

Programski Sistemi

Java i XML



XML

- XML je skrećenica za Extensible Markup Language, a definisao ga je 1998 by the World Wide Web Consortium (W3C)
- XML dokument se sastoji od elemenata; svaki element ima početni tag, sadržaj i završni tag
- XML dokument sme da ima samo jedan korenski element, tj. jedan tag koji sadrži sve ostale elemente

XML

- XML je case sensitive, tj. razlikuje velika i mala slova
- XML fajl mora da bude dobro formatiran:
 - Uvek počinje zaglavljem
 - Svaki početni tag ima završni tag
 - Svi tagovi su ugnježđeni
- XML fajl je validan ako je dobro formatiran i ako sadrži link ka XML šemi koja definiše kako on treba da izgleda

Prednosti XML-a

- XML je običan tekst
- Predstavlja podatke bez definisanja kako ti podaci treba da budu prikazani
- Može se transformisati u druge formate preko XSL-a
- Lako se obrađuje standardnim parserima

XML elementi

- XML uvek počinje zaglavljem; ono može biti minimalno, npr.

`<?xml version="1.0"?>`

a može da sadrži i druge informacije, npr.
o kodiranju

`<?xml version="1.0" encoding="UTF-8"
standalone="yes" ?>`

- Tag u kome se ne nalazi nikakav sadržaj je prazan tag, npr. `<flag/>`
- Komentari se pišu ovako: `<! KOMENTAR>`

XML i Java

- U Java SDK ugrađena su dva XML parsera: **DOM** i **SAX**; oba imaju prednosti i mane
- U verziji 1.6 uveden je **StaX** API (Streaming API for XML)
- DOM, SAX i StaX se zajednički zovu **JAXP** (Java API for XML Processing)
- Postoje i biblioteke koje nisu deo jezika, npr. **JDOM** koja olakšava rad sa DOM modelom

XML i Java

- U Javi 1.5 u paketu javax.xml.xpath uveden je **XPath** (XML Path Language)
- Java Architecture for XML Binding (**JAXB**) je Java API koji definiše kako se objekti konvertuju u XML i obrnuto
 - Na njega ćemo se vratiti u lekciji o Web servisima

Streaming i DOM

- Postoje dva modela rada sa XML podacima: **streaming i document object model (DOM)**
- U DOM modelu u memoriji se prave objekti koji predstavljaju celokupno stablo dokumenta
- U streaming modelu XML podaci se parsiraju serijski tokom izvršavanja aplikacije, često u realnom vremenu; ovaj model je efikasniji, ali u određenom trenutku pristupa samo delu podataka, a ne celom stablu

DOM

- DOM je Java XML Parser koji se najlakše koristi
- Parsira čitav XML dokument, učitava ga u memoriju, modelira tako da omogućuje lako kretanje kroz čvorove
- DOM parser je spor i zauzima mnogo memorije ako XML dokument sa kojim radi sadrži mnogo podataka

