

2021.01.12.

Bajusz Máté Bence

BMA015

① Java XML alapú parserek

SAX, DOM, XPath

A SAX egy egyszerű parser, de az alkalmazások kódja  
relatíven olthatatlan. Nem tud módosítani.

A DOM egy fát épít fel a memóriában, egyszerűen kezelni  
és több funkciót birköt mint a SAX.

Az XPath -el könnyen lehet keresni, lekérdezni  
esmáolni XML dokumentumokban.

2021. 01. 12.

- ② X.Path: Az XML dokumentumot fában kezeli, nagyban hasonlít a DOM API működéséhez. Az XML fában az elemek előre elérésére, kiválasztására használjuk.

A fa csomópontjából csomópontjainak egy halmazára tudunk szűrni az X.Path-ös kifejezésekkel, tehát megadunk egy mintát és ami teljesen elérhető amely csomópontok teljesítik a feltételeket, bekerülnek ebbe a halmazba. A kereséshez minden értelmezett egy kontextus csomópont

~~állás~~ Keresés működése: Megoldjuk a navigációs területet, majd az elérhetők lebőt szűkitni, végül egyszerű feltételeket adhatunk neg. pl. terület:: tipusszín [egyszerű feltétel]

kifejezések:

- \*
- text() a
- @ rev
- @ \*
- rev[1] a kontextus csomópont első gyerek elérhető
- rev[last()] a kontextus csomópont utolsó gyerek
- rev1 // rev2 a kontextus csomópont rev1 & rev2 gyerekeivel.

Területi típusok:

- self
- child: gyerek csomóponttal
- ancestor
- parent: szülő csomópont
- descendant:
- descendant-or-self
- attribute: a kontextus csomóponthoz tartozó elemjellemzők
- self: őnmaga

2021.01.12.

Bognár Lehel Béna

BMA015

③ A DOM a SAX-nál több funkciót biztosít.

A memóriában felépíti a teljes XML dokumentumot. Egy fa reprezentálja az XML dokumentumot. Az XML dokumentum manipulálására szolgál. Objektumból kezelhetők az elemek vagy a nemrátárgyai, egyszerű implementálva az API-t.

működése / feldolgozás: Első lépében \* átolvassa a megadott XML dokumentumot, majd a memóriában felépíti hozzá az objektum fát.  
Egy nagyobb a végrehajtás ideje mint SAX \*-nál, de fellelhető.

Interface elemek:

- Document
- Node
- NodeList
- Attr
- Element
- DomException

Document: \* adattagok:

- documentElement
- doctype

metódusok:

- createElement (String)
- getElementsByTagName (String)

Node: adattagok:

- nodeType
- childNodes
- parentNodes

metódusok:

- removeChild (Node)
- appendChild (Node)

DomException: adattagok:

- error

9.

2021. 01. 12.

Bajn Máté Benő

BMA 075

XML Schema nem annyira egyszerű mint a DTD, de több funkciót enged meg ~~nem szabott~~.

Pt. támogatja a részterek bevezetését, névenkettő típusokat.

Az XML Schema alapján validálhatunk XML dokumentumot, így jó arra hogy leírja hogyan nézzen ki egy XML dokumentum. Az XML Schema külön XML dokumentumban definiálható, ~~egy felügyezésen függő XML dokumentumban~~

DTD: A dokumentum szerkezetet korlátozza. Ez előny is mint ennek egyszerű, de hátrány is mert csak korlátozza, nem elég kölcsönösek a meghatározások. Pt. nincs részter és típusok elnevezések: element, attribute, entity

element szerkezeti részlete: (! Element elemneve szerkezet)

