

Integracija softverskih tehnologija

1

Zoran Čirović 18:14

Osnovne oblasti:

- ▶ 1. deo
 - ▶ Tehnologije:
 - ▶ XML
 - ▶ XSD, DTD
 - ▶ XPath
 - ▶ XSLT...
- ▶ Testovi (a,b,c).
- ▶ 2. deo
 - ▶ WS servisi
- ▶ Projekat.

▶ 2

Zoran Čirović 18:14

Alati koji će biti korišćeni

- ▶ 1. deo
 - ▶ XML editori
 - ▶ VisualStudio poslednja licencirana verzija
 - ▶ Pišemo (programiramo) XSLT transformacije
- ▶ 2. deo
 - ▶ VisualStudio (+ NetBeans)
 - ▶ (WS servisi i klijenti za te servise)

▶ 3

Zoran Čirović 18:14

Poželjno predznanje

- ▶ Poznavanje rada na računaru
- ▶ Poznavanje osnovnih pojmova pri radu u Internet okruženju
- ▶ Osnove HTML
- ▶ Osnove JavaScript-a, Jave ili C#
- ▶ *Bez obzira na stečeno predznanje, biće prezentovano gradivo potrebno za kurs u celini!!!*

▶ 4

Zoran Čirović 18:14

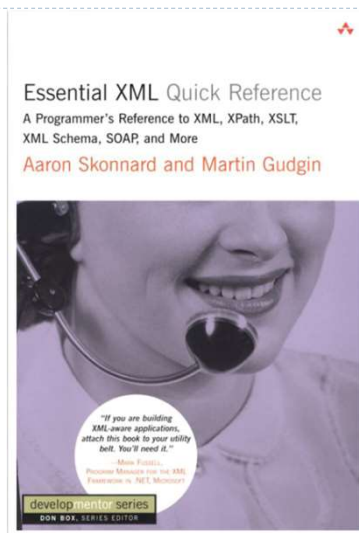
Literatura

- ▶ **Materijali koji će biti postavljeni na stranici predmeta biće dovoljni za polaganje ispita sa najvećom ocenom...uz preduslove (predavanja, vezbe, ...)**
- ▶ Materijali će biti u vidu pdf skripti/prezentacija, a sadržaće i dodatne fajlove koji će pratiti odgovarajuće primere.
- ▶ Poželjno je koristiti i dodatnu literaturu:

▶ 5

Zoran Čirović 18:14

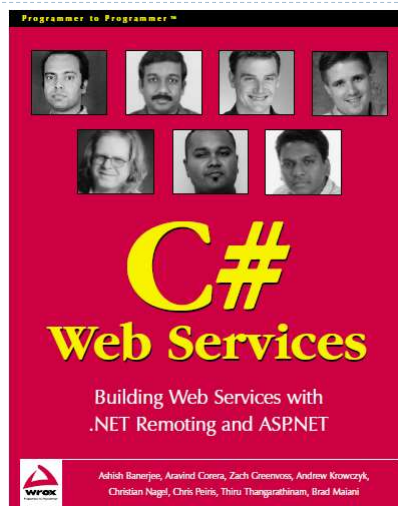
Dodatna literatura: I deo



▶ 6

Zoran Čirović 18:14

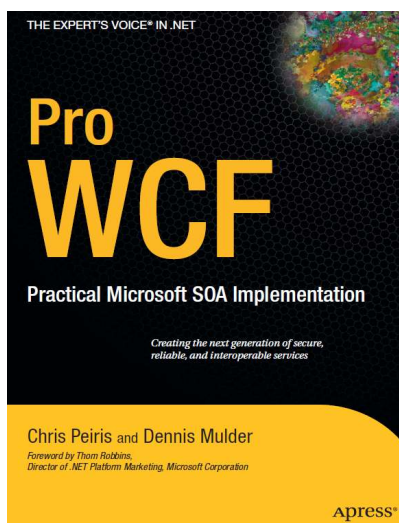
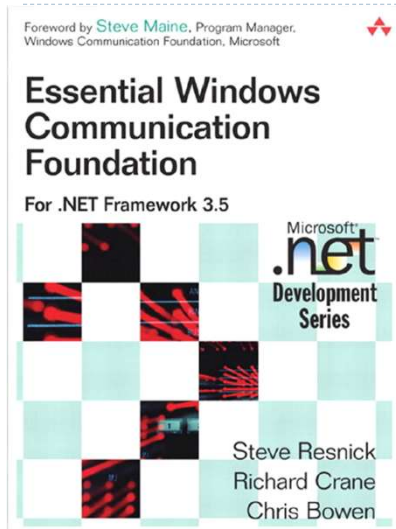
Dodatna literatura: II deo



► 7

Zoran Ćirović 18:14

Dodatna literatura: III deo



► 8

Zoran Ćirović 18:14

Polaganje ispita

- ▶ Opcija 1. Ispit u celosti
- ▶ Opcija 2. Ispit iz delova.
- ▶ Parcijalno mogu polagati samo studenti koji su redovni na predavanjima i vežbama (na više od 80%)
 - ▶ Abc testovi
 - ▶ Projekat
- ▶ Vežbe se evidentiraju
 - ▶ Vežbe su obavezne u 80%

▶ 9

Zoran Čirović 18:14

Organizacija kursa

- ▶ Predavanja
 - ▶ 3 čas nedeljno
- ▶ Vežbe
 - ▶ 2 časa nedeljno
 - ▶ Vežbe počinju posle odrađenog dela koji se odnosi na XML...
 - ▶ Vežbe su uglavnom laboratorijske tj. za računarima

▶ 10

Zoran Čirović 18:14

Organizatori kursa

► Predavanja

- Prof. dr Zoran Ćirović
- Kabinet: 514
- Konsultacije se definišu terminom koji je na Web stranicam Škole

► Vežbe

- Nemanja Cvijan

► 11

Zoran Ćirović 18:14

Pitanja?

