

Digitalni zapis podataka

Predavanje 5

29. novembar 2022.

Sadržaj

- 1 Jezici za obeležavanje
 - XML
 - JSON

XML

- eXtensible Markup Language (skr. XML) je meta jezik za obeležavanje koji je nastao sredinom 1990-tih kao rezultat potrebe za postojanjem jezika za obeležavanje koji bi bio jednostavan za parsiranje (rasčlanjavanje) i obradu
- obezbeđuje čuvanje i transport podataka
- informacije skladištene u okviru XML dokumenta su nezavisne od platforme tj. od softvera i hardvera
- xml dokumenta su samoopisna
- ekstenzija je .xml

XML

- Vremenom se pojavio se jako veliki broj XML aplikacija
 - XHTML - zapis hipertekstualnih dokumenata
 - MathML - zapis matematičkog sadržaja
 - SVG - zapis vektorskih crteža
 - najnoviji Microsoft Office formati su zasnovani na jeziku XML
- razvoj specijalnih baza podataka koje su zasnovane na skladištenju informacija u XML dokumentima i korišćenju specijalizovanih jezika (npr. XPath, XQuery) za pretraživanje

XML dokumenta

- XML se koristi da bi se obeležila struktura dokumenata određenog tipa
- oznake se koriste za obeležavanje elemenata dokumenta
- oznake određuju strukturu dokumenta
- dokumenti se sastoje od međusobno ugnježenih elemenata
- za obeležavanje elemenata se koriste etikete (tagovi) oblika `<ime-elementa>` i `</ime-elementa>` (npr. `<primalac>` i `</primalac>`, `<autor>` i `</autor>`)

XML dokumenta

- elementi sadrže tekst ili druge elemente
- elementi mogu biti dodatno okarakterisani atributima oblika `ime_atributa="vrednost"` (npr. `id="PX1"`)
- u okviru teksta mogu se pojaviti i karakterski entiteti oblika **&ime-entiteta;** koji označavaju određene karaktere

XML dokumenta - primer

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

  <poruka id="id_1">
    <posiljalac>Marko</posiljalac>
    <primalac>Petar</primalac>
    <tekst>Sastanak je u 18h</tekst>
  </poruka>

  <poruka id="id_2">
    <posiljalac>Ana</posiljalac>
    <primalac>Marko</primalac>
    <tekst>Polozila sam ispit</tekst>
  </poruka>

</poruke>
```

XML dokumenta - primer

Objašnjenje primera:

- etikete su prikazane plavom bojom (npr. poruka)
- atributi su prikazani crvenom bojom (npr. id), a njihove vrednosti ljubičastom bojom
- tekst elementa je prikazan crnom bojom
- koreni element - **poruke**
- element **poruke** se sastoji od elemenata **poruka**
- element **poruka** ima atribut **id**
- element **poruka** se sastoji od elemenata **posiljalac**, **primalac** i **tekst**

XML dokumenta

- za XML dokument sa ispravnom sintaksom se kaže da je *dobro formiran*
- XML dokument je *dobro formiran* ako
 - ima koreni element (u primeru **poruke**)
 - elementi imaju zatvarajuće etikete
 - etikete su osetljive na mala i velika slova
 - etikete su pravilno ugnježdene, tj. nema preplitanja
 - vrednosti atributa su navedene sa dvostrukim navodnicima

XML dokumenta

- sadržaj i značenje elemenata nije propisano XML-om već se svakom dokumentu pridružuje njegov tip
- tip dokumenta određuje sintaksu dokumenta tj. koji elementi, atributi i entiteti se mogu javiti u okviru dokumenta i kakav je njihov međusobni odnos
- DTD (eng. Document Type Definition) ili XML Schema se koriste za definiciju tipa dokumenta
- pripadnost XML datoteke određenom tipu dokumenta koji je definisan pomoću DTD se izražava deklaracijom `<!DOCTYPE>` koja se navodi na početku samog dokumenta. U okviru ove deklaracije nalazi se uputnica na definiciju tipa dokumenta.

DTD

- DTD definiše strukturu i dozvoljene elemente i attribute u XML dokumentima
- poželjno je koristiti DTD kada se vrši razmena podataka između različitih korisnika
- ako XML dokument zadovoljava uslove DTD onda je dokument *validan*
- DTD deklaracija se može navesti u okviru dokumenta ili kao zaseban dokument
- Ako je DTD deklaracija navedena u zasebnom dokumentu, ekstenzija DTD dokumenata je .dtd

Primer DTD

