

Inhaltsverzeichnis

Vorwort	17
Danksagung	17
Kapitel 1 Einleitung	19
1.1 Zentrale Konzepte	19
1.1.1 Das Client-Server-Modell	20
1.1.2 Mehrschichtanwendungen	20
1.1.3 Ein API und seine Implementierung	22
1.2 Aufbau des Buchs	22
1.2.1 Die Darstellungsschicht	23
1.2.2 Geschäftslogik und Kommunikation	24
1.2.3 Datenhaltung und Datenrepräsentation	26
1.2.4 ... und ist da sonst gar nichts mehr?	26
1.3 An wen richtet sich dieses Buch?	28
1.3.1 Konventionen	28
1.3.2 Disclaimer	29
1.4 Bezug und Installation der Software	29
1.4.1 Java Platform Enterprise Edition 5	30
1.4.2 Die Wahl der »richtigen« Entwicklungsumgebung	30
1.4.3 Das Kompilierungstool Apache Ant	31
1.4.4 Apache Tomcat Server	31
1.4.5 JBoss Application Server	35
1.4.6 Achtung, fertig, los!	36
Kapitel 2 JavaServer Pages	37
2.1 Webanwendungen gestern und heute	38
2.1.1 Das Hypertext Transfer Protocol (HTTP)	39
2.1.2 Der Request – eine Seite anfordern	40
2.1.3 Die Response – ein Server antwortet	43
2.1.4 Hypertext Markup Language (HTML)	44
2.1.5 HTTP vs. HTML	46
2.2 Ihre erste dynamische JSP	46
2.2.1 Einfügen von Java-Anweisungen	47
2.2.2 Einrichten der Webanwendung	48

Inhaltsverzeichnis

2.3 Bausteine für JavaServer Pages	54
2.3.1 JSP-Ausdrücke	54
2.3.2 Mehrzeilige Java-Anweisungen durch Scriptlets.	56
2.3.3 Scriptlets, Ausdrücke und der rote Faden Ihrer JSP	60
2.3.4 Definieren von Java-Methoden durch JSP-Deklarationen.	61
2.3.5 Vergleich: Ausdrücke, Scriptlets und Deklarationen	64
2.3.6 Kommentare	65
2.3.7 Entscheidungen und Schleifen	66
2.4 Die vordefinierten Variablen einer JSP	69
2.4.1 Die verschiedenen Kontexte des Servlet-Containers	70
2.4.2 Acht Variablen zur Kontrolle von JSPs.	73
2.5 Das Auslesen des HTTP-Request	74
2.5.1 Request-Parameter	75
2.5.2 Auslesen aller übermittelten Request-Parameter.	76
2.5.3 Auslesen von Request-Headern.	77
2.5.4 Besonders häufig verwendete Request-Header	79
2.6 Direktiven – Eigenschaften einer JSP	80
2.6.1 Die Seitendirektive page	80
2.6.2 Einbinden von Dateien, die Direktive include	87
2.7 Verwendung der Benutzer-Session	90
2.7.1 Was ist die Benutzer-Session?.	91
2.7.2 Ein kleines Zahlenspiel	91
2.7.3 Arbeiten mit der Session.	93
2.7.4 Kodieren von URLs	95
2.8 JSPs und JavaBeans	96
2.8.1 Grundlagen.	96
2.8.2 Eine einfache JavaBean	97
2.8.3 Eingabe der Daten in die JavaBean.	100
2.8.4 Ausgabe von Attributen einer JavaBean.	104
2.8.5 Erweitertes Erzeugen von JavaBeans	105
2.8.6 Vorteile von JavaBeans.	106
2.9 Zusammenfassung.	107
2.9.1 Goldene Regeln für die Verwendung von JSPs	107