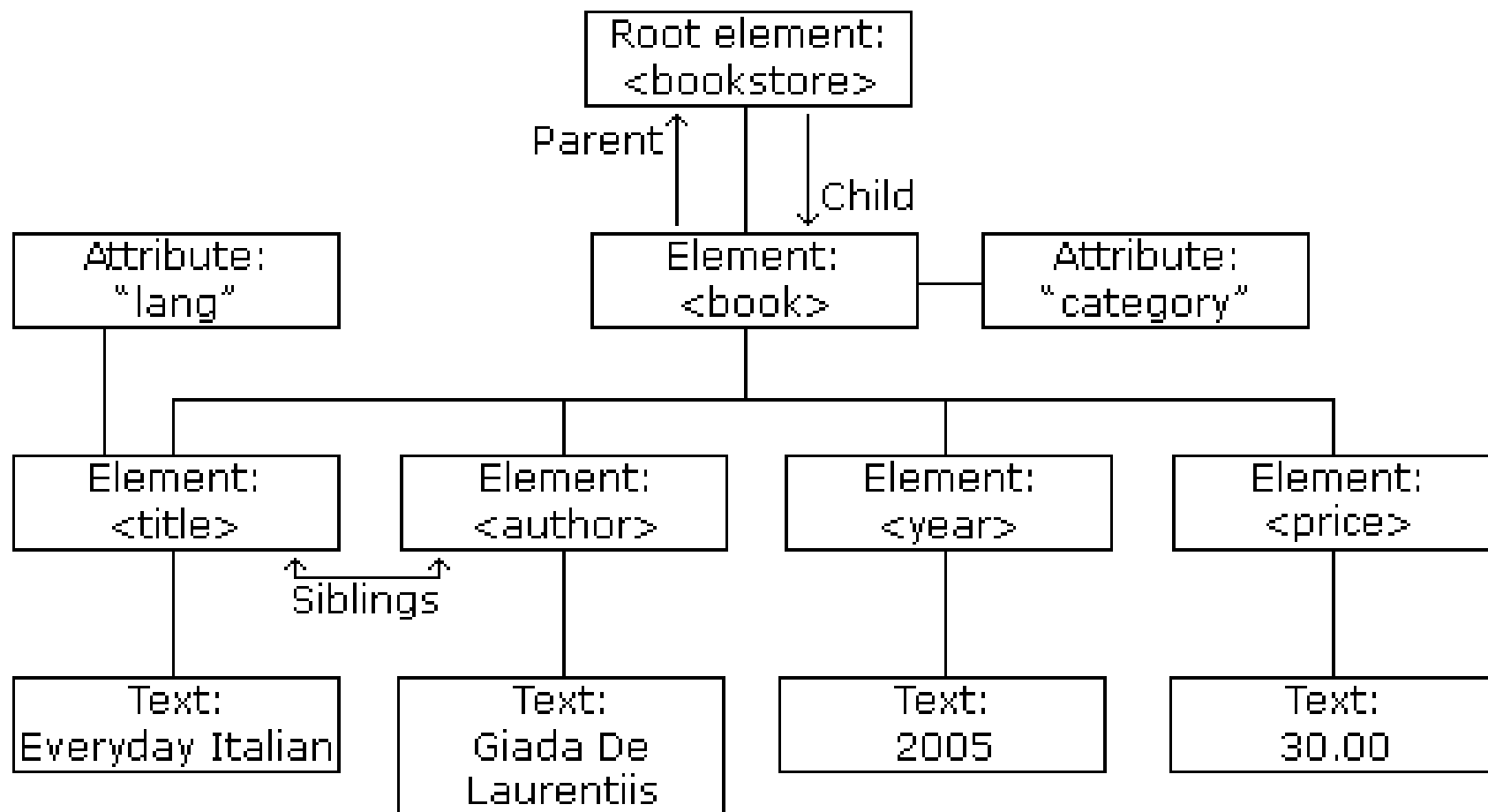
DOM čvorovi

- U DOM modelu, sve u XML dokumentu je čvor (**node**)
 - Ceo XML fajl je document node
 - Svaki XML element je element node
 - Tekst u XML elementima je text node
 - Svaki atribut je attribute node
 - Komentar je comment node

XML fajl: primer

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<bookstore>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

DOM stablo



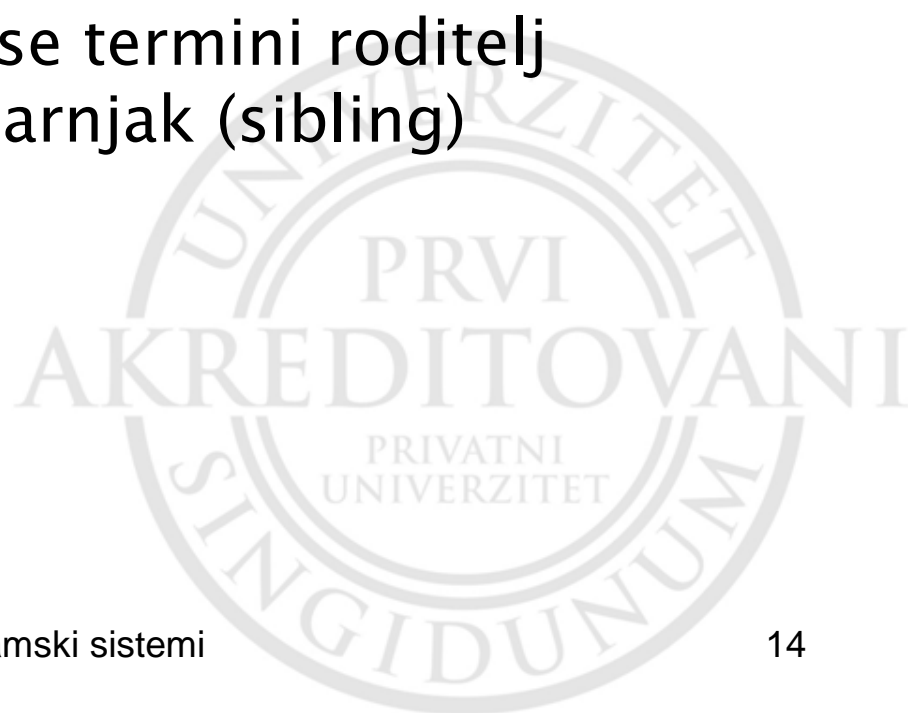
DOM stablo

- DOM model je nezavisan od programskog jezika (može se koristiti i u jezicima C i Perl), ali to znači da ne koristi prednosti objektno orijentisanog pristupa
- XML DOM vidi XML document kao stablo sa čvorovima (node tree)



