

XML i tehnologije:

računalna obrada XML dokumenta

Tehnologije elektroničkog poslovanja

54 od 94

SAX i DOM



- ♦ 2 osnovne paradigme za izgradnju aplikacijskih programskih sučelja (API) za XML:
 - SAX (Simple API for XML)
 - DOM (Document Object Model)

Tehnologije elektroničkog poslovanja

SAX – Simple API for XML



 sekvencijalno čitanje od početka do kraja dokumenta; segmenti se obrađuju redoslijedom njihovog pojavljivanja

<person>

<name>Nikola Tesla</name> <email>tesla@tesla.com</email> </person>

Ovisno o duljini spremnika (buffer) koja ovisi o implementaciji, moguće je da SAX registrira više uzastopnih tekstualnih čvorova

SAX:

- 1. Počinje XML dokument
- 2. Počinje element person
- 3. Tekst novi redak
- Počinje element name
- Tekst Nikola Tesla
- Završava element name
- Tekst novi redak
- 8. Počinje element email
- Tekst tesla@tesla.com
- 10. Završava element email
- 11. Tekst novi redak
- 12. Završava element person
- 13. Završava XML dokument

Tehnologije elektroničkog poslovanja

56 od 94

SAX – Simple API for XML

- ne postoji službena specifikacija
- ♦ standard *de facto* je izvedba u jeziku Java
- preporučuje se programski paket Xerces tvrtke Apache (otvoreni kod, javno dostupan, ugrađen u mnogim sustavima za razvoj programa)

org.apache.html.dom rg.apache.wml org.apache.wml.dom org.apache.xerces.dom org.apache.xerces.dom.events org.apache.xerces.domx org.apache.xerces.framework rg.apache.xerces.jaxp org.apache.xerces.msg org.apache.xerces.parsers org.apache.xerces.readers org.apache.xerces.utils org.apache.xerces.utils.regex org.apache.xerces.validators.commor org.apache.xerces.validators.datatype org.apache.xerces.validators.dtd org.apache.xerces.validators.schema org.apache.xerces.validators.schema.identit org.apache.xml.serialize

Packages

org.w3c.dom org.w3c.dom.events org.w3c.dom.html org.w3c.dom.ranges org.w3c.dom.traversal

org.xml.sax.ext

Tehnologije elektroničkog poslovanja

2

DOM – Document Object Model



- platforma za obradu HTML i XML dokumenata
- standard organizacije W3C, postoje 3 razine
 - Za 3. razinu vrijedi trodijelna specifikacija
 - Document Object Model (DOM) Level 3 Core Specification, Version 1.0 (W3C Recommendation 07 April 2004) http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407
 - Document Object Model (DOM) Level 3 Load and Save Specification, Version 1.0 (W3C Recommendation 07 April 2004) https://www.w3.org/TR/2004/REC-DOM-Level-3-LS-20040407
 - Document Object Model (DOM) Level 3 Validation Specification, Version 1.0 (W3C Recommendation 27 January 2004) http://www.w3.org/TR/2004/REC-DOM-Level-3-Val-20040127
 - Najnovije važeće (uključujući i buduće) verzije specifikacija nalaze se na adresama http://www.w3.org/TR/DOM-Level-3-LS http://www.w3.org/TR/DOM-Level-3-Val

Tehnologije elektroničkog poslovanja

58 od 94

DOM – Document Object Model



- DOM, slično kao XPath, promatra XML dokument kao stablo čvorova
 - 12 vrsta čvorova:

Naziv vrste čvora	Postoji u XPathu
korijenski čvor (Document)	DA
fragment (DocumentFragment)	NE
tip dokumenta (DocumentType)	NE
entitet (Entity)	NE
poveznica s entitetom (EntityReference)	NE
element (Element)	DA
atribut (Attr)	DA
procesorska instrukcija (ProcessingInstruction)	DA
komentar (Comment)	DA
tekst (Text)	DA
CDATASection	NE
zabilješka (Notation)	NE NE
rennologije elektronickog poslovanja	

