

jetzt lerne ich

J2EE

Der einfache Einstieg in die Programmierung
mit der Java 2 Enterprise Edition

THOMAS STARK



Markt+Technik

Inhaltsverzeichnis

Vorwort	17
1 Einleitung	19
1.1 Zentrale Konzepte	20
1.1.1 Das Client-Server-Modell	20
1.1.2 Mehrschichtanwendungen	21
1.1.3 Eine API und ihre Implementierung	22
1.2 Aufbau des Buches	22
1.2.1 Die Darstellungsschicht	23
1.2.2 Geschäftslogik und Kommunikation	24
1.2.3 Datenhaltung und Datenrepräsentation	25
1.2.4 ... und ist da sonst gar nichts mehr?	25
1.3 An wen richtet sich dieses Buch	27
1.4 Bezug und Installation der Software	27
1.4.1 Java 2 Platform Enterprise Edition (J2EE)	28
1.4.2 Die Wahl der Entwicklungsumgebung	28
1.4.3 Das Kompilierungstool Apache Ant	29
1.4.4 Apache Tomcat Webserver	29
1.4.5 JBoss Application Server	33
1.4.6 Achtung, fertig, los!	35
1.5 Übungen	35
1.6 Fragen zur Selbstkontrolle	35

2	Einführung in JavaServer Pages	37
2.1	Request und Response – Frage und Antwort	38
2.1.1	Das Hypertext Transfer Protokoll	39
2.1.2	Hypertext Markup Language (HTML)	42
2.1.3	HTTP vs. HTML	42
2.2	Eine einfache JSP	43
2.2.1	Einfügen von Java-Anweisungen	44
2.2.2	Installation der Webanwendung	44
2.2.3	Test der JSP	47
2.3	Bausteine für JavaServer Pages	48
2.3.1	JSP-Ausdrücke	49
2.3.2	Mehrzeilige Java-Anweisungen durch Scriptlets	50
2.3.3	Definieren von Java-Methoden durch JSP-Deklarationen	53
2.3.4	Vergleich: Ausdrücke, Scriptlets und Deklarationen	57
2.3.5	Kommentare	58
2.4	Entscheidungen und Schleifen	59
2.4.1	Die if-else-Anweisung	59
2.4.2	Die switch-Verzweigung	60
2.4.3	Schleifen	61
2.5	Zusammenfassung	62
2.6	Übungen	63
2.7	Fragen zur Selbstkontrolle	63
3	JavaServer Pages einsetzen	65
3.1	Die vordefinierten Variablen einer JSP	65
3.1.1	Die verschiedenen Kontexte des Webserver	66
3.1.2	Acht Variablen zur Kontrolle von JSPs	68
3.2	Das Auslesen des HTTP-Requests	70
3.2.1	Request-Parameter	70
3.2.2	Auslesen aller übermittelten Request-Parameter	71
3.2.3	Auslesen von Request-Headern	72
3.2.4	Besonders häufig verwendete Request-Header	74
3.3	Direktiven – Eigenschaften einer JSP	75
3.3.1	Die Seitendirektive page	76
3.3.2	Einbinden von Dateien – die Direktive include	83
3.4	Verwendung der Benutzer-Session	86
3.4.1	Was ist die Benutzer-Session	87
3.4.2	Ein kleines Zahlenspiel	87
3.4.3	Arbeiten mit der Session	90
3.4.4	Kodieren von URLs	91

