

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Dávid Rebeka**

Nepunkód: **EQ4B3D**

A feladat leírása:

A feladatban egy olyan ER modellt csináltam, ami megmutatja egy adott Étterem tulajdonosának, hogyan is működnek a munkafolyamatok a termelés beszerzésétől egészen az étel asztalra kerüléséig. Összesen 6 egyed található benne:

Termelő, Beszállító, Étterem, Személyzet, Vendég és Foglалás

A Termelő egyed tartalmazza az össze információt a termelőről, ahonnan az alapanyagok érkeznek. Minden termelőnek van név, cím és Elérhetőség tulajdonsága.

A termelőtől az élelmiszert egy Beszállító viszi el az Étterembe, ezért neki is csináltam egy külön egyedet. A Beszállító tulajdonságaihoz tartozik a Név, cím és Elérhetőség. A termelő és Beszállító közötti kapcsolatnál megtudjuk, hogy mikor, miből és mekkora mennyiséget vesz fel a termelőtől.

Az Étterem egyed tulajdonságaihoz tartozik a tulajdonos, cím, Név menü és Elérhetőség. A Beszállítóval lévő kapcsolatánál megtudjuk, hogy mikor vette át az Árut, mekkora mennyiséget vásároltak és ezért mennyit fizetett. Az ár tulajdonságot az Áru és mennyiség segítségével tudjuk kiszámítani.

A Személyzet egyednél külön azonosító tulajdonságot csináltam kulcsnak, mivel egy ember személyigazolványának száma is változhat. Ezen kívül még tároljuk a Névét, lakcímét, Elérhetőségét Születési dátumát és hogy milyen Munkakörben dolgozik. Az Étterem egyeddel való kapcsolatánál lévő tulajdonságokból megtudjuk mióta dolgozik az Étteremnél, illetve, hogy az Étterem hány főt foglalkoztat.

A Vendég egyed gyenge egyed típusú, mivel nincsen semmilyen kulcsa. Tulajdonságánál azt tudjuk meg, hogy egy bizonyos napon hány olyan ember fáradt be az Étterembe, akik előtte nem foglaltak náluk asztalt. Kiszolgál kapcsolattal kapcsolódik a Személyzet egyedhez, illetve ellátogat kapcsolattal az Étterem egyedhez, amik a gyenge egyed típus miatt gyenge kapcsolatok.

A Foglалás egyed tulajdonságai, hogy hány egy adott Foglалást melyik dátumra egyeztették és ennél a Foglалásnál hány főre kérték az asztalt.

Az Elérhetőség tulajdonság minden egyednél többértékű, mivel lehet egyszerre telefonszáma, e-mail címe, ahol lehetséges ott weboldala is. A cím tulajdonság pedig mindenhol összetett, mivel egy cím Irányítószámból Utcanévből és Házzszámból áll.

A Termelő-Beszállító között N:M kapcsolat van, milyen egy adott termelőtől több Beszállító is vehet fel Árut, illetve egy adott Beszállító is mehet több termelőhöz pl. húsert és zöldségért máshoz megy.

Fontos elmondanom, hogy mivel egy adott Étteremről van szó a kapcsolatok máshogy alakultak, mint megszokott lenne.

A Beszállító-Étterem kapcsolatot 1:N-re állítottam, mivel ennek az adott Étteremnek több Beszállítója is lehet.

