

→ Thomas Stark

# J2EE

→ Einstieg für Anspruchsvolle

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>13</b>
Danksagung . . . . .	13
<b>Kapitel 1 Einleitung</b>	<b>15</b>
1.1 Das Client/Server-Modell . . . . .	15
1.1.1 Application Programming Interface und Service Provider . . . . .	16
1.2 Aufbau dieses Buches. . . . .	17
1.2.1 Darstellung. . . . .	17
1.2.2 Business-Logik . . . . .	18
1.2.3 Datenhaltung . . . . .	20
1.2.4 Plattformübergreifende Kommunikation. . . . .	21
1.3 Ziele dieses Buches. . . . .	21
1.4 An wen richtet sich dieses Buch? . . . . .	22
1.4.1 Welche Voraussetzungen müssen Sie mitbringen? . . . . .	23
1.4.2 Disclaimer. . . . .	24
1.5 Bezug und Installation der Basis-Software. . . . .	24
1.5.1 Java 2 Plattform Enterprise Edition (J2EE) . . . . .	24
1.5.2 Die Wahl der „richtigen“ Entwicklungsumgebung (IDE) . . . . .	24
1.5.3 Ant – Jeder Aufgabe ihre Ameise . . . . .	25
1.5.4 Apaches Tomcat – Ein Kater erobert das Netz . . . . .	25
1.5.5 JBoss – Einer für alles . . . . .	25
<b>Kapitel 2 Java Server Pages</b>	<b>27</b>
2.1 Web-Applikationen gestern und heute. . . . .	27
2.2 Einrichten unserer Web-Applikation . . . . .	32
2.3 Statisches HTML. . . . .	34
2.4 Die erste dynamische HTML-Seite . . . . .	35
2.5 Die Bausteine einer JSP . . . . .	36
2.5.1 JSP-Ausdrücke . . . . .	36
2.5.2 JSP-Scriptlets . . . . .	39
2.5.3 JSP-Deklarationen . . . . .	43
2.5.4 Kommentare . . . . .	45
2.6 Die vordefinierten Variablen einer JSP . . . . .	46
2.6.1 Die verschiedenen Kontexte des Webservers. . . . .	46
2.6.2 Acht Helfer zur Kontrolle unserer JSPs. . . . .	48

2.7	Direktiven – Eigenschaften einer JSP . . . . .	50
2.7.1	page – die Seitendirektive . . . . .	51
2.7.2	Ein Beispiel für die Verwendung von page-Direktiven . . . . .	58
2.7.3	Die include-Direktive . . . . .	66
2.8	JSPs und JavaBeans . . . . .	67
2.8.1	Grundlagen von JavaBeans . . . . .	68
2.8.2	Verwendung von JavaBeans . . . . .	69
2.8.3	JavaBeans gemeinsam nutzen . . . . .	72
2.8.4	JavaBeans erzeugen . . . . .	73
2.8.5	Vorteile von JavaBeans gegenüber Variablen . . . . .	74

## **Kapitel 3 Servlets** **75**

3.1	Von JSPs zu Servlets . . . . .	75
3.1.1	Die Vergangenheit . . . . .	78
3.2	Ein einfaches Servlet . . . . .	78
3.2.1	Servlets verwenden . . . . .	80
3.2.2	Der Lebenszyklus eines Servlets . . . . .	87
3.2.3	Instanzen und Threads – Parallele Anfragen . . . . .	92
3.2.4	Logging . . . . .	93
3.3	Auf den Client hören: HTTP-Request-Header . . . . .	94
3.3.1	Die Hauptzeile lesen . . . . .	95
3.3.2	Request-Header lesen . . . . .	95
3.3.3	Ein Beispiel . . . . .	98
3.4	Mit dem Client sprechen: HTTP-Response-Header . . . . .	101
3.4.1	Die Statuszeile erzeugen . . . . .	101
3.4.2	Response-Header setzen . . . . .	102
3.5	Was Sie mit JSPs nicht machen können . . . . .	104
3.6	Ein einfacher Chatroom . . . . .	107
3.6.1	Die UserBean . . . . .	107
3.6.2	Die UserRegistry . . . . .	108
3.6.3	Der Gesprächsspeicher . . . . .	110
3.6.4	Die Anmeldung am Chat . . . . .	111
3.6.5	Das Chat-Servlet . . . . .	113
3.6.6	Die Darstellung des Chats . . . . .	121
3.6.7	Fazit . . . . .	122
3.7	Filtern von Formulardaten . . . . .	123
3.8	Zusammenfassung . . . . .	125

