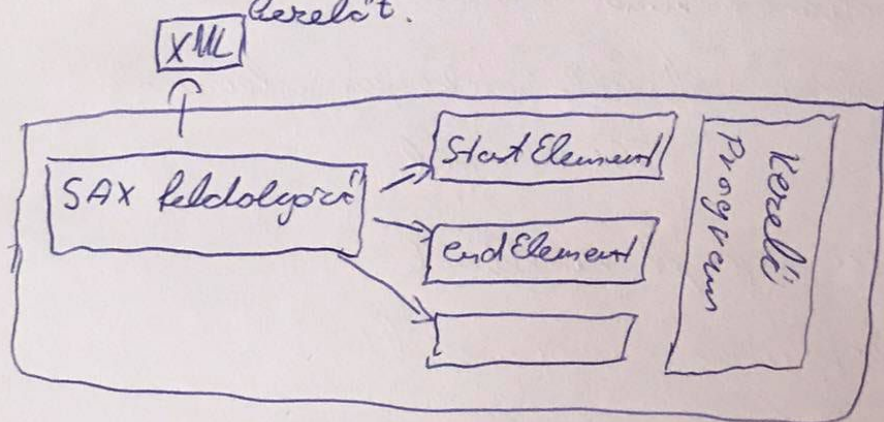


Donga Dániel
PIUKMI

2021. 01. 12

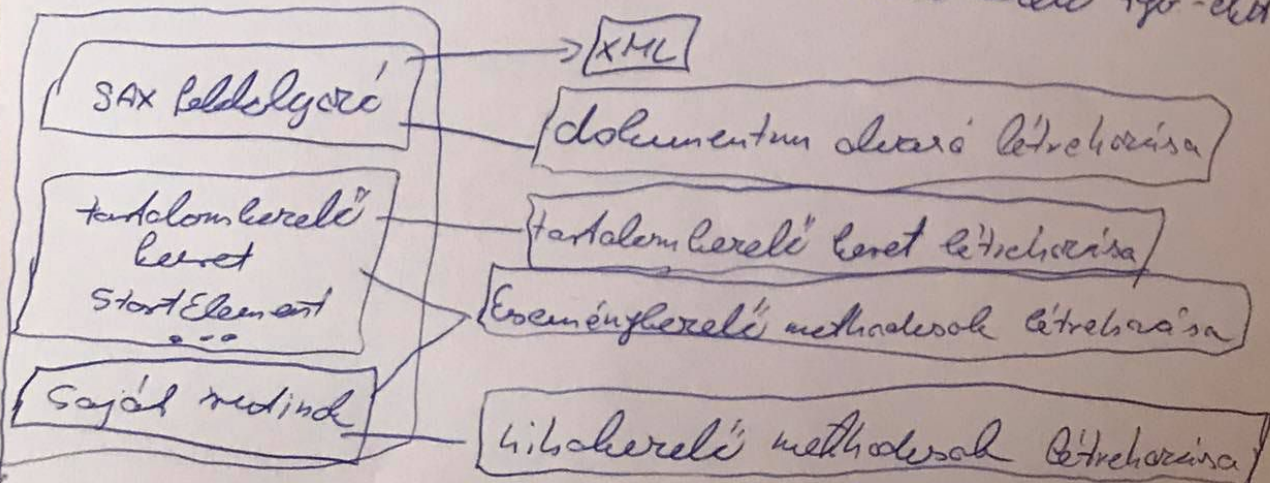
1. feladat: JAVA XML alapú Parser:
SAX modell működése (Simple API for XML)

- Az értelmező szekvenciálisan feldolgozza az XML dokumentumot.
- Minden fontosabb eseményről Callback mechanizmussal értesíti a kérést.



Elterjedt szabvány, könnyű implementálási, csak olvasási műveletet igénylő feldolgozóval támogatja kiemelkedően.

