editierbefehle	427
8.5.6	Version 6: Undo!	437
8.5.7	Fazit – Was hat es genützt?	441
8.6	Let's Swing again – weitere Komponenten	442
8.6.1	Buttons	442
8.6.2	Auswahl von Farben	444
8.6.3	Comboboxen	445
8.6.4	Fortschrittsanzeigen und Timer	446
8.6.5	Slider	448
8.6.6	Tabellen	449
8.6.7	Paßwort-Eingabe	452
8.6.8	JSpinner	453
8.6.9	Drag&Drop	456
8.6.10	Threadsicherheit	463
9	Komponenten entwickeln	465
	<i>von Christoph Lange</i>	
9.1	Grundlagen	465
9.1.1	Package-Organisation	465
9.1.2	Allgemeingültigkeit	466
9.1.3	Noch eine kleine Regel	467
9.1.4	Verschiedene Ansätze	467
9.2	Version 1: Ein Verzeichnisbaum	468
9.3	Version 2: Eine bessere Optik	472
9.4	Version 3: Stamm-/Root-Verzeichnisse anzeigen	476
9.5	Version 4: Nicht nur Ordnersymbole	478
9.6	Version 5: Fit für Wechsel (-medien)	481
9.7	Version 6: Fit für die Beanbox	492
9.8	Von der Komponente zur Bean	502
9.8.1	Schritt 1: Methodennamen anpassen	503
9.8.2	Schritt 2: Bound Properties hinzufügen	504
9.8.3	Schritt 3: Serialisierung ermöglichen	504
9.8.4	Schritt 4: Die Bean verpacken	505
9.8.5	Schritt 5: Das eigene Icon	507
9.8.6	Schritt 6: Die Bean integrieren	508



10	RMI – Verteilte Programmierung unter JAVA	513
	<i>von Anja Austermann</i>	
10.1	Programmierung der Server-Objekte	514
10.1.1	RMI-Programme ausführen	516
10.2	Programmierung eines Client-Objekts	518
10.3	Sicherheitseinstellungen mit Security Policies	519
10.3.1	3...2...1...start!	519
10.4	Aktivierbare Objekte	520
10.4.1	Eine »Implementations«-Klasse für aktivierbare Objekte	521
10.4.2	Programmierung einer »Setup«-Klasse	521
10.4.3	Sicherheit für aktivierbare Objekte	523
10.4.4	Starten	523
11	Internet-Programmierung mit dem JBuilder	525
	<i>von Anja Austermann</i>	
11.1	Netzwerkprogrammierung mit Java-Sockets	525
11.1.1	Das erste Netzwerkprogramm	527
11.1.2	Unser Client lernt »sprechen«	528
11.1.3	Unser erster Internet-Server...	531
11.1.4	Quasseln im Netz – der Chat-Server	535
11.1.5	Die Klassen URL und URLConnection	541
11.2	IP*Works – TCP/IP-Programmierung für Eilige	543
11.2.1	Ein FTP-Client mit IP*Works	544
11.2.2	Events	547
11.2.3	Mail, HTTP und weitere Dienste	549
11.2.4	Empfangen von Mails via POP3	550
11.2.5	Senden von Mails via SMTP	551
12	Sicherheit in Java	553
	<i>von Markus Wöhrensimmel</i>	
12.1	Die Java-Sandbox	553
12.2	Berechtigungen in Java	554
12.3	Dateien für die Sicherheitsrichtlinien	555
12.3.1	Verwendung des Policytools	556
12.3.2	Implementieren eigener Berechtigungsklassen	557
12.3.3	Beispiel	558
12.4	Programmierschnittstellen im Detail	561
12.4.1	java.lang.SecurityManager	561
12.4.2	java.security.Permission	561
12.4.3	java.io.FilePermission	562
12.4.4	java.security.BasicPermission	562
12.4.5	java.lang.RuntimePermission	563
12.5	Digitale Signaturen und Zertifikate	564
12.5.1	Grundlagen	564
12.5.2	Java und Kryptographie	565
12.5.3	JCA-Klassen	566
12.5.4	Verwendung von Zertifikaten unter Java	567
12.5.5	Signieren von Jar-Files	568



13	Datenbankprogrammierung mit Java	573
	<i>von Daniel Basler</i>	
13.1	Relationale Datenbanksysteme	575
13.2	Das Entity-Relationship-Modell	577
13.3	Besonderheiten einer Datenstruktur	578
13.3.1	Beziehungen	578
13.3.2	Indizes	579
13.3.3	Referenzinformation	579
13.4	Aufbau von JDBC	580
13.4.1	Zugriffsmodelle	580
13.4.2	Zugriff auf serverseitige Datenbanken	581
13.4.3	Treiber	582
13.5	Datenbanken und SQL	584
13.5.1	Datenmanipulation	585
13.5.2	Mathematische Funktionen	589
13.5.3	SQL in der Anwendung	591
13.5.4	Beispieldatenbank-Konfiguration	592
13.6	Der Datenbank-Pilot	603
13.7	Datenzugriffs-Komponenten	605
13.7.1	Datensteuerungs-Komponenten	605
13.8	Arbeiten mit DataExpress-Komponenten	606
13.8.1	Database	607
13.8.2	QueryDataSet einrichten	610
13.8.3	Tabellenausgabe	614
13.8.4	Datensätze sortieren	619
13.8.5	Arbeiten mit Parameterwerten	621
13.8.6	Arbeiten mit Stored Procedures	625
13.9	Visuelle Komponenten	631
13.9.1	Komponente jdbcTextField	631
13.9.2	Die Komponente jdbcNavField und jdbcNavComboBox	634
13.9.3	Berechnungen mit Datensteuerungs-Komponenten	636
13.9.4	Daten zur Laufzeit filtern	638
13.9.5	Arbeiten mit XML-Komponenten	640
13.10	Master-Detail-Abfragen	645
13.10.1	Beispiel	646
13.11	Datenmodule	648
13.12	Transaktionsverarbeitung	651
13.13	Arbeiten mit JDataStore	652
13.13.1	Erstellen eines JDataStore-Objekts	652
14	Web Services	657
	<i>von Daniel Basler</i>	
14.1	Was sind eigentlich Web Services?	658
14.2	SOAP	660
14.3	WSDL – Web Service Description Language	662
14.4	UDDI – Universal Description, Discovery, and Integration	662
14.4.1	Datentypen	663



