

# Jegyzőkönyv

Adatkezelés XML környezetben

Horgászciikk kölcsönző

Féléves feladat

Készítette: Ládi András Balázs

Neptun: GFMV6T

## **A feladat leírása:**

A feladathoz egy horgászcikkekkel foglalkozó fiktív cég adatbázisát hozom létre. Az ügyfél tulajdonságaiból a név és születési idő egyértelmű, a telefonszám opcionális, a személyi azonosító pedig egyedi tulajdonság. A lakcím összetett, az irányítószámból, utcából, házszámból és a városból áll. Tartozik még hozzá egy Id, amely az ügyfél táblával kapcsolja össze. Az ügyfél tábla a kölcsönzés által a kölcsönzés adataihoz kapcsolódik. A kölcsönzés az adatbázisban egy segédtábla, így csak két oszlopa van, a rendelesid és az ugyfelid. A kölcsönzés adatai tartalmazza a kölcsönzéshez tartozó adatok ügyfélhez tartozó részét, így a kikölcsönzés dátumát (datum), a lejáratot és két egyedi kulcsot, a rendelesid-t és a raktarid-t. Ehhez kapcsolódik még két tábla, a kölcsönzési lista és a raktár. A kölcsönzési lista tartalmazza a kikölcsönzött tárgyak adatait, amely a darab, az ár, a gyártó és a típus által tárolódik. A rendelesid, mint idegen kulccsal fog kapcsolódni az előbbi táblához. A raktár a céghez tartozó telephelyeket tartalmazza, amivel beazonosíthatjuk, hogy melyik helyről vagy helyekről kölcsönözték a termékeket. Tagjai: raktarid, település és irányítószám.

### **Kapcsolatok:**

Lakcím – ügyfél: egy ügyfélnek csak egy hivatalos állandó lakcíme lehet, de ugyanarra a lakcímre többen is bejelentkezhetnek, így ez egy kötelező 1:N kapcsolat.

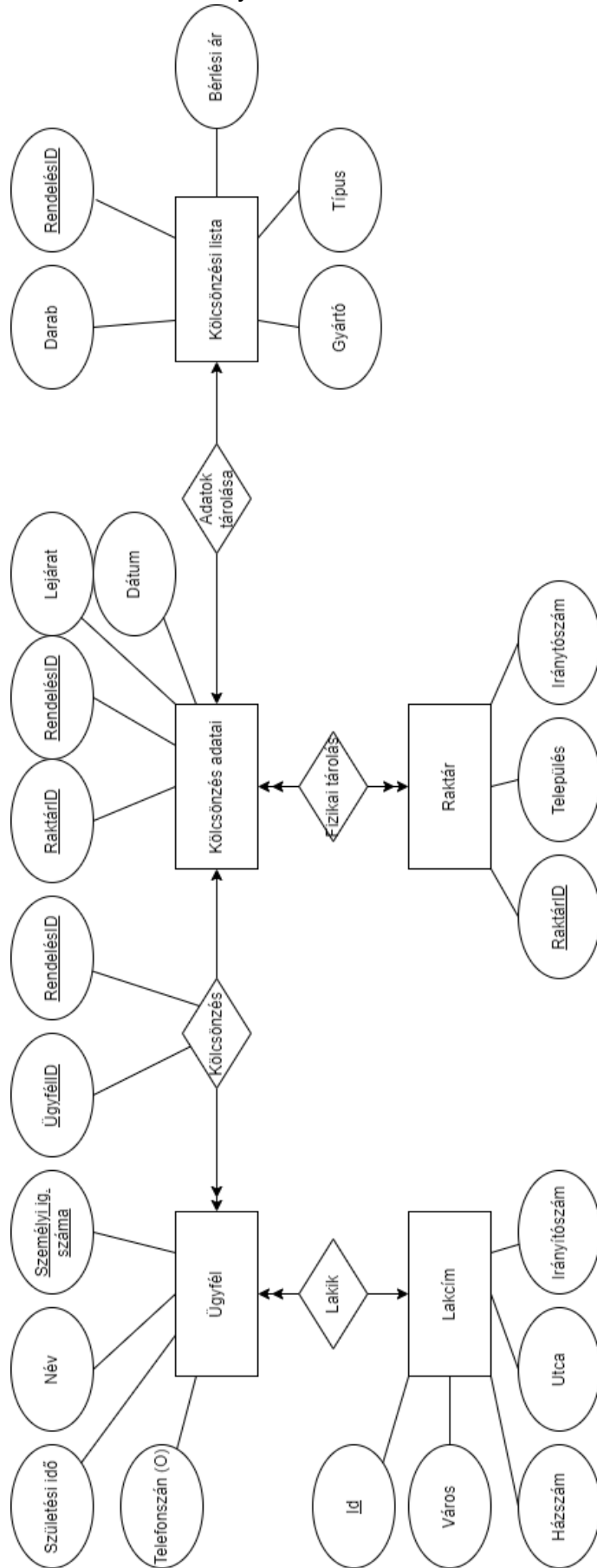
Ügyfél – kölcsönzés adatai: több ügyfél is adhat le rendelést, de amíg vissza nem vitte az eszközöket, addig nem adhat le újabb rendelést, tehát 1:N kapcsolat. Az ügyfél oldalán nem kötelező, hogy legyen kölcsönzés, így az opcionális, de ha leadja, akkor ahhoz tartozni fog egy lista az adatokkal, a másik oldal kötelező.