## **Kapitel 4 Tag-Bibliotheken** **127**

4.1	Ein eigenes Tag . . . . .	127
4.1.1	Erstellen der Java-Klasse . . . . .	128
4.1.2	Der Tag Library Descriptor . . . . .	130
4.1.3	Verwendung in JSPs . . . . .	132

4.2	Attribute verwenden	133
4.3	Vordefinierte Variablen	135
4.4	Den Rumpf manipulieren	137
4.5	Mit dem Kontext arbeiten	140
4.5.1	Entscheidungen (if-else)	140
4.6	Zusammenfassung	146
<b>Kapitel 5 Struts</b>		<b>147</b>
5.1	Frameworks	147
5.2	Struts	149
5.2.1	Struts-Komponenten	150
5.2.2	Was wir zusätzlich benötigen	152
5.3	Ein Adressbuch	152
5.3.1	Installation	152
5.3.2	Unsere Business-Objekte	156
5.3.3	JSPs	157
5.3.4	Taglibs	161
5.3.5	Resource	161
5.3.6	ActionForms	163
5.3.7	Actions	166
5.3.8	Konfiguration der Anwendung	169
5.3.9	Fazit	172
5.4	Erweiterungen	172
5.4.1	Die Datenbank-Action	173
<b>Kapitel 6 Java Naming and Directory Interface</b>		<b>177</b>
6.1	Was ist JNDI?	179
6.1.1	Naming Services	181
6.1.2	Directory Services	183
6.2	API und SPI	185
6.2.1	Download der API	185
6.2.2	Einige SPI-Implementierungen	186
6.3	Arbeiten mit der JNDI	187
6.3.1	Erzeugen eines Kontextes	188
6.3.2	Mit dem Kontext arbeiten	190
6.3.3	Datenquellen im Apache Tomcat	194
<b>Kapitel 7 Enterprise JavaBeans</b>		<b>199</b>
7.1	Vom Einschicht- zum Dreischichtsystem	199
7.2	Einsatz eines Application Servers	200
7.2.1	Aufbau eines Application Servers	201
7.2.2	Portabilität	202
7.2.3	Unterstützte APIs	203
7.2.4	Aufgaben eines EJB-Containers	203

7.3	Kurzer Überblick über EJB-Technologien . . . . .	204
7.3.1	Beans, Beans, Beans. . . . .	205
7.3.2	Methodenfernaufrufe . . . . .	206
7.3.3	Objektserialisierung . . . . .	208
7.4	Die verschiedenen Bean-Typen . . . . .	208
7.4.1	Bestandteile einer Enterprise JavaBean . . . . .	209
7.4.2	Entity-Beans . . . . .	210
7.4.3	Stateless Session-Beans . . . . .	210
7.4.4	Stateful Session-Beans. . . . .	211
7.4.5	Message Driven Beans . . . . .	212
7.4.6	Remote-Client-View vs. Local-Client-View . . . . .	212
7.5	Persistenz-Typen (BMP vs. CMP). . . . .	213
7.6	Bean Managed Persistence (BMP) am Beispiel. . . . .	214
7.6.1	Voraussetzungen . . . . .	214
7.6.2	Das Remote-Interface. . . . .	216
7.6.3	Eine Ausnahme bei Kontoüberziehung . . . . .	216
7.6.4	Das Home-Interface. . . . .	217
7.6.5	Die Implementierung . . . . .	217
7.6.6	Der Deployment Descriptor . . . . .	228
7.7	Container Managed Persistence (CMP) am Beispiel. . . . .	231
7.7.1	Der Deployment Descriptor . . . . .	235
7.7.2	Der Persistence Descriptor . . . . .	237
7.8	Relationen. . . . .	238
7.8.1	Beziehungen zwischen Entity-Beans . . . . .	239
7.8.2	Kardinalität der Beziehungen . . . . .	240
7.8.3	Implementierung von Relationen. . . . .	241
7.8.4	Die Konfiguration im Deployment Descriptor . . . . .	242
7.8.5	Die Konfiguration im Persistence Descriptor . . . . .	244
7.8.6	EJB Query Language (EJB-QL). . . . .	245

