

# Jegyzőkönyv

Adatkezelés XML környezetben

Számítógépes bolthálózat

Féléves feladat

Készítette: **Karancsi Gábor**

Nepunkód: **W4OGB7**

#### A feladat leírása:

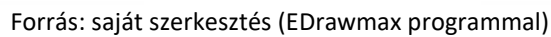
Adatbázis létrehozása számítógépes bolthálózathoz. Az adatbázis ER-modelljét az 1/a. ábra mutatja be. Az ábrán lévő egyedek és tulajdonságok jelentése a következő:

1. A **Bolt egyed** az egyes boltok **Székhelyét, Telefonszámát, Webcímét** tartalmazza. A típustulajdonságok mind egyértékűek.
2. A **Munkavállaló egyed** a **Neve, Cím, Születési idő, Jogosultsága** (Beosztása) egyértékű tulajdonság jellemzi.
3. A **Termék egyed** tulajdonságait **Elnevezés, Cikkszám, Ár, Darabszám** egyértékű tulajdonság jellemzi.
4. Az **Ügyfél egyed** a cég vásárlóit írja le, a **Név, Cím, E-mail cím** egyértékű tulajdonság jellemzi.
5. A **Rendelés egyed** az **Azonosító, Cikkszám, Darabszám** egyértékű tulajdonságok.

Az egyedek közötti kapcsolatok a következők (lásd még az 1. ábrán):

- ☞ **Eladó reláció:** A Bolt egyed és a Munkavállaló egyed közötti 1:N (egy-több) kapcsolat. 1 munkavállalóhoz 1 bolt, de 1 bolthoz több munkavállaló tartozhat.
- ☞ **Vásárló reláció:** A Ügyfél egyed és a Bolt egyed közötti N:M (több-több) kapcsolat. Egy bolthoz több ügyfél, és egy ügyfélhez több bolt kapcsolódhat.
- ☞ **Raktárkészlet reláció:** A Termék egyed és a Bolt egyed közötti N:M (több-több) kapcsolat. Egy bolt raktárában több termék is lehet, és egy termék, több bolt raktárkészletén is lehet.
- ☞ **Rendel reláció:** A Ügyfél, Bolt és Rendelés egyedek kapcsolja össze. Az egyedek közti 1:1:N kapcsolat. Egy rendelést egy adott boltban egy ügyfél ad le, de akár ugyanazon ügyfél ugyanazon boltban több rendelést is leadhat.
- ☞ **Rendel termék reláció:** A Termék egyed és a Rendelés egyed közötti N:M (több-több) kapcsolat. 1 rendelés több terméket is tartalmazhat, egy termék több rendelésen is szerepelhet.

### 1/a ábra: A feladat ER-modellje



Forrás: saját szerkesztés (EDrawmax programmal)



## Az XDM modell alapján készült XML kód:

```
<?xml version="1.0" encoding="UTF-8"?>
<bolt idno="1">
  <boltadatok>
    <id>2345</id>
    <szekhely>3713 Miskolc, Kazinczy utca 22.</szekhely>
    <telefonszam>+36/706587632</telefonszam>
    <email>vezeto@valami.hu</email>
  </boltadatok>
  <munkavallalo>
    <id>76543PA</id>
    <bolt_fk>2</bolt_fk>
    <nev>
      <v.nev>Kiss</v.nev>
      <k.nev>Gábor</k.nev>
    </nev>
    <cim>
      <ir>3512</ir>
      <varos>Kazincbarcika</varos>
      <utcahaszam>Futó utca 76</utcahaszam>
    </cim>
    <szulido>1992.04.85.</szulido>
    <jogosultsag>Elsődleges</jogosultsag>
  </munkavallalo>
  <munkavallalo>
    <id>453212PA</id>
    <bolt_fk>1</bolt_fk>
    <nev>
      <v.nev>Nagy</v.nev>
      <k.nev>József</k.nev>
    </nev>
    <cim>
      <ir>3523</ir>
      <varos>Nagykanizsa</varos>
      <utcahaszam>Halmos utca 76</utcahaszam>
    </cim>
    <szulido>1982.04.85.</szulido>
    <jogosultsag>Másodlagos</jogosultsag>
  </munkavallalo>
  <termek>
    <t_cikkszam>1267öpiu</t_cikkszam>
    <elnevezes>Processzor</elnevezes>
    <cikkszam>123456987-HU</cikkszam>
    <ar>39999</ar>
    <darab>1</darab>
  </termek>
  <termek>
    <elnevezes>Gépház</elnevezes>
    <cikkszam>789654123-HU</cikkszam>
    <ar>19999</ar>
    <darab>1</darab>
  </termek>
  <termek>
    <elnevezes>Asztal</elnevezes>
    <cikkszam>65432897-HU</cikkszam>
    <ar>59999</ar>
    <darab>10</darab>
  </termek>
</ugyfel>
```

```
<id>764TREW</id>
<nev>
  <v.nev>Nagy</v.nev>
  <k.nev>Erika</k.nev>
</nev>
<cim>
  <ir>1234</ir>
  <varos>Mályi</varos>
  <utcahaszszam>Fő utca 10.</utcahaszszam>
</cim>
<email>nagyeri@gmail.com</email>
</ugyfel>
<rendeles>
  <r_azonosito>UZTI7645</r_azonosito>
  <r_bolt_FK>2</r_bolt_FK>
  <r_ugyfel_FK>64TREW</r_ugyfel_FK>
  <cikkszam>OTZIT54</cikkszam>
  <darabszam>2</darabszam>
</rendeles>
</bolt>
```