12

Zoran Ćirović 18:14

Uvod u XML tehnologije

13

Zoran Čirović 18:14

Pojam - XML?

- ▶ Skraćenica od:
- ▶ *EX*tensible
- ▶ *Markup*
- ▶ *Language*

- ▶ **XML po W3C preporukama**

▶ 14

Zoran Čirović 18:14

- ▶ **XML dokument** predstavlja podatke koji su tekstualno formatirani u skladu sa (strogim) pravilima.
- ▶ XML dokument tj. tekst sa podacima može da bude
 - ▶ smešten u datoteku na disku
 - ▶ kao poruka koja se šalje HTTP protokolom
 - ▶ kao niz znakova, tj. string u programskom jeziku
 - ▶ kao objekat u bazi podataka
 - ▶ na bilo koji drugi način koji omogućava korišćenje tekstualnih podataka
- ▶ Zbog svog formata, omogućava razmenu podataka i između nekompatibilnih sistema,
- ▶ **Nezavisan je od korišćenih hardverskih i softverskih platformi.**

▶ 15

Zoran Čirović 18:14

Svojstva XML-a - 1

- ▶ XML je danas postao *de-facto* **standard** za opis sadržaja i strukture (tekstualnih i multimedijalnih) dokumenata i razmenu dokumenata na Web-u

Markup omogućava

- specijalno značenje podataka
- koristi se **tag** za predstavljanje markup-a

Extensible

- ▶ prošiv jezik, tj. dozvoljava definisanje novih tagova
- ▶ predstavlja meta jezik koji omogućava definisanje drugih markup jezika

▶ 16

Zoran Čirović 18:14

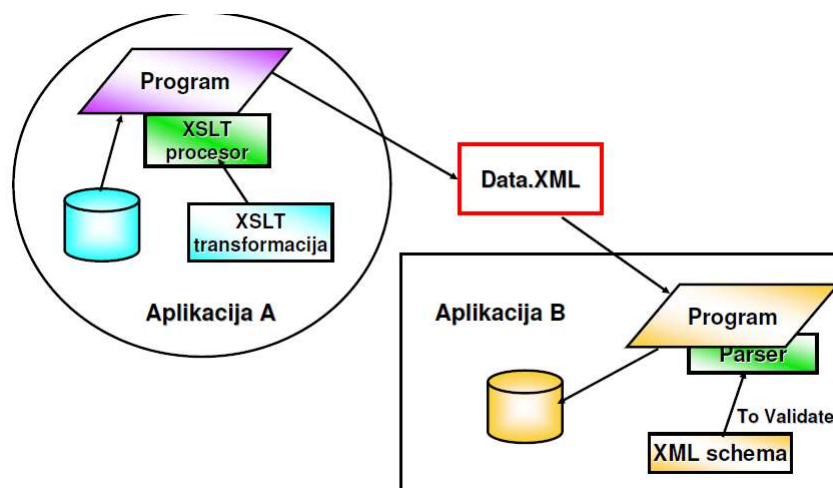
Svojstva XML-a - 2

- ▶ XML je **samoopisujući**, platformski **nezavisan** i u tekstualnom formatu. XML je veoma **pogodan kao format za razmenu podataka između heterogenih aplikacija na Web-u**.
- ▶ XML omogućava **razdvajanje struktuiranog sadržaja dokumenta od njegove prezentacije** (*Style Sheet*)
- ▶ XML je **projektovan za distribuirano okruženje**.
- ▶ XML kao format **je dovoljno formalan za mašinsko procesiranje i dovoljno razumljiv za korisnike**

▶ 17

Zoran Čirović 18:14

Primer



▶ 18

Zoran Čirović 18:14

Prateće tehnologije i pojmovi u primeni

- ▶ Procesiranje XML dokumenata:
 - ▶ XML parseri
 - ▶ transformacija XML dokumenata (XSLT jezik)
- ▶ Specifikacija logičke strukture XML dokumenata
 - ▶ DTD = *Document Type Definiton*
 - ▶ XML schema
- ▶ Upitni XML jezici
 - ▶ XPath
 - ▶ XQuery