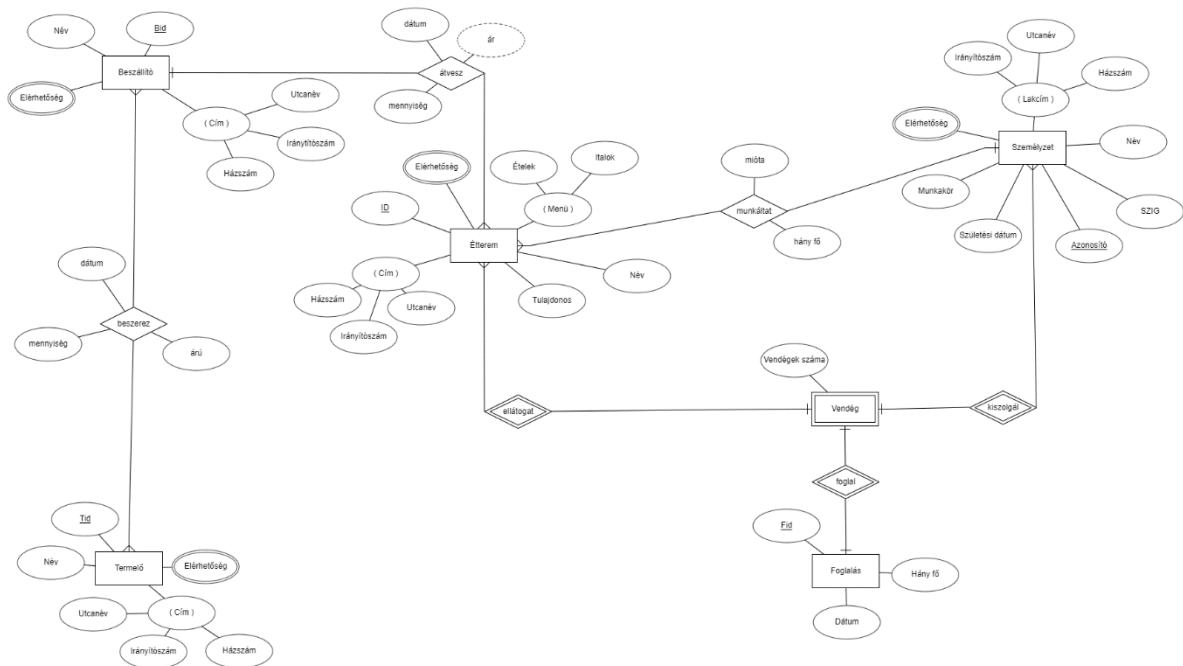
A Személyzet-Étterem kapcsolat szintén 1:N, mivel az Étterem egyszerre több személyt is foglalkoztat.

A Vendég-Személyzet kapcsolat azért 1:N, hiszen egy pincér egy este alatt több Vendéghez is szállít Ételt, illetve egy szakács is egyszerre több Vendégnek készíti el a rendelését.

Végül a Vendég-Étterem kapcsolat azért 1:N, mivel ebben az adott Étteremben egyszerre több Vendég is megfordul, viszont egy Vendég egyszerre csak egy Étteremben tartózkodhat.

1. feladat

1a, Az adatbázis ER modell:

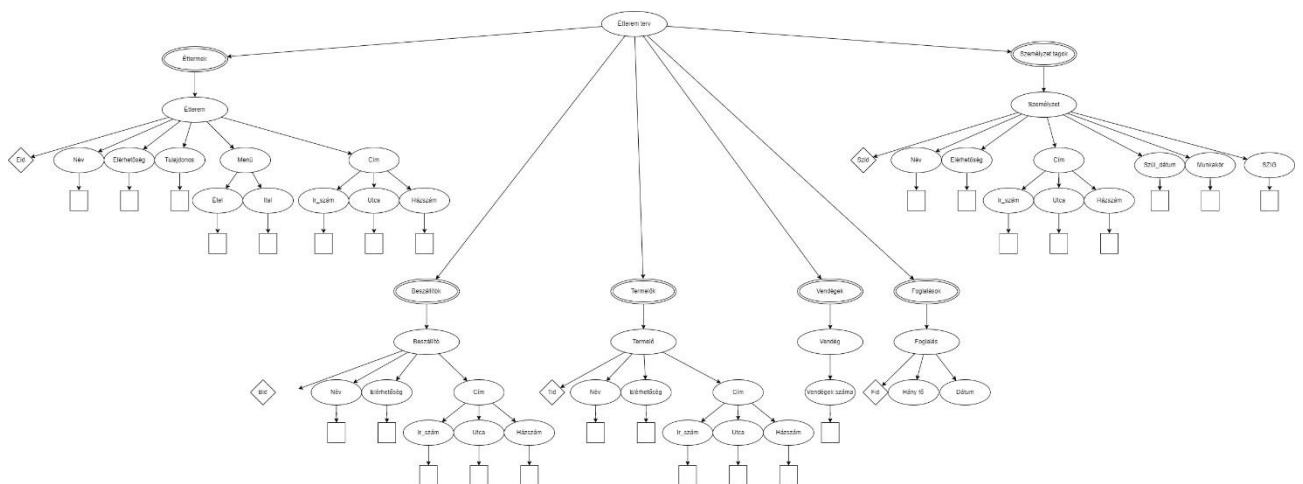


1b, Az adatbázis konvertálása XDM modellre:

Mivel az ER modellben az Étterem egyed kapcsolatban áll a Személyzet és Vendég egyeddel is, így döntést kellett hoznom, hogy a megfelelő XDM modell miatt milyen sorrendben legyenek ezek a kapcsolatok.