Kapitel 3 Servlets

	109
3.1 Ein einfaches Servlet	109
3.1.1 JSPs und Javacode	110
3.1.2 In grauer Vorzeit	112
3.1.3 Einbinden eines Servlet.	114
3.1.4 HelloWorld-Servlet vs. HelloWorld-JSP	119
3.2 Der Lebenszyklus eines Servlet	120
3.2.1 Die init()-Methode.	121
3.2.2 Die Service-Methoden.	124
3.2.3 Die Destroy-Methode	128

3.2.4	Das vollständige Datenbank-Servlet	128
3.2.5	Vergleich zwischen GET und POST	131
3.2.6	Zusammenfassung des Servlet-Lebenszyklus	131
3.3	Servlets vs. JavaServer Pages	132
3.3.1	Was Sie mit JSPs nicht machen können	133
3.3.2	Direkter Vergleich zwischen Servlets und JSPs	134
3.3.3	Fazit	135
3.4	Cookies	136
3.4.1	Cookies erstellen	136
3.4.2	Cookies lesen	139
3.5	Binäre Daten senden	140
3.5.1	Writer vs. OutputStream	140
3.5.2	Dynamische Bilder mit Servlets erstellen	141
3.5.3	Die Konfiguration des Servlet	142
3.5.4	Das Resultat	143
3.6	Weiterleiten und Einfügen	143
3.6.1	Einfügen einer Ressource	144
3.6.2	Weiterleiten eines Request	144
3.7	Fortgeschrittenes Arbeiten mit der Session	144
3.7.1	Aktive Sessions überwachen	145
3.7.2	Weitere Session-Listener	149
3.8	Servlet-Filter – Servlets light	151
3.8.1	Das Interface javax.servlet.Filter	152
3.8.2	Ein einfacher Basisfilter	153
3.8.3	Konfiguration des Filters	154
3.8.4	Manipulieren von Request und Response	155
3.8.5	Filterketten	158
3.9	Zusammenfassung	160

Kapitel 4 Tag-Bibliotheken	163	
4.1	Ein eigenes Tag	164
4.1.1	Erstellen der Java-Klasse	164
4.1.2	Konfiguration des Tag-Handler	168
4.1.3	Einbinden der Tag-Bibliothek	170
4.1.4	Verwenden des Tag in einer JSP	172
4.2	Verwendung von Attributen	173
4.2.1	Der erweiterte Tag-Handler	173
4.2.2	Registrieren des Tag-Handler im Tag Library Descriptor	175
4.2.3	Verwendung innerhalb der JSP	176
4.3	Vordefinierte Variablen in Tag-Handlers	177
4.3.1	Ein Tag-Handler für HTTPS-Verbindungen	177
4.3.2	Die Konfiguration im Tag Library Descriptor	179
4.3.3	Eine Test-JSP	179

Inhaltsverzeichnis

4.4 Den Rumpf eines Tag manipulieren	180
4.4.1 Die Klasse BodyTagSupport	180
4.4.2 Ein Iteration-Tag.	180
4.4.3 Konfiguration im Tag Library Descriptor	182
4.4.4 Test mit einer JSP	182
4.5 Definition von Script-Variablen	183
4.5.1 Der Gültigkeitsbereich von Script-Variablen.	183
4.5.2 Ein Tag-Handler für Script-Variablen	184
4.5.3 Konfiguration des Tag-Handler.	185
4.5.4 Eine Beispiel-JSP	186
4.6 Mit dem Kontext arbeiten.	187
4.6.1 Wie realisiert man den Zugriff auf Eltern-Tags?	187
4.6.2 Entscheidungen (if-then-else)	188
4.6.3 Überprüfen des Tag-Kontexts	189
4.6.4 Der <if>-Handler	189
4.6.5 <then> und <else>	190
4.6.6 Konfiguration der Tags im Tag Library Descriptor	191
4.6.7 Ein Beispiel	191
4.7 Zusammenfassung	192
Kapitel 5 Struts – die Diva unter den Frameworks	193
5.1 Was sind Frameworks?	194
5.2 Struts – ein Webframework	196
5.2.1 Struts-Komponenten	197
5.2.2 Was Sie außerdem benötigen	200
5.3 Ein Adressbuch	201
5.3.1 Bezug und Installation	201
5.3.2 Unsere Business-Objekte	206
5.3.3 Die JavaServer Pages	207
5.3.4 Struts Tag-Bibliotheken	211
5.3.5 Ressource	212
5.3.6 ActionForms	213
5.3.7 Actions	217
5.3.8 Konfiguration der Anwendung	219
5.3.9 Fazit	223
5.4 Erweiterungen	224
5.4.1 Die Datenbank-Action	224
5.5 Zusammenfassung	226
Kapitel 6 JavaServer Faces (JSF)	229
6.1 Gemeinsamkeiten mit Struts	230
6.1.1 Ein Servlet als Front-Controller	230
6.1.2 Konfiguration über eine zentrale XML-Datei	231
6.1.3 Die Business-Komponente AddressBean	231