▶ 19

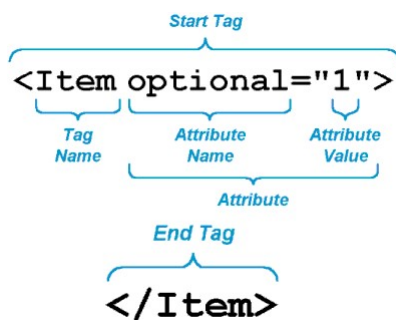
Zoran Čirović 18:14

XML struktura

20

Zoran Čirović 18:14

XML elementi - 1



- ▶ XML dokument se sastoji iz teksta oraganizovanog uz pomoć tagova u elemente

▶ 21

Zoran Čirović 18:14

XML elementi - 2

- ▶ Elementi su osnovni blokovi XML-a
- ▶ `<pozdrav> Hello XML! </pozdrav>`
- ▶ **Složeni / kontejner** element čini par tag-ova (pocetni i krajnji tag) sa sadržajem
- ▶ **Prost/prazan** element obično se za krajnji tag koristi skraćénica `/>`

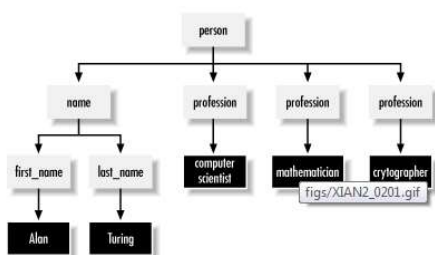
- ▶ `<poruka/>`
- ▶ `<pozdrav tekst = "Hello XML" />`

▶ 22

Zoran Čirović 18:14

Koren dokumenta

- XML dokument sadrži jedan element koji nema roditelja.
- To je prvi element i sadrži sve druge elemente.
- Taj element nazivamo korenskim (engl. *root*).



```

<person>
  <name>
    <first_name>Alan</first_name>
    <last_name>Turing</last_name>
  </name>
  <profession>computer
  scientist</profession>
  <profession>mathematician</profession>
  <job>cryptographer</job>
</person>

```

► 23

Zoran Čirović 18:14

Mešovit sadržaj

- XML se može koristiti i za narativne dokumente slobodnijeg oblika, kao što su poslovni izveštaji, članci, eseji, priče...

```

<biography> <name><first_name>Alan</first_name>
<last_name>Turing</last_name></name> was one of the
first people to truly deserve the name <emphasize>computer
scientist</emphasize>. Although his contributions to the field
are too numerous to list, his best-known are the eponymous
<emphasize>Turing Test</emphasize> and
<emphasize>Turing Machine</emphasize>. <definition>The
<term>Turing Test</term> is to this day the standard test for
determining whether a computer is truly intelligent. This test
has yet to be passed. </definition> <definition>The
<term>Turing Machine</term> is an abstract finite state
automaton with infinite memory that can be proven equivalent
to any any other finite state automaton with arbitrarily large
memory. Thus what is true for a Turing machine is true for all
equivalent machines no matter how implemented.
</definition>
<name><last_name>Turing</last_name></name> was also
an accomplished <profession>mathematician</profession>
and <profession>cryptographer</profession>. His assistance
was crucial in helping the Allies decode the German Enigma
machine. He committed suicide on
<date><month>June</month> <day>7</day>,
<year>1954</year></date> after being convicted of
homosexuality and forced to take female hormone injections.
</biography>

```

► 24

Zoran Čirović 18:14

Atributi

Elementima se mogu pridružiti atributi koji pružaju dodatne informacije o elementima



```
<poruka datum = "12.5.08." sala = "201">
<tekst>Sastanak Katedre sutra u 10</tekst>
</poruka>
```

▶ 25

Zoran Čirović 18:14

Imena

- ▶ Pravila su stroga. Važe ista za imena elemenata i za attribute i nazivamo ih XML imena (engl. *XML names*).
- ▶ XML imena mogu da sadrže:
 - ▶ Sve alfanumeričke znakove
 - ▶ Neengleska slova, brojeve i ideograme (ö, ç,...)
 - ▶ Donja crta, crtica, tačka
- ▶ XML imena ne mogu sadržati:
 - ▶ Navodnike, polunavodnike, znak za dolar, kapicu^, znak za procenat, tačka-zarez.
 - ▶ Dvotačka je dozvoljena, ali je rezervisana za prostore imena.
 - ▶ Sva imena koja počinu sa xml su rezervisana za standardizaciju.
- ▶ XML imena mogu da počnu samo slovom, donjom crtom.
- ▶ Ne mogu početi sa cifrom, crticom, tačkom.
- ▶ Dužina imena nije ograničena.

▶ 26

Zoran Čirović 18:14

Reference - 1

- ▶ Znakovni podaci unutar elementa ne smeju da sadrže znak manje od (<). Ovaj karakter se uvek tumači kao početak taga.
- ▶ Ako želite da koristite baš ovaj karakter u vašem tekstu, možete koristiti referencu <
- ▶ Kada neki parser čita dokument vrši se zamena < reference sa stvarnim znakom <
- ▶ Na primer:
- ▶

```
<SCRIPT LANGUAGE="JavaScript">
  if (location.host.toLowerCase( ).indexOf("cafeconleche") &lt; 0)
  {
    location.href="http://www.cafeconleche.org/";
  }
</SCRIPT>
```
- ▶ Znakovni podaci ne smeju sadržavati ni sirovi znak ampersand (&) koji nema svoju izlaznu sekvencu jer se ovaj znak uvek interpretira kao početak reference. Zato se za prikaz znaka ampersand koristi referenca &
- ▶ `<publisher>O'Reilly & Associates</publisher>`