```
<!DOCTYPE poruke [  
  
  <!ELEMENT poruke (poruka+)>  
  
  <!ELEMENT poruka (posiljalac, primalac, tekst)>  
  <!ATTLIST poruka id ID #REQUIRED>  
  
  <!ELEMENT posiljalac (#PCDATA)>  
  <!ELEMENT primalac (#PCDATA)>  
  <!ELEMENT tekst (#PCDATA)>  
]>
```

Iz ugla DTD XML dokumenta se sastoje od

- elemenata
- atributa
- entiteta
 - neki karakteri u XML imaju posebno značenje (npr. `< i >`) te ih nije moguće koristiti u okviru teksta, već su za njih definisani entiteti pomoću kojih je takvim karakterima dodeljeno specijalno ime koje se koristi u tekstu, a parser ih pri obradi XML dokumenta zamenjuje odgovarajućim karakterom (npr. `<` se koristi za `<`, `>` za `>`, `&` za `&`, `"` za `"` i `'` za `'`)
 - mogu se definisati dodatni entiteti u obliku `<!ENTITY ime "vrednost">` i onda se koriste navođenjem `&ime;`
- PCDATA (parsed character data) - tekst između otvorene i zatvorene etikete koji će biti parsiran i entiteti će biti zamenjeni pridruženim vrednostima
- CDATA (character data) - tekst između otvorene i zatvorene etikete neće biti parsiran

- elementi se navode u obliku `<!ELEMENT ime-elementa (sadržaj)>` gde sadržaj može da bude
 - **#PCDATA**
npr. `<!ELEMENT primalac (#PCDATA)>`
 - **(imena elemenata od kojih se sastoji)**
npr. `<!ELEMENT poruka (posiljalac, primalac, tekst)>`

- pored svake stavke sadržaja može se navesti dozvoljen broj pojavljivanja dete-elementa u roditelj elementu pri definiciji `<!ELEMENT roditelj (dete)>` navođenjem
 - (dete) - element dete mora da se pojavi tačno jedanput u roditelj elementu
 - (dete+) - element dete mora da se pojavi bar jedanput u roditelj elementu
 - (dete*) - element dete može da se pojavi 0 ili više puta u roditelj elementu
 - (dete?) - element dete može da se pojavi 0 ili jedanputa u roditelj elementu
 - (dete1|dete2) - element roditelj će sadržati element dete1 ili dete2

Odrediti za element roditelj šta može da sadrži sa narednim opisima:

- `<!ELEMENT roditelj (dete1, (dete2|dete3))>` -
- `<!ELEMENT roditelj ((dete1+), (dete2*))>` -
- `<!ELEMENT roditelj ((dete1?), dete2)>` -

Odrediti za element roditelj šta može da sadrži sa narednim opisima:

- `<!ELEMENT roditelj (dete1, (dete2|dete3))>` - mora sadržati element dete1 i mora sadržati element dete2 ili element dete3
- `<!ELEMENT roditelj ((dete1+), (dete2*))>` -
- `<!ELEMENT roditelj ((dete1?), dete2)>` -

Odrediti za element roditelj šta može da sadrži sa narednim opisima:

- `<!ELEMENT roditelj (dete1, (dete2|dete3))>` - mora sadržati element dete1 i mora sadržati element dete2 ili element dete3
- `<!ELEMENT roditelj ((dete1+), (dete2*))>` - mora sadržati element dete1 bar jedanput i element dete2 0 ili više puta
- `<!ELEMENT roditelj ((dete1?), dete2)>` -

Odrediti za element roditelj šta može da sadrži sa narednim opisima:

- `<!ELEMENT roditelj (dete1, (dete2|dete3))>` - mora sadržati element dete1 i mora sadržati element dete2 ili element dete3
- `<!ELEMENT roditelj ((dete1+), (dete2*))>` - mora sadržati element dete1 bar jedanput i element dete2 0 ili više puta
- `<!ELEMENT roditelj ((dete1?), dete2)>` - element dete1 može da se pojavi 0 puta ili jedanput i element dete2 mora da se pojavi jedanput

- atributi se navode u obliku `<!ATTLIST ime-elementa ime-atributa tip-atributa vrednost-atributa>` (ako ima više atributa nakon navođenja imena elemenata opisuje se posebno svaki od njih)
gde tip može da bude
 - `#CDATA` (npr. `<!ATTLIST naslovknjige godinaizdanja CDATA "2000">`)
 - `(en1|en2|..)` - spisak mogućih vrednosti
 - `ID` - vrednost je jedinstvena za elemente tog tipa
 - `ENTITY` - vrednost je entitet
- gde vrednost atributa može da bude
 - vrednost - podrazumevana vrednost
 - `#REQUIRED` - atribut je obavezno navesti pri korišćenju elementa za koji je vezan u XML dokumentu
 - `#IMPLIED` - navođenje atributa nije obavezujuće pri korišćenju elementa za koji je vezan u XML dokumentu
 - `#FIXED value` - vrednost atributa je fiksna

Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