## Az XML dokumentum alapján készült XML Schema modell:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bolt" id="boltname">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="boltadatok" id="boltadatok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id">
                <xs:key name="bolt_id">
                  <xs:selector xpath="./boltadatok" />
                  <xs:field xpath="id" />
                </xs:key>
              </xs:element>
              <xs:element name="szekhely" id="bszekhely" type="xs:string" />
              <xs:element name="telefonszam" id="btel" type="xs:int" />
              <xs:element name="email" id="bemail" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="munkavallalo" id="mid">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id">
                <xs:key name="munka_id">
                  <xs:selector xpath="./munkavallalo" />
                  <xs:field xpath="id" />
                </xs:key>
              <xs:keyref name="bolt_FK" refer="bolt_id">
                <xs:selector xpath="./bolt" />
                <xs:field xpath="id" />
              </xs:keyref>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="nev" id="mnev">
```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element name="v.nev" type="xs:string" />
                <xs:element name="k.nev" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="cim" id="mcim">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ir" type="xs:unsignedShort" />
                <xs:element name="varos" type="xs:string" />
                <xs:element name="utcahaszszam" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="szulido" type="xs:string" />
    <xs:element name="jogosultsag" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="termek" id="btermek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="t_cikkszam">
                <xs:key name="tcsz_id">
                    <xs:selector xpath="./termek" />
                    <xs:field xpath="tcsz_id" />
                </xs:key>
            </xs:element>
            <xs:element name="elnevezes" type="xs:string" />
            <xs:element name="ar" type="xs:string" />
            <xs:element name="darab" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ugyfel" id="vasarlo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="id">
                <xs:key name="u_id">
                    <xs:selector xpath="ugyfel" />
                    <xs:field xpath="@u_id" />
                </xs:key>
            </xs:element>
            <xs:element name="nev">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="v.nev" type="xs:string" />
                        <xs:element name="k.nev" type="xs:string" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="cim" id="vcim">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="ir" type="xs:string" />
                        <xs:element name="varos" type="xs:string" />
                        <xs:element name="utcahaszszam" type="xs:string" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```
        </xs:complexType>
        </xs:element>
        <xs:element name="email" type="xs:string" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rendeles" id="rid">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="r_azonosito" id="raz">
                <xs:key name="r_id">
                    <xs:selector xpath="./rendeles" />
                    <xs:field xpath="r_id" />
                </xs:key>
                <xs:keyref name="r_bolt_FK" refer="bolt_id">
                    <xs:selector xpath="../munkavallalo" />
                    <xs:field xpath="@boltFK" />
                </xs:keyref>
                <xs:keyref name="r_ugyfel_FK" refer="ugyfel_id">
                    <xs:selector xpath="../munkavallalo" />
                    <xs:field xpath="ugyfelFK" />
                </xs:keyref>
            </xs:element>
            <xs:element name="r_cikkszam" id="ecksz"/>
            <xs:element name="darabszam" id="rdb" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## DOM program kódja adatolvasás céllal (DomReadW4OGB7.java):

Létrehoztam egy Java projektet, benne egy package-t. Ebben a package-ben valósítom meg a feladatokat. A kód írásának első lépése, hogy importáljuk az alap dolgokat, mint a java.io.File.

Az adatbeolvasás egy Try-Catch blokkon belül történik. Az adatok beolvasása után a kiírás egy összefűzött String-ben történik az adatok megjelenésének formázása érdekében.