▶ 27

Zoran Čirović 18:14

Reference - 2

▶ <

&lt;	<
&gt;	>
&amp;	&
&apos;	'
&quot;	"

▶ 28

Zoran Čirović 18:14

Znaci navoda

- ▶ Vrednosti atributa moraju uvek biti unutar znaka navoda. Moguće je koristiti **jednostruke ili dvostruke** znake navoda:

```
<ime="Krcun"> ili:
<ime='Krcun'>
```

- ▶ Dupli znaci navoda su češći. Nekada je neophodno koristiti jednostruke kao u sledećem primeru:

```
<ime='Slobodan "Krcun" Penezić'>
```

▶ 29

Zoran Čirović 18:14

Primeri

- ▶ `<Person name='Martin' age='33' />`
- ▶ `<age base='16' units='years' >20</age>`
- ▶ `<age base="10" units="years" >32</age>`

▶ 30

Zoran Čirović 18:14

Šta koristiti - element ili atribut?

Primer 1.

```
<ime>Krcun</ime>
<nesto ime="Krcun">
```

Primer 2a.

```
<partner tip="nabavljač">
  <ime>Pera</ime>
  <prezime>Perić</prezime>
</partner>
```

Primer 2b.

```
<partner> nabavljač </partner>
  <ime>Pera</ime>
  <prezime>Perić</prezime>
```

Podaci se mogu skladištiti u elementima ili u atributima.

U prvom primeru *tip* je atribut. U drugom primeru *tip* je element. Oba primera daju iste informacije. Ne postoje određena pravila kada koristiti attribute a kada elemente. Neka načelna preporuka je da se elementi koriste kada je u pitanju nešto što je samo po sebi celokupna informacija, a ne neki njen pomoćni deo.

► 31

Zoran Čirović 18:14

Potencijalni problemi (ograničenja) prilikom korišćenja atributa

1. Atributi ne mogu sadržati višestruke vrednosti (elementi mogu)
2. Atributi nisu lako proširivi
3. Atributi ne opisuju strukturu
4. Atributima se teže manipuliše u programskom kodu
5. Vrednosti atributa se teško testiraju u odnosu na DTD (eng. Document Type Definition) – definicija tipa dokumenta

► 32

Zoran Čirović 18:14

Komentari

▶ `<!-- comment text -->`

- ▶ Koristi se za dodatne informacije radi čitljivosti. Sadržaj u komentaru ne može imati --, sve ostale markup karaktere može.
- ▶ Aplikacije za parsiranje mogu a ne moraju da prosledjuju informacije o komentarima (misli se na sam sadržaj komentara).

▶ **Primeri:**

- ▶ `<!-- This is a comment about how to open ( <![CDATA[ ]> ) CDATA sections -->`
- ▶ `<!-- I really like having elements called <fred> in my markup languages -->`
- ▶ `<!-- Comments can contain all sorts of character literals including &, <, >, ' and". -->`
- ▶ `<!-- If entities are used inside comments ( < for example ) they are not expanded. -->`

▶ 33

Zoran Čirović 18:14

▶ Primeri ne ispravnih komentara

- ▶ `<!-- Comments cannot contain the -- char sequence -->`
- ▶ `<!-- Comments cannot end with a hyphen --->`
- ▶ `<!-- Comments cannot <!-- be nested --> -->`

▶ 34

Zoran Čirović 18:14

Odeljak CDATA

- ▶ XML dokumenti imaju strogo definisana pravila. Ukoliko je potrebno da dokument sadrži običan tekst i specijalne zankove, dokument postaje teško čitljiv. U takvim slučajevima može se koristiti odeljak CDATA koji obezbeđuje da se sadržaj tretira kao sirovi tekst. Na primer.

```
<p>ovde ide neki tekst </p>
<pre>
<![CDATA[
  <svg xmlns="http://www.w3.org/2000/svg"
    width="12cm" height="10cm">
    <ellipse rx="110" ry="130" />
    <rect x="4cm" y="1cm" width="3cm" height="6cm" />
  ]]>
</pre>
```

▶ 35

Zoran Čirović 18:14

Instrukcije obrade

- ▶ **<?target data?>**
- ▶ Instrukcije se koriste kako bi pružile informaciju aplikacijama za obradu XML dokumenata. Mogu sadržati informacije kako dokument obraditi, prikazati i slično.
- ▶ Sastoje se iz dva dela: ime instrukcije /target/ i od podataka za instrukciju /data/
- ▶ **<?target data?>**
- ▶ Target mora da zadovolji ista pravila za imena kao i elementi i atributi, osim završnog karaktera za sekvencu (**?>**)
- ▶ Sav markap deo se ignoriše unutar konteksta instrukcije, ali po preporukama W3C konzorcijuma ova sekvenca **ne sme počinjati sa xml sekvencom**.
- ▶ **Imenski prostori se ne primenjuju na instrukcije**. Zato je teško garantovati jedinstvenost instrukcija i to predstavlja potencijalni problem.

