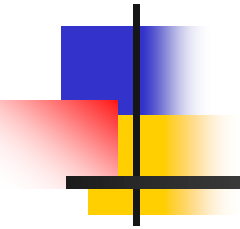


Interneto technologijos

Duomenų formatai
XML

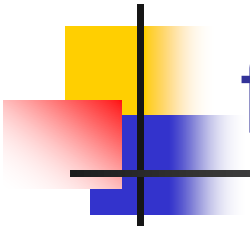
Duomenys ir duomenų aprašai

- Paprastai sakant, *duomenys* – tai kažkam įdomi tam tikra simbolių seka, tinkamame kontekste tampanti informacija:
  - Jonas Jonaitis 1976-10-11 37610111234 Vilnius
- Kad trečiai šaliai padėti „susiorientuoti", kokius duomenis aprašo tam tikra simbolių seka, dažnai naudojami *duomenų aprašai* (*meta-duomenys*)
  - **Vardas**: Jonas Jonaitis, **Gimimo data**: 1976-10-11
  - Kitas pavyzdys: visos lentelės paprastai turi antraštes, kad būtų aiškiau, kokie duomenys yra stulpeliuose

Duomenų formatas

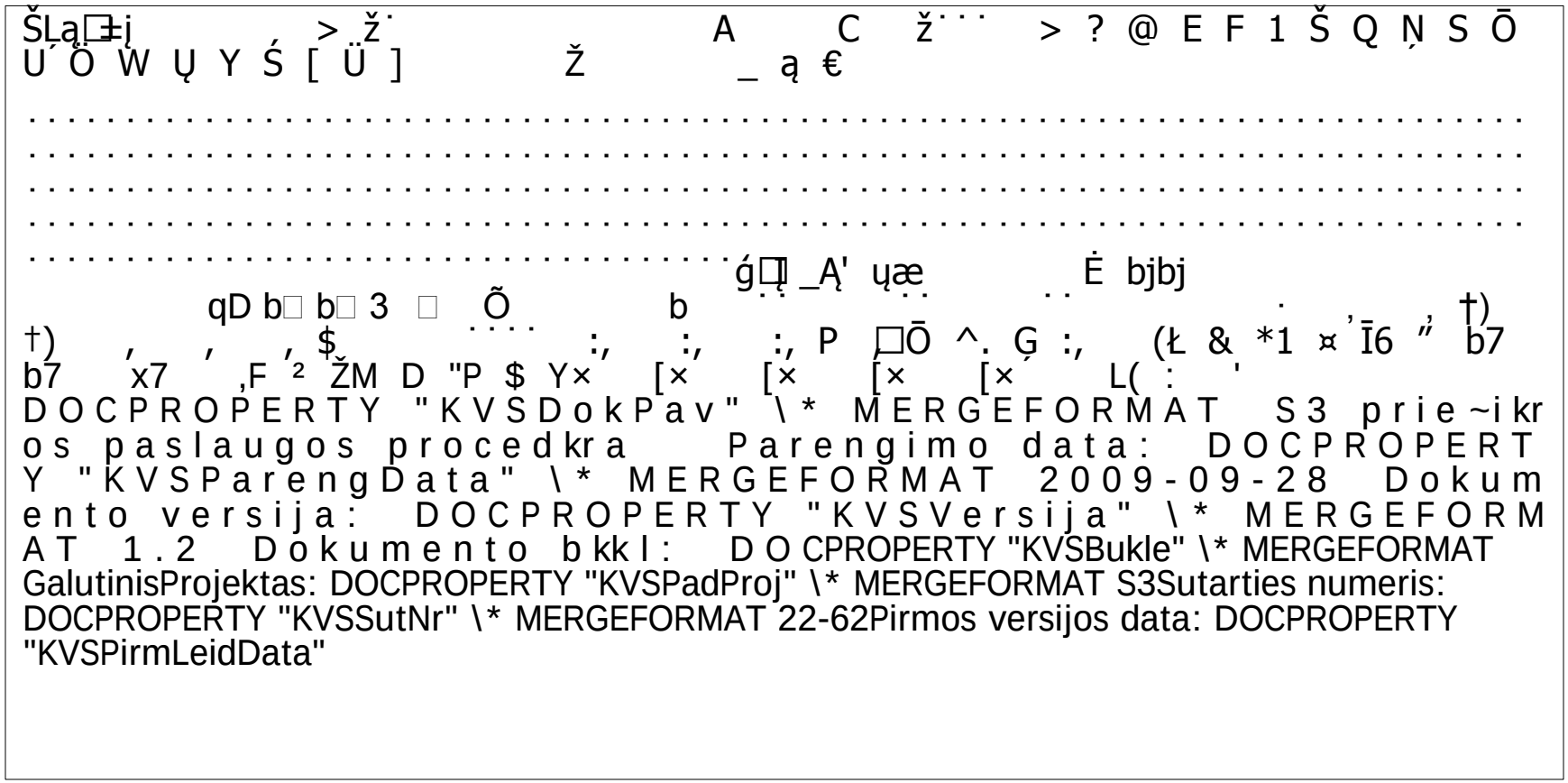
- Susitarimas, kaip duomenis (ir galbūt duomenų aprašus) užrašyti tam tikra simbolių ir *skirtukų* eilute, vadinamas *duomenų formatu*
- Pvz.: dažnai sutinkamas CSV (Comma-separated values, kableliais atskirtos reikšmės) formatas:
  - Vieno įrašo duomenų reikšmės atskiriamos kableliais
  - Naujas įrašas pradedamas naujoje eilutėje
  - Pvz.:
    - Jonas Jonaitis, 1976-10-11, 37610111234, Vilnius
    - Petras Petraitis, 1980-02-03, 38002039999, Kaunas