Kölcsönzés adatai – kölcsönzési lista: egy kölcsönzéshez csak egy bizonyos lista tartozik kötelezően, így 1:1 kapcsolat.

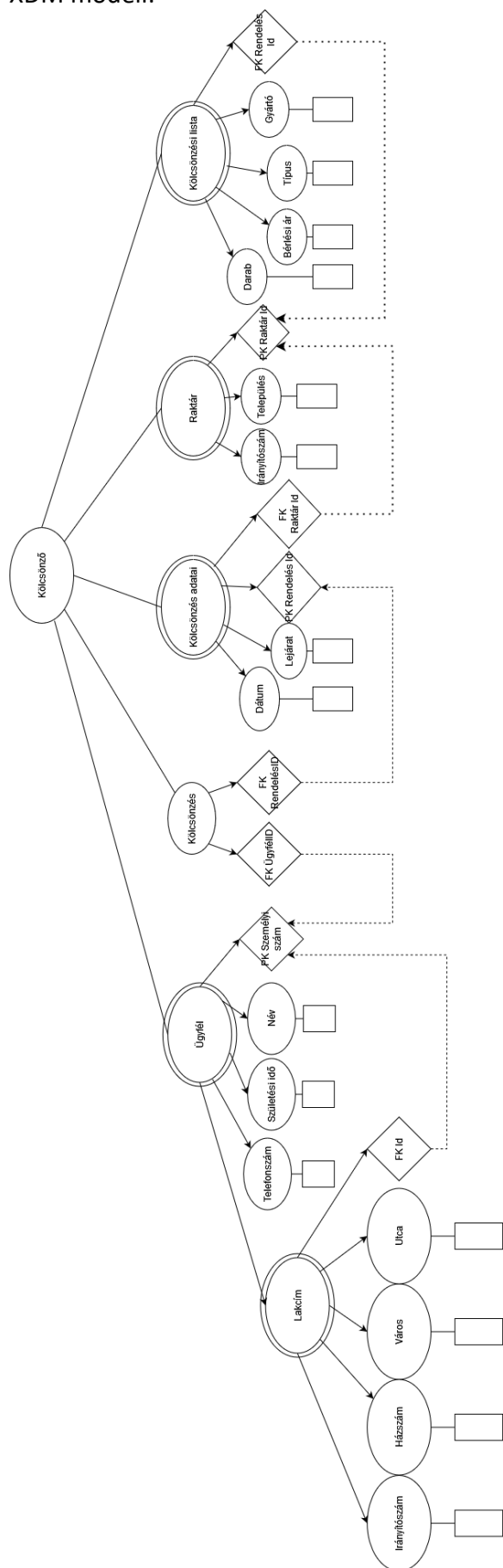
Kölcsönzés adatai – raktár: egy raktárhoz több adat is tartozhat, hiszen egyszerre nem csak egy eszközt lehet rendelni, és vissza fele is igaz, ha nem található egy termék az adott raktárban, akkor egy másiktól megkaphatja hozzá, így N:M kapcsolat.