- entitet i poveznica s entitetom
 - prije pojave XML Scheme "posebna" slova nacionalnih alfabeta definirala su se kao entiteti, npr. é u riječi Montréal

บช บน ฮ4

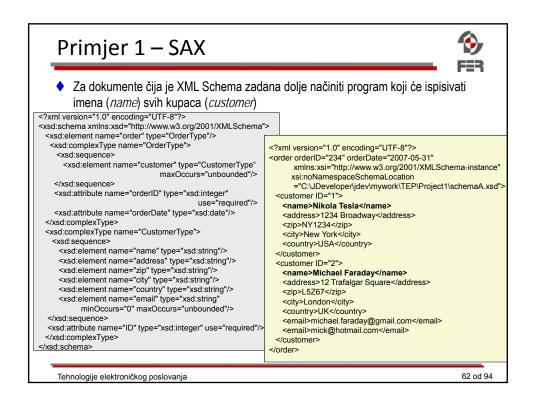
Packages DOM – programske izvedbe org.apache.html.dom org.apache.wml org.apache.wml.dom org.apache.xerces.dom ◆ preporučuje se ranije spomenuti org.apache.xerces.dom.events programski paket Xerces org.apache.xerces.domx org.apache.xerces.frameworl org.apache.xerces.jaxp org.apache.xerces.msg org.apache.xerces.parsers org.apache.xerces.readers org.apache.xerces.utils org.apache.xerces.utils.regex org.apache.xerces.validators.common org.apache.xerces.validators.datatype org.apache.xerces.validators.dtd org.apache.xerces.validators.schema org.apache.xerces.validators.schema.identity org.apache.xml.serialize org.w3c.dom org.w3c.dom.events org.w3c.dom.html org.w3c.dom.traversal org.xml.sax org.xml.sax.ext Tehnologije elektroničkog poslovanja

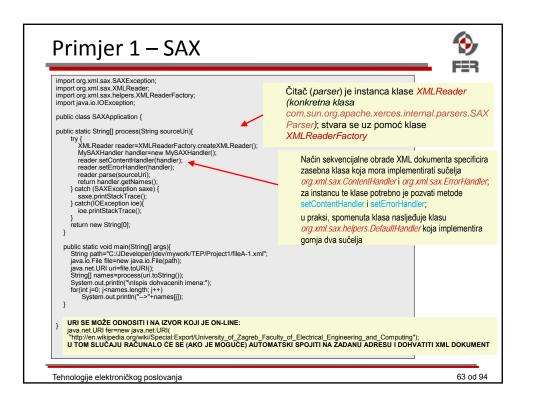
SAX ili DOM

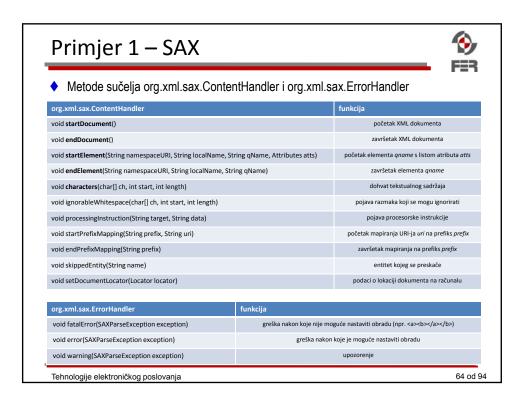


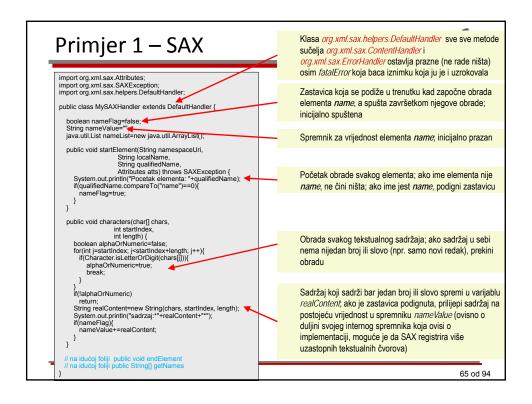
- SAX je brži i lako se postigne da se ne gubi vrijeme na nevažne dijelove XML dokumenata, DOM zahtijeva da se učita cijelo stablo
- DOM omogućuje kretanje po stablu s čvora na čvor i laganu malipulaciju čvorovima (dodavanje novih čvorova, brisanje čvorova)

