

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Könyvtári nyilvántartás létrehozása

Készítette: **Fekete Máté**

Neptunkód: **JR9KY7**

Dátum: **2022.11.30.**

Tartalomjegyzék

Feladat leírása	3
ER modell leírása	3
ER modell konvertálása XDM modellre	5
Adatbázis relációs sémája	5
XML dokumentum létrehozása XDM alapján	6
XML Schema létrehozása	9
Adatok beolvasása és kiírása (Java DOM).....	14
Futási eredmények:.....	25
Adatok módosítása (Java DOM)	31
Futási eredmények:.....	36
Adatok lekérdezése (Java DOM).....	40
Futási eredmények:.....	47

Feladat leírása

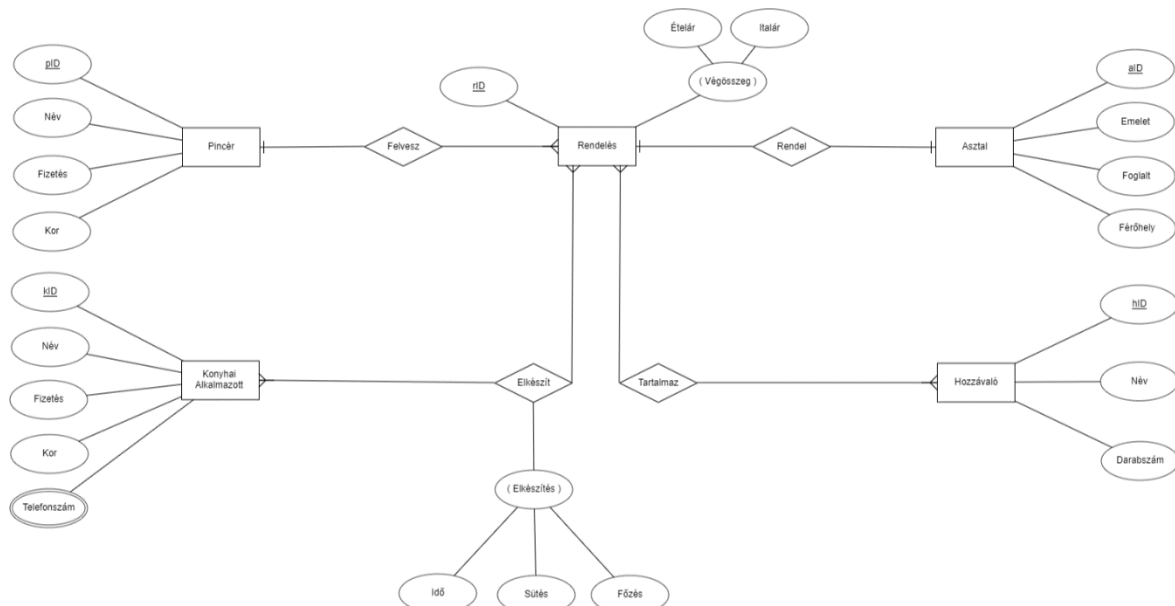
A feladat egy hiteles adatbázis létrehozása, különböző lekérdezések kialakítása XML nyelv felhasználásával. A feladat megoldásának feltétele, hogy az adatbázis tartsa legalább 5 egyedet, egyedenként 4 tulajdonságot, valamint többféle kapcsolatot. A feladatban egy étteremet választottam témának. Ezt követően az adatbázis szerkezetét leíró ER, valamint XDM modelleket kell kialakítani. Amint a megfelelő konvertálások megtörténnek a feladat folytatódik az xml fájl létrehozásával és annak feltöltésével. Miután feltöltöttük a dokumentumot adatokkal, létre kell hozni hozzá egy XML Schema fájlt az előzőleg létrehozott XML dokumentum alapján. A feladat befejezéshez szükséges egy jegyzőkönyv, illetve a feladat megvédése.