### **1. feladat:**

# Adatbázis ER modellje:



XDM modell:



XDM modell alapján készült nyers XML kód:

```
<?xml version="1.0" encoding="UTF-8"?>
<kolcsonzo>
  <kolcsonzes ugyfelid="" rendelesid=""></kolcsonzes>
  <ugyfel szigszam=''>
    <nev></nev>
    <szulido></szulido>
    <telefon></telefon>
    <lakcim id=''>
      <varos></varos>
      <iranyitoszam></iranyitoszam>
      <utca></utca>
      <hazszam></hazszam>
    </lakcim>
  </ugyfel>
  <kolcslista rendelesid="">
    <gyarto></gyarto>
    <tipus></tipus>
    <darab></darab>
    <nyilvantart></nyilvantart>
    <ar></ar>
  </kolcslista>
  <kolcsadatok raktarid="" rendelesid="">
    <datum></datum>
    <lejarat></lejarat>
  </kolcsadatok>
  <raktar raktarid="">
    <isz></isz>
    <telepules></telepules>
  </raktar>
</kolcsonzo>
```

XML Schema modell kódja:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="kolcsonzo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='ugyfel' />
        <xs:element ref='kolcsadatok' />
        <xs:element ref='kolcsonzes' />
        <xs:element ref="kolcslista" />
        <xs:element ref="raktar" />
      </xs:sequence>
    </xs:complexType>
```

```

</xs:element>
<xs:element name="ugyfel">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nev"
type="xs:string"></xs:element>
      <xs:element name="szulido"
type="xs:date"></xs:element>
      <xs:element name="telefon"
type="xs:integer"></xs:element>
      <xs:element ref="lakcim"/>
    </xs:sequence>
    <xs:attribute name="szigszam" use="required"
type="xs:string"/>
  </xs:complexType>
  <xs:key name="szigszamKulcs">
    <xs:selector xpath="ugyfel"/>
    <xs:field xpath="@szigszam"/>
  </xs:key>
</xs:element>
<xs:element name="lakcim">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="varos"
type="xs:string"></xs:element>
      <xs:element name="iranyitoszam"
type="xs:integer"></xs:element>
      <xs:element name="hazszam"
type="xs:integer"></xs:element>
      <xs:element name="utca"
type="xs:string"></xs:element>
    </xs:sequence>
    <xs:attribute name="id" use="required"
type="xs:string"/>
  </xs:complexType>
  <xs:keyref name="szigszamIdegenKulcs"
refer="szigszamKulcs">
    <xs:selector xpath="ugyfel"/>
    <xs:field xpath="szigszam"/>
  </xs:keyref>
</xs:element>
<xs:element name="kolcsadatok">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="datum"></xs:element>
    <xs:element name="lejarat"></xs:element>
  </xs:sequence>
  <xs:attribute name="raktarid" use="required"
type="xs:string"/>
  <xs:attribute name="rendelesid"
use="required" type="xs:string"/>
</xs:complexType>
<xs:key name="rendelesidKulcs">
  <xs:selector xpath="kolcsadatok"/>
  <xs:field xpath="@rendelesid"/>
</xs:key>
<xs:keyref name="rendelesidAdatokIdegenKulcs"
refer="rendelesidKulcs">
  <xs:selector xpath="kolcsadatok"/>
  <xs:field xpath="rendelesid"/>
</xs:keyref>
</xs:element>
<xs:element name="kolcslista">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="gyarto"
type="xs:string"></xs:element>
      <xs:element name="tipus"
type="xs:string"></xs:element>
      <xs:element name="darab"
type="xs:integer"></xs:element>
      <xs:element name="nyilvantart"
type="xs:string"></xs:element>
      <xs:element name="ar"
type="xs:integer"></xs:element>
    </xs:sequence>
    <xs:attribute name="rendelesid"
use="required" type="xs:string"/>
  </xs:complexType>
  <xs:keyref name="rendelesidListaIdegenKulcs"
refer="rendelesidKulcs">
    <xs:selector xpath="kolcslista"/>
    <xs:field xpath="rendelesid"/>
  </xs:keyref>

```