DOM stablo

- Svi čvorovi stabla su u međusobnoj vezi: stablo počinje u korenskom čvoru i grana se do tekstualnih čvorova koji se nalaze na dnu stabla
- Čvorovi u stablu imaju hijerarhiju
 - Za opis relacija koriste se termini roditelj (parent), dete (child) i parnjak (sibling)



DOM stablo

- Roditeljski čvorovi imaju decu
- Deca na istom nivou su parnjaci (siblings)
- Čvor na vrhu je korenski čvor (root)
- Svaki čvor osim korenskog ima tačno jedan roditeljski čvor
- Čvor može da ima proizvoljan broj dece
- List je čvor bez dece
- Parnjaci su čvorovi koji imaju istog roditelja

DOM stablo

- Pošto su XML podaci strukturirani u obliku stabla, kroz njih se može kretati bez poznavanja stvarne strukture stabla i tipova podataka koji se u njemu nalaze



Streaming: push i pull parsiranje

- Streaming **pull parsiranje** je model u kome klijentska aplikacija poziva metode XML parsera samo kada su joj XML podaci potrebni
 - Klijent dobija (izvlači) XML podatke kada ih eksplicitno zatraži (StaX)
- Streaming **push parsiranje** je model u kome XML parser šalje XML data klijentu čim naiđe na njih
 - parser šalje podatke bez obzira da li je klijent u tom trenutku spreman da ih koristi (SAX)

SAX (Simple API for XML)

- SAX parser za razliku od DOM parsera ne učitava XML dokument u memoriju niti pravi predstavu XML dokumenta u obliku objekta
- Umesto toga, SAX parser koristi callback funkciju `org.xml.sax.helpers.DefaultHandler` da bi obavestio klijente o strukturi XML dokumenta (Observer design pattern)
- Zasnovan na dva interfejsa
 - `XMLReader` (parser)
 - `ContentHandler` (prima podatke iz parsera)

StaX

- StaX API ima dva ogranka:
 - **Cursor API:** kursor pomoću koga se kroz XML dokument može kretati od početka do kraja. Kursor pokazuje na jednu stavku i uvek se kreće samo napred, jednu po jednu stavku (kao SAX).
 - **Event Iterator API:** predstavlja XML dokument kao skup diskretnih događaja koje izvlači aplikacija u redosledu čitanja u dokumentu.
- Za parsiranje XML dokumenata može se koristiti bilo koji od njih

JAXP: poređenje

	StaX	SAX	DOM
Tip API-ja	Pull, streaming	Push, streaming	Stablo u memoriji
Lakoća korišćenja	lako	srednje	lako
Podrška za XPath	Ne	Ne	Da
Efikasnost	Dobra	Dobra	promenljiva
Parsiranje XML-a	Da	Da	Da
Kreiranje XML-a	Da	Ne	Da



XML i klasa Properties

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd>
<properties> <entry key="jdbc.username">student</entry>
               <entry key="jdbc.password">student</entry>
```



XML i klasa Properties

```
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Properties;

public class XMLToProperties
{
    public static void main(String[] args) throws IOException
    {
        Properties props = new Properties();
        InputStream is = new FileInputStream("C:db.xml");
        props.loadFromXML(is);
        props.list(System.out);
    }
}
```

XML i klasa Properties

```
import java.io.*;
import java.util.Properties;