```
<!ATTLIST student indeks CDATA "000/2021">
```

Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

```
<!ATTLIST student indeks CDATA "000/2021">
```

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut , a atribut pol može imati vrednosti m ili z , a podrazumevana je z

Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

```
<!ATTLIST student indeks CDATA "000/2021">
```

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut , a atribut pol može imati vrednosti *m* ili *z*, a podrazumevana je *z*

```
<!ATTLIST student  
indeks CDATA "000/2021"  
pol (m|z) "z">
```

Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

```
<!ATTLIST student indeks CDATA "000/2021">
```

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut , a atribut pol može imati vrednosti *m* ili *z*, a podrazumevana je *z*

```
<!ATTLIST student  
indeks CDATA "000/2021"  
pol (m|z) "z">
```

- tip jedinstven identifikator i obavezno je navođenje vrednosti atributa

Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

```
<!ATTLIST student indeks CDATA "000/2021">
```

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut , a atribut pol može imati vrednosti *m* ili *z*, a podrazumevana je *z*

```
<!ATTLIST student  
indeks CDATA "000/2021"  
pol (m|z) "z">
```

- tip jedinstven identifikator i obavezno je navođenje vrednosti atributa

```
<!ATTLIST student indeks ID #REQUIRED>
```


Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

```
<!ATTLIST student indeks CDATA "000/2021">
```

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut , a atribut pol može imati vrednosti *m* ili *z*, a podrazumevana je *z*

```
<!ATTLIST student  
indeks CDATA "000/2021"  
pol (m|z) "z">
```

- tip jedinstven identifikator i obavezno je navođenje vrednosti atributa

```
<!ATTLIST student indeks ID #REQUIRED>
```

- tip zadata niska, a navođenje atributa u elementu je opciono

Napisati opis atributa indeks za element student tako da je:

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut

```
<!ATTLIST student indeks CDATA "000/2021">
```

- tip zadata niska, a podrazumevana vrednost indeksa 000/2021 ako se ne navede atribut , a atribut pol može imati vrednosti *m* ili *z*, a podrazumevana je *z*

```
<!ATTLIST student  
indeks CDATA "000/2021"  
pol (m|z) "z">
```

- tip jedinstven identifikator i obavezno je navođenje vrednosti atributa

```
<!ATTLIST student indeks ID #REQUIRED>
```

- tip zadata niska, a navođenje atributa u elementu je opciono