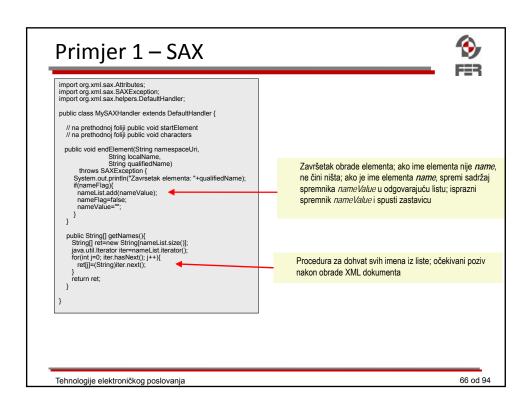
Tehnologije elektroničkog poslovanja

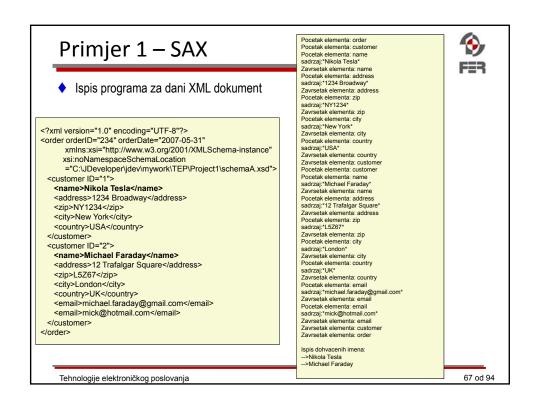


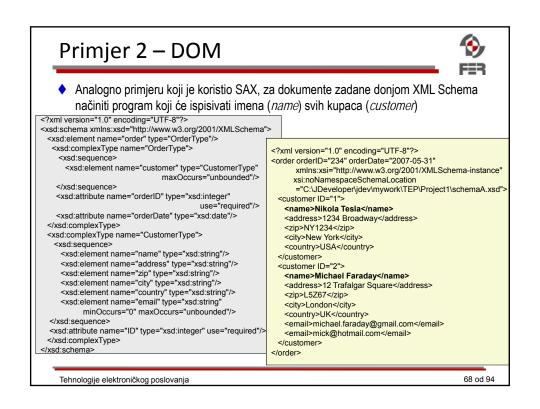


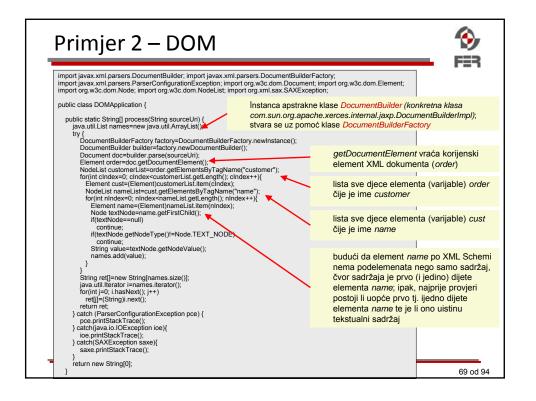












Primjer 2 – DOM



```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Document;
import org.w3c.dom.Node:
import org.w3c.dom.Nodel.ist;
import org.w3c.AXException;

public class DOMApplication {

//na prethodnoj foliji public static String[] process(String sourceUri)

public static void main(String[] args){

// obradjujem datoteku fileA-1.xml
String path="C:JJDeveloper/Idev/mywork/TEP/Project1/fileA-1.xml";
java.io.File file=ney java.io.File(path);
java.not.File file=ney java.io.File(path);
java.not.File file=ney java.io.File(path);
String[] names=process(url.oString());
System.out.println("Ispis dohvacenih imena:");
for(int j=0; j-names.length; j++)
System.out.println(""->"+names[jj));
}
```