public class XMLToProperties
{
    public static void main(String[] args) throws IOException
    {
        Properties props = new Properties();
        props.setProperty("email.support", "donot-spam-me@nospam.com");
        OutputStream os = new FileOutputStream("F:/email-configuration.xml");
        props.storeToXML(os, "email", "UTF-8");
    }
}
```

XPath

- Navigacija kroz strukturu XML dokumenta da bi se pronašao određeni čvor je veoma osetljiva, a kod podložan greškama
- Potreban je način za zadavanje čvorova koji su programu potrebni bez eksplicitnog navođenja kako će program stići do njih
- Xpath je deklarativni jezik za pronalaženje čvorova u XML dokumentu
- Uveden u verziji Jave 1.5 u paketu `javax.xml.xpath`

XPath

- XPath location path navodi čvorove iz dokumenta koji se traže, ali ne govori ništa o tome koji algoritam će biti korišćen za njihovo pronalaženje
- Korišćenje Xpatha u Java programu slično je korišćenju SQL-a; umesto izdvajanja informacija iz baze podataka, one se pronalaze u XML dokumentu

Xpath: model podataka

- root node
 - dokument
- element node
- attribute node
- text node
 - Neprekinuti tekst između tagova, komentara i naredbi za obradu
- comment node
- processing instruction node
- namespace node

Razlike između DOM i XPath modela podataka

- U Xpath modelu samo atributi, elementi, naredbe za obradu i imenici imaju imena, podeljena na lokalni deo i URI imenika (namespace)
- U XPath modelu, vrednost korenskog čvora je nadovezan string svih vrednosti njegove tekstualne dece, a ne null kao u DOM-u

Xpath: Biranje čvorova

Izraz	Opis
<i>"imecvora"</i>	Bira sve čvorove sa imenom <i>"imecvora"</i>
/	Bira sve počev od korenskog čvora
//	Bira čvorove u dokumentu počev od tekućeg čvora koji odgovaraju kriterijumu, bez obzira gde se nalaze u dokumentu
.	Bira tekući čvor
..	Bira roditeljski čvor tekućeg čvora
@	Bira atribut

Xpath: primeri izraza

XPath izraz	Opis
"bookstore"	Svi čvorovi sa imenom "bookstore"
/bookstore	Korenski element bookstore Napomena: Ako putanja počinje kosom crtom(/), ona uvek predstavlja apsolutnu putanju do elementa!
bookstore/book	Svi elementi book koji su deca elementa bookstore
//book	Svi elementi book, bez obzira gde se oni nalaze u dokumentu
bookstore//book	Svi elementi book koji su deca elementa bookstore, bez obzira gde se nalaze u dokumentu
//@lang	Svi atributi koji se zovu lang

Xpath: predikati

- Predikati se koriste za pronalaženje specifičnog čvora ili nekog čvora koji sadrži određenu vrednost
- Predikati se uvek nalaze u uglastim zagradama



Xpath: primer predikata

XPath izraz	Opis
/bookstore/book[1]	Prvi element book koji je dete elementa bookstore
/bookstore/book[last()]	Poslednji element book koji je dete elementa bookstore
/bookstore/book[last()-1]	Pretposlednji element book koji je dete elementa bookstore
/bookstore/book[position()<3]	Prva dva elementa book koji su deca elementa bookstore
//title[@lang]	Svi elementi title koji imaju atribut lang
//title[@lang='eng']	Svi elementi title koji imaju atribut lang sa vrednošću eng
/bookstore/book[price>35.00]	Svi elementi book, deca elementa bookstore, koji imaju element price sa vrednošću većom od 35

Xpath: džoker znaci

XPath izraz	Opis
*	Bilo koji element
@*	Bilo koji atribut
node()	Čvor bilo koje vrste
/bookstore/*	Sva deca elementa bookstore
//*	Svi elementi u dokumentu
//title[@*]	Svi elementi title koji imaju bilo kakav atribut

Xpath i Java: primer

```
import java.io.*;
import org.w3c.dom.*;
import org.xml.sax.SAXException;
import javax.xml.parsers.*;
import javax.xml.xpath.*;

public class XPathExample {
    public static void main(String[] args)
        throws ParserConfigurationException, SAXException,
            IOException, XPathExpressionException {

        DocumentBuilderFactory domFactory = DocumentBuilderFactory.newInstance();
        domFactory.setNamespaceAware(true);
        DocumentBuilder builder = domFactory.newDocumentBuilder();
        Document doc = builder.parse("book.xml");
        XPath xpath = XPathFactory.newInstance().newXPath();
        XPathExpression expr = xpath.compile("/bookstore/book");
        Object result = expr.evaluate(doc, XPathConstants.NODESET);
        NodeList nodes = (NodeList) result;
        for (int i = 0; i < nodes.getLength(); i++) {
            System.out.println(nodes.item(i).getNodeValue());
        }
    }
}
```

Xpath: tipovi podataka

- XPath ima samo četiri osnovna tipa podataka: node-set, number, boolean i string
- Metod evaluate() vraća Object; ono što stvarno vraća zavisi od rezultata XPath izraza, kao i od tipa koji zadajemo

XPath tip	Java tip
number	java.lang.Double
boolean	java.lang.Boolean
string	java.lang.String
node-set	org.w3c.dom.NodeList

Povratni tip metode evaluate()

- Kada se XPath izraz proverava u Javi, drugi argument zadaje povratni tip koji želimo
- Ima pet mogućnosti, definisanih kao nabrojane konstante u klasi XPathConstants
 - XPathConstants.NODESET
 - XPathConstants.BOOLEAN
 - XPathConstants.NUMBER
 - XPathConstants.STRING
 - XPathConstants.NODE

XPathConstants.NODE

- XPathConstants.NODE ne odgovara XPath tipu, već se koristi kada znamo da će izraz vratiti samo jedan čvor
- Ako XPath izraz vrati više od jednog čvora kada je zadat povratni tip XPathConstants.NODE, onda evaluate() vraća prvi čvor u redosledu dokument
- Ako ne nađe ništa, evaluate() vraća null
- Ako zahtevana konverzija ne može da se izvrši, baca se izuzetak XPathException