6.1.4	Eine Welcome-Seite als Einstiegspunkt in die Applikation	232
6.1.5	Internationalisierung	233
6.2	Eine einfache JSF-Anwendung	233
6.2.1	Darstellung: JavaServer Pages	233
6.2.2	Datenbindungssyntax	236
6.2.3	Controller	237
6.2.4	Konfiguration	238
6.3	Navigationsregeln	239
6.3.1	Verwendung von Wildcards	241
6.3.2	Standardverhalten	242
6.3.3	Prioritäten der Verarbeitung	242
6.4	Validieren von Daten	242
6.4.1	Standardvalidatoren	243
6.4.2	Pflichtfelder deklarieren	243
6.4.3	Validierungsfehler ausgeben	244
6.4.4	Überschreiben der vorgefertigten Fehlermeldungen	246
6.4.5	Implementieren eines eigenen Validators	247
6.5	Konverter	249
6.5.1	Das Converter-Attribut	250
6.5.2	Das Converter-Tag	250
6.6	Zusammenfassung	253
Kapitel 7 Java Naming and Directory Interface		255
7.1	Einführung in Namensdienste	257
7.1.1	Ein API – zwei Services	258
7.1.2	Der Kontext	259
7.1.3	Namensdienste	259
7.1.4	Verzeichnisdienste	262
7.2	API und SPI – Download und Installation	264
7.2.1	Download des API	264
7.2.2	Einige SPI-Implementierungen	265
7.3	Arbeiten mit dem JNDI	267
7.3.1	Erzeugen eines initialen Kontexts	268
7.3.2	Auf den Dienst zugreifen	271
7.3.3	Ausgabe der Elemente eines Kontexts	272
7.3.4	Zugreifen auf gebundene Elemente	274
7.3.5	Umbenennen von Objektbindungen	276
7.3.6	Entfernen von Objekten	277
7.3.7	Verschieben von gebundenen Objekten	277
7.4	Speichern einer Datenbankverbindung	279
7.4.1	Vom DriverManager zur DataSource	279
7.4.2	Ablegen der Datenquelle	280
7.4.3	Auslesen der Datenbankverbindung	282

