

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>15</b>
<b>Kapitel 1 Ein- und Ausgabe (I/O)</b>	<b>21</b>
1.1 Standardein- und -ausgabe .....	23
1.1.1 Vordefinierte I/O-Objekte .....	23
1.1.2 Redirection .....	26
1.1.3 Standard-Fehlerausgabe .....	32
1.2 Byteströme .....	33
1.2.1 Abstrakte Basisklassen .....	33
1.2.2 Lesen und Schreiben .....	34
1.2.3 Pufferung .....	37
1.2.4 Schließen von Streams .....	38
1.2.5 ARM .....	40
1.3 File-I/O .....	43
1.3.1 Konkrete Datenquellen und -senken .....	43
1.3.2 Filestreams .....	44
1.3.3 Blockweise Übertragung .....	48
1.3.4 Abstrakte und konkrete Methoden .....	52
1.4 Transformationen mit Filterklassen .....	53
1.4.1 Ableitung und Komposition .....	54
1.4.2 Abstrakte Filterklassen .....	59
1.4.3 Konkrete Filterklassen .....	62
1.4.4 Filter der Laufzeitbibliothek .....	64
1.4.5 Zip-Dateien .....	69
1.5 Decorator-Pattern .....	72
1.6 Umgang mit Textdateien .....	74
1.6.1 Interne und externe Darstellung .....	74
1.6.2 Basisklassen Reader und Writer .....	76

1.6.3	Konkrete Quellen und Senken .....	77
1.6.4	Textfilter .....	79
1.6.5	Ersetzen der Standardein- und -ausgabe-Objekte .....	82
1.6.6	Zeichensätze und Encodings .....	83
1.7	Definition neuer I/O-Klassen .....	88
1.8	Datei-Operationen und Directories .....	93
1.8.1	Datei-Objekte .....	93
1.8.2	Eigenschaften von Dateien .....	94
1.8.3	Zugriffsrechte .....	97
1.8.4	Pfade .....	99
1.8.5	Directories .....	102
1.8.6	Schreiben und Lesen .....	105
1.8.7	Filesystem-Operationen .....	107
<b>Kapitel 2</b>	<b>Serialisierung</b>	<b>119</b>
2.1	Fragen .....	121
2.2	Object-Streams .....	122
2.2.1	Methoden readObject und writeObject .....	122
2.2.2	Interface Serializable .....	124
2.2.3	Objektgraphen .....	125
2.2.4	Stream-Struktur .....	126
2.2.5	Aufruf von Konstruktoren .....	127
2.2.6	Cloning .....	130
2.2.7	Caching .....	132
2.2.8	Nicht serialisierte Variablen .....	133
2.2.9	Serialisierungsversionen .....	136
2.2.10	Eingriff in die Serialisierung .....	137
2.3	JavaBeans .....	139
2.3.1	Properties .....	140
2.3.2	XMLEncoder und XMLDecoder .....	142
2.3.3	Speicherformat XML .....	144

2.3.4	Inkonsistenzen .....	145
2.3.5	Objektgraphen .....	146
2.3.6	Serialisierer-Objekte .....	146
2.3.7	Neue Serialisierer .....	148
2.3.8	Deserialisierung .....	150
2.4	XStream.....	151
2.4.1	toXML und fromXML .....	152
2.4.2	Referenzen .....	153
2.4.3	Converter .....	155
2.4.4	Neue Converter, statische und transiente Variablen .....	155
2.4.5	Encoding .....	159
2.4.6	Kompatibilität mit Object-Streams .....	160
<b>Kapitel 3</b>	<b>XML</b>	<b>167</b>
3.1	Struktur, Grammatik und Validierung.....	168
3.1.1	Aufbau eines XML-Dokuments .....	168
3.1.2	Validierung .....	175
3.2	Arbeit mit dem DOM.....	185
3.2.1	DOM-Parser .....	185
3.2.2	Validierung .....	186
3.2.3	Objektbaum .....	189
3.2.4	Arbeiten mit dem DOM.....	194
3.2.5	Ausgabe .....	196
3.2.6	Modifikation.....	197
3.2.7	Fehlerbehandlung .....	200
3.3	SAX-Parser.....	201
3.3.1	Initialisierung .....	201
3.3.2	Handler.....	202
3.3.3	Validierung .....	204
3.3.4	Umgang mit Text .....	207
3.3.5	Fehlerbehandlung .....	211

