

Inhalt

Vorwort	15
---------------	----

1 XML	17
1.1 Was ist XML?	17
1.2 Das XML-Dokument	18
1.2.1 XML-Namen	19
1.2.2 Elemente	19
1.2.3 Attribute	20
1.2.4 Text	21
1.2.5 Kommentare	22
1.2.6 XML-Deklaration	23
1.2.7 Processing Instructions	24
1.2.8 Knoten und ihre Beziehungen zueinander	25
1.2.9 Wohlgeformtheit	26
1.2.10 XML-Parser	27
1.3 Die Namensräume	27
1.4 XML beschränken mit DTD	30
1.4.1 Ein einfaches Beispiel	31
1.4.2 Elemente	32
1.4.3 Attribute	33
1.4.4 Entities	34
1.4.5 DTD einbinden	35
1.5 XML beschränken mit XML Schema	37
1.5.1 Ein XML Schema referenzieren	38
1.5.2 Die XML-Schema-Datei	38
1.5.3 Definition von einfachen Typen	39
1.5.4 Definition von komplexen Typen	41
1.5.5 Definition von Attributen	46
1.5.6 Definition von Elementen	47
1.5.7 Globale Definitionen und Referenzen	47
1.5.8 XML Schema und Namensräume	49
1.6 Navigieren mit XPath	51
1.6.1 Knotentypen	52
1.6.2 Lokalisierungspfad	53
1.6.3 Prädikate	54
1.6.4 XPath-Funktionen	56
1.6.5 Einsatzgebiete von XPath	57

Inhalt

1.7	XML transformieren mit XSLT	57
1.7.1	Das XSLT-Stylesheet	58
1.7.2	Wichtige XSLT-Instruktionen	61
1.8	XSL-FO	63
1.8.1	Der Aufbau eines XSL-FO Dokuments	63
1.8.2	XSL-FO und XSLT-Stylesheet	68
1.8.3	Der FO-Prozessor	69
1.8.4	Ein kleines Beispiel	70
1.9	Zusammenfassung	71
2	XML mit Java – Grundbegriffe	73
2.1	Parsen und Serialisieren	74
2.1.1	Begriffsklärung	74
2.1.2	Aufgaben von XML-Bibliotheken	76
2.2	Validieren	76
2.3	Streambasierte Verarbeitung	77
2.3.1	SAX	78
2.3.2	StAX	81
2.4	Modellbasierte Verarbeitung	84
2.4.1	DOM	87
2.4.2	Binding	92
2.5	Navigieren mit XPath und Transformieren mit XSLT	100
2.6	APIs und Factories	101
2.6.1	APIs in der Java-Welt	102
2.6.2	Probleme abstrakter APIs	103
2.6.3	Die Lösung: Factories	104
2.6.4	Einziges Problem: die Factory finden	105
2.6.5	Zusammenfassung	106
2.7	Die Rolle von JAXP	106
2.8	Zusammenfassung	108
3	DOM	109
3.1	Einführungsbeispiel	109
3.1.1	Öffnen des Beispiels	110
3.1.2	Beispiel 1: Parsen und Navigieren	112
3.1.3	Beispiel 2: Dokumente erzeugen, ändern und serialisieren	122
3.1.4	Zusammenfassung	129

3.2	Der Ausgangspunkt: die DOMImplementation	130
3.2.1	Features abfragen	131
3.2.2	Features abrufen	132
3.2.3	Dokument erzeugen	133
3.2.4	Erzeugen einer DOM-Implementierung	133
3.2.5	Zusammenfassung	135
3.3	Das Objektmodell	135
3.3.1	Knoten- und Hilfsinterfaces	135
3.3.2	DOM-eigene Datenstrukturen für Listen und Maps	138
3.3.3	Node – die Basis aller Knoten-Interfaces	139
3.3.4	Document	143
3.3.5	Element	144
3.3.6	Attr	146
3.3.7	Text, CDATASEction und Comment	147
3.3.8	ProcessingInstruction	149
3.3.9	Sonstige Knoten-Interfaces	149
3.3.10	Zusammenfassung	150
3.4	Navigieren	151
3.4.1	Allgemeine Navigationsfunktionen	152
3.4.2	Spezifische Navigationsfunktionen	153
3.4.3	Zusammenfassung	155
3.5	Neue Knoten generieren	155
3.6	Modifizieren	157
3.7	Dokumente parsen und serialisieren	159
3.7.1	Überblick	159
3.7.2	Parsen	160
3.7.3	Serialisieren	164
3.7.4	Zusammenfassung	166
3.8	Namensräume	166
3.8.1	Namensraum-URI, Präfix, lokaler Name und qualifizierter Name	166
3.8.2	Zugriff auf Knoten über ihren Namen	168
3.8.3	Knoten in Namensräumen erzeugen	169
3.8.4	Manuelles Erzeugen von Präfix- und Default-Namespace-Bindungen	170
3.8.5	Zusammenfassung	172
3.9	Alternative DOM-Implementierungen	172