Inhaltsverzeichnis

7.5 JNDI und Verzeichnisdienste	283
7.5.1 Attribute des Verzeichnisdienstes LDAP	283
7.5.2 Erzeugen eines DirContext-Objekts	284
7.5.3 Binden von Objekten	285
7.5.4 Suche nach Objekten mit bestimmten Attributen	286
7.6 Eine JNDI Lookup-Klasse	287
7.7 JNDI und Webanwendungen	289
7.7.1 Datenquelle mit Apache Tomcat im Standalone-Betrieb	289
7.7.2 Datenquelle im JBoss Application Server	291
7.7.3 Zugriff aus der Applikation	292
7.7.4 Vorteile für Webanwendungen	293
7.8 Zusammenfassung	293
Kapitel 8 Enterprise JavaBeans (EJB)	295
8.1 Aufgabe von Enterprise JavaBeans	296
8.1.1 Kapselung der Geschäftslogik	296
8.1.2 Transaktionsmanagement	297
8.1.3 Loadbalancing	298
8.2 Vom Webserver zum Application-Server	299
8.2.1 Portabilität von Enterprise JavaBeans	300
8.2.2 Aufgaben eines EJB-Containers	301
8.3 Beans, Beans, Beans	302
8.3.1 Beans	302
8.3.2 JavaBeans	302
8.3.3 Enterprise JavaBeans (EJB)	303
8.4 Methodenfernaufruf	304
8.4.1 Verschiedene Broker-Architekturen	305
8.4.2 Austausch von Objekten durch Serialisierung	306
8.5 Verschiedene Typen von Enterprise JavaBeans	307
8.6 EJBs in freier Wildbahn	308
8.6.1 Stateless Session Beans	308
8.6.2 Stateful Session Beans	309
8.6.3 Hello World – eine Stateless Session Bean	309
8.6.4 Hello World – eine Stateful Session Bean	314
8.6.5 Ein Client	316
8.7 Restriktionen bei der Implementierung von EJBs	317
8.8 Zusammenfassung	318
Kapitel 9 Java Message Service	319
9.1 Asynchrone Kommunikation	320
9.1.1 Download und Installation	321
9.1.2 Message Oriented Middleware	321
9.1.3 Vorteile asynchroner Kommunikation	321

9.2 Das Konzept	322
9.2.1 Kommunikationspartner	323
9.2.2 Nachrichtenkonzepte	324
9.2.3 Konfiguration und Eigenschaften der Dienste	325
9.3 Bestandteile des API	326
9.3.1 javax.jms.ConnectionFactory	326
9.3.2 javax.jms.Connection	326
9.3.3 javax.jms.Session	327
9.3.4 javax.jms.Destination	328
9.3.5 javax.jms.Message	328
9.3.6 Zusammenfassung des API	330
9.4 Senden einer Nachricht	330
9.4.1 Der schematische Ablauf	331
9.4.2 Ein Nachrichtensender (Message Producer)	332
9.4.3 Konfiguration der Queue	334
9.4.4 Test der Anwendung	335
9.5 Empfangen einer Nachricht	335
9.6 Empfangsverfahren: Pull vs. Push	337
9.6.1 Push, wenn der Postmann zweimal klingelt	338
9.7 Topic vs. Queue	339
9.7.1 Ein konsolenbasierter Chat	339
9.7.2 Konfiguration und Test des Chat	342
9.7.3 Topic vs. Queue	342
9.8 Optionen für Nachrichten	343
9.8.1 Priorität einer Nachricht	343
9.8.2 Das Verfallsdatum einer Nachricht festlegen	343
9.8.3 Identifizierung einer Nachricht	344
9.9 Filtern einer Nachricht	345
9.9.1 Setzen und Auslesen von Eigenschaften	345
9.9.2 Ausgabe von Message-Attributten	346
9.9.3 Filtern anhand von Attributen	347
9.9.4 Filterelemente	349
9.10 Transaktionen und Empfangsbestätigungen	351
9.10.1 Transaktionen beim Senden	352
9.10.2 Transaktionen beim Nachrichtenempfang	352
9.10.3 Empfangsbestätigungen	353
9.11 Hin und zurück – Synchrone Kommunikation mit JMS	354
9.11.1 Ein QueueRequestor	355
9.11.2 Ein Empfänger, der antwortet	356
9.11.3 Request und Reply vs. Remote Procedure Call	358
9.12 Message Driven Beans	358
9.12.1 Drei erfolgreich verheiratete Java-EE-Technologien	361
9.12.2 Alternative Konfiguration mit Deployment Descriptor	361
9.13 Zusammenfassung	362