▶ 36

Zoran Čirović 18:14

Primeri instrukcija obrade

- ▶ `<?robots index="yes" follow="no"?>`
- ▶ `<?display table-view?>`
- ▶ `<?sort alpha-ascending?>`
- ▶ `<?textinfo whitespace is allowed ?>`
- ▶ `<?elementnames <fred>, <bert>, <harry> ?>`

▶ 37

Zoran Čirović 18:14

- ▶ Druge instrukcije obrade mogu imati potpuno drugačiju sintaksu i semantiku. Na primer, instrukcije obrade mogu imati efektivno neograničenu količinu teksta. PHP smešta velike programe u instrukcije za obradu. Na primer:

```

▶ <?php
    mysql_connect("database.unc.edu", "clerk",
        "password");
    $result = mysql("HR", "SELECT LastName,
        FirstName FROM Employees ORDER BY LastName,
        FirstName");
    $i = 0;
    while ($i < mysql_numrows ($result))
    {
        $fields = mysql_fetch_row($result);
        echo "<person>$fields[1] $fields[0]
        </person>\r\n"; $i++;
    }
    mysql_close( );
?>

```

▶ 38

Zoran Čirović 18:14

- ▶ Instrukcije obrade spadaju u označavanje, tj **nisu elementi**. Mogu se pisati na bilo kom mestu u dokumentu osim unutar oznaka, kao i komentari što se pisu. Mogu se pisati i pre korenskog elementa i posle.

- ▶ Primer primene stilova

- ▶

```
<?xml-stylesheet href="person.css"
type="text/css"?>
<person>
    Alan Turing
</person>
```

▶ 39

Zoran Čirović 18:14

XML deklaracija

Svaki XML dokument mora da sadrži XML **deklaraciju**, tj. **instrukciju obrade kojom se** dokument identifikuje kao XML dokument. Ovo je prva konstrukcija u dokumentu.

Osnovni oblik XML deklaracije: `<?xml version ="1.0"?>`

Opcioni oblik XML deklaracije:

```
<?xml version ="1.0" encoding= "UTF-8"?>
<?xml version ="1.0" encoding= "UTF-16"?>
```

Opšti oblik XML deklaracije:

```
<?xml          version='1.0'
               encoding='character encoding'
               standalone='yes|no'?>
```

▶ 40

Zoran Čirović 18:14

Primeri

- ▶ `<?xml version='1.0' ?>`
- ▶ `<?xml version='1.0' encoding='US-ASCII' ?>`
- ▶ `<?xml version='1.0' encoding='US-ASCII' standalone='yes' ?>`
- ▶ `<?xml version='1.0' encoding='UTF-8' ?>`
- ▶ `<?xml version='1.0' encoding='UTF-16' ?>`
- ▶ `<?xml version='1.0' encoding='ISO-10646-UCS-2' ?>`
- ▶ `<?xml version='1.0' encoding='ISO-8859-1' ?>`
- ▶ `<?xml version='1.0' encoding='Shift-JIS' ?>`

▶ 41

Zoran Čirović 18:14

Atribut *version*

- ▶ Atribut *version* treba da ima vrednost 1.0. U veoma retkim slučajevima može da dobije vrednost 1.1. Pošto zadavanje verzije 1.1 ograničava upotrebu na manji broj analizatora, ne bi trebalo postavljati ovu verziju bez istinskog razloga.
- ▶ Ako ne govorite burmanski, mongolski, kamodžanski, amharski ili diverhi, ako ne koristite netekstualne C0 kontrolne znakove (vertikalni tabulator, nova stranica, zvonice) nemate razloga da koristite verziju 1.1

▶ 42

Zoran Čirović 18:14

Atribut *encoding*

- ▶ XML je dokument od čistog teksta.
- ▶ Kakvi su znaci u tom tekstu? ASCII, Latin-1, Unicode?
- ▶ Svi XML dokumenti su podrazumevano kodirani sa UTF-8 kodovima promenljive dužine u skupu Unicode.
- ▶ Pri analizi XML fajla, analizator čita iz metapodataka vrstu kodovanja tog fajla i primenjuje ga. Međutim ne pružaju svi sistemi takve podatke i nekada ih je neophodno navesti i u samom dokumentu pomoću gore navedene deklaracije kodovanja.
- ▶ Ako podaci o kodovima ne postoje u samoj datoteci, a ne navedu se u dokumentu, analizator pretpostavlja Unicode.
- ▶ Ukoliko postoje oba podatka, a u suprotnosti su, analizator veruje sistemu tj. onako kako je dokument kodovan.