Mint látható azt a megoldást választottam, hogy az Étterem a Személyzet egyeddel marad kapcsolatban és innen csatlakozik a Vendég egyed.

Mivel az ER modellben a Vendég egyednek gyenge az egyed típusa, így az XDM modellben a Foglalkás mezőben csak a FID, azaz FoglalkásID szerepel.



1c, Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<etteremterv>
  <etterem eid="1">
    <elerhetoseg>06301593682</elerhetoseg>
    <menu>
      <etel>Húsleves, Spagetti, Túrógombóc, Rántott hús</etel>
      <ital>Cola, Víz, Sör</ital>
    </menu>
    <tulajdonos>Kovács Ákos</tulajdonos>
    <nev>Szőlőskert Étterem</nev>
    <cim>
      <ir_szam>8800</ir_szam>
      <utca>Szőlősgazda út</utca>
      <hazszam>1</hazszam>
    </cim>
  </etterem>
  <beszallito bid="1" tid="1">
    <nev>nagypista</nev>
    <elerhetoseg>06352569852</elerhetoseg>
    <cim>
      <ir_szam>8778</ir_szam>
      <utca>Kossuth Lajos utca</utca>
      <hazszam>47</hazszam>
    </cim>
  </beszallito>
  <beszallito bid="2" tid="2">
    <nev>kispista</nev>
    <elerhetoseg>06301475658</elerhetoseg>
    <cim>
      <ir_szam>8400</ir_szam>
      <utca>Ipari park</utca>
      <hazszam>15</hazszam>
    </cim>
  </beszallito>
  <beszallito bid="3" tid="3">
    <nev>Lencse Trans Kft</nev>
    <elerhetoseg>06703692584</elerhetoseg>
    <cim>
      <ir_szam>2900</ir_szam>
      <utca>Török Ignác utca</utca>
      <hazszam>87</hazszam>
    </cim>
  </beszallito>
  <termelo tid="1">
    <nev>Újudvar termelői Piac</nev>
    <elerhetoseg>06706583654</elerhetoseg>
    <cim>
```

```
        <ir_szam>8778</ir_szam>
        <utca>Fő út</utca>
        <hazszam>73</hazszam>
    </cim>
</termelo>
<termelo tid="2">
    <nev>Hévízi termelői Piac</nev>
    <elerhetoseg>06304862596</elerhetoseg>
    <cim>
        <ir_szam>8380</ir_szam>
        <utca>Király út</utca>
        <hazszam>12</hazszam>
    </cim>
</termelo>
<termelo tid="3">
    <nev>Őriszentpéter Helyi termelői Piac</nev>
    <elerhetoseg>06203488785</elerhetoseg>
    <cim>
        <ir_szam>9941</ir_szam>
        <utca>Városszer</utca>
        <hazszam>94</hazszam>
    </cim>
</termelo>
<szemelyzet fid="1" szid="1">
    <munkakor>szakács</munkakor>
    <elerhetoseg>0641526378</elerhetoseg>
    <SZIG>156934AS</SZIG>
    <szul_datum>1979.10.25.</szul_datum>
    <nev>Kiss Péter</nev>
    <cim>
        <ir_szam>8800</ir_szam>
        <utca>Rózsa utca</utca>
        <hazszam>25</hazszam>
    </cim>
    <vendeg>
        <vendeg_szam>2000</vendeg_szam>
    </vendeg>
</szemelyzet>
<szemelyzet fid="2" szid="2">
    <munkakor>takarító</munkakor>
    <elerhetoseg>0674515621</elerhetoseg>
    <SZIG>364825SA</SZIG>
    <szul_datum>1998.05.13.</szul_datum>
    <nev>Példa Péter</nev>
    <cim>
        <ir_szam>1534</ir_szam>
        <utca>Fő út</utca>
        <hazszam>15</hazszam>
    </cim>
```