```
<!ATTLIST student indeks CDATA #IMPLIED>
```

Zadatak

Napisati DTD dokument za XML dokumente koji sadrže podatke o dečijim pesmicama uzimajući u obzir sledeće uslove:

- pesma ima jedan naslov koji mora da se navede
- pesma ima autora čije ime ne mora da se navede
- pesma se sastoji od jedne ili više strofa
- strofa pesme se sastoji od jednog ili više stihova
- stih sadrži tekst

Napisati i XML dokument sa podacima o dečijim pesmicama.
Rešenje je u zasebnoj datoteci.

JSON

- JavaScript Object Notation (JSON) predstavlja standardni format teksta koji služi za reprezentaciju podataka
- zasnovan na JavaScript sintaksi zapisivanja objekata
- JSON se često koristi za razmenu podataka između veb aplikacija (npr. za slanje podataka od serverske aplikacije ka klijentskoj aplikaciji, kako bi se ti podaci prikazali na veb stranici, ili obrnuto, za slanje podataka od klijentske aplikacije ka serverskoj aplikaciji kako bi se ti podaci skladištili u nekoj bazi podataka)
- jednostavan za čitanje od strane čoveka
- JSON datoteke mogu lako da se parsiruju u različitim programskim jezicima
- ekstenzija .json

JSON

```
{  
  "id": "id_2",  
  "posiljalac": "Ana",  
  "primalac": "Marko",  
  "tekst": "Polozila sam ispit"  
}
```

JSON

U JSON datoteci

- podaci su dati u obliku ime:vrednost, gde je ime niska koja se navodi pod dvostrukim navodnicima (npr. "autor": "Jovan Jovanovic Zmaj")
- vrednost podatka može biti
 - niska koja se navodi pod dvostrukim navodnicima (npr. "autor": "Jovan Jovanovic Zmaj")
 - broj (npr. "godina": 1980)
 - niz vrednosti (npr. "deca": ["Marko", "Petar", "Ana"])
 - ...

JSON

U JSON datoteci

- vitičaste zagrade `{}` sadrže podatke o jednom objektu (instanci, entitetu) koji je opisan sa skupom parova `ime:vrednost`
(npr. o jednoj osobi `{"ime":"Petar", "pol":"m", "starost":45, "deca":["Marko", "Ana"]}`)
- uglaste `[]` zagrade sadrže niz vrednosti koje su razdvojene sa `,`
(npr. `deca: ["Marko", "Petar", "Ana"]`)

JSON primer

```
{
  "poruke": [
    {
      "id": "id_1",
      "posiljalac": "Marko",
      "primalac": "Petar",
      "tekst": "Sastanak je u 18h"
    },
    {
      "id": "id_2",
      "posiljalac": "Ana",
      "primalac": "Marko",
      "tekst": "Polozila sam ispit"
    }
  ]
}
```

- Struktura JSON datoteke se može definisati pomoću JSON Scheme

Zadatak

Zapisati u JSON datoteci podatke o sledećim osobama:

- Marko ima 25 godina, student je, a od sportova se bavi plivanjem i košarkom
- Petar je programer po zanimanju, a od sportova se bavi trčanjem i skijanjem
- Ana je balerina i ima 30 godina, a od sportova se bavi jahanjem, plesom i plivanjem

Zadatak

Zapisati u XML datoteci podatke o sledećim osobama i napraviti odgovarajući DTD:

- Marko ima 25 godina, student je, a od sportova se bavi plivanjem i košarkom
- Petar je programer po zanimanju, a od sportova se bavi trčanjem i skijanjem
- Ana je balerina i ima 30 godina, a od sportova se bavi jahanjem, plesom i plivanjem

Korisni alati i veb strane

- XML validator
<https://www.xmlvalidation.com/>
- JSON validator
<https://jsonformatter.curiousconcept.com>

Korišćen materijal za pripremu slajdova

- Stefan Mišković, Materijali za kurs Uvod u organizaciju i arhitekturu računara 1 na Matematičkom fakultetu
- Filip Marić, Predrag Janičić, Programiranje 1, 2021.
- Filip Marić, Skripta za kurs Uvod u veb i internet tehnologije na Matematičkom fakultetu
- Materijali za vežbe iz kursa Uvod u veb i internet tehnologije na Matematičkom fakultetu
<https://matfuvit.github.io/UVIT/vezbe/>