Inhaltsverzeichnis

Kapitel 10 Persistenz – JavaBeans und Datenbanken	365
10.1 Etablierte Persistenz-Standards	365
10.1.1 Java Database Connectivity (JDBC)	366
10.1.2 EJBs Entity Beans	367
10.1.3 Java Data Objects	367
10.1.4 Freie OpenSource Frameworks	368
10.1.5 Das Persistenz API	368
10.2 Bezug und Installation	369
10.2.1 Hypersonic SQL Database (HSQLDB)	369
10.3 Eine einfache JavaBean	370
10.3.1 Verfeinerungen	371
10.3.2 Konfiguration	376
10.4 Arbeiten mit der JavaBean	378
10.4.1 Anlegen eines neuen Datensatzes	378
10.4.2 Suche anhand des Primärschlüssels	381
10.4.3 Zustände einer JavaBean	382
10.4.4 Manipulieren einer JavaBean	383
10.4.5 Löschen eines vorhandenen Datensatzes	384
10.5 Das Query API	385
10.5.1 Suche über den Nachnamen	385
10.5.2 Benannte Anfragen hinterlegen	387
10.5.3 Anspruchsvolle Anfragen	388
10.6 Datensätze verknüpfen	389
10.6.1 Die E-Mail-Bean	389
10.6.2 Relationen zwischen Datensätzen	390
10.6.3 Verknüpfen zweier JavaBeans über Annotationen	392
10.6.4 Unterschiedliche Fetch-Typen	397
10.7 Persistenz im EJB-Container	398
10.7.1 Grundlagen	398
10.7.2 Eine einfache JavaBean	399
10.7.3 Konfiguration der Persistenz-Unit	400
10.7.4 Anpassen der Session Bean	400
10.7.5 Abschließende Bemerkungen	402
10.8 Zusammenfassung	403
Kapitel 11 eXtensible Markup Language (XML)	405
11.1 Kurze Einführung in XML	405
11.1.1 Kurze Geschichte der Markup-Sprachen	407
11.1.2 Die Geburtsstunde für XML	409
11.1.3 Element für Element zum Dokument – Regeln für XML	411
11.1.4 Dokumentzentriert vs. datenzentriert	412
11.1.5 Wichtige auf XML basierende Standards	413
11.1.6 Vorteile von XML	414

11.2 Elemente eines XML-Dokuments	414
11.2.1 Der Prolog und XML-Anweisungen	414
11.2.2 Das Element	415
11.2.3 Zeichenketten	418
11.2.4 Kommentare	419
11.2.5 Namensräume	420
11.2.6 CDATA-Rohdaten	421
11.3 Verarbeitungsmodelle für XML	421
11.3.1 DOM vs. JDOM vs. DOM4J	422
11.4 Arbeiten mit dem JDOM	424
11.4.1 Erzeugen eines neuen XML-Dokuments	424
11.4.2 Hinzufügen von weiteren Elementen	425
11.4.3 Kurzform für die Definition des Beispieldokuments	426
11.4.4 Mischen von Text und Elementen	428
11.4.5 Einführen von Namensräumen	429
11.4.6 Einlesen eines vorhandenen XML-Dokuments	430
11.4.7 Traversieren eines Dokuments	432
11.4.8 Ausgabe der direkten Kindelemente	433
11.4.9 Löschen von Elementen	434
11.4.10 Herauslösen und Klonen eines Elements	434
11.4.11 Einsatz von Filtern	435
11.4.12 Ausgabe des Dokuments	437
11.4.13 Umwandeln von JDOM in DOM	439
11.5 Mit SAX-Events arbeiten	440
11.5.1 Arbeitsweise von SAX	440
11.5.2 Callback-Methoden	441
11.5.3 Ein Dokument via SAX parsen	442
11.5.4 SAX-Events in der Pipeline	446
11.5.5 Einer für alles	451
11.5.6 Wofür eignet sich die Verarbeitung mit SAX-Events?	451
11.5.7 Zusätzliche Handler	452
11.6 Zusammenfassung	453
Kapitel 12 XSL, XPath und Co.	455
12.1 XPath – eine Einführung	455
12.2 Arbeiten mit XPath	456
12.2.1 Einfache Ausdrücke	457
12.2.2 Suche nach Elementen	457
12.2.3 Navigation über »Arrays« von Elementen	458
12.2.4 Verwenden des Operators or	458
12.2.5 Referenzieren von Attributen	459
12.2.6 Komplexe Pfade definieren	459
12.2.7 Operationen und Relationen	461
12.2.8 XPath-Funktionen	462