Tekstiniai ir binariniai duomenų formatai

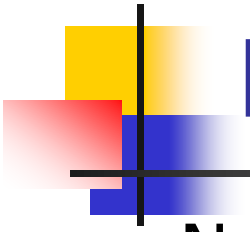


- Duomenų formatas yra *tekstinis*, jei:
  - šiuo formatu užrašytiems duomenims perskaityti žmogui užtenka turėti paprastą tekstinį redaktorių (pvz.: Notepad)
    - Pvz.: aukščiau rodytas CSV formatas yra tekstinis
- Duomenų formatas yra *dvejetainis/binarinis*, jei:
  - šiuo formatu užrašytiems duomenims perskaityti reikia turėti specializuotą programinę įrangą
    - Binarinio formato pavyzdys kitoje skaidrėje

Binarinio formato pavyzdys: seno Microsoft Word „doc“ formatas

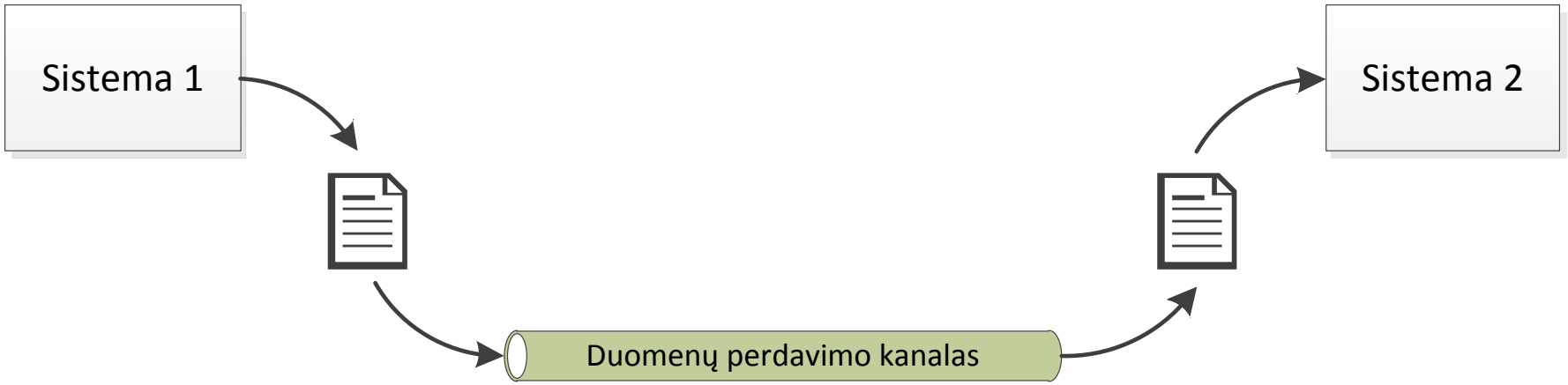


Duomenų struktūrizavimo laipsnis



- Nestruktūrizuoti duomenys
  - Pvz.: laisvos formos tekstas
- Dalinai struktūrizuoti
  - Tekstas išskaidytas į skyrius, duomenys pateikiami lentelėmis, sąrašais, bet yra ir laisvos formos teksto
- Griežtai struktūrizuoti
  - Struktūra apibrėžiama iš anksto, duomenų pateikimo forma (dokumentas) turi šią struktūrą griežtai atitikti
- XML leidžia aprašyti dalinai ir griežtai struktūrizuotus duomenis
  - Mes šiame kurse nagrinėsime tik griežtos struktūros duomenų užrašymą XML formatu

Kontekstas – duomenų apsikeitimas tarp sistemų



1. Sistema 1 duomenis patalpina į tam tikro formato dokumentą
2. Dokumentas per duomenų perdavimo kanalą nusiunčiamas sistemai 2
3. Sistema 2 „skaito“ dokumentą – paima dokumentu atsiųstus duomenis

Reikalavimai duomenų formatui

- Klausimas – koks formatas yra tinkamiausias, kad **sistemos** galėtų patogiai apsikeisti duomenimis?
- Būtų gerai, jei duomenų formatas būtų:
  - *Formalus* – negali būti tos pačios simbolių ir skirtukų eilutės dviejų skirtingų interpretacijų
  - *Paprastas* – sistemai (tiksliau, programuotojams) neturi būti sunku sukurti/skaityti tokio formato dokumentus
  - *Atviras, standartizuotas* – formato aprašymas būtų nemokamas, visiems prieinamas ir standartizuotas
  - *Skaitomas ir mašinai, ir žmogui* – ir žmogus turi galėti skaityti šiuo formatu užrašytus duomenis nenaudodamas specialios programinės įrangos
    - Formatas turi būti tekstinis
  - *Įgalintų aprašomų duomenų evoliuciją* (formato plečiamumas) – laikui bėgant, duomenų struktūra paprastai yra linkusi sudėtingėti. Būtų negerai, jei kiekvieną kartą tektų sistemoje perprogramuoti dokumento formavimą/skaitymą

Kodėl netinka CSV?

- CSV formatas tenkina tik pirmus keturis reikalavimus
- CSV formatas nėra *plečiamas* – šiuo formatu užrašytus duomenis skaitančios sistemos prisiriša prie stulpelių pozicijų (*pozicinis formatas*)
  - Negalime įterpti naujų stulpelių į vidurį
  - Pabaigoje naujus stulpelius pridėti galime, bet to ne visada pakanka
  - Ne visus duomenis pavyksta aprašyti lentelės struktūra, pvz. hierarchinių duomenų CSV formatu užrašyti nepavyks
- Išvada: CSV formatas tinka tik santykinai paprastiems duomenims užrašyti

XML – E**x**tensible **M**ark-up **L**anguage

- Tiesioginis vertimas: “plečiama duomenų aprašų kalba”
- Pirmą versiją sukūrė W3C konsorciumas 1998 m. vasario 10 d.
- Tenkinami visi aukščiau išvardinti reikalavimai
- Aktuali yra:
  - versijos 1.0 penkta redakcija, priimta 2008 m. lapkričio 26 d.
  - versijos 1.1 antra redakcija, priimta 2006 m. rugpjūčio 16 d.