- A feldolgozó biztosít egy szabálykönyvtárt, melyen amin keresztül definiálhatók a Callback kérések fgy-el.



Donga Dániel

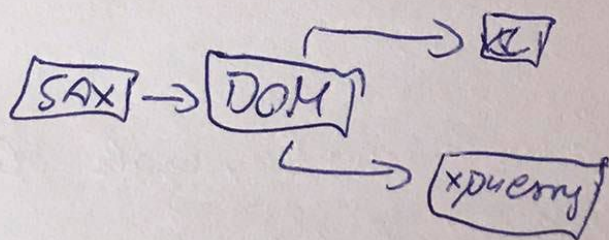
2021. 01. 12

PIUMMI

1. feladat folytatása:

II, DOM (document object Model)

- Mind az olvasási mind a módosítási műveleteket támogatja
- elterjedt szabvány, wherebb többek között implementációi, SAX-ra épül.
- Nagy memória igényel rendelkezésre
- több forráshoz is implementált



Donga Dániel
PIUMMI

2021.01.12

2, XPATH: Rendszeres elnevezések biztosítása név, pozíció, érték alapján.

- tengelytípusok:
- self: maga a kontextus csomópont
 - child: gyermek csomópont
 - descendant: befoglat csomópontok
 - descendant-or-self: befoglat csomópontok + kontextus
 - Parent: szülő tartalmazó csomópont
 - ancestor: a befoglaló csomópontok
 - ancestor-or-self: a befoglaló csomópontok és kontextus
 - namespace: névtér
 - attribute: elemjelhez
 - following: következő csomópontok.

- Kapcsolatok:
- Navigációs tengely: Megadja a keresés fő irányát.
 - Csomópont szűrés: Csomópontok halmazából szűkíteni lehet csomópont név vagy típus alapján.
 - Selektálás feltétel: további szűrés

Megadása:

tengely:: típus - szűrés [szelők]

összetett kifejezés: elem-hf / elemi-hf / ...

Adatmodellje:

- gyökér csomópont: tartalmaz a dokumentum szöveg csomópont kálendenciáját
- elem csomópont: tartalmaz a befoglalt szöveg csomópontok kálendenciáját
- elemjelhez csomópont: tartalmaz a jelhez értéke
- szöveg csomópont: tartalmaz szövegrész
- névtér csomópont: névtér alias neve

Operátorai

- kivonás
- és
- mod
- összekapcsolás
- csomópontokhoz expozíció |
- át
- szűrés
- logikai operátorok (kisebbségi, .. és, vagy)

Donga Dániel
PIUMVI

2021.01.12

2. feladat Selyfában

A tengelyekre vonatkozó szűrés történelme

new alapján: new

newer: new

*
newer: *

tipus alapján:

node()

text()

Processing instruction()

A megadott tengely mentén lévő csomópontok halmozása
történi sorrendben, kapcsolódó elemekre is kiterjedő szűrés.

Egy szűrésben lévő kifejezés is tartalmazhat szűrést.

Darga Dániel
PIUMMI

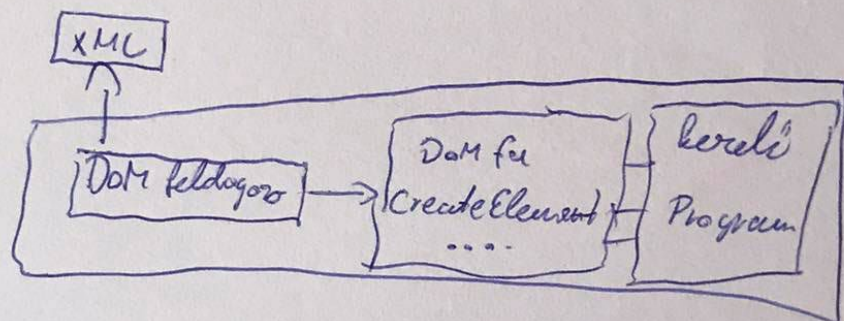
2021.01.12

Dom (Document Object Model)

