JEGYZŐKÖNYV

Adatkezelés XML környezetben
Féléves feladat
Vállalat nyílvántartás

Készítette: **Harsányi Balázs**

Neptunkód: **HD9DOJ**

Dátum: 2024.12.02.

Tartalomjegyzék

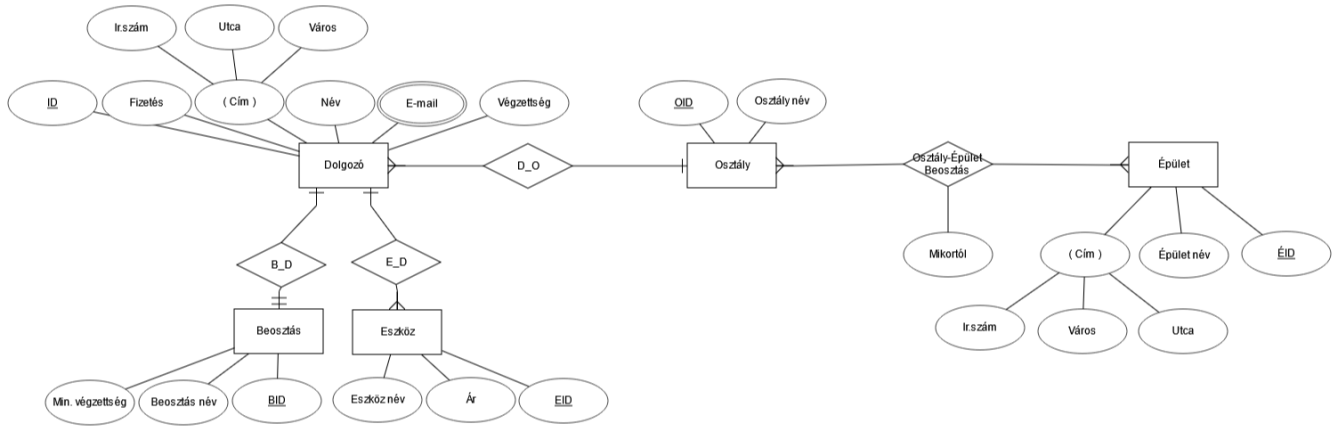
Bevezetés

A feladatban egy vállalat alkalmazottainak adatai nyilvántartására egy XML adatkezelő rendszert kellet készíteni. Ebben a rendszerben dolgozók, a vállalati eszközök, beosztások, a vállalati osztályok és a vállalat épületeiről találhatóak információk. A dolgozókról a nevük, fizetésük, legmagasabb végzettségük szintje, email címeik (mind vállalati és személyes) és a címük adatai találhatóak. A cím adataiba beletartozik a település, irányítószám és az utca a házszámmal együtt. Az eszközökről a nevük és az árukról található adat. A beosztásról az álláshoz betöltendő minimum végzettség és a beosztás neve kell. Az osztályokról csak a név szükséges. Az épület adataiban az épület neve és címeiknek adatai találhatóak. Még az az adat is szükséges, hogy az osztály mióta használja a megadott épületet. A rendszerben továbbá benne kell lennie, hogy melyik dolgozók vannak egy osztályban, egy dolgozónak milyen vállalati eszközök vannak biztosítva, egy dolgozó melyik beosztást látja el, és az melyik osztályok dolgoznak melyik épületekben. A rendszer megvalósításához alkotni kell elsőnek egy adatbázishoz tartozó ER modell. Ez után az ER modell alapján kell készíteni egy XDM modellt. Az XDM modell alapján egy XML dokumentumot kell készíteni, amiben már ki vannak töltve az adatok. Az XML dokumentumhoz kell készíteni egy XMLSchema dokumentumot, amivel validálni kell az XML dokumentumot.