ER modell leírása

Itt egy reális ER adatbázist hozok létre, melynek jellemzői a következők:

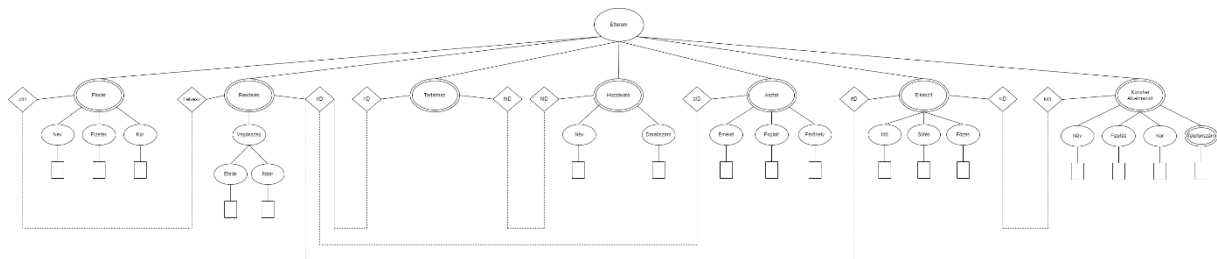
- RENDELÉS EGYED:
 - **rID**: egyedi tulajdonság és azonosító.
 - **Végösszeg** származtatott tulajdonság, ami az **Ételár**, **Italár** tulajdonságokból áll össze
- ASZTAL EGYED:
 - **aID**: egyedi tulajdonság és azonosító.
 - **Emelet** tulajdonság: az asztal hányadik emeleten található az étteremben.
 - **Foglalt** tulajdonság: azt mutatja, hogy éppen foglalt-e az asztal vagy sem.
 - **Férőhely** tulajdonság: az asztalnál egyszerre elférő emberek száma.
- PINCÉR EGYED:
 - **pID** tulajdonság: egyedi tulajdonság és azonosító.
 - **Név** tulajdonság: a pincér neve.
 - **Fizetés** tulajdonság: a pincér fizetése.
 - **Kor** tulajdonság: a pincér életkora.
- KONYHAI ALKALMAZOTT EGYED:
 - **kID** tulajdonság: egyedi tulajdonság és azonosító.
 - **Név** tulajdonság: a konyhai alkalmazott neve.
 - **Fizetés** tulajdonság: a konyhai alkalmazott fizetése.
 - **Kor** tulajdonság: a konyhai alkalmazott életkora.
 - **Telefonszám** tulajdonság: többértékű tulajdonság, ami a konyhai alkalmazott telefonszámát jelzi, ebből több is lehet.

- HOZZÁVALÓ EGYED:
 - **hID** tulajdonság: egyedi tulajdonság és azonosító.
 - **Név** tulajdonság: a hozzávaló neve.
 - **Darabszám** tulajdonság: hány darab van az adott hozzávalóból raktáron.
- RENDEL KAPCSOLAT:
 - Egy-egy (1:1) kapcsolat, ami a Rendelés és az Asztal egyedeket köti össze.
- FELVESZ KAPCSOLAT:
 - Egy-több (1:N) kapcsolat, ami a Pincér (N) és Rendelés (1) egyedeket köti össze.
- TARTALMAZ KAPCSOLAT:
 - Több-több (N:M) kapcsolat, a Rendelés és a Hozzávaló egyedeket köti össze.
- ELKÉSZÍT KAPCSOLAT:
 - Több-több (N:M) kapcsolat, ami a Rendelés és a Konyhai Alkalmazott egyedeket köti össze.
 - Elkészítés összetett tulajdonsága, melyhez kapcsolódnak a következő tulajdonságok:
 - **Idő:** a rendelés elkészítési ideje percben
 - **Sütés:** tartalmaz-e a rendelés sült ételt.
 - **Főzés:** tartalmaz-e a rendelés főtt ételt.



ER modell konvertálása XDM modellre

- A „**Pincér**” egyedből egy elemet készítünk a meglévő tulajdonságaival, kulcsa „pID” lesz.
- A „**Rendelés**” egyedből egy elemet készítünk a meglévő tulajdonságaival, kulcsa „rID” lesz és kap egy idegen kulcsot „pID”, ami a rendeléshez kapcsolódó pincért reprezentálja.
- A „**Hozzávaló**” egyedből egy elemet készítünk a meglévő tulajdonságaival, kulcsa „hID” lesz.
- A „**Tartalmaz**” kapcsolatból egy elemet készítünk, ennek nincsennek tulajdonságai, a „Rendelés” és a „Hozzávaló” elemeket köti össze az „rID” és „hID” segítségével.
- Az „**Asztal**” egyedből egy elemet készítünk a meglévő tulajdonságaival, kulcsa „aID” lesz, és ez a kulcs köti össze a „Rendeléssel”.
- A „**Konyhai alkalmazott**” egyedből egy elemet készítünk a meglévő tulajdonságaival, kulcsa „kID” lesz.
- Az „**Elkészít**” kapcsolatból egy elemet készítünk, a meglévő tulajdonságaival. A „Rendelés” és a „Konyhai alkalmazott” elemeket köti össze az „rID” és „kID” segítségével.