Ispis dohvacenih imena -->Nikola Tesla -->Michael Faraday

Tehnologije elektroničkog poslovanja

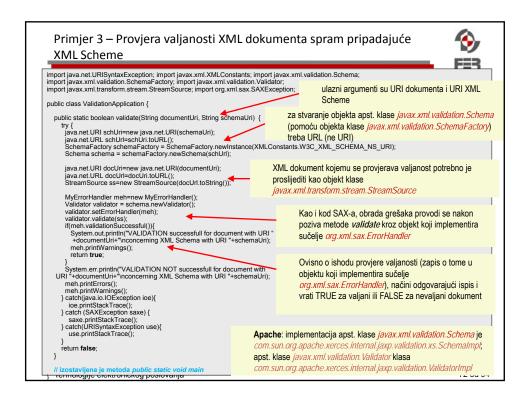
70 od 94

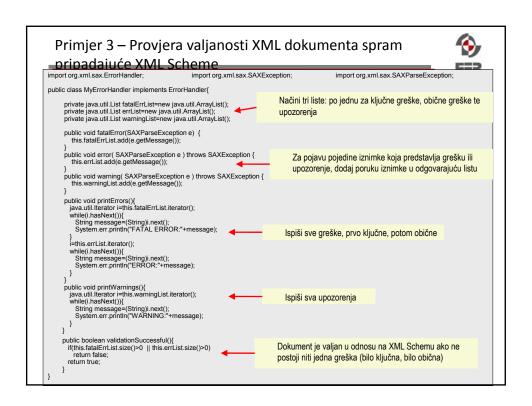
Primjer 3 – Provjera valjanosti XML dokumenta spram pripadajuće XML Scheme



- programski paket Xerces nudi i podršku za provjeru valjanosti XML dokumenta
- U nadolazećem primjeru za objekt klase javax.xml.validation.Validator poziva se metoda setErrorHandler(org.xml.sax.ErrorHandler h)
 - Objekt h je instanca klase koja implementira sučelje org.xml.sax.ErrorHandler, najčešće naslijeđuje ranije prikazanu klasu org.xml.sax.helpers.DefaultHandler

Tehnologije elektroničkog poslovanja





<?xml version="1.0" encoding="UTF-8"?> Primjer 3 <order orderID="234" orderDate="2007-05-31"</pre> xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation ="C:\JDeveloper\jdev\mywork\TEP\Project1\schemaA.xsd" <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema <customer ID="1"> <xsd:element name="order" type="OrderType"/> <name>Nikola Tesla</name> <xsd:complexType name="OrderType"> <address>1234 Broadway</address> <xsd:sequence> <zip>NY1234</zip> <xsd:element name="customer" type="CustomerType" <city>New York</city> maxOccurs="unbounded"/> <country>USA</country> </xsd:sequence> </customer> <xsd:attribute name="orderID" type="xsd:integer"</pre> customer ID="2"> use="required"/> <name>Michael Faraday</name> <xsd:attribute name="orderDate" type="xsd:date"/> <address>12 Trafalgar Square</address> </xsd:complexType> <zip>L5Z67</zip> <xsd:complexType name="CustomerType"> <city>London</city> <xsd:sequence> <country>UK</country> <xsd:sequences <xsd:element name="name" type="xsd:string"/> <xsd:element name="address" type="xsd:string"/> <email>michael.faraday@gmail.com</email> <email>mick@hotmail.com</email> <xsd:element name="zip" type="xsd:string"/> <xsd:element name="city" type="xsd:string"/> </customer> </order> <xsd:element name="country" type="xsd:string"/> <xsd:element name="email" type="xsd:string"</p> public static void main(String[] args)(// obradjujem datoteku fileA-1.xml String docPath="C:JJDeveloperi/dev/mywork/TEP/Project1/fileA-1.xml"; java.io.File docFile=new java.io.File(docPath); java.net.URl docUri=docFile.toURl(); String schPathA="C:JJDeveloperi/dev/mywork/TEP/Project1/schemaA.xsd"; java.ie.tile.gocFileA-peri/dev/mywork/TEP/Project1/schemaA.xsd"; java.ie.tile.gocFileA-peri/dev/mywork/TEP/Project1/schemaA.xsd"; minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attribute name="ID" type="xsd:integer" use="required"/> </xsd:complexType> </xsd:schema> java.io.File schFileA-new java.io.File(schPathA); java.net.URI schUriA=schFileA.toURI(); validate(docUri.toString(), schUriA.toString()); Prikazani dokument odgovara danoj XML Schemi VALIDATION successful for document with URI file:/C:/JDeveloper/jdev/mywork/TEP/Project1/fileA-1.xml concerning XML Schema with URI file:/C:/JDeveloper/jdev/mywork/TEP/Project1/schemaA.xsd Tehnologije elektroničkog poslo