14.5	Arbeiten mit dem Web Services Kit for Java	664
14.5.1	Entwickeln von SOAP-Servern	664
14.5.2	Erzeugen einer Java-Klasse	666
14.5.3	Der Web Services Explorer	671
14.5.4	Fazit	674
14.5.5	Quellen	674
15	Java Native Interface	675
	<i>von Michael Spörl</i>	
15.1	Erstellen von Java-Klassen mit nativen Methoden	675
15.1.1	Schreiben der Java-Klasse	676
15.1.2	Erzeugen der Header-Datei	677
15.1.3	Implementierung der nativen Methoden	678
15.2	Mapping zwischen Java und nativen Datentypen	679
15.2.1	Basisdatentypen	679
15.2.2	Java-Klassen	679
15.3	Zugriff auf Java-Objekte in nativen Methoden	680
15.3.1	Manipulation von Strings	680
15.3.2	Manipulation von Arrays	682
15.3.3	Aufruf von Java-Methoden	684
15.3.4	Zugriff auf Java-Variablen	689
15.3.5	Exceptions	689
15.3.6	Threads	691
A	Kurzreferenz der JDK-Tools	693
	<i>von Michael Spörl</i>	
A.1	appletviewer – Betrachter für Applets	693
A.1.1	Beschreibung	693
A.1.2	Aufruf	694
A.1.3	Optionen	694
A.2	extcheck – Verifizierung installierter Erweiterungen	694
A.2.1	Beschreibung	694
A.2.2	Aufruf	694
A.2.3	Optionen	695
A.3	idlj – Der IDL-Compiler	695
A.3.1	Beschreibung	695
A.3.2	Aufruf	695
A.3.3	Optionen	695
A.4	jar – Das Java-Archivtool	696
A.4.1	Beschreibung	696
A.4.2	Aufruf	697
A.4.3	Optionen	697
A.5	jarsigner – Signierer für JAR-Dateien	699
A.5.1	Beschreibung	699
A.5.2	Aufruf	699
A.5.3	Optionen	699



A.6	java – Der Java-Interpreter	700
A.6.1	Beschreibung	700
A.6.2	Aufruf	700
A.6.3	Optionen	700
A.7	javac – Der Java-Compiler	702
A.7.1	Beschreibung	702
A.7.2	Aufruf	703
A.7.3	Optionen	703
A.8	javadoc – Der Dokumentationsgenerator	704
A.8.1	Beschreibung	704
A.8.2	Aufruf	706
A.8.3	Optionen	706
A.9	javah – Generator für C-Header- und Stub-Dateien	709
A.9.1	Beschreibung	709
A.9.2	Aufruf	710
A.9.3	Optionen	710
A.10	javap – Der Java-Disassembler	711
A.10.1	Beschreibung	711
A.10.2	Aufruf	711
A.10.3	Optionen	711
A.11	jdb – Der Java-Debugger	712
A.11.1	Beschreibung	712
A.11.2	Aufruf	714
A.11.3	Optionen	715
A.12	keytool – Verwalten von Schlüsseln und Zertifikaten	715
A.12.1	Beschreibung	715
A.12.2	Aufruf	715
A.12.3	Befehle	715
A.12.4	Optionen	717
A.13	native2ascii – Konvertierung von Quelldateien	718
A.13.1	Beschreibung	718
A.13.2	Aufruf	718
A.13.3	Optionen	718
A.14	policytool – Das Policy-Tool	718
A.14.1	Beschreibung	718
A.15	rmic – Der RMI-Compiler	719
A.15.1	Beschreibung	719
A.15.2	Aufruf	719
A.15.3	Optionen	719
A.16	rmid – Der RMI-Dämon	721
A.16.1	Beschreibung	721
A.16.2	Aufruf	721
A.16.3	Optionen	721
A.17	rmiregistry – Registry für Remote-Objekte	721
A.17.1	Beschreibung	721
A.17.2	Aufruf	722
A.17.3	Optionen	722



A.18	serialver – Ermittlung der »serialVersionUID«	722
A.18.1	Beschreibung	722
A.18.2	Aufruf	722
A.18.3	Optionen	722
A.19	tnameserv – Der Transient Name Server	723
A.19.1	Beschreibung	723
A.19.2	Aufruf	723
A.19.3	Optionen	723
Stichwortverzeichnis		725

Auf der Buch-CD befindet sich eine Trial-Version des JBuilder 7 für Windows für eine Nutzungsdauer von 30 Tagen.

Leser aus Deutschland können die Trial durch einen Anruf bei 0800 182 7043 (kostenfrei) um weitere 30 Tage verlängern.

Leser aus Österreich und der Schweiz verlängern den Lizenzschlüssel unter 00800 50 55 7777.

Ihr C&L Verlag