▶ 43

Zoran Čirović 18:14

Atribut *standalone*

- ▶ Ako ovaj atribut ima vrednost “no” onda aplikacija može učitati spoljni DTD (učimo o DTD sledeći čas) preko koga se utvrđuju vrednosti i značenje delova dokumenta.
- ▶ Dokument koji nema DTD može imati ovaj atribut postavljen na vrednost “yes”.
- ▶ Ako je izostavljen, pretpostavlja se da ima vrednost “no”

▶ 44

Zoran Čirović 18:14

Praznine u XMLu

- ▶ U XML-u je sačuvan prazan prostor. Korišćenjem XML-a prazan prostor je prikazan u parsiranom dokumentu. Na primer:

```
<body>Puno      pozdrava iz Beograda</body>
```

će u parseru biti:

Puno pozdrava iz Beograda

(to sa HTML-om nije slučaj)

▶ 45

Zoran Čirović 18:14

Primer ispravno korišćenih praznina

- ▶

```
<pre:Vehicle xmlns:pre='urn:example-org:Transport'
```
- ▶

```
type='car'>
```
- ▶

```
<seats>4</seats>
```
- ▶

```
<colour>White</colour>
```
- ▶

```
<engine>
```
- ▶

```
<petrol/>
```
- ▶

```
<capacity units='cc'>1598</capacity>
```
- ▶

```
</engine>
```
- ▶

```
</pre:Vehicle>
```

- ▶ Šta možete zaključiti na osnovu ovih primera?

▶ 46

Zoran Čirović 18:14

Primer neispravno korišćenih praznina

- ▶ `<pre:Vehicle xmlns:pre='urn:example-org:Transport'`
- ▶ `Type>='car'>`
- ▶ `<seats>4</seats>`
- ▶ `</pre:Vehicle>`

▶ 47

Zoran Čirović 18:14

Novi red

- ▶ U XML-u, CR / LF karakteri se pretvaraju u LF karakter.

U XML-u, nov red u tekstu je uvek sačuvan kao LF (eng. line feed). U Windows aplikacijama nov red je par CR (eng. carriage return) i LF karaktera. Kod UNIX sistema karakter nov red je LF, mada neke aplikacije koriste i samo CR. Ova razlika među operativnim sistemima često za posledicu ima da se podaci vraćaju u obliku toka (eng. stream) a ne u željenom formatu.

▶ 48

Zoran Čirović 18:14

Prostor imena (imenski prostor)

49

Zoran Čirović 18:14

Prostor imena - *namespace*

- ▶ Uloga prostora imena je:
 - ▶ 1. da omogući razlikovanje istoimenih elemenata
 - ▶ 2. da grupiše srodne elemente
- ▶ Mnogi XML dokumenti kombinuju više grupa elemenata koji potiču od različitih aplikacija. Na primer, neki XHTML dokument može da sadrži i SVG slike i MathML jednačine.
- ▶ U ovim slučajevima, dokument sadrži elemente koji moraju biti jedinstveni koristeći imenski prostor kome pripadaju.
- ▶ Takođe, XML dozvoljava dizajneru da odabere sopstvene elemente, pa je verovatna pojava istoimenih elemenata.

▶ 50

Zoran Čirović 18:14

Kvalifikovano ime

- ▶ XML prostor imena omogućava razdvajanje XML elemenata koji imaju isto lokalno (kratko) ime, koristeći **jedinstveni identifikator resursa - URI** (engl. *Uniform Resource Identifier*).
- ▶ URI predstavlja jedinstveni string za definisanje punih imena na osnovu lokalnih imena i URLa.
- ▶ Dakle, prostor imena i lokalno ime zajedno čine globalno jedinstveno ime.

▶ 51

Zoran Čirović 18:14

Korišćenje prostora imena

- ▶ **Deklaracija prostora imena** se vrši unutar početnog taga.
- ▶ **Mapiranje prostora imena** se vrši u jedan drugi string, obično kraći, poznat kao *prefiks* prostora imena.
- ▶ **xmlns:prefix='URI'**
- ▶ Ako se izostavi prefiks, na primer, **xmlns='URI'**, onda se mapiranje vrši u **podrazumevani prostor imena**.

▶ 52

Zoran Čirović 18:14

Primer

```

▶ <rdf:RDF xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax#">
  <rdf:Description
    about="http://www.cafeconleche.org/examples/impressionists.xml">
    <title> Impressionist Paintings </title>
    <creator> Elliotte Rusty Harold </creator>
    <description> A list of famous impressionist paintings organized by painter and date
    </description>
    <date>2000-08-22</date>
  </rdf:Description>
▶ </rdf:RDF>

```

▶ 53

Zoran Čirović 18:14

Opseg važenja prostora imena

- ▶ Opseg važenja (eng. scope) prostor imena je skup elemenata na koje se odnosi taj prostor imena.
- ▶ **Prostor imena važi za elementat u kome je deklarisan i u svim elementima koji su hijerarhijski ispod tog elementa.**

▶ 54

Zoran Čirović 18:14

Pojmovi vezani za prostore imena

- ▶ Kvalifikovana imena – Qnames (eng. qualified names)
- ▶ Imena svih elemenata u nekom dokumentu koji odgovaraju prostorima imena neke XML specifikacije nazivaju se QNames. **QNames** imaju lokalni naziv i neobavezno prefiks.
- ▶ Primer:
- ▶ `<?xml version='1.0' ?>`
- ▶ `<doc xmlns:x="http://example.com/ns/foo">`
`<x:p/>`
- ▶ `</doc>`