<?xml version="1.0" encoding="UTF-8"?> Primjer 3 <order orderID="234" orderDate="2007-05-31"</p> xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation <?xml version="1.0" encoding="UTF-8"?> ="C:\JDeveloper\jdev\mywork\TEP\Project1\schemaA.xsd"> <customer ID="1"> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema <name>Nikola Tesla</name> <xsd:element name="order" type="OrderType"/> <xsd:complexType name="OrderType"> <address>1234 Broadway</address> <zip>NY1234</zip> <xsd:sequence> <city>New York</city> <xsd:element name="customer" type="CustomerType"/> <country>USA</country> </xsd:sequence> </customer> <xsd:attribute name="orderID" type="xsd:integer"</p> <customer ID="2"> <name>Michael Faraday</name> <xsd:attribute name="orderDate" type="xsd:date"/> <address>12 Trafalgar Square</address> </xsd:complexType> <zip>L5Z67</zip> <xsd:complexType name="CustomerType"> <city>London</city> <country>UK</country> <xsd:element name="name" type="xsd:string"/> <xsd:element name="address" type="xsd:string"/> <email>michael.faraday@gmail.com <xsd:element name="zip" type="xsd:string"/> <email>mick@hotmail.com</email> </customer> <xsd:element name="city" type="xsd:string"/> </order> <xsd:element name="country" type="xsd:string"/> <xsd:element name="email" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/> public static void main(String[] args) in Static Volu Tilani, Guingji argsh // obradjujem datoteku fileA-1.xml String docPath="C:/JDeveloper/jdev/mywork/TEP/Project1/fileA-1.xml"; java.io.File docFile=new java.io.File(docPath); </xsd:sequence> <xsd:attribute name="ID" type="xsd:integer" use="required"/> </xsd:complexType> java.net.URI docUri=docFile.toURI(); String schPathB="C:/JDeveloper/jdevlmywork/TEP/Project1/schemaB.xsd" java.io.File schFileB=new java.io.File(schPathB); java.net.URI schUnB=schFileB.toURI(); Dokument ne odgovara danoj XML Schemi validate(docUri.toString(), schUriB.toString()); (element *order* mora imati točno jedan VALIDATION NOT successful for document with URI file:/C:/JDeveloper/jdev/mywork/TEP/Project1/fileA-1.xn concerning XML Schema with URI file:/C:/JDeveloper/jdev/mywork/TEP/Project1/schemaB.xsd ERROR:cvc-complex-type.2.4.d: Invalid content was found starting with element 'customer'. No child element is expected at this point. podelement customer) Tehnologije elektroničkog p



XSLT (Extensible Stylesheet Language Transformations)