```
package domparsew4ogb7;

import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
```



```
import org.w3c.dom.NodeList;

public class DomReadW4OGB7 {



    public static void main(String[] args) {
        try{
            File xmlDoc = new File("XMLW4OGB7.xml");
            DocumentBuilderFactory dbFact =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuild = dbFact.newDocumentBuilder();
            Document doc = dBuild.parse(xmlDoc);
```

**Ellenőrzés céljából kiírom a gyökér elemet, valamint a bolt Node elemet.**

```
//root element
System.out.println("Root element: " + doc.getDocumentElement().getNodeName());

//read bolt element
NodeList nList = doc.getElementsByTagName("bolt");
```

Az olvasás eredménye:

 Console 

<terminated> DomReadW4OGB7 [Java Application] C:\Program  
Root element: bolt  
Node name:bolt 1bolt Id #: 1

**A bolt adatait olvasom valamint kiíratom az alábbiak szerint:**

- Bolt székhelye
- Bolt telefonszáma
- Bolt e-mail címe

```
for(int i=0; i<nList.getLength();i++)
{
    Node nNode = nList.item(i);
    System.out.print("Node name:" + nNode.getNodeName() + " " +
(i+1));

    if(nNode.getNodeType() == Node.ELEMENT_NODE)
    {
        Element eElement = (Element) nNode;
        //Bolt adatai
        System.out.println("bolt Id #: " +
eElement.getAttribute("idno"));
        System.out.println("Bolt székhelye: " +
eElement.getElementsByTagName("szekhely").item(0).getT
extContent());
```

A székhely kiírásának eredménye:

```
Console [X] [redacted]  
<terminated> DomReadW4OGB7 [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\jav  
Bolt székhelye: 3713 Miskolc, Kazinczy utca 22.
```

```
System.out.println("Bolt telefonszáma: " +  
    eElement.getElementsByTagName("telefonszam").item(0).g  
etTextContent());
```

A telefonszám kiírásának eredménye:

```
Console [X] [redacted]  
<terminated> DomReadW4OGB7 [Java Application] C:\Program Files\Ja  
Bolt telefonszáma: +36/706587632
```

```
System.out.println("Bolt email címe: " +  
    eElement.getElementsByTagName("email").item(0).getText  
Content());
```

```
Console [X] [redacted]  
<terminated> DomReadW4OGB7 [Java Application] C:\Program Files  
Bolt email címe: vezeto@valami.hu
```


**A munkavállaló adatait olvasom be, valamint kiíratom az alábbiak szerint:**

- Munkavállaló címe
- Munkavállaló születési ideje
- Munkavállaló jogosultsága

Az adatokat egy összefűzött Strignben írtam ki, az egyszerűbb kiírás érdekében. Az elválasztáshoz a „\n” sortötétst használtam:

```
System.out.println("Munkavállaló adatai: " + "\n"+"Neve:"+  
    eElement.getElementsByTagName("v.nev").item(0).getText  
Content()+" "+eElement.getElementsByTagName("k.nev").item(0).getTextContent()  
    +"\n"+"Munkavállaló címe:  
"+eElement.getElementsByTagName("varos").item(0).getTextContent()+  
    eElement.getElementsByTagName("utcahazszam").item(0).g  
etTextContent()+"\n"+"Születési  
ideje:"+eElement.getElementsByTagName("szulido").item(0).getTextContent()+"\n"  
+  
    "Jogosultsága:  
"+eElement.getElementsByTagName("jogosultsag").item(0).getTextContent()  
    );
```

A lekérdezés eredménye:

```
Console 
<terminated> DomReadW4OGB7 [Java Application] C:\Program Files\Java\jdk-13.0.2\
Munkavállaló adatai:
Neve:Kiss Gábor
Munkavállaló címe: KazincbarcikaFutó utca 76
Születési ideje:1992.04.85.
Jogosultsága: Elsődleges
```


### A termék adatait olvasom be, valamint kiíratom az alábbiak szerint:

- Termék elnevezése
- Termék cikkszáma
- Termék ára
- Termék darabszáma

Az adatokat egy összefűzött Stringben írtam ki, az egyszerűbb kiírás érdekében. Az elválasztáshoz a „\n” sortörést használtam:

```
System.out.println("Termék adatai: " + "\n" + "Elnevezés:" +
    eElement.getElementsByTagName("elnevezes").item(0)
    .getTextContent() + "\n" + "Cikkszám:" + eElement.getElementsByTagName("cikkszam").i
    tem(0).getTextContent()
    + "\n" + "Ár:
" + eElement.getElementsByTagName("ar").item(0).getTextContent() + " Ft
\n" + "Raktárkészlet:
" + eElement.getElementsByTagName("darab").item(0).getTextContent() + " db");
```

A lekérdezés eredménye:

```
Console 
<terminated> DomReadW4OGB7 [Java Ap
Termék adatai:
Elnevezés:Processzor
Cikkszám:123456987-HU
Ár: 39999 Ft
Raktárkészlet: 1 db
```

### Az ügyfél adatait egyben olvastam be, valamint egyben írtam ki az összes adatot:

Az adatok formázás nélkül kerültek megjelenítésre:

```
System.out.println("Ügyfél adatai: " + "\n" + "Neve:" +
    eElement.getElementsByTagName("ugyfel").item(0).ge
    tTextContent());
```

A lekérdezés eredménye:

```
Console
<terminated> DomReadW4OGB7 [Java Application] C:\F
Ügyfél adatai:
Neve:

        Nagy
        Erika

        1234
        Mályi
        Fő utca 10.

        nagyeri@gmail.com
```

## DOM program adatok módosítására:(DomModifyW4OGB7.java):

Az XML adatok módosítása szintén egy Try-Catch blokkban történik. Az alábbi feladatokat végeztem el a dokumentumban:

- Adatmódosítás
- Adat törlés
- Adat hozzáadás
- Adatok kiírása egy új XML file-ba

```
package domparsew4ogb7;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerFactoryConfigurationError;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

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 DomModifyW4OGB7 {
    public static void main(String[] args) {
        // A szerkeszteni kívánt XML file megnyitása
```

```
String filePath = "XMLW4OGB7.xml";
File xmlFile = new File(filePath);
DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder;
try {
    dBuilder = dbFactory.newDocumentBuilder();
    Document doc = dBuilder.parse(xmlFile);
    doc.getDocumentElement().normalize();
    // Függvények használata az adatmódosításra
    // Módosítja a bolt székhelyének címét
    updateElementValue(doc);

    // Törli a munkavállaló címét
    deleteElement(doc);

    // Hozzáad egy "tipus" elemet a termékhez
    addElement(doc);

    // Kiírja egy fileba a módosításokat
    writeXMLFile(doc);

} catch (SAXException | ParserConfigurationException | IOException |
TransformerException e1) {
    e1.printStackTrace();
}
}
```

## Adatmódosítás:

**A bolt székhelyének módosítása az XML dokumentumban:**

Az eredeti székhely cím: 3713 Miskolc, Kazinczy utca 22.

```
private static void updateElementValue(Document doc) {
    NodeList bolt = doc.getElementsByTagName("boltadatok");
    Element boltadatok = null;
    for (int i = 0; i < bolt.getLength(); i++) {
        boltadatok = (Element) bolt.item(i);
        Node szekhely =
boltadatok.getElementsByTagName("szekhely").item(0).getFirstChild();
        szekhely.setNodeValue("5879 Magyaralföld, Dózsa u 63.");
    }
}
```

A módosítás után pedig az alábbi olvasható a módosított XML Fileban:

```
Updated_XMLW4OGB7.xml – Jegyzettömb
Fájl Szerkesztés Formátum Nézet Súgó
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<bolt idno="1">

    <boltadatok>

        <szekhely>5879 Magyaralföld, Dózsa u 63.</szekhely>

        <telefonszam>+36/706587632</telefonszam>

        <email>vezeto@valami.hu</email>

    </boltadatok>
```

## Adat törlés:

A munkaválló címét szerettem volna kitörölni az XML dokumentumból, mégpedig az egész elem eltávolításával:

Az eredeti XML Fileban ez látható:

```
<munkavallalo>
  <nev>
    <v.nev>Nagy</v.nev>
    <k.nev>József</k.nev>
  </nev>
  <cim>
    <ir>3523</ir>
    <varos>Nagykanizsa</varos>
    <utcahazszam>Halmos utca 76</utcahazszam>
  </cim>
```

A program kódja:

```
private static void deleteElement(Document doc) {
    NodeList bolt = doc.getElementsByTagName("munkavallalo");
    Element munkavallalo = null;
    // loop for each user
    for (int i = 0; i < bolt.getLength(); i++) {
        munkavallalo = (Element) bolt.item(i);
        Node cim = munkavallalo.getElementsByTagName("cim").item(0);
        munkavallalo.removeChild(cim);
    }
}
```

A törlés után pedig látható, hogy nincs más <cím> elem a módosított XML Fileban:



## Adat hozzáadása:

A termék elemhez adtam hozzá egy „tipus” nevű elemet, amelyet feltöltöttem adattal:  
Az eredeti XML dokumentumban az alábbi látható:


```
<termek>
  <elnevezes>Processzor</elnevezes>
  <cikkszam>123456987-HU</cikkszam>
  <ar>39999</ar>
  <darab>1</darab>
</termek>
```

Az program kódja:

```
private static void addElement(Document doc) {
    NodeList bolt = doc.getElementsByTagName("termek");
    Element termék = null;

    // loop for each user
    for (int i = 0; i < bolt.getLength(); i++) {
        termék = (Element) bolt.item(i);
        Element id = doc.createElement("tipus");
        id.appendChild(doc.createTextNode("PC Alkatrészek"));
        termék.appendChild(id);
    }
}
```

A hozzáadás után pedig látható, hogy a <tipus> elem hozzáadásra került, valamint „PC alkatrészek” adattal fel is lett töltve:


 Updated\_XMLW4OGB7.xml – Jegyzettömb

Fájl Szerkesztés Formátum Nézet Súgó

```
<termek>
  <elnevezes>Processzor</elnevezes>
  <cikkszam>123456987-HU</cikkszam>
  <ar>39999</ar>
  <darab>1</darab>
  <tipus>PC Alkatrész</tipus>
</termek>
```

## Adatok kiírása egy új XML file-ba:

A módosított adatokat egy új XML file-ba írtam ki, hogy az eredeti adatok ne változhassanak. Ennek a dokumentumnak „Updated\_XMLW4OGB7.xml” nevet adtam. Az alábbi képek szemléltetik a változást.

 XMLW4OGB7.xml – Jegyzettömb

Fájl Szerkesztés Formátum Nézet Súgó

```
<?xml version="1.0" encoding="UTF-8"?>
  <bolt idno="1">
    <boltadatok>
      <szekhely>3713 Miskolc, Kazinczy utca 22.</szekhely>
      <telefonszam>+36/706587632</telefonszam>
      <email>vezeto@valami.hu</email>
    </boltadatok>
    <munkavallalo>
      <nev>
        <v.nev>Kiss</v.nev>
        <k.nev>Gábor</k.nev>
      </nev>
      <cim>
        <ir>3512</ir>
        <varos>Kazincbarcika</varos>
        <utcahaszszam>Futó utca 76</utcahaszszam>
      </cim>
      <szulido>1992.04.85.</szulido>
      <jogosultsag>Elsődleges</jogosultsag>
    </munkavallalo>
  </bolt idno="1">
```

(Az eredeti XML dokumentum amiből az adatok származnak)



Updated\_XMLW4OGB7.xml – Jegyzetömb

Fájl Szerkesztés Formátum Nézet Súgó

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<bolt idno="1">
  <boltadatok>
    <szekhely>5879 Magyaralföld, Dózsa u 63.</szekhely>
    <telefonszam>+36/706587632</telefonszam> |
    <email>vezeto@valami.hu</email>
  </boltadatok>
  <munkavallalo>
    <nev>
      <v.nev>Kiss</v.nev>
      <k.nev>Gábor</k.nev>
    </nev>
    <szulido>1992.04.85.</szulido>
    <jogosultsag>Elsődleges</jogosultsag>
  </munkavallalo>
```

(A módosított XML dokumentum amiben jól látszik, hogy a bolt székhelye módosult, valamint a munkavállaló címe törlésre került)