1. Feladat

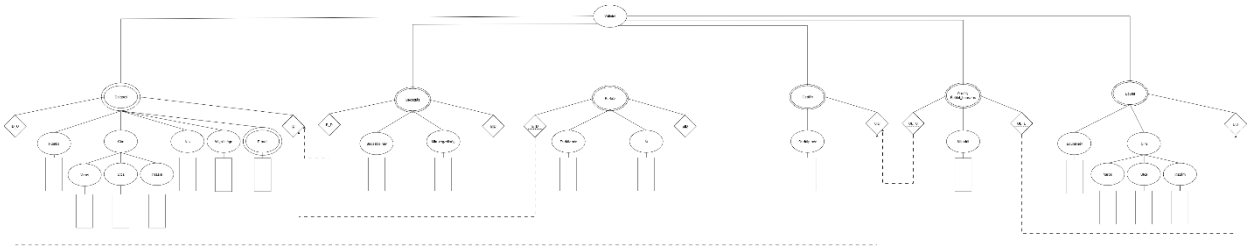
1.1 Az adatbázis ER modell tervezése

A ER modellbe dolgozók, az eszközök, beosztások, az osztályok és az épületek vannak egyedként megjelenítve. Ezekhez tartozó adatok tulajdonságokként vannak megjelenítve. A dolgozó email tulajdonsága több értékű tulajdonságként van megjelenítve és a dolgozó és épület cím tulajdonsága összetett tulajdonság. Minden egyedhez tartozik még egy ID tulajdonság. A dolgozó és osztály között N:1 kapcsolat, a dolgozó és eszköz között 1:N kapcsolat, a dolgozó és beosztás között 1:1 kapcsolat, az osztály és épület között N:N kapcsolat van. Az osztály és épület kapcsolathoz tartozik a mikortól tulajdonság.



1.2 Az adatbázis konvertálása XDM modellre

Az XDM modell egy vállalat elemmel kezdődik, amiből származnak a dolgozó, az eszköz, beosztás, az osztály és az épület elem, ezeken túl a több-több kapcsolat megvalósításához osztály-épület beosztás is szerepel, mindez dupla ellipszissel jelölve, mert többször is szerepelhetnek. Ezekből az elemekből még az ER modellben szereplő tulajdonságaik elemekként származnak belőlük (kivéve a ID tulajdonságok), és a E-mail elem dupla ellipszissel van jelölve, mert többször is szerepelhet a dokumentumban. A tulajdonság elemek alatt téglalappal van jelölve a szöveg, kivéve a cím tulajdonságok, amik részei további elemekként vannak jelen, és azok alatt vannak a szövegek jelölve. A direkt vállalat elemből származó elemeknek van rombusszal jelölve egy ID attribútuma, kivéve az osztály-épület beosztásnak. A dolgozónak, eszköznek, beosztásnak van egy idegen kulcs attribútuma és a osztály-épület beosztásnak két idegen kulcsa van, amik az ER modellben szereplő kapcsolatok alapján a megfelelő helyre mutatnak.



1.3 Az XDM modell alapján XML dokumentum készítése

Az XML dokumentum írása a szövegfeldolgozó utasítással, majd a vállalat gyökérelem írásával, annak a névterének és (a még nem létező) XMLSchema helyének megadásával kezdődik. Ez után XDM fában lefelé haladva, megy a dokumentum készítése, vagyis a dolgozó elem írása, annak attribútumainak megadása, utána a dolgozóból származó elemek írásával történt. Így történt minden ágnál.

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

<!-- A gyokerelem nevterrel es a schema helyevel  -->

<Vallalat xmlns:xs=http://www.w3.org/2001/XMLSchema-instance
xs:noNamespaceSchemaLocation="XMLSCHEMAHD9DOJ.xsd">

   <!-- A dolgozok adatai  -->

   <Dolgozo id="d1" D_O="o1">
      <Fizetes>300000</Fizetes>
      <Nev>Lakatos Ottó</Nev>
      <Cim>
         <Iranyito_szam>3744</Iranyito_szam>
         <Varos>Kazincbarcika</Varos>
         <Utca>Tancsics Utca 22</Utca>
      </Cim>
      <Vegzettseg>Érettségi</Vegzettseg>
      <E-mail>lakott@gmail.com</E-mail>
      <E-mail>lakO@ceges.com</E-mail>
      <E-mail>lakotthoni@freemail.com</E-mail>
   </Dolgozo>