3.5	JSPs und JavaBeans	93
3.5.1	Grundlagen	93
3.5.2	Eine JavaBean	94
3.5.3	Eingabe der Daten in die JavaBean	96
3.5.4	Ausgabe von Attributen einer JavaBean	102
3.5.5	Erweitertes Erzeugen von JavaBeans	103
3.5.6	Vorteile von JavaBeans	104
3.6	Zusammenfassung	106
3.6.1	Goldene Regeln für die Verwendung von JSPs	106
3.7	Übungen	107
3.8	Fragen zur Selbstkontrolle	107
4	Servlets	109
4.1	Ein einfaches Servlet	109
4.1.1	JSPs und Java-Code	109
4.1.2	In grauer Vorzeit ...	110
4.1.3	Einbinden eines Servlets	111
4.1.4	HelloWorld-Servlet vs. HelloWorld-JSP	116
4.2	Der Lebenszyklus eines Servlets	117
4.2.1	Die Init-Methode	117
4.2.2	Die Service-Methoden	121
4.2.3	Die Destroy-Methode	125
4.2.4	Das vollständige Datenbank-Servlet	126
4.2.5	Vergleich zwischen GET und POST	129
4.2.6	Zusammenfassung des Servlet-Lebenszyklus	130
4.3	Servlet vs. JavaServer Pages	130
4.3.1	Was Sie mit JSPs nicht machen können	131
4.3.2	Ein direkter Vergleich zwischen Servlets und JSPs	132
4.3.3	Fazit	133
4.4	Cookies	134
4.4.1	Cookies erstellen	135
4.4.2	Cookies lesen	138
4.5	Binäre Daten senden	139
4.5.1	Writer vs. OutputStream	139
4.5.2	Ein Beispiel-Servlet	140
4.5.3	Die Konfiguration des Servlets	142
4.5.4	Das Resultat	143
4.6	Weiterleiten und Einfügen	143
4.6.1	Einfügen einer Ressource	144
4.6.2	Weiterleiten eines Requests	144

4.7	Zusammenfassung	144
4.8	Übungen	145
4.9	Fragen zur Selbstkontrolle	145
5	Tag-Bibliotheken	147
5.1	Ein eigenes Tag	148
5.1.1	Erstellen der Java-Klasse	148
5.1.2	Konfiguration des Tag-Handlers	151
5.1.3	Einbinden der Tag-Bibliothek	154
5.1.4	Verwenden des Tags in einer JSP	156
5.2	Verwendung von Attributen	157
5.2.1	Der erweiterte Tag-Handler	157
5.2.2	Registrieren des Tag-Handlers im TLD	158
5.2.3	Verwendung innerhalb der JSP	159
5.3	Vordefinierte Variablen	159
5.3.1	Ein Secure-Tag-Handler	160
5.3.2	Die Konfiguration	161
5.3.3	Eine Test-JSP	162
5.4	Den Rumpf eines Tags manipulieren	162
5.4.1	Die Klasse BodyTagSupport	162
5.4.2	Ein Iteration-Tag	163
5.4.3	Konfiguration im Tag Library Descriptor	164
5.4.4	Test mit einer JSP	165
5.5	Zusammenfassung	166
5.6	Übungen	166
5.7	Fragen zur Selbstkontrolle	167
6	Java Naming & Directory Interface	169
6.1	Einführung	169
6.1.1	Eine API – zwei Services	171
6.1.2	Der Kontext	171
6.1.3	Namensdienste	172
6.1.4	Verzeichnisdienste	174
6.2	API und SPI – Download und Installation	175
6.2.1	Download der API	176
6.2.2	Einige SPI-Implementierungen	176
6.3	Arbeiten mit dem JNDI	178
6.3.1	Erzeugen eines initialen Kontextes	179
6.3.2	Auf den Dienst zugreifen ...	183
6.3.3	Ausgabe der Elemente eines Kontextes	184
6.3.4	Zugreifen auf gebundene Elemente	186

6.3.5	Umbenennen von Objekt-Bindungen	188
6.3.6	Entfernen von Objekten	189
6.3.7	Verschieben von gebundenen Objekten	190
6.4	Speichern einer Datenbankverbindung	192
6.4.1	Vom DriverManager zur DataSource	192
6.4.2	Ablegen der Datenquelle	193
6.4.3	Auslesen der Datenbankverbindung	195
6.5	JNDI und Verzeichnisdienste	196
6.5.1	Attribute des Verzeichnisdienstes LDAP	197
6.5.2	Erzeugen eines DirContext-Objekts	198
6.5.3	Binden von Objekten	199
6.5.4	Suche nach Objekten mit bestimmten Attributen	199
6.6	JNDI und Webanwendungen	201
6.6.1	Datenquelle im Apache Tomcat	202
6.6.2	Konfiguration des Apache Tomcat	202
6.6.3	Zugriff aus der Applikation	204
6.6.4	Vorteile für Webanwendungen	205
6.7	Zusammenfassung	206
6.8	Übungen	206
6.9	Fragen zur Selbstkontrolle	207
7	Enterprise JavaBeans (EJB)	209
7.1	Wofür benötigen Sie JavaBeans	210
7.1.1	Kapselung der Geschäftslogik	210
7.1.2	Transaktions-Management	211
7.1.3	Lastverteilung	212
7.1.4	Unterschiede zu JavaBeans	213
7.2	Vom Webserver zum J2EE Application Server	214
7.2.1	Unterstützte APIs eines Application Servers	215
7.2.2	Aufgaben eines EJB-Containers	215
7.3	Verschiedene Typen von EJBs	217
7.3.1	Entity-Beans	217
7.3.2	Session-Beans	217
7.3.3	Message Driven Beans	218
7.4	Bestandteile einer Enterprise JavaBean	218
7.4.1	Das Home-Interface	218
7.4.2	Das Remote-Interface	220
7.4.3	Die EJB-Implementierung	220
7.4.4	Der Bean-Kontext	222
7.4.5	Der Deployment Descriptor	223
7.4.6	Restriktionen bei der Implementierung von EJBs	223