Inhaltsverzeichnis

12.3	Zusammenfassung XPath	465
12.4	Die eXtensible Stylesheet Language	466
12.4.1	Anwendungen für die eXtensible Stylesheet Language	467
12.4.2	Ein einfaches Template	468
12.5	eXtensible Stylesheet Language Transformation (XSLT)	470
12.5.1	Die Qual der Wahl der Implementierung	470
12.5.2	Transformation über die Kommandozeile	472
12.5.3	Transformation eines DOM	472
12.5.4	Verschiedene Transformationsquellen und Ziele	474
12.5.5	SAX vs. DOM	475
12.5.6	Laden eines referenzierten Stylesheet	475
12.5.7	Template vs. Transformer	476
12.5.8	Transformieren eines JDOM	477
12.5.9	Zusammenfassung XSL-Transformationen	477
12.6	XSL Stylesheets	478
12.6.1	Ein einfaches Template	478
12.6.2	Importieren anderer Stylesheets	478
12.6.3	Einfügen eines Stylesheet	479
12.6.4	Attribut-Sets	479
12.6.5	Variablen	479
12.6.6	Weiterführende Literatur	480
12.7	Ein Stylesheet für die Bibliothek	481
12.7.1	Das Root-Element	481
12.7.2	Das Library-Element und eine konstante Kopfzeile	482
12.7.3	Transformation von Book-Elementen	483
12.7.4	Entscheidungen	484
12.7.5	Das vollständige XSL Stylesheet	485
12.7.6	Zusammenfassung XSL Stylesheet	487
12.8	XSL Formatting Objects	487
12.8.1	Geschichte	488
12.8.2	Grundlagen und Installation	488
12.9	Aufbau eines FO-Dokuments	490
12.9.1	Die Definition einer A4-Seite	491
12.9.2	Das Auffüllen mit Inhalt	492
12.9.3	Rendern mit dem Formatting Objects Processor (FOP)	493
12.9.4	Resultat	494
12.10	Von XML zum PDF – ein Beispiel	494
12.10.1	Das Stylesheet	494
12.10.2	Attribut-Sets	497
12.10.3	Das Root-Element	497
12.10.4	Library und Book	498
12.10.5	Das Rendering	498
12.11	Zusammenfassung	500

Kapitel 13 Webservices	503
13.1 Die Idee hinter Webservices	504
13.2 XML – Remote Procedure Call (XML-RPC).....	504
13.2.1 Grundlagen	505
13.2.2 Aufbau eines XML-RPC-Dokuments.....	506
13.2.3 Übergabe der Parameter.....	507
13.2.4 Fehlerbehandlung.....	508
13.2.5 Zusammenfassung XML-RPC.....	508
13.3 Simple Object Access Protocol (SOAP).....	510
13.3.1 Die Idee hinter SOAP	510
13.3.2 SOAP und Java	510
13.4 Erstellen eines einfachen Webservice.....	512
13.5 Einen Webservice ansprechen	514
13.6 Webservice Description Language (WSDL).....	515
13.7 Universal Description and Discovery Interface (UDDI).....	516
13.8 Zusammenfassung	517
13.8.1 Allgemeine Abschlussbemerkungen	517
J2EE Software License	521
Stichwortverzeichnis	527