Adatbázis relációs sémája

- Asztal [aID, Férőhely, Emelet, Foglalt]
- Rendelés [rID, Ételár, Italár, pID]
- Pincér [pID, Név, Fizetés, Kor]
- Hozzávaló [hID, Név, Darabszám]
- Konyhai alkalmazott [kID, Név, Fizetés, Kor]
- Elkészít [Idő, Sütés, Főzés, rID, kID]
- Tartalmaz [rID, hID]

XML dokumentum létrehozása XDM alapján

Az XML fájlt fejlesztőkörnyezet segítségével hoztam létre az XDM dokumentum alapján és feltöltöttem adatokkal. Három példányt hoztam létre mindenhez, ezeket kommenttel jelöltem, hogy jobban el lehessen különíteni őket.

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

```
<etterem>
```

```
  <!-- 1. feltöltés -->
```

```
  <pincer pID = "1">
```

```
    <nev>Kiss László</nev>
```

```
    <fizetes>230000</fizetes>
```

```
    <kor>23</kor>
```

```
  </pincer>
```

```
  <rendeles rID = "1" pID = "1">
```

```
    <etelAr>6000</etelAr>
```

```
    <italAr>2500</italAr>
```

```
  </rendeles>
```

```
  <hozzavalo hID = "1">
```

```
    <nev>Burgonya</nev>
```

```
    <darabszam>4</darabszam>
```

```
  </hozzavalo>
```

```
  <asztal aID = "1">
```

```
    <emelet>1</emelet>
```

```
    <foglalt>igen</foglalt>
```

```
    <ferohely>2</ferohely>
```

```
  </asztal>
```

```
  <kAlk kID = "1">
```

```
    <nev>Nagy Sándor</nev>
```

```
    <fizetes>300000</fizetes>
```

```
    <kor>26</kor>
```

```
    <telefonszam>06204793472</telefonszam>
```

```
  </kAlk>
```

```
<elkeszit rID = "1" kID = "1">
```

<ido>40</ido>
<sutes>igen</sutes>
<fozes>igen</fozes>
</elkeszit>
<tartalmaz rID = "1" hID = "1">
</tartalmaz>
<!-- 2. feltöltés -->
<pincer pID = "2">
 <nev>Szabó Norbert</nev>
 <fizetes>280000</fizetes>
 <kor>28</kor>
</pincer>
<rendeles rID = "2" pID = "2">
 <etelAr>5000</etelAr>
 <italAr>2000</italAr>
</rendeles>
<hozzavalo hID = "2">
 <nev>Csirkehús</nev>
 <darabszam>2</darabszam>
</hozzavalo>
<asztal aID = "2">
 <emelet>0</emelet>
 <foglalt>igen</foglalt>
 <ferohely>2</ferohely>
</asztal>
<kAlk kID = "2">
 <nev>Tóth József</nev>
 <fizetes>290000</fizetes>
 <kor>26</kor>
 <telefonszam>06207355921</telefonszam>
</kAlk>

<elkeszit rID = "2" kID = "2">

<ido>45</ido>
<sutes>igen</sutes>
<fozes>nem</fozes>
</elkeszit>
<tartalmaz rID = "2" hID = "2">
</tartalmaz>
<!-- 3. feltöltés -->
<pincer pID = "3">
 <nev>Almási Péter</nev>
 <fizetes>280000</fizetes>
 <kor>25</kor>
</pincer>
<rendeles rID = "3" pID = "3">
 <etelAr>18000</etelAr>
 <italAr>6500</italAr>
</rendeles>
<hozzavalo hID = "3">
 <nev>Marhahús</nev>
 <darabszam>3</darabszam>
</hozzavalo>
<asztal aID = "3">
 <emelet>1</emelet>
 <foglalt>igen</foglalt>
 <ferohely>2</ferohely>
</asztal>
<kAlk kID = "3">
 <nev>Szilva Béla</nev>
 <fizetes>310000</fizetes>
 <kor>30</kor>
 <telefonszam>06206395629</telefonszam>
</kAlk>

<elkeszit rID = "3" kID = "3">


```

        <ido>70</ido>

        <sutes>igen</sutes>

        <fozes>nem</fozes>

    </elkeszit>

    <tartalmaz rID = "3" hID = "3">

    </tartalmaz>

</etterem>