<b>Kapitel 4</b>	<b>Rekursion</b>	<b>219</b>
4.1	Arbeitsweise .....	221
4.1.1	Selbstaufruf und Abbruchkriterium .....	221
4.1.2	Parameter und Ergebnis .....	224
4.1.3	Indirekte Rekursion .....	227
4.1.4	Laufzeitstack .....	228
4.1.5	Rekursive Datenstrukturen .....	230
4.2	Rekursion und Iteration .....	230
4.2.1	Endrekursion .....	231
4.2.2	Eliminierung der Endrekursion .....	234
4.3	Klassifizierung .....	234
4.3.1	Lineare Rekursion .....	235
4.3.2	Verzweigte Rekursion.....	237
4.3.3	Geschachtelte Rekursion .....	240
4.4	Anwendungen .....	242
4.4.1	Filesystem .....	242
4.4.2	Parser.....	247
4.4.3	Permutationen .....	250
4.5	Memoizing .....	257
4.5.1	Redundante Methodenaufrufe .....	257
4.5.2	Bedingungen für Memoizing .....	264
4.5.3	Implementierung von Caches .....	264
<b>Kapitel 5</b>	<b>Geschachtelte Klassen</b>	<b>281</b>
5.1	Statisch geschachtelte Klassen .....	283
5.1.1	Definition und Syntax .....	283
5.1.2	Bedeutung von private.....	283
5.1.3	Anwendungsbeispiel: Factory-Methoden .....	285
5.1.4	Bytecode-Dateien .....	288
5.1.5	Statisch geschachtelte Interfaces.....	289

5.2	Innere Klassen .....	289
5.2.1	Anwendungsbeispiel: Iterierbare Strings .....	291
5.3	Lokale Klassen .....	293
5.4	Anonyme Klassen .....	296
5.4.1	Anwendungsbeispiele .....	297
5.5	Lambda-Ausdrücke (Java 8) .....	301
5.5.1	Funktionsinterfaces .....	301
5.5.2	Lambda-Ausdrücke .....	302
5.5.3	Lambda-Ausdrücke im Typsystem .....	306
5.5.4	Closure .....	308
5.5.5	Gültigkeitsbereiche .....	309
5.5.6	Exceptions in einem Lambda-Ausdruck .....	311
5.5.7	Geschachtelte Lambda-Ausdrücke .....	311
5.5.8	Funktionsinterfaces in der Bibliothek .....	314
5.6	Default-Methoden (Java 8) .....	314
5.6.1	Definition von Default-Methoden .....	316
5.6.2	Implementierung von Default-Methoden .....	316
5.6.3	Einschränkungen von Default-Methoden .....	318
5.6.4	Vererbung und Default-Methoden .....	318
5.6.5	Evolution von Interfaces .....	322
5.6.6	ABCs und Default-Methoden .....	323
<b>Kapitel 6</b>	<b>Nebenläufigkeit</b>	<b>331</b>
6.1	Paralleler Programmablauf .....	333
6.1.1	Basisklasse Thread .....	333
6.1.2	Start eines neuen Threads .....	335
6.1.3	Eigenschaften von Threads .....	339
6.1.4	Ende eines Threads .....	342
6.1.5	Interface Runnable .....	344
6.1.6	Threads am Programmende .....	346

6.2	Kommunikation mit Interrupts .....	348
6.2.1	sleep und InterruptedException .....	352
6.3	Scheduling und Multiprozessoren .....	354
6.3.1	Lebenslauf eines Threads .....	354
6.3.2	Scheduler .....	356
6.3.3	Expliziter Zustandswechsel .....	357
6.3.4	Threads und Prozessoren .....	360
6.3.5	Hintergrund-Threads .....	363
6.3.6	Threads und Prozesse .....	365
6.4	Konkurrierender Zugriff und Synchronisation .....	365
6.4.1	Thread-lokale und geteilte Daten .....	365
6.4.2	Gleichzeitiger Zugriff .....	367
6.4.3	Zusammengesetzte Operationen .....	369
6.4.4	Synchronisation auf einem Monoprozessor .....	373
6.4.5	Atomare Operationen .....	373
6.4.6	Länge von synchronized-Blöcken .....	374
6.4.7	synchronized-Methoden .....	377
6.5	volatile und Deadlocks .....	379
6.5.1	Anwendungsbeispiel Parkhaus .....	379
6.5.2	Optimierung von Variablenzugriffen .....	382
6.5.3	Modifier volatile .....	384
6.5.4	Deadlocks .....	385
6.5.5	Unveränderliche Klassen .....	391
6.6	Bedingtes Warten .....	391
6.6.1	Aktives Warten .....	391
6.6.2	wait und notify .....	392
6.6.3	Test der wait-Bedingung .....	394
6.6.4	notify und notifyAll .....	396