- W3C – pasaulinio tinklo konsorciumas (šiai organizacijai priklauso beveik visos didžiosios IT kompanijos)
- http://www.w3.org/
- W3C kuria technologijas (specifikacijas, įrankius, ir t.t.), vienaip ar kitaip susijusias su Internetu
- Specifikacijos pereina šiuos paruošimo etapus:
  - Working draft – darbinis juodraštis
  - Candidate recommendation – kandidatas į rekomendaciją
  - Proposed recommendation – siūloma rekomendacija
  - Recommendation – rekomendacija (priimtas standartas)

W3C specifikacijos/darbo grupės

- Accessibility
- Amaya
- Annotea
- CC/PP
- Compound Document Formats
- CSS
- CSS Validator
- Databinding
- Device Independence
- DOM
- Efficient XML Interchange
- Health Care and Life Sciences
- HTML
- HTML Tidy
- HTML Validator
- HTTP
- Incubator
- InkML
- Internationalization
- Jigsaw
- Libwww
- MathML
- Mobile Web Initiative (W3C-MWI)
- Multimodal Interaction
- OWL
- Patent Policy
- PICS
- PNG
- Privacy and P3P
- Quality Assurance (QA)
- RDF
- Rich Web Clients
- Rules
- Semantic Web
- SMIL
- SOAP/XMLP
- SPARQL
- Style
- SVG
- Timed Text
- URI/URL
- Validators
- Voice
- WAI
- Web APIs
- Web Application Formats
- Web Architecture (TAG)
- WebCGM
- Web Services
- XForms
- XHTML
- XLink
- XML
- XML Base
- XML Encryption
- XML Key Management
- XML Processing
- XML Query
- XML Schema
- XML Signature
- XPath
- XPointer

XML šiandieniniame pasaulyje

- XML formatas labai sparčiai išplito visame pasaulyje – didžioji dauguma sistemų šiandien duomenimis apsikeičia būtent šiuo formatu
- XML taikymo pavyzdžiai:
  - Dokumentų rengimo programinė įranga:
    - Microsoft Office – dokumentai saugomi XML formatu nuo 2007 ofiso versijos
    - OpenOffice – dokumentai irgi saugomi XML formatu
  - Programuotojams skirtos technologijos:
    - XHTML, SOAP (web services), RSS, Atom, ir daug kitų

XML dokumento pavyzdys

```
<?xml version="1.0" encoding="UTF-8"?>
<KnygųSąrašas>
   <knyga kalba="en">
      <autorius>Eric van der Vlist</autorius>
      <pavadinimas>XML Schema</pavadinimas>
      <metai>2002</metai>
      <ISBN>0-596-00252-1</ISBN>
   </knyga>
   <knyga>
      <!-- Čia komentaras -->
      ...
   </knyga>
   ...
</KnygųSąrašas>
```

Atributas

Žymė (elementas)

XML standartas nusako taisykles, kurioms turi paklusti visi XML dokumentai

Duomenų aprašai XML formate

- XML formatas turi dvi duomenų aprašų rūšis:
  - žymės (angliškai *tag / element*), pvz.:
    - **<autorius>**Jonas Jonaitis**</autorius>**
    - čia „Jonas Jonaitis“ yra duomuo, „autorius“ yra duomens aprašas – žymė, „<“ ir „>“ yra skirtukai
  - atributai, pvz.:
    - <knyga **kalba**="lt">
    - čia „lt“ yra duomuo, „kalba“ yra duomens aprašas – atributas, „=“ yra skirtukas
- Atributai privalo būti paskelbti žymės viduje