```

XML Schema létrehozása

Elsőnek létrehozom a modellhez szükséges saját típusokat a kulcsok, illetve a dátumok kezeléséhez. Ezt követően létrehozom az elsődleges, valamint az idegen kulcsokat. Beállítom a fő elemet, amiben hivatkozok az alelemekre „ref” segítségével, és beállítom az előfordulásukat korlátlanra. Ezt követően létrehozom a szükséges alelemeket.

```

<?xml version="1.0" encoding="UTF-8"?>

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

    <xs:simpleType name = "idTipus">
        <xs:restriction base = "xs:string">
            <xs:minLength value = "1"/>
            <xs:maxLength value = "3"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name = "telszamTipus">
        <xs:restriction base = "xs:string">
            <xs:pattern value = "\d{11}"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name = "igenNemTipus">
        <xs:restriction base = "xs:string">
            <xs:enumeration value = "igen"/>
            <xs:enumeration value = "nem"/>
        </xs:restriction>
    </xs:simpleType>

```

</xs:restriction>

</xs:simpleType>

<xs:key name = "pincer_kulcs">

<xs:selector xpath = "pincer"/>

<xs:field xpath = "@pID" />

</xs:key>

<xs:key name = "rendeles_kulcs">

<xs:selector xpath = "rendeles"/>

<xs:field xpath = "@rID" />

</xs:key>

<xs:key name = "hozzavalo_kulcs">

<xs:selector xpath = "hozzavalo"/>

<xs:field xpath = "@hID" />

</xs:key>

<xs:key name = "asztal_kulcs">

<xs:selector xpath = "asztal"/>

<xs:field xpath = "@aID" />

</xs:key>

<xs:key name = "kAlk_kulcs">

<xs:selector xpath = "kAlk"/>

<xs:field xpath = "@kID" />

</xs:key>

<xs:keyref refer = "pincer_kulcs" name = "rendeles_pincer_fkey">

<xs:selector xpath = "rendeles" />

<xs:field xpath = "felvesz" />

</xs:keyref>

<xs:keyref refer = "rendeles_kulcs" name = "elkeszit_rendeles_fkey">

```
    <xs:selector xpath = "elkeszit" />
    <xs:field xpath = "elkeszit_r" />
</xs:keyref>
```

```
<xs:keyref refer = "kAlk_kulcs" name = "elkeszit_kAlk_fkey">
    <xs:selector xpath = "elkeszit" />
    <xs:field xpath = "elkeszit_k" />
</xs:keyref>
```

```
<xs:keyref refer = "rendeles_kulcs" name = "tartalmaz_rendeles_fkey">
    <xs:selector xpath = "tartalmaz" />
    <xs:field xpath = "tartalmaz_r" />
</xs:keyref>
```

```
<xs:keyref refer = "hozzavalo_kulcs" name = "tartalmaz_hozzavalo_fkey">
    <xs:selector xpath = "tartalmaz" />
    <xs:field xpath = "tartalmaz_h" />
</xs:keyref>
```

```
<xs:element name = "etterem">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref = "pincer" maxOccurs="unbounded"></xs:element>
            <xs:element ref = "rendeles" maxOccurs="unbounded"></xs:element>
            <xs:element ref = "asztal" maxOccurs="unbounded"></xs:element>
            <xs:element ref = "hozzavalo" maxOccurs="unbounded"></xs:element>
            <xs:element ref = "kAlk" maxOccurs="unbounded"></xs:element>
            <xs:element ref = "elkeszit" maxOccurs="unbounded"></xs:element>
            <xs:element ref = "tartalmaz" maxOccurs="unbounded"></xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

```
<xs:element name = "pincer">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name = "nev" type = "xs:string" maxOccurs = "unbounded"/>
    <xs:element name = "fizetes" type = "xs:integer" maxOccurs =
"unbounded"/>
    <xs:element name = "kor" type = "xs:integer" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:attribute name = "pID" type = "idTipus" use = "required"></xs:attribute>
</xs:complexType>
</xs:element>

<xs:element name = "rendeles">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "etelAr" type = "xs:integer" maxOccurs =
"unbounded"/>
      <xs:element name = "italAr" type = "xs:integer" maxOccurs =
"unbounded"/>
    </xs:sequence>
    <xs:attribute name = "rID" type = "idTipus" use = "required"></xs:attribute>
    <xs:attribute name = "pID" type = "idTipus" use = "required"></xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name = "hozzavallo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "nev" type = "xs:string" maxOccurs = "unbounded"/>
      <xs:element name = "darabszam" type = "xs:integer" maxOccurs =
"unbounded"/>
    </xs:sequence>
    <xs:attribute name = "hID" type = "idTipus" use = "required"></xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name = "asztal">

```