## Kapitel 8 Persistenz (JDO, Hibernate und Co.)

249

8.1	Java Data Objects (JDO) . . . . .	250
8.1.1	Die Technik dahinter . . . . .	251
8.1.2	Konfiguration . . . . .	252
8.1.3	Verwendung der Referenzimplementierung . . . . .	252
8.2	Ein Beispiel für JDO . . . . .	253
8.2.1	Konfiguration der Eigenschaften . . . . .	254
8.2.2	Verwendung eines Primary Keys . . . . .	256
8.2.3	Herunterladen, übersetzen und anreichern. . . . .	257
8.2.4	Ausprobieren . . . . .	258
8.3	JDO Query Language (JDOQL). . . . .	267
8.3.1	Basics . . . . .	268
8.3.2	Definition von Bedingungen . . . . .	268
8.3.3	Sortieren der Ergebnismenge. . . . .	269
8.3.4	Tausend weitere Möglichkeiten . . . . .	270

8.4	Zusammenfassung	270
8.5	Hibernate	271
8.5.1	Unterschiede zum JDO-Konzept	272
8.5.2	Das Datenbank-Mapping	272
8.5.3	Unsere Adressverwaltung via Hibernate	273
8.5.4	Ausblick	276
8.6	Zusammenfassung	277
<b>Kapitel 9 Java Message Service</b>		<b>281</b>
9.1	Das Konzept	284
9.1.1	Partner im System	284
9.1.2	Messaging-Konzepte	285
9.1.3	Konfiguration	286
9.2	Bestandteile der API	286
9.2.1	Zusammenfassung API	291
9.3	Senden einer Nachricht	292
9.3.1	JNDI-Lookup-Klasse	292
9.3.2	Der schematische Aufbau	292
9.3.3	Die Implementierung	294
9.4	Empfangen einer Nachricht	295
9.4.1	Pull – manuelles Abholen	296
9.4.2	Push – Wenn der Postmann zweimal klingelt	297
9.4.3	Topic vs. Queue	299
9.5	Nachrichten	300
9.5.1	Reihenfolge der Nachrichten	301
9.5.2	Auslieferungsmodus	301
9.5.3	Das Verfallsdatum unserer Nachricht	302
9.5.4	Korrelation und Identifikation	303
9.6	Filtern einer Nachricht	303
9.6.1	Setzen und Auslesen von Eigenschaften	303
9.6.2	Ausgabe von Attribut-Properties	304
9.6.3	Filtern anhand von Attributen	305
9.6.4	Filterelemente	307
9.7	Transaktionen und Acknowledgement	308
9.7.1	Transaktionen beim Senden	309
9.7.2	Transaktion beim Nachrichtenempfang	309
9.7.3	Empfangsbestätigungen	310
9.8	Hin und zurück	311
9.8.1	Der QueueRequestor	312
9.8.2	Ein Empfänger mit Antwort	313
9.8.3	Request-Reply vs. klassischem Remote Procedure Call	315
9.9	Ein Blick über den Tellerrand	315
9.9.1	Message Driven Beans	315
9.10	Zusammenfassung	321