4 SAX	173
4.1 Einführungsbeispiel	174
4.1.1 Öffnen des Beispiels	174
4.1.2 Das Beispiel starten	175
4.1.3 Das Hauptprogramm	175
4.1.4 Die Handler-Klasse – PersonHandler	177
4.1.5 Zusammenfassung	183
4.2 Parsen	184
4.2.1 Parser-Instanz erzeugen	185
4.2.2 Konfigurieren des XMLReaders	186
4.2.3 Eingabedokument vorbereiten	188
4.2.4 Starten des Parse-Vorgangs	189
4.2.5 Zusammenfassung	190
4.3 Handler	190
4.3.1 Den Dokumentinhalt abgreifen mit dem ContentHandler	190
4.3.2 Erweiterte Inhaltsinformation abgreifen mit dem LexicalHandler	198
4.3.3 Zugriff auf DTD-Inhalte mit dem DTDHandler und dem DeclHandler	199
4.3.4 Fehlerbehandlung mit dem ErrorHandler	201
4.3.5 Verweise auflösen mit dem EntityResolver	202
4.3.6 Hilfsklassen zur Erstellung von Handler-Klassen	204
4.3.7 Zusammenfassung	206
4.4 Filter und Pipelines	206
4.5 Ein Wort zu SAX 1	213
4.6 Zusammenfassung	214
5 JAXP	215
5.1 Aufbau und Installation von JAXP	216
5.1.1 Die Package-Struktur von JAXP	216
5.1.2 JAXP-Versionen und die JAXP-Referenzimplementierung	216
5.2 Grundlegende Klassen und Mechanismen	218
5.2.1 Die Klasse QName	218
5.2.2 Der Factory-Mechanismus von JAXP	219
5.3 Einstieg in die SAX-API	221
5.3.1 Einführungsbeispiel	222

5.3.2	Der Ausgangspunkt: die SAXParserFactory	225
5.3.3	Die Parser-Implementierung: der SAXParser ...	229
5.3.4	Zusammenfassung	231
5.4	Einstieg in die DOM-API	231
5.4.1	Einführungsbeispiel	232
5.4.2	Der Ausgangspunkt: die DocumentBuilderFactory	235
5.4.3	Dokumente parsen und generieren mit dem DocumentBuilder	240
5.4.4	Zusammenfassung	243
5.5	Dokumente transformieren	243
5.5.1	Einführungsbeispiel	244
5.5.2	Der Ausgangspunkt: die TransformerFactory ...	248
5.5.3	Transformationen durchführen mit dem Transformer	252
5.5.4	Transformationsquellen	257
5.5.5	Transformationsergebnisse	262
5.5.6	SAX-Erweiterungen	265
5.5.7	Zusammenfassung	271
5.6	Dokumente validieren	271
5.6.1	Validierung mit klassischen Mitteln	272
5.6.2	Schnelleinstieg in die neue Validation-API	274
5.6.3	Der Ausgangspunkt: die SchemaFactory	277
5.6.4	Schemainformationen wiederverwenden mit dem Schema-Objekt	281
5.6.5	Validierungen durchführen mit dem Validator	282
5.6.6	Validierung von Dokumenten, die als SAX-Events vorliegen	284
5.6.7	Zusammenfassung	286
5.7	Navigieren in Dokumenten	286
5.7.1	Einführungsbeispiel	287
5.7.2	Der Ausgangspunkt: die XPathFactory	290
5.7.3	XPath-Ausdrücke auswerten mit dem XPath-Objekt	295
5.7.4	Alternative XPath-Implementierungen	301
5.7.5	Zusammenfassung	301
5.8	Dokumente serialisieren	302
5.9	Zusammenfassung	303