```

<xs:complexType>
    <xs:sequence>
        <xs:element name = "emelet" type = "xs:integer" maxOccurs =
"unbounded"/>
        <xs:element name = "foglalt" type = "igenNemTipus" maxOccurs =
"unbounded"/>
        <xs:element name = "ferohely" type = "xs:integer" maxOccurs =
"unbounded"/>
    </xs:sequence>
    <xs:attribute name = "aID" type = "idTipus" use = "required"></xs:attribute>
</xs:complexType>
</xs:element>

<xs:element name = "kAlk">
    <xs:complexType>
        <xs:sequence>
            <xs:element name = "nev" type = "xs:string" maxOccurs = "unbounded"/>
            <xs:element name = "fizetes" type = "xs:integer" maxOccurs =
"unbounded"/>
            <xs:element name = "kor" type = "xs:integer" maxOccurs = "unbounded"/>
            <xs:element name = "telefonszam" type = "telszamTipus" maxOccurs =
"unbounded"/>
        </xs:sequence>
        <xs:attribute name = "kID" type = "idTipus" use = "required"></xs:attribute>
    </xs:complexType>
</xs:element>

<xs:element name = "elkeszit">
    <xs:complexType>
        <xs:sequence>
            <xs:element name = "ido" type = "xs:integer" maxOccurs = "unbounded"/>
            <xs:element name = "sutes" type = "igenNemTipus" maxOccurs =
"unbounded"/>
            <xs:element name = "fozes" type = "igenNemTipus" maxOccurs =
"unbounded"/>
        </xs:sequence>

```

```

        <xs:attribute name = "rID" type = "idTipus" use = "required"></xs:attribute>
        <xs:attribute name = "kID" type = "idTipus" use = "required"></xs:attribute>
    </xs:complexType>
</xs:element>

<xs:element name = "tartalmaz">
    <xs:complexType>
        <xs:attribute name = "rID" type = "idTipus" use = "required"></xs:attribute>
        <xs:attribute name = "hID" type = "idTipus" use = "required"></xs:attribute>
    </xs:complexType>
</xs:element>

</xs:schema>

```

Adatok beolvasása és kiírása (Java DOM)

Elsőként a megfelelő csomagokat importáltam a Java osztályba, majd megnyitottam a szükséges fájlokat, amikkel dolgoznom kellett. Létrehoztam a buildereket és meghatároztam a gyökérelemet. Ezt követően a gyermekelemek hozzáadása a gyökérelemhez, majd formázott output-ba rendezés, és kiírás.

```

public class DOMReadJR9KY7 {

    public static void main(String[] args)
        throws ParserConfigurationException, SAXException, IOException,
        TransformerException {

        // Meglévő xml fájl megnyitása és új létrehozása

        File xmlFile = new File("XMLJR9KY7.xml");
        File myFile = new File("XMLJR9KY71.xml");

        // Dokumentum builderek létrehozása

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
    }
}

```

// Dokumentumok beállítása

Document doc = dBuilder.parse(xmlFile);

Document doc2 = dBuilder.newDocument();

// Gyökérelem meghatározása

Element root = doc2.createElementNS("XMLJR9KY7", "etterem");

doc2.appendChild(root);

// Elemek létrehozása a meglévő XML dokumentum és séma alapján

root.appendChild(createPincer(doc2, "1", "Kis Zoltan", "240000", "25"));

root.appendChild(createPincer(doc2, "2", "Nagy Albert", "245000", "24"));

root.appendChild(createPincer(doc2, "3", "Toth Balazs", "228000", "22"));

root.appendChild(createRendeles(doc2, "1", "1", "5600", "2500"));

root.appendChild(createRendeles(doc2, "2", "2", "4100", "1500"));

root.appendChild(createRendeles(doc2, "3", "3", "5900", "3500"));

root.appendChild(createHozzavalo(doc2, "1", "Csirkehus", "2"));

root.appendChild(createHozzavalo(doc2, "2", "Marhahus", "3"));

root.appendChild(createHozzavalo(doc2, "3", "Borjuhus", "2"));

root.appendChild(createAsztal(doc2, "1", "1", "igen", "2"));

root.appendChild(createAsztal(doc2, "2", "0", "igen", "2"));

root.appendChild(createAsztal(doc2, "3", "0", "igen", "4"));

root.appendChild(createKalk(doc2, "1", "Kiss Anita", "290000", "28", "06703579415"));

root.appendChild(createKalk(doc2, "2", "Nagy Anita", "320000", "32", "06200252585"));

root.appendChild(createKalk(doc2, "3", "Toth Anita", "210000", "21", "06305548644"));

root.appendChild(createElkeszit(doc2, "1", "1", "40", "igen", "nem"));

root.appendChild(createElkeszit(doc2, "2", "2", "70", "igen", "igen"));

root.appendChild(createElkeszit(doc2, "3", "3", "45", "igen", "nem"));

root.appendChild(createTartalmaz(doc2, "1", "1"));

root.appendChild(createTartalmaz(doc2, "2", "2"));

root.appendChild(createTartalmaz(doc2, "3", "3"));

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transf = transformerFactory.newTransformer();

// Output kialakítása