Detaliau apie žymes

- Angliškai: *tag* / *element*
- Žymė susideda iš trijų dalių:
  - *atidarančios* žymės, apskliaustos skirtukais „<“ ir „>“
    - pvz.: <autorius>
  - žymės *turinio*
    - duomenys ir/arba kitos žymės
    - pvz.: Jonas Jonaitis
  - ir *uždarančios* žymės, apskliaustos skirtukais „</“ ir „>“
    - pvz.: </autorius>
- Atidaranti žymė žymi duomenų pradžią, uždaranti žymė – duomenų pabaigą
  - `<autorius>Jonas Jonaitis</autorius>`

Žymės gali apimti kitas žymes

- Žymės viduje gali būti arba duomenys, arba kitos žymės
- Pvz.:

```
<knyga>
    <autorius>Jonas Jonaitis</autorius>
    <pavadinimas>Raudonkepuraitė</pavadinimas>
    <metai>2002</metai>
    <ISBN>0-596-00252-1</ISBN>
</knyga>
```

- Čia žymė *knyga* viduje turi keturias žymes: *autorius*, *pavadinimas*, *metai* ir *ISBN*
- Kitaip sakant, žymės gali formuoti *hierarchinę struktūrą*

Detaliau apie atributus

- Atributas yra nebūtina žymės sudėtinė dalis, susidedanti iš trijų dalių:
  - atributo pavadinimo
    - pvz.: kalba
  - skirtuko „=“
  - duomenų
- Pvz.: `kalba="lt"`
- Atributai privalo būti paskelbti kokios nors žymės viduje:
  - Pvz.: `<knyga kalba="lt">`

Tuščios žymės

- Jei žymė neturi duomenų (žinios, kad duomenų nėra, irgi yra informacija), galima sutrumpinta notacija
- Vietoj:
  - `<pageidavimai></pageidavimai>`
- Galima rašyti:
  - `<pageidavimai/>`
- Tuščios žymės (kaip ir normalios) gali turėti atributus:
  - `<pageidavimai kalba="lt"/>`

Apribojimai žymėms ir atributams

- XML formatas reikalauja, kad:
  - žymės ir atributų pavadinimuose nebūtų tarpų
  - atidarančios žymės pavadinimas sutaptų su uždarančios žymės pavadinimu
  - viena žymė neturėtų dviejų atributų tuo pačiu pavadinimu
  - egzistuotų *viena ir tik viena* šakninė žymė
  - daug kitų techninių reikalavimų, kurių neaptarinėsime
- Apribojimas apie šakninę žymę yra labai svarbus – jis reiškia, kad XML formato dokumentas yra *griežtos medžio struktūros*

Detalesnis XML apibrėžimas

- XML – žmogui ir kompiuteriui suprantama *hierarchinė plečiama* duomenų aprašymo *meta-kalba*
  - Žmogui suprantama – XML dokumentai yra tekstinio (ne binarinio) formato
  - Kompiuteriui suprantama – visi XML dokumentai turi atitikti tam tikrą *reguliarią gramatiką* (turi tenkinti XML standarto taisykles)
- Meta-kalba – kalba, skirta kitų kalbų kūrimui
  - Pirmoje praktinėje užduotyje jūs turite susikurti **savo** kalbą – savo žymes ir atributus, bei taisykles, kokios žymės turi būti kokių žymių viduje

Hierarchinė, plečiama kalba

- XML – hierarchinė kalba
  - Kiekviena žymė gali turėti vaikines žymes
  - Gali būti tik viena šakninė žymė
    - XML dokumentas yra žymių *medis*
- XML – plečiama kalba
  - Jau sukurtai kalbai galima pridėti naujų žymių, t.y., galima plėsti savo kalbą
  - **Pastaba**: plečiamas yra ne pats XML standartas, o tik autoriaus susikurta kalba!