   <Dolgozo id="d2" D_O="o2">
      <Fizetes>2000000</Fizetes>
```

```
    <Nev>Tóth Balázs</Nev>
    <Cim>
        <Iranyito_szam>2252</Iranyito_szam>
        <Varos>Miskolc</Varos>
        <Utca>Hosok ut 1</Utca>
    </Cim>
    <Vegzettseg>Diploma</Vegzettseg>
    <E-mail>tthB@freemail.hu</E-mail>
    <E-mail>toThB@ceges.com</E-mail>
    <E-mail>tthB@yahoo.hu</E-mail>
</Dolgozo>

<Dolgozo id="d3" D_O="o3">
    <Fizetes>120000</Fizetes>
    <Nev>Tóth Krisztina</Nev>
    <Cim>
        <Iranyito_szam>2252</Iranyito_szam>
        <Varos>Miskolc</Varos>
        <Utca>Hosok ut 1</Utca>
    </Cim>
    <Vegzettseg>Diploma</Vegzettseg>
    <E-mail>ttkr@freemail.hu</E-mail>
    <E-mail>toKrB@ceges.com</E-mail>
    <E-mail>ttKrB@yahoo.hu</E-mail>
</Dolgozo>

<!-- A beosztasok adatai  -->

<Beosztas BID="b1" B_D="d1">
    <Minimum_vegzettseg>Érettségi</Minimum_vegzettseg>
    <Beosztas_nev>Leltár Menedzser</Beosztas_nev>
</Beosztas>

<Beosztas BID="b2" B_D="d2">
    <Minimum_vegzettseg>Diploma</Minimum_vegzettseg>
    <Beosztas_nev>Emberi erőforrások Menedzser</Beosztas_nev>
```

```
</Beosztas>

<Beosztas BID="b3" B_D="d3">
    <Minimum_vegzettseg>Diploma</Minimum_vegzettseg>
    <Beosztas_nev>Programozó</Beosztas_nev>
</Beosztas>

<!-- Az eszkozok adatai  -->

<Eszkoz EID="e1" E_D="d1">
    <Eszkoz_nev>Telefon</Eszkoz_nev>
    <Ar>100000</Ar>
</Eszkoz>

<Eszkoz EID="e2" E_D="d2">
    <Eszkoz_nev>Laptop</Eszkoz_nev>
    <Ar>300000</Ar>
</Eszkoz>

<Eszkoz EID="e3" E_D="d3">
    <Eszkoz_nev>Laptop</Eszkoz_nev>
    <Ar>300000</Ar>
</Eszkoz>

<!-- Az osztalyok adatai  -->

<Osztaly OID="o1">
    <Osztaly_nev>Leltár</Osztaly_nev>
</Osztaly>

<Osztaly OID="o2">
    <Osztaly_nev>Emberi erőforrások</Osztaly_nev>
</Osztaly>

<Osztaly OID="o3">
    <Osztaly_nev>Informatika</Osztaly_nev>
```

```
</Osztaly>

<!-- Az osztalyok és az epuletek osszekotteteseinek adatai  -->

<Osztaly-Epulet_beosztas OE_O="o1" OE_E="ep1">
   <Mikortol>2001-12-31</Mikortol>
</Osztaly-Epulet_beosztas>

<Osztaly-Epulet_beosztas OE_O="o2" OE_E="ep2">
   <Mikortol>1999-04-12</Mikortol>
</Osztaly-Epulet_beosztas>

<Osztaly-Epulet_beosztas OE_O="o3" OE_E="ep3">
   <Mikortol>2000-08-10</Mikortol>
</Osztaly-Epulet_beosztas>

<!-- Az epuletek adatai  -->

<Epulet ÉID="ep1">
   <Epulet_nev>Raktár 1</Epulet_nev>
   <Cim>
      <Iranyito_szam>3704</Iranyito_szam>
      <Varos>Berente</Varos>
      <Utca>Berente út 5</Utca>
   </Cim>
</Epulet>

