VERES MARCELL EDWIN

JAVA XML alapú parserek

SAX Simple API for XML

Legegyszerűbb API Programozó API

XML documentumot részegtárolva jelent meg
bonyolultabb (összetettebb) API-k épültek rá
~~DOM~~
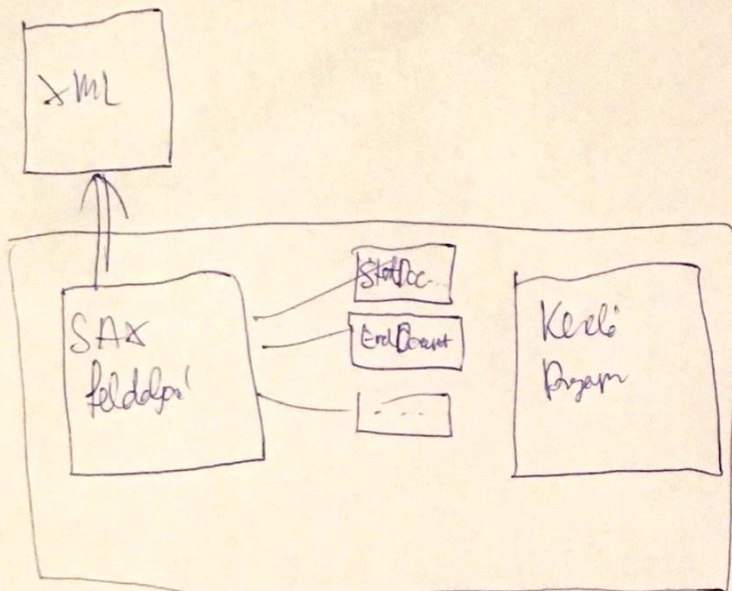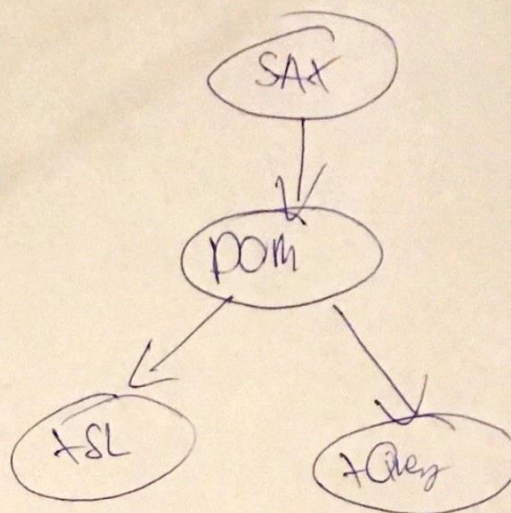
Metódusok: Start Element Stet Dccument
end Element end Document attribútum
Pozició kezelő adapterem előrése nevek

Dom Parser

Lépések JAVA-ban
Document dtó létrehozása
Tartalomkezelő" leak létrehozása
~~Feldolg~~ Eseménykezelő metódusok létehozása
Hibakezelő" metódusok létrehozása

SAXException

XPATH

Részdokumentumok elérése. Kydras műveS (érték, hej. tartalom alapján)

adattípusok: operátorok:

- numerikus   + -   * div mod

- logikai   = != < >   and or not()

- szöveg   Csomópontok tulajdonsága:

- csomópont - halmazok   - neve
                                          - gyerek
csomópontok: - testvér
                                          - ősök (... szülő nila nilage ...)
document van neve, tartalom az összes lefoglalt szövegcsomópont
                                          - leszármazottak (gyerek gyerekek ...)
elemek van neve

entity

attribútum van neve tartalma a tulajdonság érteke

comment   nincs neve

PI   nincs neve

elemi XPATH

  tengelytípus :: csomópont - művelet [szelekció]

  /   \   \   \ további művelet tartalom (érték)
  tágy regenálós   művelet (név/típus alapján)   alapján

Tengelytípusok:

- Self: barkesek csp (csomópont)

- ancestor barckelen ős

- child barckelen leszármazott

- following sibling barckelő testvér csp

- following barckelő / barckelen

- descendant or self leszármazott (befoglalt) és barkeles csp

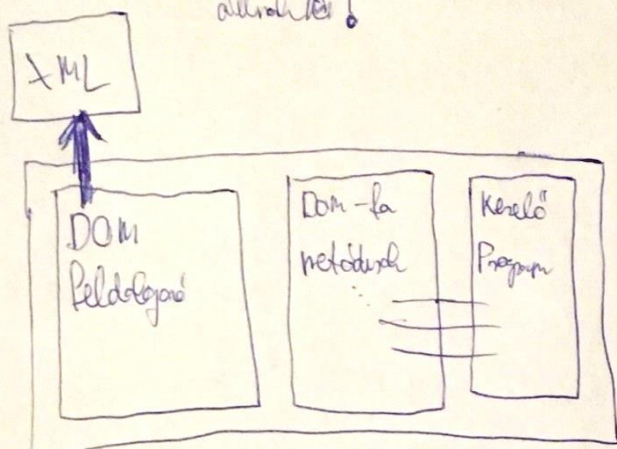- ancestor or self lefoglaló és barkeles csp

③ DOM Szabvány

DOCUMENT OBJECT MODEL

A dokumentumból egy objektum-fát állít elő. Ebből előállítható XML.
SAX-ra épül. Nehezebb hatékonyan implementálni. Memóriaigényes. Elterjedt.
Navigációs jellemű. ~~írás + olvasás~~ Olvasás + módosítás (SAx alapból csak olvasás)

Az elemek tartalmazási relációban állnak !