```

        <vendeg>
            <vendeg_szam>152</vendeg_szam>
        </vendeg>
    </szemelyzet>
    <szemelyzet fid="3" szid="3">
        <munkakor>pincér</munkakor>
        <elerhetoseg>0674185296</elerhetoseg>
        <SZIG>187263LK</SZIG>
        <szul_datum>1994.06.01.</szul_datum>
        <nev>Mihály Krisztina</nev>
        <cim>
            <ir_szam>8496</ir_szam>
            <utca>Táncsics utca</utca>
            <hazszam>82</hazszam>
        </cim>
        <vendeg>
            <vendeg_szam>1896</vendeg_szam>
        </vendeg>
    </szemelyzet>
    <foglalas fid="1">
        <hany_fo>10</hany_fo>
        <datum>2021.12.12.</datum>
    </foglalas>
    <foglalas fid="2">
        <hany_fo>20</hany_fo>
        <datum>2031.9.20.</datum>
    </foglalas>
    <foglalas fid="3">
        <hany_fo>10</hany_fo>
        <datum>2020.01.01.</datum>
    </foglalas>
</etteremterv>

```

1d, Az XML dokumentum alapján XMLSchema készítése:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="etteremterv">
        <xs:complexType>
            <xs:sequence>
                <xs:choice maxOccurs="unbounded">
                    <xs:element name="etterem" type="etterem_tipus" />
                    <xs:element name="beszallito" type="beszallito_tipus" />
                    <xs:element name="termelo" type="termelo_tipus" />
                    <xs:element name="szemelyzet" type="szemelyzet_tipus" />
                    <xs:element name="foglalas" type="foglalas_tipus" />
                </xs:choice>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

```

</xs:complexType>
<xs:key name="eid">
  <xs:selector xpath="etterem" />
  <xs:field xpath="@eid" />
</xs:key>
<xs:key name="bid">
  <xs:selector xpath="beszallito" />
  <xs:field xpath="@bid" />
</xs:key>
<xs:key name="tid_pk">
  <xs:selector xpath="termelo" />
  <xs:field xpath="@tid" />
</xs:key>
<xs:key name="szid">
  <xs:selector xpath="szemelyzet" />
  <xs:field xpath="@szid" />
</xs:key>
<xs:key name="fid_pk">
  <xs:selector xpath="foglalas" />
  <xs:field xpath="@fid" />
</xs:key>
<xs:keyref name="tid_fk" refer="tid_pk">
  <xs:selector xpath="beszallito" />
  <xs:field xpath="@tid" />
</xs:keyref>
<xs:keyref name="fid_fk" refer="fid_pk">
  <xs:selector xpath="szemelyzet" />
  <xs:field xpath="@fid" />
</xs:keyref>
</xs:element>

<xs:complexType name="etterem_tipus">
  <xs:sequence>
    <xs:element name="elerhetoseg" type="elerhetoseg_tipus" />
    <xs:element name="menu" type="menu_tipus" />
    <xs:element name="tulajdonos" type="tulajdonos_tipus" />
    <xs:element name="nev" type="nev_tipus" />
    <xs:element name="cim" type="cim_tipus" />
  </xs:sequence>
  <xs:attribute type="xs:string" name="eid" use="required" />
</xs:complexType>

<xs:complexType name="beszallito_tipus">
  <xs:sequence>
    <xs:element name="elerhetoseg" type="elerhetoseg_tipus" />
    <xs:element name="nev" type="nev_tipus" />
    <xs:element name="cim" type="cim_tipus" />
  </xs:sequence>
  <xs:attribute type="xs:string" name="bid" use="required" />

```

```

        <xs:attribute type="xs:string" name="tid" use="required" />
    </xs:complexType>

    <xs:complexType name="termelo_tipus">
        <xs:sequence>
            <xs:element name="elerhetoseg" type="elerhetoseg_tipus" />
            <xs:element name="nev" type="nev_tipus" />
            <xs:element name="cim" type="cim_tipus" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="tid" use="required" />
    </xs:complexType>