▶ 55

Zoran Čirović 18:14

Kvalifikovano ime – <http://www.w3.org/TR/REC-xml-names/#NT-QName>

- ▶ **Qname** ::= PrefixedName | UnprefixedName
- ▶ PrefixedName ::= Prefix ':' LocalPart
- ▶ UnprefixedName ::= LocalPart
- ▶ Prefix ::= NCName
- ▶ LocalPart ::= NCName
- ▶ NCName ::= Name - (Char* ':' Char*) /* XML Ime, bez ":" */
- ▶ Name ::= NameStartChar (NameChar)*
- ▶ NameStartChar ::= ":" | [A-Z] | "_" | [a-z] | [#xC0-#xD6] | [#xD8-#xF6] | [#xF8-#x2FF] | [#x370-#x37D] | [#x37F-#x1FFF] | [#x200C-#x200D] | [#x2070-#x218F] | [#x2C00-#x2FEF] | [#x3001-#xD7FF] | [#xF900-#xFDCF] | [#xFDF0-#xFFFD] | [#x10000-#xEFFFF]
- ▶ NameChar ::= NameStartChar | "-" | "." | [0-9] | #xB7 | [#x0300-#x036F] | [#x203F-#x2040]
- ▶ Char ::= /* any Unicode char, excluding surrogate blocks FFFE and FFFF. */
- ▶ #x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]

▶ 56

Zoran Čirović 18:14

Nekvalifikovani elementi

- ▶ **Ako element nema prefiks, onda pripada podrazumevanom prostoru imena, ako postoji.**
- ▶ Elementi koji nisu u poznatom prostoru imena nazivaju se **nekvalifikovanim elementima**. Prostor imena takvih elemenata je **prazan string** - " ".
- ▶
- ▶ Ako postoji podrazumevani prostor imena za neki elemenat, a želi se da taj elemenat bude nekvalifikovan, tada se taj (podrazumevani) prostor imena može maskirati koristeći deklaraciju oblika **xmlns=" "** u tom elementu.

▶ 57

Zoran Čirović 18:14

Kvalifikovani i nekvalifikovani elementi

- ▶ Primer 1.

```
<pre:Person xmlns:pre='urn:example-org:People' >
  <name>Martin</name>
  <age>33</age>
</pre:Person>
```

- ▶ Element *Person* ima prefiks *pre* koji je mapiran u prostor imena *urn:example-org:People*. Ovaj elemenat sadrži podelemente sa lokalnim imenima *name*, *age*. Oba podelementa su nekvalifikovana; zato što nisu u nijednom prostoru imena. Razlikovati oblast vazenja prostora imena od primene na neko ime!

▶ 58

Zoran Čirović 18:14

Kvalifikovani i nekvalifikovani elementi

► Primer 2.

```
<Person xmlns='urn:example-org:People' >
  <name xmlns="">Martin</name>
  <age xmlns="">33</age>
</Person>
```

- Element *Person* nema prefiks. Ujedno isti elemenat definiše podrazumevani prostor imena *urn:example-org:People*. Ovaj elemenat sadrži podelemente sa imenima *name* odnosno *age*. Međutim, oba podelementa maskiraju podrazumevani prostor imena sa prostorom '' (prazan string). Dakle, oba podelementa su nekvalifikovana, zato što nisu u nijednom prostoru imena (tj. pripadaju prostoru imena koji je prazan string). Ovaj primer je ekvivalentan primeru 1.

► 59

Zoran Čirović 18:14

Kvalifikovani elementi

► Primer 3.

```
<pre:Person xmlns:pre='urn:example-org:People' >
  <pre:name>Martin</pre:name>
  <pre:age">33</pre:age>
</pre:Person>
```

- Element *Person* ima prefiks, kao i podelementi. Elemenat definiše *pre* prostor imena *urn:example-org:People*.

► 60

Zoran Čirović 18:14

Kvalifikovani elementi

► Primer 4.

```
<Person xmlns='urn:example-org:People' >
  <name>Martin</name>
  <age>33</age>
</Person>
```

- Element *Person* definiše podrazumevani prostor imena *urn:example-org:People*. Taj prostor imena koriste podelementi. Ovaj primer je ekvivalent primeru 3.

► 61

Zoran Čirović 18:14

Atributi i prostor imena

- Nazivi atributa su QNames.

- prefix:localname='value'

► Primer 1. Kvalifikovani atributi

```
<Person xmlns='urn:example-org:People'
  xmlns:b='urn:example-org:People:base'
  xmlns:u='urn:example-org:units' >
  <name>Martin</name>
  <age b:base='10' u:units='years' >33</age>
</Person>
```

► 62

Zoran Čirović 18:14

Atributi i prostor imena

- ▶ **Napomena:** Atributi bez prefiksa ne pripadaju nijednom imenskom prostoru, čak i kada su u oblasti važenja nekog podrazumevanog imenskog prostora.

- ▶ **Primer 2. Nekvalifikovani atributi**

- ▶ `<Person xmlns='urn:example-org:People' >`
- ▶ `<name>Martin</name>`
- ▶ `<age base='10' units='years' >33</age>`
- ▶ `</Person>`

- ▶ **Mada postoji podrazumevani imenski prostor, atributi su nekvalifikovani.**

▶ 63

Zoran Čirović 18:14

Dobro formiran dokument

64

Zoran Čirović 18:14

Da bi document bio dobro-formiran XML document treba da:

1. Postoji XML deklaracija.
2. Dokument sadrži jedan i samo jedan koreni element u kome su ugnježdjeni svi ostali elementi i njihovi sadržaji.
3. Svi elementi i atributi u dokumentu moraju da budu sintaksno ispravni.
4. Elementi moraju imati završni tag (<...> ... </...>). Jedini izuzetak predstavlja *empty* tag koji nema ni sadržaj ni telo, a označava se <.../>.
5. Elementi moraju biti ugnježdjeni.
6. Imena tagova u XML-u su case-sensitive (zavise od velikih i malih slova). Pri dodeljivanju imena se moraju poštovati određena pravila.
7. Sve vrednosti atributa moraju biti u okviru navodnika.