<Epulet ÉID="ep2">
   <Epulet_nev>Fő épület</Epulet_nev>
   <Cim>
      <Iranyito_szam>3744</Iranyito_szam>
      <Varos>Kazincbarcika</Varos>
      <Utca>Kazinc út 5</Utca>
   </Cim>
</Epulet>
```

```
  <Epulet ÉID="ep3">
    <Epulet_nev>Informatika épület</Epulet_nev>
    <Cim>
      <Iranyito_szam>3744</Iranyito_szam>
      <Varos>Kazincbarcika</Varos>
      <Utca>Petőfi út 2</Utca>
    </Cim>
  </Epulet>

</Vallalat>
```

1.4 Az XML dokumentum alapján XMLSchema készítése

Az XMLSchema dokumentum írása a szövegfeldolgozó utasítással kezdtem, majd a schema gyökérelemként szerepel a dokumentumban. Ez után a XML dokumentum gyökéreleme szerepel, és ennek a típusában szereplő elemeknek a típusai saját típusként vannak megadva, és ezek csak később lesznek a dokumentumban. Ez után a gyökér elem elemeinek kulcsai vannak, majd az idegen kulcsok vannak megadva. A kulcsok után a dokumentum úgy van felépítve, hogy egy komplex sajáttípus van leírva, és az ebben lévő elemek, ha complex típusok, akkor sajáttípusként, ha egyszerű típusok akkor referenciaként vannak megadva. A simple típusú elemek közvetlenül a complex típus után vannak felsorolva.

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!-- A gyokerelem elemei  -->

  <xs:element name="Vallalat">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Dolgozo" type="dolgozoTipus" maxOccurs="unbounded"/>
        <xs:element name="Beosztas" type="beosztasTipus" maxOccurs="unbounded"/>
        <xs:element name="Eszkoz" type="eszkozTipus" maxOccurs="unbounded"/>
        <xs:element name="Osztaly" type="osztalyTipus" maxOccurs="unbounded"/>
        <xs:element name="Osztaly-Epulet_beosztas" type="osztaly-epulet_beosztasTipus"
maxOccurs="unbounded"/>
        <xs:element name="Epulet" type="epuletTipus" maxOccurs="unbounded"/>
      </xs:sequence>
```

```
    </xs:complexType>


<!--a fo kulcsok-->

    <xs:key name="dolgozo_kulcs">
        <xs:selector xpath="Dolgozo" />
        <xs:field xpath="@id" />
    </xs:key>

    <xs:key name="beosztas_kulcs">
        <xs:selector xpath="Beosztas" />
        <xs:field xpath="@BID" />
    </xs:key>

    <xs:key name="eszkoz_kulcs">
        <xs:selector xpath="eszkoz" />
        <xs:field xpath="@EID" />
    </xs:key>

    <xs:key name="osztaly_kulcs">
        <xs:selector xpath="Osztaly" />
        <xs:field xpath="@OID" />
    </xs:key>

    <xs:key name="epulet_kulcs">
        <xs:selector xpath="Epulet" />
        <xs:field xpath="@ÉID" />
    </xs:key>

    <!--az idegen kulcsok-->

    <xs:keyref refer="dolgozo_kulcs" name="eszkoz_dolgozo_idegen_kulcs">
        <xs:selector xpath="Eszkoz" />
        <xs:field xpath="@E_D" />
    </xs:keyref>
```

```
    <xs:keyref refer="osztaly_kulcs" name="dolgozo_osztaly_idegen_kulcs">
        <xs:selector xpath="Dolgozo" />
        <xs:field xpath="@D_O" />
    </xs:keyref>

    <xs:keyref refer="osztaly_kulcs" name="OE_osztaly_idegen_kulcs">
        <xs:selector xpath="Osztaly-Epulet_beosztas" />
        <xs:field xpath="@OE_O" />
    </xs:keyref>

    <xs:keyref refer="epulet_kulcs" name="OE_epulet_idegen_kulcs">
        <xs:selector xpath="Osztaly-Epulet_beosztas" />
        <xs:field xpath="@OE_E" />
    </xs:keyref>

    <!-- 1-1 kapcsolat megvalositasa-->

    <xs:unique name="unique_beosztas">
        <xs:selector xpath="Beosztas" />
        <xs:field xpath="@B_D" />
    </xs:unique>

</xs:element>

<!-- tipusok-->


<!--a dolgozo tipus es benne levo elemek-->
<xs:complexType name="dolgozoTipus">