    <xs:complexType name="szemelyzet_tipus">
        <xs:sequence>
            <xs:element name="munkakor" type="munkakor_tipus" />
            <xs:element name="elerhetoseg" type="elerhetoseg_tipus" />
            <xs:element name="szig" type="szig_tipus" />
            <xs:element name="szul_datum" type="szul_datum_tipus" />
            <xs:element name="nev" type="nev_tipus" />
            <xs:element name="cim" type="cim_tipus" />
            <xs:element name="vendeg" type="vendeg_tipus" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="szid" use="required" />
        <xs:attribute type="xs:string" name="fid" use="required" />
    </xs:complexType>

    <xs:complexType name="foglalas_tipus">
        <xs:sequence>
            <xs:element name="hany_fo" type="hanyfo_tipus" />
            <xs:element name="datum" type="datum_tipus" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="fid" use="required" />
    </xs:complexType>

    <xs:complexType name="menu_tipus">
        <xs:sequence>
            <xs:element name="etel" type="etel_tipus" />
            <xs:element name="ital" type="ital_tipus" />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="vendeg_tipus">
        <xs:sequence>
            <xs:element name="vendeg_szam" type="vendeg_szam_tipus" />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="cim_tipus">
        <xs:sequence>

```



```
        <xs:element name="ir_szam" type="ir_szam_tipus" />
        <xs:element name="utca" type="utca_tipus" />
        <xs:element name="hazszam" type="hazszam_tipus" />
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="nev_tipus">
    <xs:restriction base="xs:string">
        <xs:minLength value="3" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tulajdonos_tipus">
    <xs:restriction base="xs:string">
        <xs:minLength value="3" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="etel_tipus">
    <xs:restriction base="xs:string">
        <xs:minLength value="3" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ital_tipus">
    <xs:restriction base="xs:string">
        <xs:minLength value="3" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="elerhetoseg_tipus">
    <xs:restriction base="xs:integer">
        <xs:minInclusive value="1" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="munkakor_tipus">
    <xs:restriction base="xs:string">
        <xs:minLength value="3" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="szig_tipus">
    <xs:restriction base="xs:string">
        <xs:minLength value="3" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="utca_tipus">
```

```
<xs:restriction base="xs:string">
  <xs:minLength value="3" />
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="szul_datum_tipus">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="hazszam_tipus">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ir_szam_tipus">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="hanyfo_tipus">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="vendeg_szam_tipus">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="datum_tipus">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

2. feladat

2a, Adatolvasás:

```
package hu.domparse.eq4b3d;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;

public class DOMReadEQ4B3D {

    public static void main(String[] args) {
        try {
            File xmlFile = new File("XMLEQ4B3D.xml"); // fájlt beolvassa
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            Document doc = dBuilder.parse(xmlFile);
            doc.getDocumentElement().normalize();
            Read(doc);
            //ExceptionHandling
        } catch (ParserConfigurationException pce) {
            pce.printStackTrace();
        } catch (IOException ioe) {
            ioe.printStackTrace();
        } catch (SAXException sae) {
            sae.printStackTrace();
        }
    }

    public static void Read(Document doc) {

        NodeList nList = doc.getElementsByTagName("etterem_terv"); //létrehozzuk a NoteListet,
        amiben eltároljuk az elemeket

        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) { //az ID-k kiszedése
                String eid = element.getAttribute("eid");
            }
        }
    }
}
```

```

        String bid = element.getAttribute("bid");
        String szid = element.getAttribute("szid");
        ReadEtteremById(doc, eid);
        ReadBeszallitoById(doc, bid);
        ReadSzemelyzetById(doc, szid);
    }
}

}

public static void ReadEtteremById(Document doc, String eid) {
    NodeList nList = doc.getElementsByTagName("etterem"); //az étterem adatait tartalmazó
nodeList létrehozása
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("eid").equals(eid)) { //kiírandó adatok
kiszedése
                String elerhetoseg =
element.getElementsByTagName("elerhetoseg").item(0).getTextContent();
                String menu =
element.getElementsByTagName("menu").item(0).getTextContent();
                String tulajdonos =
element.getElementsByTagName("tulajdonos").item(0).getTextContent();
                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();
                String cim =
element.getElementsByTagName("cim").item(0).getTextContent();
                System.out.println("Étterem adatok: \n\tNév:\t" + nev +
"\n\tTulajdonos:\t" + tulajdonos
+ "\n\tElérhetőség:\t" + elerhetoseg +
"\n\tMenü:\t" + menu + "\n\tCím:\t" + cim); //kiíratás
            }
        }
    }
}

public static void ReadBeszallitoById(Document doc, String bid) {
    NodeList nList = doc.getElementsByTagName("beszallito"); //a szállító adatait
tartalmazó nodeList létrehozása
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("bid").equals(bid)) { //kiírandó adatok
kiszedése
                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();