XML standartas (specifikacija)

- Apibrėžia, kas yra:
  - Žymė (elementas), atributas, komentaras
  - Deklaracija, apdorojimo instrukcija (processing instruction), nuoroda į simbolį (character reference), nuoroda į esybę (entity reference) – apie šiuos nešnekėsim
- Pateikia *reguliarią gramatiką* (EBNF – Extended Backus-Naur Form), kurią turi atitikti kiekvienas XML dokumentas
  - Analogija iš transliavimo metodų kurso: tekstinis dokumentas yra Pascal programa, jei atitinka Pascal kalbos reguliarią gramatiką (turi prasidėti žodžiu `program`, pasibaigti tašku, ir t.t.)
  - Panašiai ir su XML: tekstinis dokumentas yra XML dokumentas, jei jis atitinka XML reguliarią gramatiką

Supaprastinta XML gramatika
(nereikia egzaminui)

```
document ::= prolog element Misc*
prolog   ::= XMLDecl? Misc* (dtd Misc*)?
XMLDecl ::= '<?xml' VersionInfo EncodingDecl?
  S? '?>'
VersionInfo::=S 'version=' '"' 1.0 '"'
EncodingDecl::= S 'encoding=' '"' EncName '"'
Misc    ::= Comment | S
Comment ::= '<!--' Char* '-->'
S       ::= (#x20 | #x9 | #xD | #xA)+
```

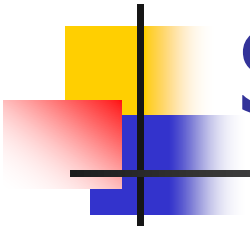
Supaprastinta XML gramatika (2)

```
element ::= EmptyElemTag | STag content Etag
EmptyElemTag ::= '<'  Name Attribute* '/>'
STag    ::=      '<'  Name Attribute* '>'
ETag    ::=      '</' Name '>'
content ::= CharData? (
              (element | Comment) CharData?
            )*
Attribute ::= Name '=' AttValue
AttValue  ::= '"' Char* '"'
Name ::= (Letter | '_' | ':') (NameChar)*
NameChar ::= Letter|Digit|'.'|'-'|'_'|':'
```

Dokumento tipo deklaracija

- XML dokumento pradžioje gali būti to dokumento struktūros apibrėžimas – nuoroda į autoriaus taisykles, užrašytas tam tikra kalba (XML Schema)
  - Kurdami savo kalbą jūs pasakote, kokias žymes jūsų dokumente galima naudoti, bei kokia tvarka, t.y., apibrėžiate taisykles, kurias turi tenkinti jūsų dokumentai
- Taigi yra du taisyklių rinkiniai:
  - XML standarto taisyklės (XML gramatika, ...)
  - XML dokumento autoriaus taisyklės
    - Nusakomos XML Schema arba kitomis kalbomis

Savo XML kalbos kūrimas



- Ką reiškia susikurti *savo kalbą*?
  - Sugalvoti žymių/atributų vardus
  - Sugalvoti, kokios reikšmės bus saugomos žymėse/atributuose
    - Žymėse gali būti kitos žymės ir/arba *duomenys* (tekstas, skaičiai, base64 užkoduoti binariniai duomenys, ...)
    - Atributuose gali būti tik duomenys
  - Nusakyti taisykles, kokios žymės/atributai kur ir kada gali būti naudojami (t.y., apibrėžti jūsų kalba rašomų dokumentų *struktūrą*)
    - Tokios taisyklės aprašomos su DTD, XML Schema arba kitomis dokumento struktūros aprašymo kalbomis