    <xs:sequence>
        <xs:element ref="Fizetes" />
        <xs:element ref="Nev" />
        <xs:element name="Cim" type="cimTipus"/>
        <xs:element ref="Vegzettseg" />
        <xs:element ref="E-mail" maxOccurs="unbounded" />
```

```
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required" />
    <xs:attribute name="D_O" type="xs:string" use="required" />
</xs:complexType>

<xs:element name="Fizetes" type="xs:positiveInteger" />
<xs:element name="Nev" type="xs:string" />
<xs:element name="Vegzettseg" type="xs:string" />
<xs:element name="E-mail" type="xs:string" />

<!--a cimtipus es a benne levo elemek-->
<xs:complexType name="cimTipus">
    <xs:sequence>
        <xs:element ref="Iranyito_szam" />
        <xs:element ref="Varos" />
        <xs:element ref="Utca" />
    </xs:sequence>
</xs:complexType>

<!--az iranyitoszam elem csak magyar iranyitoszamokat fogad el-->
<xs:element name="Iranyito_szam">
    <xs:simpleType>
        <xs:restriction base="xs:positiveInteger">
            <xs:minInclusive value="1000"/>
            <xs:maxInclusive value="9999"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="Varos" type="xs:string" />
<xs:element name="Utca" type="xs:string" />


<!-- a beosztastipus es a benne levo elemek-->
<xs:complexType name="beosztasTipus">
    <xs:sequence>
```

```
        <xs:element ref="Minimum_vegzettseg" />
        <xs:element ref="Beosztas_nev" />
    </xs:sequence>
    <xs:attribute name="BID" type="xs:string" use="required" />
    <xs:attribute name="B_D" type="xs:string" />
</xs:complexType>

<!--az minimum vegzettsegbe csak a Diploma es Érettségi ertekeket fogadja el-->
<xs:element name="Minimum_vegzettseg" >
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="Diploma"/>
            <xs:enumeration value="Érettségi"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="Beosztas_nev" type="xs:string" />

<!-- az eszkoztipus es a benne levo elemek-->
<xs:complexType name="eszkozTipus">
    <xs:sequence>
        <xs:element ref="Eszkoz_nev" />
        <xs:element ref="Ar" />
    </xs:sequence>
    <xs:attribute name="EID" type="xs:string" use="required" />
    <xs:attribute name="E_D" type="xs:string" use="required" />
</xs:complexType>

<xs:element name="Eszkoz_nev" type="xs:string" />
<xs:element name="Ar" type="xs:integer" />

<!-- az osztalytipus es a benne levo elemek-->
<xs:complexType name="osztalyTipus">
    <xs:sequence>
        <xs:element ref="Osztaly_nev" />
```

```
        </xs:sequence>
        <xs:attribute name="OID" type="xs:string" use="required" />
    </xs:complexType>

    <xs:element name="Osztaly_nev" type="xs:string" />

    <!-- az osztaly-epulet_beosztasTipus es a benne levo elemek-->
    <xs:complexType name="osztaly-epulet_beosztasTipus">
        <xs:sequence>
            <xs:element ref="Mikortol" />
        </xs:sequence>
        <xs:attribute name="OE_O" type="xs:string" use="required" />
        <xs:attribute name="OE_E" type="xs:string" use="required" />
    </xs:complexType>

    <xs:element name="Mikortol" type="xs:date" />


    <!-- az epuletTipus_beosztasTipus es a benne levo elemek-->
    <xs:complexType name="epuletTipus">
        <xs:sequence>
            <xs:element ref="Epulet_nev" />
            <xs:element name="Cim" type="cimTipus"/>
        </xs:sequence>
        <xs:attribute name="ÉID" type="xs:string" use="required" />
    </xs:complexType>

    <xs:element name="Epulet_nev" type="xs:string" />


</xs:schema>
```