```

```

        String elerhetoseg =
element.getElementsByTagName("elerhetoseg").item(0).getTextContent();

        String cim =
element.getElementsByTagName("cim").item(0).getTextContent();

        System.out.println("Beszállító adatok: \n\tNév:\t" + nev +
"\n\tElérhetőség:\t" + elerhetoseg
                                + "\n\tCím:\t" + cim); //kiíratás

        String tid = element.getAttribute("tid");
        ReadTermeloById(doc, tid);
    }
}

}

}

public static void ReadTermeloById(Document doc, String tid) {
    NodeList nList = doc.getElementsByTagName("termelo"); //a termelő adatait tartalmazó
nodeList létrehozása

    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("tid").equals(tid)) { //kiírandó adatok
kiszedése

                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();

                String elerhetoseg =
element.getElementsByTagName("elerhetoseg").item(0).getTextContent();

                String cim =
element.getElementsByTagName("cim").item(0).getTextContent();

                System.out.println("Termelő adatok: \n\tNév:\t" + nev +
"\n\tElérhetőség:\t" + elerhetoseg
                                + "\n\tCím:\t" + cim); //kiíratás
            }
        }
    }
}

public static void ReadSzemelyzetById(Document doc, String szid) {
    NodeList nList = doc.getElementsByTagName("szemelyzet"); //a személyzet adatait
tartalmazó nodeList létrehozása

    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("szid").equals(szid)) { //kiírandó adatok
kiszedése

                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();

```

```

        String elerhetoseg =
element.getElementsByTagName("elerhetoseg").item(0).getTextContent();

        String cim =
element.getElementsByTagName("cim").item(0).getTextContent();

        String munkakor =
element.getElementsByTagName("munkakor").item(0).getTextContent();

        String SZIG =
element.getElementsByTagName("SZIG").item(0).getTextContent();

        String szul_datum =
element.getElementsByTagName("szul_datum").item(0).getTextContent();

        String vendeg =
element.getElementsByTagName("vendeg").item(0).getTextContent();

        System.out.println("Személyzet adatok: \n\tNév:\t" + nev +
"\n\tElérhetőség:\t" + elerhetoseg
                                + "\n\tCím:\t" + cim + "\n\tMunkakör:\t" +
munkakor + "\n\tSzemélyigazolvány szám:\t" + SZIG
                                + "\n\tSzületési dátum:\t" + szul_datum +
"\n\tVendégek száma:\t" + vendeg); //kiíratás

        String fid = element.getAttribute("fid");
        ReadFoglalasById(doc, fid);
    }
}

}

}

public static void ReadFoglalasById(Document doc, String fid) {
    NodeList nList = doc.getElementsByTagName("foglalas"); //a foglalás adatait tartalmazó
nodeList létrehozása

    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("fid").equals(fid)) { //kiírandó adatok
kiszedése

                String hany_fo =
element.getElementsByTagName("hany_fo").item(0).getTextContent();

                String datum =
element.getElementsByTagName("datum").item(0).getTextContent();

                System.out.println("Foglalás adatok: \n\tHány főre:\t" +
hany_fo + "\n\tDátum:\t" + datum); //kiíratás
            }
        }
    }
}

}

}
}