<b>Kapitel 7</b>	<b>Netzwerkprogrammierung</b>	<b>407</b>
7.1	Grundlagen .....	408
7.1.1	ISO/OSI-Schichtenmodell .....	408
7.1.2	Protokollstapel .....	410
7.1.3	MAC-Adressen (Ethernet) .....	410
7.1.4	Hostadressen (IP) .....	412
7.1.5	Loopback-Device .....	415
7.1.6	Symbolische Hostnamen .....	416
7.1.7	Domainname-Server (DNS) .....	418
7.1.8	Ports .....	418
7.1.9	Dienste und Protokolle .....	420
7.1.10	Client-Server-Systeme .....	421
7.1.11	Peer-to-Peer-Netzwerke .....	422
7.2	Clients und Server .....	423
7.2.1	Sockets .....	423
7.2.2	Portscanner .....	426
7.2.3	Lesen und Schreiben .....	428
7.2.4	Beenden einer Verbindung .....	430
7.2.5	Pufferung .....	431
7.2.6	Beispiel: Echo-Client .....	431
7.2.7	Telnet-Client .....	432
7.2.8	Serversockets .....	434
7.2.9	Beispiele: Time- und Echo-Server .....	436
7.2.10	Verbindung auf einzelnen Netzwerkschnittstellen .....	437
7.2.11	Teilweiser Abbau .....	439
7.2.12	Verschlüsselte Verbindung .....	441
7.3	HTML, HTTP und Webserver .....	446
7.3.1	Idee .....	446
7.3.2	Webserver und Browser .....	446
7.3.3	URLs .....	447
7.3.4	HTML .....	448

7.3.5	DOM .....	450
7.3.6	Aufbau eines Dokuments .....	451
7.3.7	HTTP-Request .....	452
7.3.8	HTTP-Response .....	454
7.3.9	HTTP-Zugriff mit Bibliotheksklassen .....	457
7.3.10	Minimaler Webserver .....	459
7.3.11	Statischer Webserver .....	460
7.3.12	Dynamische Webseiten .....	463
7.3.13	Nebenläufiger Webserver .....	465
7.3.14	Gefährlicher dynamischer Server .....	467
7.3.15	Webserver-Basisklasse .....	468
<b>Kapitel 8</b>	<b>Reflection</b>	<b>479</b>
8.1	Factory-Methoden .....	481
8.1.1	Typobjekte und die Methode <code>forName</code> .....	481
8.1.2	Erzeugen von Objekten mit <code>newInstance</code> .....	483
8.1.3	Beispielanwendung: Pizzas .....	484
8.1.4	Konstruktor-Objekte .....	486
8.1.5	Beispielanwendung: Pizzas, weitere Verbesserung .....	487
8.2	Analyse der Codestruktur .....	488
8.2.1	Modifier .....	490
8.2.2	Konstruktoren .....	491
8.2.3	Variablen .....	492
8.2.4	Methoden .....	492
8.2.5	Anwendung: UML-artige Skizze einer Klassendefinition ...	493
8.3	Analyse und Modifikation von Objekten .....	495
8.3.1	Zugriff auf Variablen .....	496
8.3.2	Umgang mit Arrays .....	498
8.3.3	Verändern von Objekten .....	499
8.3.4	Anwendung: Reflektive <code>toString</code> -Methode .....	500
8.3.5	Reflektive Methodenaufrufe .....	501
8.3.6	Erzeugen neuer Objekte .....	503



<b>Kapitel 9</b>	<b>Annotationen</b>	<b>513</b>
9.1	Idee .....	514
9.2	Vordefinierte Annotationen .....	517
9.2.1	Warnungen des Java-Compilers .....	520
9.2.2	Annotation SafeVarargs .....	523
9.3	Neue Annotationen .....	526
9.3.1	Definition .....	526
9.3.2	Annotation-Werte .....	527
9.3.3	Meta-Annotation Target .....	528
9.3.4	Meta-Annotation Retention .....	529
9.3.5	Meta-Annotation Inherited und Documented .....	531
9.3.6	Marker-Annotationen .....	532
9.4	Auswertung zur Laufzeit .....	532
9.4.1	Anwendung: Reflektive toString-Methode .....	535
9.5	Prozessoren .....	540
9.5.1	Arbeitsweise .....	540
9.5.2	Diagnoseausgabe des Compilers .....	543
9.5.3	Anwendung: Generierte Bean-Klassen .....	543
9.6	Packages .....	548
<b>Anhang A</b>	<b>Voraussetzungen</b>	<b>553</b>
<b>Anhang B</b>	<b>I/O-Pipelines</b>	<b>559</b>
<b>Anhang C</b>	<b>Beispiel Decorator-Pattern</b>	<b>565</b>
<b>Anhang D</b>	<b>Java-8-Entwicklerversion</b>	<b>571</b>
D.1	Lambda-Ausdrücke .....	571
D.2	Default-Methoden .....	573
	<b>Literaturverzeichnis</b>	<b>575</b>
	<b>Index</b>	<b>577</b>