Tehnologije elektroničkog poslovanja

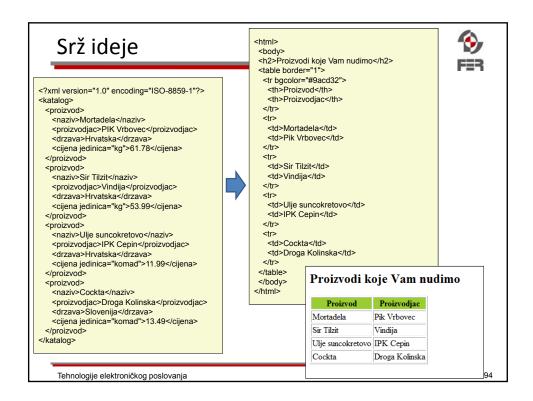
76 od 94

Obrada XML dokumenata



- Primarna područje upotrebe XML-a
 - 1. razmjena sadržaja putem mreže
 - početni format →XML → početni ili neki drugi format
 - ⇒ baza podataka → XML → druga baza podataka
 - baza podataka → XML → ERP
 - ► ERP → XML → drugi ERP
 - predložak XML dokumenta određen standardom ili bilateralnim/multilateralnim dogovorom
 - 2. XML kao posrednik pri transformaciji sadržaja u HTML
 - početni format →XML → (X)HTML
 - baza podataka → XML → (X)HTML
 - moguće je pretvaranje izvesti direktno, bez XML-a, XML postaje nužan ako je potrebno u jedinstveni format uklopiti više različitih izvora

Tehnologije elektroničkog poslovanja



Srž ideje



- načiniti transformacijski predložak u posebnoj datoteci
- ako se promijeni sadržaj izvorne datoteke (npr. dodavanje novog proizvoda), potrebno je da se promjena sadržaja odrazi i na izlaznu XML (XHTML) datoteku bez promjene sadžaja samog predloška

Tehnologije elektroničkog poslovanja

Skupina jezika XSL



- XSL (Extensible Stylesheet Language) skupina jezika za rad s XML dokumentima, razvoj pod okriljem W3C
 - XSL Transformations (XSLT)
 - jezik zasnovan na XML-u za transformiranje XML dokumenata
 - XML u XML (XHTML)
 - XSL Formatting Objects (XSL-FO)
 - jezik zasnovan na XML-u za specificiranje vizualnog formata (XML) dokumenata
 - XML u PDF/PS
 - XML Path Language (XPath)
 - jezik koji NIJE zasnovan na XML-u
 - za dohvat pojedinih dijelova XML dokumenta
 - koristi ga XSLT

Tehnologije elektroničkog poslovanja

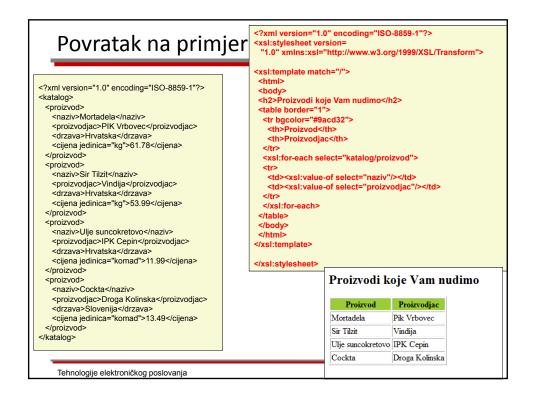
80 od 94

XSLT - specifikacija



- ♦ 3 verzije: 1.0, 1.1 i 2.0
 - XSL Transformations (XSLT) Version 1.0, W3C Recommendation 16 November 1999 http://www.w3.org/TR/1999/REC-xslt-19991116
 - XSL Transformations (XSLT) Version 1.1, W3C Working Draft 24 August 2001 http://www.w3.org/TR/2001/WD-xslt11-20010824/
 - XSL Transformations (XSLT) Version 2.0, W3C Recommendation 23 January 2007 http://www.w3.org/TR/2007/REC-xslt20-20070123/
- verzija 1.1 ostala working draft
- verzija 2.0 jest preporuka, ali je ne podržavaju preglednici Interneta
- u praksi se koristi samo verzija 1.0
 - Mozilla Firefox: uključuje XSLT od verzije 3
 - MS internet Explorer: uključuje XSLT od verzije 6
 - Opera: uključuje XSLT od verzije 9
 - Google Chrome: uključuje XSLT od verzije 1
 - Apple Safari: uključuje XSLT od verzije 3