6 StAX	305
6.1 StAX besorgen und installieren	305
6.1.1 StAX und Java 6	306
6.1.2 StAX und ältere Java-Versionen	306
6.1.3 Die StAX-Referenzimplementierung	307
6.2 Der Factory-Mechanismus von StAX	307
6.3 Die API-Varianten von StAX	307
6.3.1 Die Cursor-API	308
6.3.2 Die Event-Iterator-API	309
6.3.3 Zusammenfassung	311
6.4 Einführungsbeispiel	311
6.4.1 Öffnen des Beispiels	312
6.4.2 Parson	314
6.4.3 Serialisieren	326
6.4.4 Wie geht es weiter?	335
6.5 Der Ausgangspunkt: Die Factory-Klassen	336
6.5.1 Erzeugen von Parson mit der XMLInputFactory	337
6.5.2 Erzeugen von Serializern mit der XMLEventFactory	343
6.5.3 Bausteininstanzen erzeugen mit der XMLEventFactory	344
6.6 Die Cursor-API im Detail	346
6.6.1 Parson mit der Cursor-API über den XMLStreamReader	346
6.6.2 Serialisieren mit der Cursor-API über den XMLStreamWriter	362
6.7 Die Event-Iterator-API im Detail	371
6.7.1 Die Baustein-Interfaces	372
6.7.2 Parson mit der Event-Iterator-API über den XMLEventReader	385
6.7.3 Serialisieren mit der Event-Iterator-API über den XMLEventReader	387
6.8 Zusammenfassung	389
7 JAXB	391
7.1 XML-Binding mit JAXB	391
7.1.1 XML-Binding – ein Rückblick	391
7.1.2 Vorstellung von JAXB	392

7.2	Einführungsbeispiel	395
7.2.1	Öffnen des Beispiels	396
7.2.2	Generierung von Modellklassen mit dem XJC-Binding Compiler	397
7.2.3	Die Compiler-Ausgabe und die Abbildungsregeln	399
7.2.4	Die Anwendung und die JAXB-API	402
7.2.5	Generierung eines Schemas mit dem Schema-Gen Binding Compiler	410
7.3	Abbildung von XML nach Java	413
7.3.1	Standardregeln, Binding Declarations und Mapping Annotations	414
7.3.2	Das Standardverhalten anhand eines Beispiels	416
7.3.3	Konvertierung von Namen	417
7.3.4	Komplexe Typen – ComplexTypes	418
7.3.5	Einfache Typen – SimpleTypes	428
7.3.6	Typ-Binding auf einen Blick	435
7.3.7	ObjectFactory	435
7.3.8	Binding Declarations	436
7.4	Abbildung von Java nach XML	441
7.4.1	Konvertierung von Namen	442
7.4.2	Das Standardverhalten anhand eines Beispiels	443
7.4.3	Klassen	445
7.4.4	Primitive Typen und Standard-Java-Typen	447
7.4.5	Enum-Typen	448
7.4.6	Collection-Typen und eindimensionale Arrays	449
7.4.7	Maps	450
7.4.8	Mehrdimensionale Arrays	452
7.4.9	Mapping Annotations	455
7.4.10	Sonstige Aufgaben	459
7.5	Die JAXB-API	462
7.5.1	Package-Überblick	463
7.5.2	Der Ausgangspunkt: JAXBContext	464
7.5.3	Unmarshalling	467
7.5.4	Marshalling	473
7.5.5	Binden an ein XML-Objektmodell mit dem Binder	477
7.5.6	Validierung	486
7.5.7	Exception-Klassen	490
7.5.8	Nützliche Hilfsklassen	491
7.6	Zusammenfassung	493