Teisingai struktūrizuoti ir validūs XML dokumentai

- Tekstinis dokumentas vadinamas *teisingai struktūrizuotu* (angl. well-formed) XML dokumentu, jei jis atitinka XML gramatiką (t.y., XML standarto reikalavimus)
- Teisingai struktūrizuotas XML dokumentas, kurio struktūra atitinka išreikštinai nurodytą DTD/XML Schemą, vadinamas *validžiu* (angl. valid) dokumentu
- Pirma pratybų užduotis – sukurti teisingai struktūrizuotą XML dokumentą, antra – patikrinti, ar jis validus (t.y., tenkina jūsų susikurtas autoriaus taisykles)
- Tikrinimą, ar tekstinis dokumentas yra teisingai struktūrizuotas / validus XML dokumentas atlieka programinė įranga, vadinama XML *parseriais* (leksiniais analizatoriais)
  - Šiai dienai dauguma XML redaktorių parserius turi viduje ir dokumentų validumą tikrina automatiškai

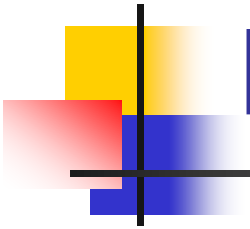
XML 1.0 ir XML 1.1 skirtumai

- Kokius simbolius galima naudoti žymių varduose?
  - XML 1.0 pateikia sąrašą Unicode simbolių, kuriuos galima naudoti (kitus – draudžiama)
  - XML 1.1 teigia, kad galima naudoti visus simbolius, kurių nėra uždraustų sąraše (netgi ir tokius, kurių dabar nėra, bet atsiras ateityje)
- XML 1.1 atpažįsta naujus eilutės pabaigos simbolius - #x85 ir #x2028
- Rekomenduojama naudoti XML 1.0 versiją (nes kol kas didžioji dauguma įrankių palaiko būtent šią versiją)
  - Versija paskelbiama XML dokumento antraštėje:
    `<?xml version="1.0" encoding="UTF-8"?>`

Pavyzdys, kaip nereikia daryti

- Bandoma aprašyti geografinę šalių informaciją:

```
<Lietuva>
  <Vilnius>
    <GyventojųSk>100000</GyventojųSk>
  </Vilnius>
  <Kaunas>
    ...
  </Kaunas>
  ...
</Lietuva>
<Graikija>
  ...
</Graikija>
```



Kodėl?

- Kaip savo kalbos autoriai mes pasiūlome **baigtinį** žymių rinkinį, kurio turi pakakti mūsų kalbos naudotojams
  - Šioje kalboje mes turime tokias žymes:
    - `Lietuva`, `Graikija`, `GyventojųSk`, ...
- Mūsų kalbos naudotojai **negalės** aprašyti Latvijos ar Bulgarijos, nes mes **neduodame** tokių žymių!!!
  - T.y., mūsų kalba visiškai netinkama žmonėms, kurie nori aprašyti kitas šalis

Pataisytas galimas variantas

```
<Valstybė pav="Lietuva">
  <Miestas pav="Vilnius">
    <GyventojųSk>...</GyventojųSk>
  </Miestas>
  <Miestas pav="Kaunas">
    ...
  </Miestas>
</Valstybė>
<Valstybė pav="Graikija">
  ...
</Valstybė>
...
```

Paaiškinimai

- Dabar jau kalba susideda iš tokių žymių:
  - `Valstybė`, **`Miestas`**, `GyventojųSk`, ...
- Panašiai kaip programavimo kalbose yra *kintamųjų tipai* ir jų *reikšmės*, taip ir savo kalboje turite turėti žymes kaip kintamųjų tipus, o reikšmes kaip duomenis (žymių/atributų reikšmes)

Taip daryti irgi yra negerai

```
<Miestas_1>Vilnius</Miestas_1>
<Miestas_2>Kaunas</Miestas_2>
...
<Miestas_15>Marijampolė</Miestas_15>
```

- Problema ta pati: jei jūsų kalbos naudotojas norės turėti daugiau nei 15 miestų, jis negalės pasinaudoti jūsų kalba, nes jūs leisite turėti TIK 15 miestų
- Trumpai sakant, numeracijos daryti nereikia – jei bus reikalinga numeracija, sunumeruos programinė įranga (dažniausiai interneto naršyklė) automatiškai (apie tai kalbėsime kitose paskaitose)