Tehnologije elektroničkog poslovanja



Element stylesheet



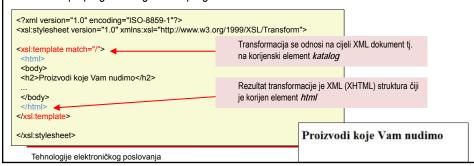
- svaki XSLT dokument ima korijenski element stylesheet
 - sadrži poziv na standardni imenik za XSLT, http://www.w3.org/1999/XSL/Transform
- element stylesheet ima 1 atribut
 - version (opisuje verziju predloška; vrijednost "1.0")
- element stylesheet ima jedan ili više podelemenata template

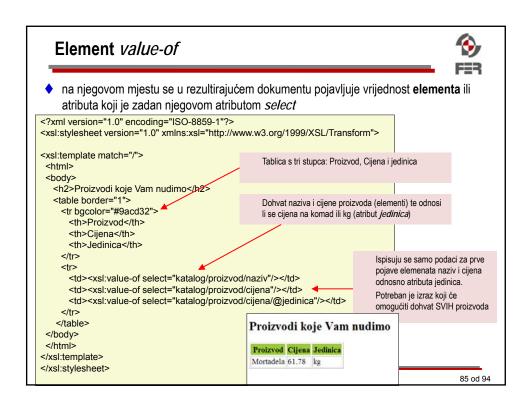
Tehnologije elektroničkog poslovanja

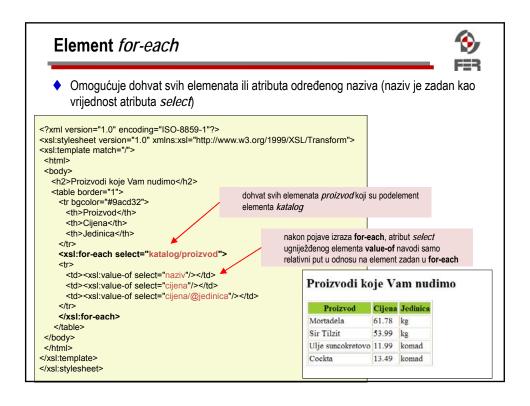
Element template

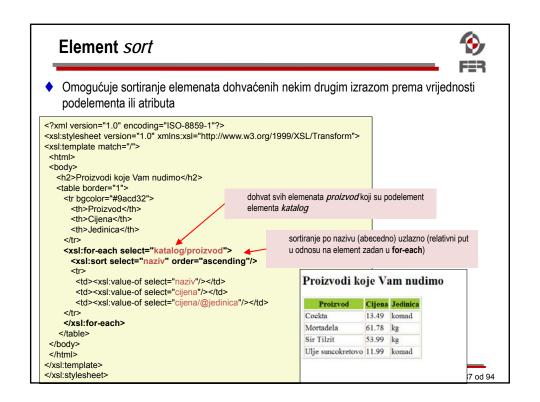


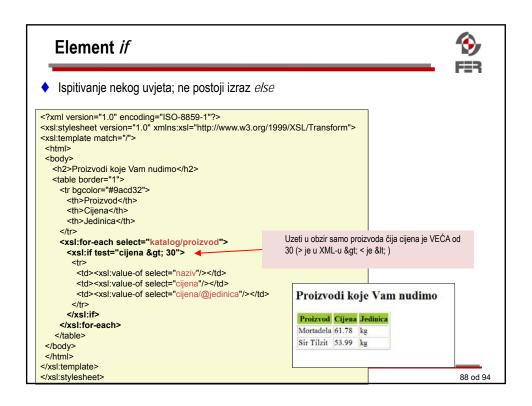
- svaki element template odnosi se na dio dokumenta određen XPath izrazom koji je vrijednost pripadnog atributa match
 - dijete elementa *template* je korijen rezultata transformacije
 - u pravilu, ako element stylesheet sadrži samo jedan podelement template, on se odnosi na cijeli dokument (match="/")
 - stylesheet može sadržavati više podelemenata template; u tom slučaju jedna transformacija u sebi uključuje poziv druge transformacije, analogno pozivanju potprograma u glavnom programu