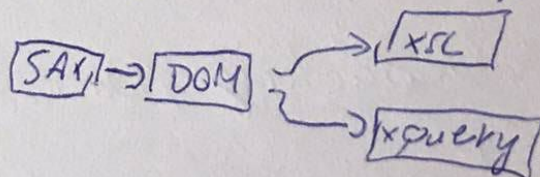
3. DOM

Egy rugalmasan programozható API felület, amelyben az értelmező beolvassa az XML dokumentumot, és egy fát épít fel belőle.

Az előállított fát a módszersein keresztül lehet írni és olvasni, módosítani.

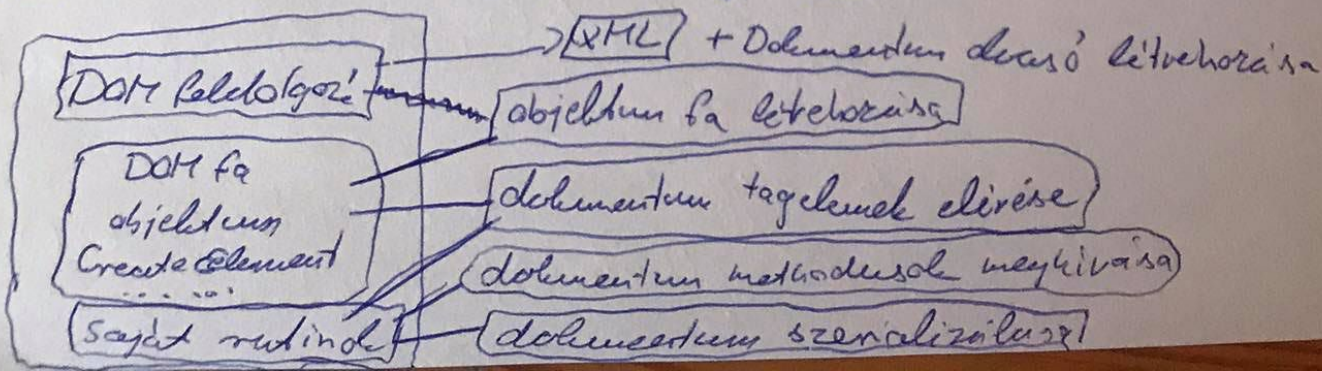


Elterjedt szabvány, amelyet sokszor több hatékonyabban implementáció, a SAX-ra épül. Támogatja az olvasási, módosítási műveleteket.



Több fordításnyelvre is implementált.

A feldolgozó létrehoz egy DOM objektumot, amin keresztül lehet az XML dokumentum tartalmát lekérdezni és módosítani.

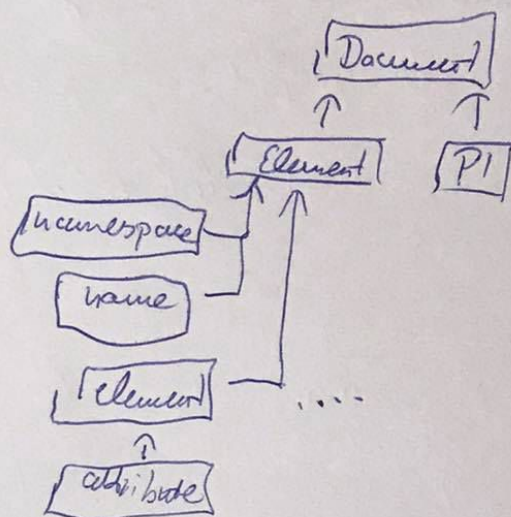


Donga Dániel
PIUMMI

2021.01.12.