```

</xs:element>
<xs:element name="raktar">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="isz"
type="xs:integer"></xs:element>
      <xs:element name="telepules"
type="xs:string"></xs:element>
    </xs:sequence>
    <xs:attribute name="raktarid" use="required"
type="xs:string"/>
  </xs:complexType>
  <xs:key name="raktaridKulcs">
    <xs:selector xpath="raktar"/>
    <xs:field xpath="@raktarid"/>
  </xs:key>
</xs:element>
<xs:element name="kolcsonzes">
  <xs:complexType>
    <xs:attribute name="ugyfelid" use="required"
type="xs:string"/>
    <xs:attribute name="rendelesid"
use="required" type="xs:string"/>
  </xs:complexType>
  <xs:keyref name="ugyfelidIdegenKulcs"
refer="szigszamKulcs">
    <xs:selector xpath="kolcsonzes"/>
    <xs:field xpath="ugyfelid"/>
  </xs:keyref>
  <xs:keyref name="raktarIdegenKulcs"
refer="rendelesidKulcs">
    <xs:selector xpath="kolcsonzes"/>
    <xs:field xpath="rendelesid"/>
  </xs:keyref>
</xs:element>
</xs:schema>

```

## 2. Feladat:

DOM program adatkezeléshez, a commentek a kódban olvashatóak:

Olvasás: a WithData.xml fájlból, ami nem más, mint a fentebb említett nyers xml adatokkal feltöltött változata.

```
package domparsegfmv6t;
```



```

import java.io.File;
import java.util.Iterator;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

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

public class DOMReadGFMV6T {
    public static void main (String [] args) {

        File xml = new File("WithData.xml");
        //
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        try {
            //Dokumentum fa felépítése
            DocumentBuilder db = dbf.newDocumentBuilder();
            //xml beolvasás
            Document doc = db.parse(xml);
            //Gyökér beolvasása próbaként
            System.out.println(doc.getDocumentElement().getNodeName());

            //Minden adat kiírása
            //összeszedi az elemeket a fából
            NodeList nlUgyfel = doc.getElementsByTagName("ugyfel");
            NodeList nlRaktar = doc.getElementsByTagName("raktar");
            System.out.println("Ügyfelek száma: " + nlUgyfel.getLength());
            System.out.println("Raktárak száma: " + nlRaktar.getLength());

            //lista elemeire ciklus
            for (int i = 0; i < nlUgyfel.getLength(); i++) {
                Node n1 = nlUgyfel.item(i);

                if(n1.getNodeType() == Node.ELEMENT_NODE) {
                    Element el1 = (Element) n1;

                    //ügyfél adatainak kiírása
                    System.out.println((i+1) + ". ügyfél neve: " +
el1.getElementsByTagName("nev").item(0).getTextContent());

```

```

        System.out.println((i+1) + ". ügyfél születési ideje: " +
el1.getElementsByTagName("szulido").item(0).getTextContent());
        System.out.println((i+1) + ". ügyfél telefonszáma: " +
el1.getElementsByTagName("telefon").item(0).getTextContent());
        System.out.println((i+1) + ". ügyfél lakcíme: " +
el1.getElementsByTagName("iranyitoszam").item(0).getTextContent() + ". "
+
el1.getElementsByTagName("varos").item(0).getTextContent()
+ ", " +
el1.getElementsByTagName("utca").item(0).getTextContent() +
" " +
el1.getElementsByTagName("hazszam").item(0).getTextContent());