2. Feladat

2.1 Adatolvasás

A programban elsőnek az XML dokumentumot olvassa be egy doc változóba. Ez után lekéri és kiírja a gyökérelemet. Ez után a gyökérelemből származó elemnek a nevével listába teszem, amivel egy for loopal megyek végig. A for loopban a lista elemét egy node változóba teszem, amit majd element-é konvertálok.

```
Node nNode = nList.item(i);
Element elem = (Element) nNode;
```

Ennek lekérdezem az elemeinek értékét egy-egy node-ba, amiről lekérdezem, hogy mi van beleírva.

```
Node node1 = elem.getElementsByTagName("Fizetes").item(0);
String fizetes = node1.getTextContent();
```

https://github.com/HarsanyiBalazs/HD9DOJ_XML/blob/main/XMLTaskHD9DOJ/DomParseHD9DOJ/src/hu/domparse/hd9doj/DomReadHD9DOJ.java

2.2 Adatírás

A programban elsőnek egy új dokumentumot alkot egy doc változóba. Ez után a gyökérelemet készít, amit beilleszt a dokumentumba.

```
Element root = doc.createElementNS("http://www.w3.org/2001/XMLSchema-instance","Vallalat");

doc.appendChild(root);
```

Ez után a gyökérhez illeszt egy metódust, amiben átadódnak az elem elemeinek az írott tartalma és tulajdonságainak értékei.

```
root.appendChild(createBeosztas(doc,"b1","d1","Érettségi","Leltár Menedzser"));
```

Minden komplex elemnek saját metódusa van. Ezek a metódusok egy új node-t alkotnak és adnak vissza. Erre a node-ra újabb appendChild() metódust hívnak, amiben megint egy metódus van, amiben egy node adódik vissza. Amikor a node-nak nincs gyereke, csak írott tartalma, akkor hívja meg az appendChild()-ban a createBaseElement()-t, amiben az element-be írást illeszt.

```
Element node = doc.createElement(name);

node.appendChild(doc.createTextNode(value));
```

Ezek után kezdődik a fájlba írás. Egy transformert alkot, amivel beállítja a kimenet kódolását és behúzását, egy DOMsource változóba beviszi a dokumentumot.

```
DOMSource source = new DOMSource(doc);
```

Végül a megadott kimenetbe a transformer kiviszi.

https://github.com/HarsanyiBalazs/HD9DOJ_XML/blob/main/XMLTaskHD9DOJ/DomParseHD9DOJ/src/hu/domparse/hd9doj/DomWriteHD9DOJ.java

2.3 Adat lekérdezés

A programban elsőnek az XML dokumentumot olvassa be egy doc változóba. Elsőnek a dolgozók e-mailjeit kérem le. Az összes dolgozó elemet egy listába teszem, amin egy for loopal végig megyek. Egy dolgozó elemben lévő email elemeket egy listába.

```
NodeList emailLista = eElement.getElementsByTagName("E-mail");
```

Az email elemeken egy for loopal megyek végig, amik kiírják a lista elemeinek tartalmát.

```
Element email = (Element) node1;

System.out.print("email: ");

System.out.println(email.getTextContent());
```

A feladat második részében az összes dolgozó nevét iratom ki, akinek 100000-nél többe kerül a céges eszköze. Elsőnek lekérem az összes dolgozó elemet egy listába és az összes eszköz elemet egy listába. Az eszköz elemek ár elemén végigmegy egy for loopal, és ahol az ár értéke

```
Node ar =eElement.getElementsByTagName("Ar").item(0);

int arValue = Integer.valueOf(ar.getTextContent());
```

nagyobb 100000-nél, ott lekéri az eszköz elem idegen kulcsát és egy for loopal végigmegy a dolgozók listáján. Itt megnézi, hogy a dolgozónak az elsődleges kulcsa megegyezik-e az idegen kulccsal, és ha igen lekéri a név elem értékét és kiírja.