▶ 65

Zoran Čirović 18:14

Primer

- ```

▶ <?xml version='1.0' encoding='UTF-8' ?>
▶ <p:Person xmlns:p='urn:example-org:People' >
▶ <name>Martin</name>
▶ <!-- Young and spritely -->
▶ <age>33</age>
▶ <height units='inches' >64</height>
▶ </p:Person>

▶ Dokument je dobro formiran.

```

▶ 66

Zoran Čirović 18:14

## Primer loše formiranog dokumenta

```

▶ <?xml version='1.0' encoding='UTF-8' ?>
▶ <p:Person>
▶ <name>Martin</name>
▶ <age value='33' >A young <i>and</i> spritely
▶ person</age>
▶ <height units='inches' units='in'>64</height>
▶ <weight xmlns:x1='urn:example-org:People'
▶ xmlns:x2='urn:example-org:People'
▶ x1:units='stone' x2:units='shekels' >10</weight>
▶ </p:Person>
▶ <p:Person/>

```

▶ 67

Zoran Čirović 18:14

## Primer: XML vs. HTML

### HTML

```

<html>
<title>Course Roster</title>
<body>
<center>
<h1>Course Roster</h1>
<h2>XML Programming</h2>
<h3>Department: EECS</h3>
<p>
<table border="2">
<tr>
<th>Teacher</th>
<td>Paul Thompson</td>
</tr><tr>
<th>Student
List</th>
<td>Paul Jones

Uma Abingdon

Lindsay Garmon
</td>
</tr>
</table>
</center>
</body>
</html>

```

▶ 68

### XML

```

<?xml version="1.0"?>
<course>
<name>Java Programming</name>
<department>EECS</department>
<teacher>
<name>Paul Thompson</name>
</teacher>
<student>
<name>Ron Jones</name>
</student>
<student>
<name>Uma Abingdon</name>
</student>
<student>
<name>Lindsay Garmon</name>
</student>
</course>

```

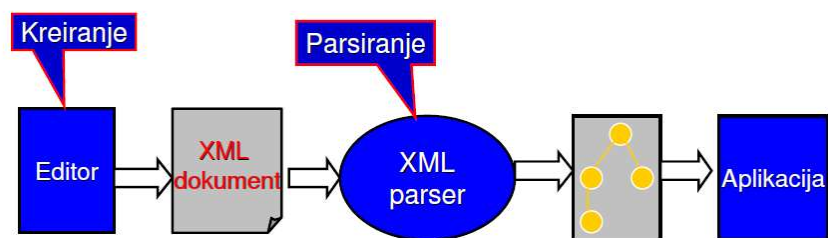
Zoran Čirović 18:14

## Obrada XML dokumenata

69

Zoran Čirović 18:14

### Schema tipične obrade XML dokumenta



▶ 70

Zoran Čirović 18:14

## XML parser

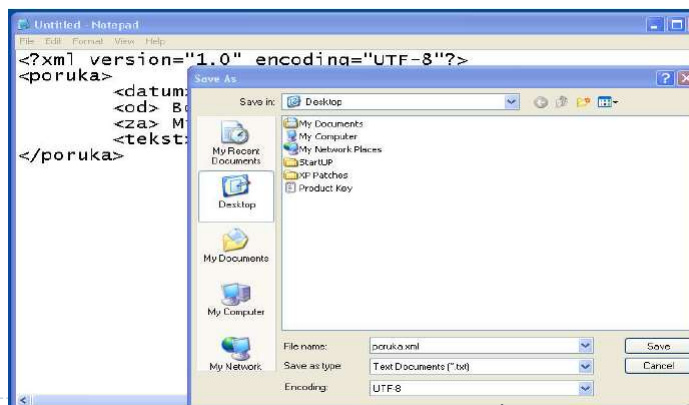
- ▶ XML parser je aplikacija ili deo aplikacije koji se može izvršavati samostalno ili kao deo druge aplikacije.
- ▶ **XML parser verifikuje da li je XML** dokument dobroformljen
- ▶ XML parser čita dokument i konvertuje ga u hijerarhijsku strukturu
- ▶ XML parser je obično deo aplikacije i prenosi parsirani dokument do same aplikacije

▶ 71

Zoran Čirović 18:14

## Kreiranje XML dokumenta

- Tekst editori (na primer Notepad)
- VS.NET XML Designer
- XML Spy – razvojno okruženje za XML



▶ 72

Zoran Čirović 18:14

## Pregled XML dokumenta (source)

- Pomoću web browser-a koji podržavaju XML (Internet Explorer 5.0 i više verzije)