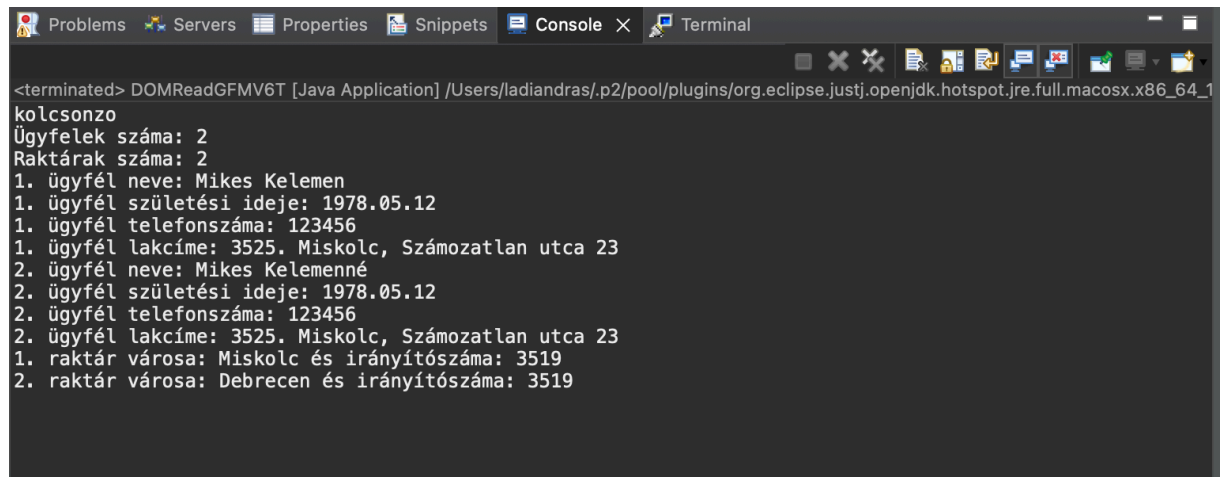
    }
}

//raktár adatainak kiírása
for (int i = 0; i < n1Raktar.getLength(); i++) {
    Node n2 = n1Raktar.item(i);

    if(n2.getNodeType() == Node.ELEMENT_NODE) {
        Element el2 = (Element) n2;
        System.out.println((i+1) + ". raktár városa: " +
el2.getElementsByTagName("telepuless").item(0).getTextContent()
+ " és irányítószáma: " +
el2.getElementsByTagName("isz").item(0).getTextContent());
    }
}
//kivételkezelés
} catch (Exception e) {
    System.out.println("Hiba történt: " +e.getMessage());
}
}
}

```

Kód lefutásakor a kiírás:



```
<terminated> DOMReadGFMV6T [Java Application] /Users/ladiandras.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_1
kolcsonzo
Ügyfelek száma: 2
Raktárak száma: 2
1. ügyfél neve: Mikes Kelemen
1. ügyfél születési ideje: 1978.05.12
1. ügyfél telefonszáma: 123456
1. ügyfél lakcíme: 3525. Miskolc, Számozatlan utca 23
2. ügyfél neve: Mikes Kelemenné
2. ügyfél születési ideje: 1978.05.12
2. ügyfél telefonszáma: 123456
2. ügyfél lakcíme: 3525. Miskolc, Számozatlan utca 23
1. raktár városa: Miskolc és irányítószáma: 3519
2. raktár városa: Debrecen és irányítószáma: 3519
```

Módosítás: szintén a WithData.xml fájlból olvassa be az adatokat, majd kiírja egy új fájlba, aminek a neve WithModifiedData.xml.

```
package domparsegfmv6t;
```

```
import java.io.File;
```

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
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;
```

```
public class DOMModifyGFMV6T {
```

```
    public static void main(String[] args) {
```

```

        //fájl helyének megadása
        String path = "WithData.xml";
        File xml = new File (path);
        //
        DocumentBuilder db;
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        try {
            db = dbf.newDocumentBuilder();
            Document doc = db.parse(xml);
            doc.getDocumentElement().normalize();

            // Módosítás
            updateElement(doc);

            // Törlés
            deleteElement(doc);

            // Element hozzáadása
            addElement(doc);

            // Módosított fájl létrehozása
            writeFile(doc);

        } catch (Exception e){
            e.printStackTrace();
        }
    }

    private static void writeFile(Document doc)
        throws TransformerFactoryConfigurationError, TransformerConfigurationException,
        TransformerException {
        doc.getDocumentElement().normalize();
        TransformerFactory transformerFactory = TransformerFactory.newInstance();

```

```
Transformer transformer = transformerFactory.newTransformer();
DOMSource ds = new DOMSource(doc);
StreamResult result = new StreamResult(new File("WithModifiedData.xml"));
transformer.setOutputProperty(OutputKeys.INDENT, "yes");
transformer.transform(ds, result);
System.out.println("XML módosítva");
}
```

//új elem hozzáadása a raktárhoz

```
private static void addElement(Document doc) {
    NodeList kolcsonzo = doc.getElementsByTagName("raktar");
    Element raktar = null;

    //minden darabra
    for (int i = 0; i < kolcsonzo.getLength(); i++) {
        raktar = (Element) kolcsonzo.item(i);
        Element id = doc.createElement("kapacitas");
        id.appendChild(doc.createTextNode("500"));
        raktar.appendChild(id);
    }
}
```