...  
milyen elemek fordulnak elő  
előfordulási sorrend

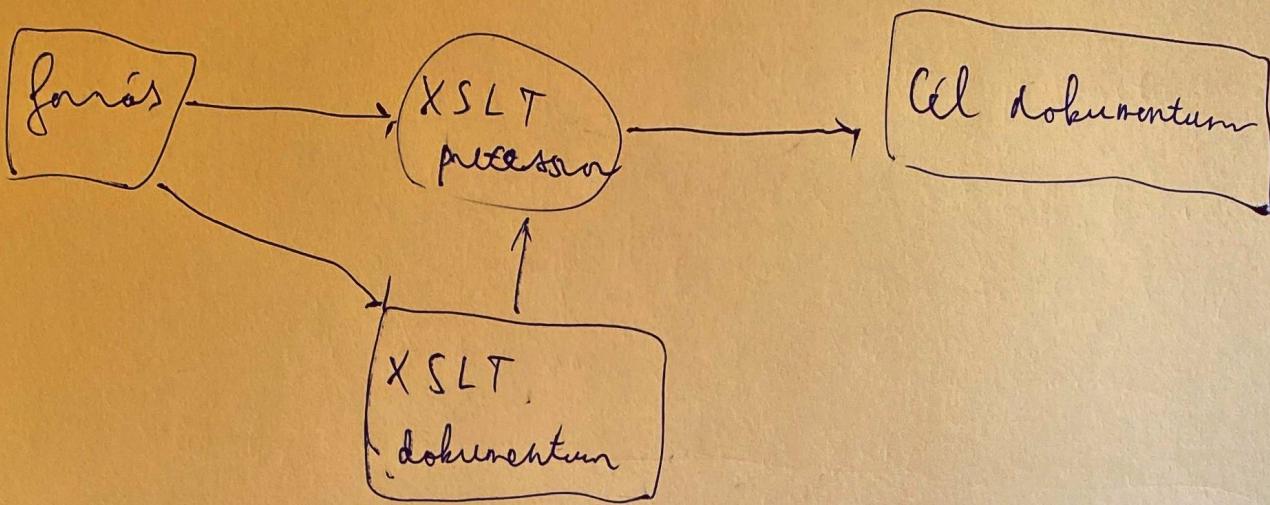
5

2020

2021. 01. 12.

Bajnok Máté

BM40 15



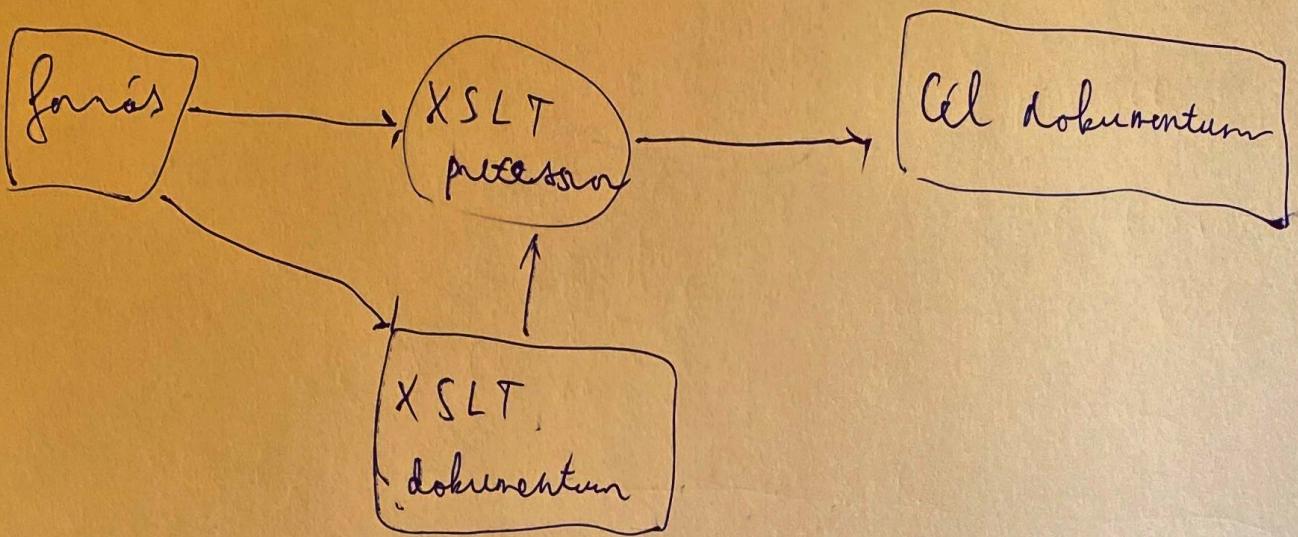
- Van módont
- fa alapú dokumentum
- \* Mintakellentes alapján ~~söderk~~ létrehozva a cél dokumentumot

5

2020

2021. 01. 12.

Bajusz Máté  
BMA015



- Nem módosít
- fa alapú dokumentum
- \* Mintaillesztés alegyén ~~szöveg~~ letrihozza a cél dokumentumot