7.4.7	Zusammenfassung	223
7.5	Eine erste Session-Bean	224
7.5.1	Das Remote-Interface	224
7.5.2	Das Home-Interface	225
7.5.3	Die Implementierung	226
7.5.4	Erstellen des Deployment Descriptors	229
7.5.5	Konfiguration des EJB-Containers	232
7.5.6	Der HelloWorld-Client	235
7.5.7	Test der Applikation	238
7.6	Zusammenfassung	238
7.7	Übungen	239
7.8	Fragen zur Selbstkontrolle	240
8	Java Message Service	241
8.1	Asynchrone Kommunikation	243
8.1.1	Download und Installation	243
8.1.2	Message Oriented Middleware	243
8.1.3	Vorteile asynchroner Kommunikation	244
8.2	Das Konzept	245
8.2.1	Kommunikationspartner	245
8.2.2	Nachrichten-Konzepte	246
8.2.3	Konfiguration und Eigenschaften der Dienste	248
8.3	Bestandteile der API	249
8.3.1	javax.jms.ConnectionFactory	249
8.3.2	javax.jms.Connection	249
8.3.3	javax.jms.Session	250
8.3.4	javax.jms.Destination	251
8.3.5	javax.jms.Message	252
8.3.6	Zusammenfassung der API	254
8.4	Senden einer Nachricht	254
8.4.1	Eine JNDI Lookup-Klasse	254
8.4.2	Der schematische Ablauf	256
8.4.3	Ein Nachrichten-Sender (Message Producer)	257
8.4.4	Konfiguration der Queue	259
8.4.5	Test der Anwendung	260
8.5	Empfangen einer Nachricht	260
8.6	Empfangsverfahren: Pull vs. Push	263
8.6.1	Push, wenn der Postmann zweimal klingelt	264
8.7	Topic vs. Queue	266
8.7.1	Ein konsolen-basierter Chat	266
8.7.2	Konfiguration und Test des Chat	269

8.8	Optionen für Nachrichten	270
8.8.1	Priorität einer Nachricht	270
8.8.2	Das Verfallsdatum einer Nachricht festlegen	271
8.8.3	Identifizierung einer Nachricht	272
8.9	Transaktionen und Empfangsbestätigungen	272
8.9.1	Transaktionen beim Senden	273
8.9.2	Transaktionen beim Nachrichteneingang	273
8.9.3	Empfangsbestätigungen	274
8.10	Message Driven Beans	276
8.10.1	Message Driven Beans aus Sicht des Client	276
8.10.2	Message Driven Beans aus Sicht des Servers	276
8.10.3	Kommunikation mit dem EJB-Container	278
8.10.4	Ein Beispiel	280
8.11	Zusammenfassung	282
8.12	Übungen	283
8.13	Fragen zur Selbstkontrolle	284
9	Java Data Objects	285
9.1	Eine Einführung in die Technik	286
9.1.1	Bytecode-Anreicherung	286
9.1.2	Verschiedene Service Provider	287
9.1.3	Die Konfiguration	289
9.2	Installation der Software	289
9.2.1	Bezug der Implementierung	289
9.2.2	Die Datenbank bTree	290
9.2.3	Der Compiler Generator ANTLR	290
9.3	Ein einfaches Beispiel	291
9.3.1	Die Konfiguration des Data Object Descriptors	292
9.3.2	Definition eines Primary Key	294
9.3.3	Übersetzen und Anreichern	295
9.4	Mit dem PersistenceManager arbeiten	296
9.4.1	PersistenceManagerFactory für den FO Store	296
9.4.2	PersistenceManagerFactory für MySQL	298
9.4.3	Anlegen eines neuen Datensatzes	299
9.4.4	Gemeinsames Speichern mehrerer Datensätze	302
9.4.5	Suche nach Datensätzen	303
9.4.6	Eingeschränkte Suche nach Datensätzen	304
9.4.7	Rollback vom Manipulationen	306
9.4.8	Arbeit ohne Netz und doppelten Boden	308
9.4.9	Löschen von Objekten	309
9.5	JDO Query Language (JDO-QL)	310