//Egy elem törlése

```
private static void deleteElement(Document doc) {
    NodeList kolcsonzo = doc.getElementsByTagName("raktar");
    Element raktar = null;

    //minden elementre
    for (int i = 0; i < kolcsonzo.getLength(); i++) {
        raktar = (Element) kolcsonzo.item(i);
        Node isz = raktar.getElementsByTagName("isz").item(0);
        raktar.removeChild(isz);
    }
}
```

```
//Egy ügyfél nevének megváltoztatása (első elem használatával)
private static void updateElement(Document doc) {
    NodeList kolcsonzo = doc.getElementsByTagName("ugyfel");
    Element ugyfel = null;
    // minden elementre
    for (int i = 0; i < kolcsonzo.getLength(); i++) {
        ugyfel = (Element) kolcsonzo.item(i);
        Node nev = ugyfel.getElementsByTagName("nev").item(0).getFirstChild();
        //megfelelő ügyfél megkeresése és adat megváltoztatása
        if (ugyfel.getAttribute("szigszam").contentEquals("654321CD")) {
            nev.setNodeValue("Szabó Magda");
        }
    }
}
}
```

Kód lefutása után láthatjuk, hogy a két fájl különbözik. Megváltoztatta a megfelelő ügyfél nevét, illetve a raktár elementjeit:

```
<terminated> DOMModifyGFMV6T [Java Application] /Users/ladiandras/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64
XML módosítva
```

Eredeti:

```
*WithData.xml X XMLSchemaGFM... DOMReadGFMV6... DOMModifyGFM... »1
26 <raktar raktarid="3519A">
27   <isz>3519</isz>
28   <telepules>Miskolc</telepules>
29 </raktar>
30 <kolcsonzes ügyfelid="123456AB" rendelesid="111111"></kolcsonzes>
31 <ügyfel szigszam='654321CD'>
32   <nev>Mikes Kelemenné</nev>
33   <szulido>1978.05.12</szulido>
34   <telefon>123456</telefon>
35   <lakcim id='123456AB'>
36     <varos>Miskolc</varos>
37     <iranyitoszam>3525</iranyitoszam>
38     <utca>Számozatlan utca</utca>
39     <hazszam>23</hazszam>
40   </lakcim>
41 </ügyfel>
42 <kolcslista rendelesid="111111">
43   <gyarto>Botgvár</gyarto>
44   <tipus>Horgászbot</tipus>
45   <darab>2</darab>
46   <nyilvantart>456ABC</nyilvantart>
47   <ar>3500</ar>
48 </kolcslista>
49 <kolcsadatok raktarid="4000A" rendelesid="111111">
50   <datum>2021.09.10</datum>
51   <lejarat>2021.11.01</lejarat>
52 </kolcsadatok>
53 <raktar raktarid="4000A">
54   <isz>3519</isz>
55   <telepules>Debrecen</telepules>
56 </raktar>
57 </kolcsonzo>
```

Módosított:

```
*WithData.xml DOMReadGFMV6... DOMModifyGFM... WithModified... »2
82 <kolcslista rendelesid="111111">
83   <gyarto>Botgvár</gyarto>
84   <tipus>Horgászbot</tipus>
85   <darab>2</darab>
86   <nyilvantart>456ABC</nyilvantart>
87   <ar>3500</ar>
88 </kolcslista>
89 <kolcsadatok raktarid="4000A" rendelesid="111111">
90   <datum>2021.09.10</datum>
91   <lejarat>2021.11.01</lejarat>
92 </kolcsadatok>
93 <raktar raktarid="4000A">
94   <telepules>Debrecen</telepules>
95   <kapacitas>500</kapacitas>
96 </raktar>
97 </kolcsonzo>
```

```
*WithData.xml  DOMReadGFMV6...  DOMModifyGFM...  WithModified...  ×  ”2
51
52
53     <telepules>Miskolc</telepules>
54
55     <kapacitas>500</kapacitas>
56 </raktar>
57
58 <kolcsontes rendelesid="111111" ugyfelid="123456AB"/>
59
60 ● <ugyfel szigszam="654321CD">
61
62     <nev>Szabó Magda</nev>
63
64     <szulido>1978.05.12</szulido>
65
66     <telefon>123456</telefon>
67
68 ● <lakcim id="123456AB">
69
70     <varos>Miskolc</varos>
71
72     <iranyitoszam>3525</iranyitoszam>
73
74     <utca>Számozatlan utca</utca>
75
76     <hazszam>23</hazszam>
77
78 </lakcim>
79
80 </ugyfel>
81
82 ● <kolcslista rendelesid="111111">
```