Lépések:

Dokumentum olvasó létrehozása
Objektum-fa létrehozása
elemek elérése (navigáció)
műveletek végrehajtása (metódusok)
Serializálás

Metódusok: ⓐ document

Normalize()
getDocumentElement()
getElementsByType()

ⓑ csomópontok:

getTextContent   SetTextContent
getTextSize   hasAttributes
getFirstChild   ~~~~
getNodeType
:

Serializálás lépései:

forrásleírás létrehozása
célleírás létrehozása
Serializer objektum létrehozása
Serialize() meghívása

Navigáció
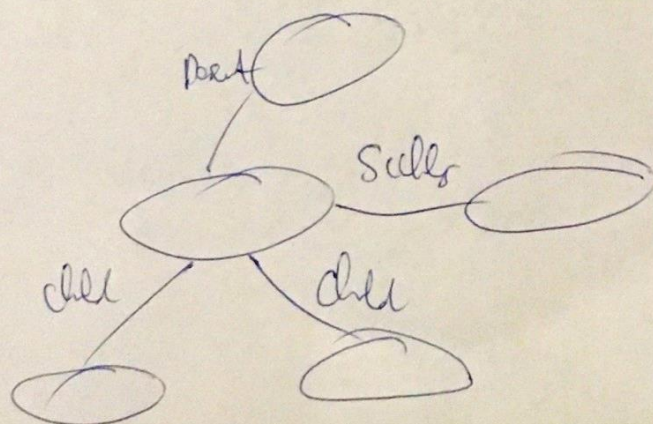
VEDES
MARCELL
JEPU 1W

④ XML Sémák Nevelése

Sémák szerepe:
- műveletek korlátozása
- integritás őrzés (validáció)
- előfordulások korlátozása
- előfordulások ellenőrzés

Sémák Típusai:
- statikus (szerkezet + értékek ellenőrzése)
- dinamikus (műveletek ellenőrzése)
- lokális elemre építő
- globális elemre építő
- sorrend kereszthetlen (sorrendnek nincs nele)
- kevert (kód optimalizálás)
- szigorú (kötelező)
- opcionális

elemi : DTD
belátó : XSD
spec : Schematron

XML Schema:
korán kapcsolódik az XML-hez
opcionális

DTD

egyszerű szerkezeti elemek
nincs
globális elemek
nincs sorrend
korlátozott integritási feltételek
nincs adattípus megkötés
nincsen abszolút típusok

VS

XSD

összetett szerkezeti elemek
kevert
lokális elemek
van sorrend
gazdag integritási feltételek
van adattípus megkötés
abszolút típusok vannak
megalapozott struktúra
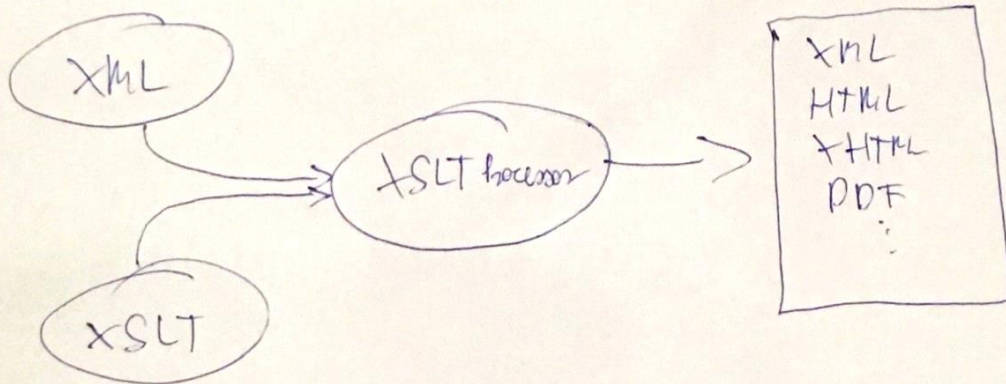modularitás

VERES
MARCELL
JEDVIN

XSLT nyelvy részei:

XSLT (eXtensible Stylesheet Transformation Language)
XSL-FO (XSL Formatting Object) – formázás
XPATH   navigáció
+Query   lekérdezések

Konvertálás történik, de egy új fileba. XSLT processzer régi.

Parancsok:

1, Csoport kiválasztás:
   `<xsl:value-of select="lft"/>`

2, Csoportképzés:
   `<xsl:     select="sel.lhrd's-lhepzés" group-by="..."/>`

3, Rendezés:
   `<xsl:sort select="..."/>`

4 Kimenet beállítása:
   `<xsl:output></xsl:output>`