8 XML-Webservices mit JAX-WS	495
8.1 Webservices – eine Einführung	496
8.1.1 Was sind eigentlich Webservices?	496
8.1.2 Webservices vs. andere Technologien	498
8.1.3 SOAP und die Rolle von XML	498
8.1.4 Webservices und XML-Binding	500
8.1.5 Webservices beschreiben mit WSDL	500
8.1.6 Aufbau eines WSDL-Dokuments	501
8.1.7 Contract-First oder Code-First?	508
8.1.8 Kompatibilität von Webservices	510
8.1.9 Zusammenfassung	511
8.2 JAX-WS besorgen und installieren	511
8.3 Code-First – Webservices aus Java-Klassen generieren ...	512
8.3.1 Das Beispiel	513
8.3.2 Die Vorgehensweise	514
8.3.3 Die Service-Klasse schreiben	515
8.3.4 Hilfsklassen generieren	527
8.3.5 Testbetrieb	531
8.3.6 Deployment in einem Application Server	536
8.3.7 Die Ansätze kombinieren: Contract-First mit Code-First-Mitteln	539
8.3.8 Zusammenfassung	541
8.4 Contract-First – Das WSDL-Dokument als Ausgangspunkt	542
8.4.1 Das Beispiel	543
8.4.2 Die Vorgehensweise	543
8.4.3 Das WSDL-Dokument erstellen	545
8.4.4 Service-Interface und Binding-Klassen generieren	556
8.4.5 Service-Interface implementieren	562
8.4.6 Testbetrieb	564
8.4.7 Deployment in einem Application Server	566
8.4.8 Zusammenfassung	567
8.5 Webservice-Clients erstellen	568
8.5.1 Verwandtschaft zwischen Server- und Client-Betrieb	568
8.5.2 Die Vorgehensweise	569
8.5.3 Service-Interface, Binding-Klassen und Client generieren	569

8.5.4	Die Aufruf- und Verarbeitungslogik implementieren	572
8.5.5	Zusammenfassung	578
8.6	Ausblick	578
8.6.1	JAX-WS-Dokumentation	579
8.6.2	JAX-WS-Beispiele	581
8.6.3	Alternative Implementierungen: Apache Axis2 und Apache CXF	582
8.6.4	Zusammenfassung	583
9	Publishing mit Java und XML	585
9.1	Grundbegriffe	585
9.2	XML-Publishing-Frameworks	587
9.3	Apache Cocoon	588
9.3.1	Grundsätzliches zu Cocoon	590
9.3.2	Maven 2	592
9.3.3	Cocoon-Block erstellen	593
9.3.4	Cocoon-Block starten	596
9.3.5	Cocoon-Block in Eclipse importieren	596
9.3.6	Cocoon-Block deployen	597
9.3.7	Die Sitemap	598
9.3.8	Die Pipeline	600
9.3.9	Registrieren einer Sitemap-Komponente	602
9.3.10	Verwenden einer Sitemap-Komponente	604
9.3.11	Sitemap-Komponente: Matcher	605
9.3.12	Sitemap-Komponente: Generator	610
9.3.13	Sitemap-Komponente: Transformer	612
9.3.14	Sitemap-Komponente: Serializer	614
9.3.15	Sitemap-Komponente: Reader	615
9.3.16	Ein kleines Beispiel	617
9.3.17	Zusammenfassung	623
Anhang	625	
A	Eclipse-Grundlagen	627
A.1	Eclipse installieren	627
A.2	Erste Gehversuche	628
A.2.1	Eclipse starten	628
A.2.2	Einen Workspace anlegen	629

A.2.3	Die Oberfläche: Menüs, Symbolleisten, Views und Perspectives	629
A.3	Java-Installationen einbinden	632
A.3.1	Öffnen des Konfigurationsdialogs	632
A.3.2	Neue Installation hinzufügen	633
A.4	Projekte	634
A.4.1	Importieren von Projekten	634
A.4.2	Aufbau von Java-Projekten	635
A.4.3	Dateien editieren	637
A.4.4	Projekte öffnen und schließen	639
A.5	Programme starten	639
A.5.1	Starten über den Package Explorer	639
A.5.2	Start über den Editor oder die Outline-View ...	640
A.5.3	Run Configurations	640
A.5.4	Ant-Scripts starten	643
B	Apache Ant	645
B.1	Ant besorgen und installieren	646
B.1.1	Ant in der IDE	646
B.1.2	Ant separat installieren	646
B.2	Erstellen und Ausführen von Ant-Scripts	647
B.2.1	Ein erstes Beispiel	647
B.2.2	Weitere Ant-Tasks	649
B.2.3	Target-Abhängigkeiten	651
B.2.4	Dateimengen	653
B.2.5	Properties	655
B.3	Externe Tasks einbinden	660
B.3.1	Ein kleines Beispiel	662
B.4	Zusammenfassung	663
C	Properties und Features von SAX	665
C.1	Wichtige Standard-Properties	665
C.2	Wichtige Standard-Features	666
D	JAXB-Binding Compiler	669
D.1	Der XJC Binding Compiler	669
D.2	Der SchemaGen Binding Compiler	677
E	Wichtige Quellen	683
F	Buch-CD	689
	Index	691