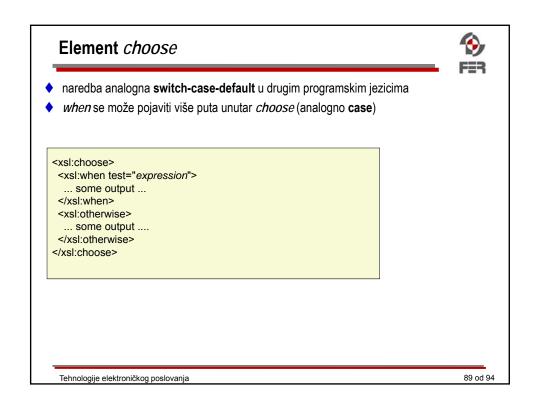


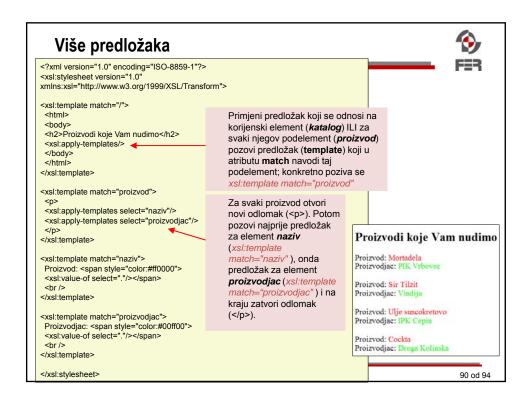


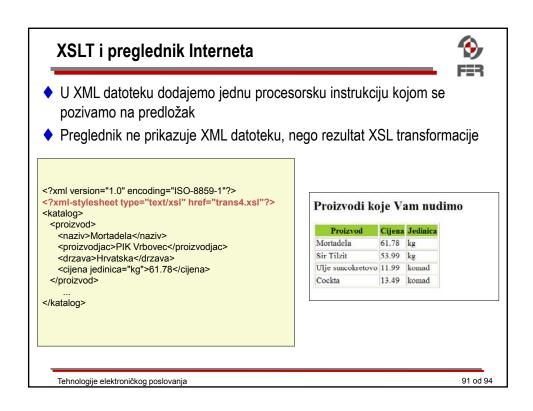










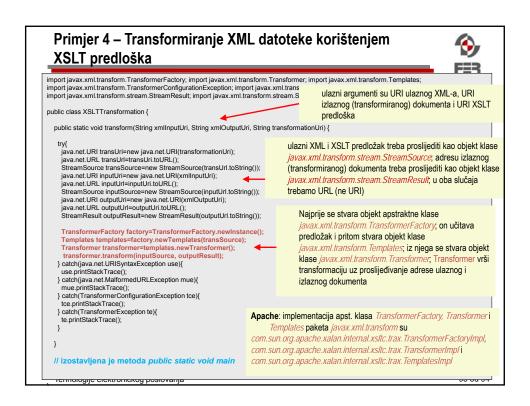


Programska izvedba



 preporučuje se programski paket Xalan tvrtke Apache (otvoreni kod, javno dostupan, ugrađen u mnogim sustavima za razvoj programa)

Tehnologije elektroničkog poslovanja



Primjer 4



public static void main(String[] args){
 String inputPath="C:/Documents and Settings/mbanek.ZOEM/My Documents/Nastava/TEP/xsit/katalog.xml";
 java.io.File inputFile=new java.io.File(inputPath);
 java.net.URI inputUri=inputFile.toURI();

String stylesheetPath="C:/Documents and Settings/mbanek.ZOEM/My Documents/Nastava/TEP/xslt/trans1.xsl"; java.io.File styleshhetFile=new java.io.File(stylesheetPath); java.net.URI stylesheetUri=styleshhetFile.toURI();

 $String\ outputPath="C:/Documents\ and\ Settings/mbanek. ZOEM/My\ Documents/Nastava/TEP/xslt/izlaz.html"; java.io.File\ outputFile=new\ java.io.File(outputPath); java.net.URI\ outputUri=outputFile.toURI(); \\$

transform(inputUri.toString(), outputUri.toString(), stylesheetUri.toString());

Tehnologije elektroničkog poslovanja