3. feladat Fogytatás (DOM)

Az objektumok közötti tartalmi viszonyok relációi áll fent



Documentum keresés

DocumentBuilderFactory, DocumentBuilder
segítségével

Legfontosabb elemek a document osztályban:

CreateElement(str), CreateTextNode(),
getDocumentElement(), CreateAttribute(), CreateEntityReference(),
insertBefore(), getXMLVersion(),

Gainápon (node) osztály legfontosabb elemek: getChildNodes(),
appendChild(Node), getParentNode(), getAttribute(),
removeChild(), hasChildNodes(),

Darga Dániel
PIUMNI

2021.01.12

4. feladat

- jelölé" elem megadása:
DTD elemei: $\langle !ELEMENT \text{ elem-név } \text{szorheret} \rangle$

ahol a szorheret típusai:

EMPTY, ANY, PCDATA

- Elemjellőr megadása
 $\langle !ATTLIST \text{ elem elemjell-név típus érté} \rangle$
ilyen típus lehet:

ID, IDREF, CDATA, IDREFS,
NOTATION

- Szimbólum megadása

$\langle !ENTITY \text{ szimbólum "érték"} \rangle$

$\langle ENTITY \text{ szimbólum SYSTEM "állomány"} \rangle$

- Notation külső programot [feldolgozót] azonosító.

$\langle !NOTATION \text{ név System feldolgozó} \rangle$

- feltétel nél semavész

előnye:

- egyszerű
- kevés megjelölés
- elemi funkcióvalitást
teljesíti.

hátránya: nem támogatja a
különböző adattípus hierarchiát

- túl kicsi ID és IDREF expressiók
- szintaktikája nem felel meg XML szintak-
tikájának

Donga Dániel

2021.01.12

PIUMNI

4. bevezetés

XML Schema : nevezett típusok, gyakori struktúra elemek
szerkezet (első), integritási elemeket tartalmaz

Elem csoport szerkezet megadása

`<xs:element jellenő>` tartalmaz `</xs:element>`

jellenő : name, type, ref, abstract, ...

Elemi szerkezetek: atom, lista, unió

Donga Daniel
PIUMMI

5. XSLT

- Csoporthézés

```
xsl:for-each-group select="clench" group-by="csoporthézés  
kéfére">
```

```
<!-- tartalom -->
```

```
</xsl:for-each-group>
```

- Feldolgozás címlere

```
<xsl:for-each select="seller's list">
```

```
<!-- feldolgozás identitása -->
```

```
</xsl:for-each>
```

- Csomópontok feldolgozásának sorrendje állítása:

```
<xsl:sort select="list1" order="list2" collection="list3"/>
```

- Input állományok additional atributei:

```
<xsl:value-of select="list1"/>
```

- Függvények feldolgozása specifikus felé:

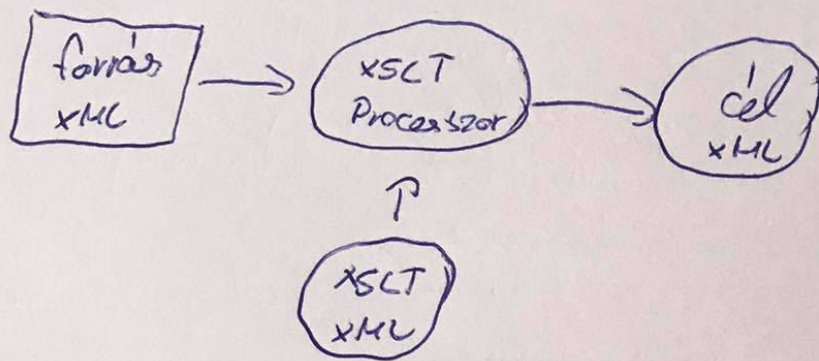
```
<xsl:apply-templates select="seller's list" />
```

render

```
</xsl:apply-templates>
```


Douga Daniel
PIUMVI

5. feladat



- megismerhető dokumentum fa felépítés
- megismerhető variáció
- struktúra és tartalom szétválasztás