A feladat harmadik részében a városok nevét iratom ki. Az összes épület elemet egy listába teszem, amin egy for loopal végig megyek. Egy épület elemben lévő cím elemeket egy listába teszem, amin egy for loopal végig megyek. Egy cím elemben lévő város elemeket egy listába teszem, amin egy for loopal végig megyek. A város elemeken egy for loopal megyek végig, amik kiírják a lista elemeinek tartalmát.

A feladat negyedik részében az osztályok neveit és a főkulcsait írja ki. Az összes osztály elemet egy listába teszem, amin egy for loopal végig megyek. Egy dolgozó elemben lévő email elemeket egy listába. Az email elemeken egy for loopal megyek végig, amik kiírják a lista elemeinek tartalmát. Ez után osztály elem id tulajdonságát kiiratom.

```
String id = oElement.getAttribute("OID");

System.out.print("id: ");

System.out.println(id);
```

https://github.com/HarsanyiBalazs/HD9DOJ_XML/blob/main/XMLTaskHD9DOJ/DomParseHD9DOJ/src/hu/domparse/hd9doj/DomQueryHD9DOJ.java

2.4 Adatmódosítás

A programban elsőnek az XML dokumentumot olvassa be egy doc változóba. Elsőnek a második és harmadik dolgozó kulcsát cserélem át egymáséra. Lekérem node változókba a második és harmadik dolgozó elemeket. Az elemek attribútumait NamedNodeMap váltózókba teszem, ezekből a változókból lekérem az id-kat node-kba, amiknek megváltoztatom az írott tartalmát.

```
Node dolgoz1 = doc.getElementsByTagName("Dolgozo").item(1);
NamedNodeMap attr1 = dolgoz1.getAttributes();
Node nodeAttr1 = attr1.getNamedItem("id");
nodeAttr1.setTextContent("d3");
```

A második részében egy dolgozó nevét írtam át. Az összes épület elemet egy listába teszem, amin egy for loopal végig megyek. Itt listába teszem a gyerek node-kat. Ezen a listán is egy for loopal végig megyek, és amelyik elemnek a neve az nev és az írott tartalmi egyezik az adott dolgozó nevével annak más írott tartalmat adok meg névnek.

```
if("Nev".equals(nElement.getNodeName())){

if("Lakatos Ottó".equals(nElement.getTextContent())){

nElement.setTextContent("Lakatos Ferenc");

}

}
```

A harmadik részében az összes olyan eszköz nevét írtam át számítógéppé, ahol laptop volt. Az összes eszköz elemet egy listába teszem, amin egy for loopal végig megyek. Itt listába teszem a gyerek node-kat. Ezen a listán is egy for loopal végig megyek, és megnézem, hogy az eszkoznev elemnek írott tartalma laptop, és ha igen számítógépet adok meg neki.

A negyedik részében egy betöltve attribútumot adok az összes beosztás elemnek, aminek az értéke igen, ha van idegen kulcsa és nem, ha nincs. Az összes beosztás elemet egy listába teszem, amin egy for loopal végig megyek. itt megnézem, hogy van-e idegen kulcsa és ha igen, akkor betöltve-nek igen lesz az értéke, egyébként nem.

```
if(node.getNodeType() == Node.ELEMENT_NODE){

Element nElement = (Element) node;

if(nElement.hasAttribute("B_D")) {

nElement.setAttribute("betoltve", "igen");

}else {

nElement.setAttribute("betoltve", "nem");

}

}
```

https://github.com/HarsanyiBalazs/HD9DOJ_XML/blob/main/XMLTaskHD9DOJ/DomParseHD9DOJ/src/hu/domparse/hd9doj/DomModifyHD9DOJ.java