<b>Kapitel 10 eXtensible Markup Language</b>	<b>323</b>
10.1 Einleitung	323
10.1.1 Geschichte	324
10.1.2 Auftritt für die eXtensible Markup Language	325
10.1.3 Einige XML-basierende Standards	329
10.1.4 Vorteile von XML	331
10.2 XML-Elemente	332
10.2.1 Prolog und XML-Anweisungen	332
10.2.2 Tags	333
10.2.3 Zeichenketten	335
10.2.4 Kommentare	336
10.2.5 Namensräume	337
10.2.6 CData-Rohdaten	338
10.2.7 Syntaxüberprüfung per Document Type Definition (DTD)	338
10.2.8 Syntax-Schema-Definitionen	344
10.2.9 Zusammenfassung	349
10.3 Mit XML-Dokumenten arbeiten	350
10.3.1 Das Document Object Model des W3C	350
10.3.2 Das Java Document Object Model (JDOM)	351
10.3.3 Erzeugen eines XML-Dokuments von Grund auf	352
10.3.4 Einlesen eines XML-Dokuments	357
10.3.5 Verarbeiten eines XML-Dokuments	358
10.3.6 Erzeugen eines XML-Dokuments	362
10.4 Mit XML-Events arbeiten	364
10.4.1 Callback-Methoden	365
10.4.2 XML lesen mit SAX	366
10.4.3 XML verarbeiten mit SAX	370
10.4.4 Zusätzliche Handler	377
10.4.5 Einer für alles	378
10.4.6 Zusammenfassung SAX	379
10.5 Zusammenfassung	380
<b>Kapitel 11 XPath, XSL &amp; Co.</b>	<b>381</b>
11.1 XPath	381
11.1.1 Referenzieren von Elementen	382
11.1.2 Erweiterte Pfade	385
11.1.3 Operationen und Relationen	386
11.1.4 XPath-Funktionen	387
11.1.5 Zusammenfassung XPath	390
11.2 eXtensible Stylesheet Language	391
11.2.1 Ein einfaches Template	392
11.2.2 eXtensible Stylesheet Language Transformation (XSLT)	394
11.2.3 Aufbau eines Stylesheets	401

11.2.4	Ein Beispieldokument . . . . .	403
11.2.5	Eine alternative Implementierung . . . . .	408
11.2.6	Erweiterungsfunktionen für XSLT . . . . .	410
11.2.7	Zusammenfassung XSL . . . . .	410
11.3	XSL Formatting Objects . . . . .	411
11.3.1	Geschichte . . . . .	411
11.3.2	Grundlagen und Installation. . . . .	412
11.3.3	Aufbau eines Dokuments . . . . .	413
11.4	Von XML zum PDF – ein Beispiel . . . . .	418
11.4.1	Das Stylesheet . . . . .	418
11.4.2	Das Rendering . . . . .	422
11.4.3	Das Ergebnis. . . . .	423
11.5	Zusammenfassung . . . . .	424
<b>Kapitel 12</b>	<b>Web Services</b>	<b>425</b>
12.1	XML-Remote Procedure Calls (XML-RPC) . . . . .	425
12.1.1	Das Protokoll . . . . .	426
12.1.2	Aufbau eines XML-RPC Dokuments . . . . .	427
12.1.3	Übergabe der Parameter . . . . .	427
12.1.4	Fehlerbehandlung . . . . .	428
12.1.5	Zusammenfassung XML-RPC . . . . .	429
12.2	Simple Object Access Protocol (SOAP) . . . . .	430
12.2.1	Die Idee hinter SOAP . . . . .	430
12.2.2	SOAP und Java. . . . .	430
12.2.3	Erstellen einer SOAP-Message . . . . .	431
12.2.4	Verschicken unserer SOAP-Nachricht. . . . .	434
12.2.5	Empfangen einer SOAP-Nachricht . . . . .	435
12.3	JAX-Remote Procedure Call (JAX-RPC) . . . . .	436
12.3.1	Das Interface . . . . .	437
12.3.2	Die Implementierung . . . . .	437
12.3.3	Die Konfiguration. . . . .	437
12.3.4	Der Client . . . . .	439
12.4	... und darüber hinaus . . . . .	439
12.4.1	Web Service Description Language (WSDL) . . . . .	439
12.4.2	Universal Description and Discovery Interface (UDDI) . . . . .	440
12.4.3	eXtensible Interaction System (Axis) . . . . .	440
12.5	Zusammenfassung . . . . .	441
<b>Kapitel A</b>	<b>Weiterführende Quellen und Literatur</b>	<b>443</b>
<b>Kapitel B</b>	<b>Apache Software License</b>	<b>445</b>
<b>Kapitel C</b>	<b>J2EE Software License</b>	<b>449</b>
	<b>Stichwortverzeichnis</b>	<b>455</b>