9.5.1	Eine Einführung	311
9.5.2	Syntax-Elemente der Anfragesprache	311
9.6	Zusammenfassung	314
9.7	Übungen	315
9.8	Fragen zur Selbstkontrolle	316
10	eXtensible Markup Language	317
10.1	Kurze Einführung in XML	317
10.1.1	Die Geschichte der Markup-Sprachen	319
10.1.2	Die Geburtsstunde von XML	320
10.1.3	Tag für Tag zu den Dokument-Regeln für XML	321
10.1.4	Dokumentzentriert vs. Datenzentriert	323
10.1.5	Wichtige auf XML basierende Standards	323
10.1.6	Vorteile von XML	324
10.2	Elemente eines XML-Dokuments	325
10.2.1	Der Prolog und XML-Anweisungen	325
10.2.2	Das Tag	327
10.2.3	Zeichenketten	329
10.2.4	Kommentare	331
10.2.5	Namensräume	332
10.2.6	CDATA-Rohdaten	333
10.3	Verarbeitungsmodelle für XML	333
10.3.1	DOM vs. JDOM vs. DOM4J	334
10.4	Arbeiten mit dem JDOM	336
10.4.1	Erzeugen eines neuen XML-Dokuments	336
10.4.2	Hinzufügen von weiteren Elementen	337
10.4.3	Mischen von Text und Elementen	341
10.4.4	Einführen von Namensräumen	342
10.4.5	Einlesen eines vorhandenen XML-Dokuments	343
10.4.6	Traversieren eines Dokuments	345
10.4.7	Ausgabe der direkten Kind-Knoten	347
10.4.8	Löschen von Elementen	347
10.4.9	Herauslösen und Klonen eines Elements	348
10.4.10	Ausgabe des Dokuments	349
10.4.11	Umwandeln von JDOM in DOM	352
10.5	Mit SAX-Events arbeiten	352
10.5.1	Arbeitsweise von SAX	353
10.5.2	Callback-Methoden	353
10.5.3	Ein Dokument via SAX parsen	355
10.5.4	Vorteile von SAX-Events	360
10.6	Zusammenfassung	361

10.7	Übungen	362
10.8	Fragen zur Selbstkontrolle	362
11	XPath, XSL und FO	363
11.1	XPath	363
11.1.1	Referenzieren von Elementen	364
11.1.2	Komplexe Pfade definieren	368
11.1.3	Operationen und Relationen	370
11.1.4	XPath-Funktionen	371
11.1.5	Zusammenfassung XPath	375
11.2	eXtensible Stylesheet Language	376
11.2.1	Anwendungen für die eXtensible Stylesheet Language	376
11.2.2	Ein einfaches Template	377
11.2.3	eXtensible Stylesheet Language Transformation (XSLT)	379
11.2.4	Aufbau eines Stylesheets	384
11.2.5	Ein Beispieldokument	387
11.2.6	Zusammenfassung XSL	394
11.3	XSL Formatting Objects	394
11.3.1	Geschichte	395
11.3.2	Grundlagen und Installation	395
11.3.3	Aufbau eines Dokuments	397
11.4	Von XML zum PDF – ein Beispiel	402
11.4.1	Das Stylesheet	402
11.4.2	Das Rendering	405
11.5	Zusammenfassung	407
11.6	Übungen	408
11.7	Fragen zur Selbstkontrolle	408
	Anhang: Antworten zu den Fragen	409
	Stichwortverzeichnis	419