```

2b, Adatmódosítás:

```
package hu.domparse.eq4b3d;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;

public class DOMModifyEQ4B3D {

    public static void main(String[] args) {
        try {
            File inputFile = new File("XMLEQ4B3D.xml");
            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(inputFile);
            Node személyzet = doc.getElementsByTagName("szemelyzet").item(0);

            // 10-re módosítjuk a szid-t 1-ről
            NamedNodeMap attr = személyzet.getAttributes();
            Node nodeAttr = attr.getNamedItem("szid");
            nodeAttr.setTextContent("10");

            NodeList list = személyzet.getChildNodes();

            // a hibaüzenetet megváltoztatjuk
            for (int temp = 0; temp < list.getLength(); temp++) {
                Node node = list.item(temp);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element hElement = (Element) node;
                    if ("SZIG".equals(hElement.getNodeName())) {
                        if ("156934AS".equals(hElement.getTextContent()))
                            hElement.setTextContent("156934SE");
                    }
                }
            }
            NodeList list2 = doc.getElementsByTagName("beszallito");
            for (int j = 0; j < list2.getLength(); j++) {
                Node beszallito = list2.item(j);
                if (beszallito.getNodeType() == Node.ELEMENT_NODE) {
                    String id =
beszallito.getAttributes().getNamedItem("bid").getTextContent();
                    if ("1".equals(id.trim())) {
                        NodeList gyerekNodes = beszallito.getChildNodes();
                        for (int k = 0; k < gyerekNodes.getLength(); k++) {
                            Node item = gyerekNodes.item(k);
                            if (item.getNodeType() == Node.ELEMENT_NODE) {
                                // kitöröljük az ügyfélből a kód mezőt
                                if
("elerhetoseg".equalsIgnoreCase(item.getNodeName())) {
                                    beszallito.removeChild(item);
                                }
                                // módosítjuk az 1-es ügyféllal
                                if
("nev".equalsIgnoreCase(item.getNodeName())) {
                                    item.setTextContent("XYZ");
                                }
                            }
                        }
                    }
                }
            }
            // kiírás
            TransformerFactory transformerFactory = TransformerFactory.newInstance();
```

```

        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        System.out.println("-----Módosított File-----");
        StreamResult consoleResult = new StreamResult(System.out);
        transformer.transform(source, consoleResult);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

2c, Adat lekérdezés:

```

package hu.dompars.eq4b3d;

import java.io.File;
import java.io.IOException;
import java.util.Scanner;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.TransformerException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMQueryEQ4B3D {

    public static void main(String[] args)
        throws ParserConfigurationException, IOException, SAXException,
        TransformerException {
        File xmlFile = new File("XMLEQ4B3D.xml"); //fájl beolvasás
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();
        LoadtermeloQuery(doc);
    }

    public static void LoadtermeloQuery(Document doc) throws TransformerException {
        NodeList nodeList = doc.getElementsByTagName("termelo"); //termelő noteList létrehozása
        String termelo;
        Element element = null;
        Node nNode = null;
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            String nev = element.getElementsByTagName("nev").item(0).getTextContent();
            //lekérdezés név szerint
            System.out.println(nev);
        }
        System.out.println("\nÍrja be annak a termelőnek a nevét, amelyiknek adatait szeretné
        látni:");

        Scanner sc = new Scanner(System.in); //consolról beolvasás
        termelo = sc.nextLine();
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                if (termelo.equals("Újudvar termelői Piac")) {
                    LoadBeszallitoQuery(doc, "1");
                    break;
                }

                if (termelo.equals("Hévízi termelői Piac")) {
                    LoadBeszallitoQuery(doc, "2");
                    break;
                }
            }
        }
    }
}

```



```

        if (termelo.equals("Óriszentpéter Helyi termelői Piac")) {
            LoadBeszallitoQuery(doc, "3");
            break;
        }
    }
    sc.close();
}
}
public static void LoadBeszallitoQuery(Document doc, String bid) throws TransformerException {
    NodeList nodeList = doc.getElementsByTagName("beszallito"); // beszállító notelist
    létrehozása

    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        Element element = (Element) nNode;
        String tid = element.getAttribute("tid"); // a tid alapján kiszedjük az elemet
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            if (bid.equals(tid)) {
                String Bid = element.getAttribute("bid"); // bid alapján
                kiszedjük az elemet
                beszállító függvényt
                DOMReadEQ4B3D.ReadBeszallitoById(doc, Bid); // meghívjuk a
            }
        }
    }
}
}

```