```
transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");  
transf.setOutputProperty(OutputKeys.INDENT, "yes");  
transf.setOutputProperty("{http://xml.apache.org/xslt}indent-amount", "2");
```

```
DOMSource source = new DOMSource(doc2);
```

// Kiírás

```
StreamResult console = new StreamResult(System.out);  
StreamResult file = new StreamResult(myFile);
```

```
transf.transform(source, console);  
transf.transform(source, file);
```

// Megnyitott xml file kezelése

```
doc.getDocumentElement().normalize();
```

// Gyökér elem kiírás

```
System.out.println("\nGyokerelem: " + doc.getDocumentElement().getNodeName());
```

// Gyerek elemek listába rendezése

```
NodeList pincerList = doc.getElementsByTagName("pincer");  
NodeList rendelesList = doc.getElementsByTagName("rendeles");  
NodeList hozzavaloList = doc.getElementsByTagName("hozzavalo");  
NodeList asztalList = doc.getElementsByTagName("asztal");  
NodeList kAlkList = doc.getElementsByTagName("kAlk");  
NodeList elkeszitList = doc.getElementsByTagName("elkeszit");  
NodeList tartalmazList = doc.getElementsByTagName("tartalmaz");
```

// Fájlba írás

```
StringWriter sw = new StringWriter();  
transf.transform(source, new StreamResult(sw));  
FileWriter fw = new FileWriter("file.txt");  
fw.write(sw.toString());
```



```
fw.close();
```

```
// Listák feltöltése
```

```
for (int i = 0; i < pincerList.getLength(); i++) {
```

```
    Node pincerNode = pincerList.item(i);
```

```
    System.out.println("\nAktualis elem: " + pincerNode.getNodeName());
```

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

```
        Element elem = (Element) pincerNode;
```

```
        String pID = elem.getAttribute("pID");
```

```
        Node node1 = elem.getElementsByTagName("nev").item(0);
```

```
        String nev = node1.getTextContent();
```

```
        Node node2 = elem.getElementsByTagName("fizetes").item(0);
```

```
        String fizetes = node2.getTextContent();
```

```
        System.out.println("Pincer azonosító: " + pID);
```

```
        System.out.println("Név: " + nev);
```

```
        System.out.println("Fizetes: " + fizetes);
```

```
    } // end if
```

```
} // end for
```

```
for (int i = 0; i < rendelesList.getLength(); i++) {
```

```
    Node rendelesNode = rendelesList.item(i);
```

```
    System.out.println("\nAktualis elem: " + rendelesNode.getNodeName());
```

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

```
        Element elem = (Element) rendelesNode;
```

```
        String rID = elem.getAttribute("rID");
```

```

        String pID = elem.getAttribute("pID");

        Node node1 = elem.getElementsByTagName("etelAr").item(0);
        String etelAr = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("italAr").item(0);
        String italAr = node2.getTextContent();

        System.out.println("Rendeles azonosito: " + rID);
        System.out.println("Pincer azonosito: " + pID);
        System.out.println("Etel(ek) ara: " + etelAr);
        System.out.println("Ital(ok) ara: " + italAr);
    } // end if
} // end for

for (int i = 0; i < hozzavalolist.getLength(); i++) {

    Node hozzavalonode = hozzavalolist.item(i);

    System.out.println("\nAktualis elem: " + hozzavalonode.getNodeName());

    if (hozzavalonode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) hozzavalonode;
        String hID = elem.getAttribute("hID");

        Node node1 = elem.getElementsByTagName("nev").item(0);
        String nev = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("darabszam").item(0);
        String darabszam = node2.getTextContent();

        System.out.println("Hozzavalolist azonosito: " + hID);
        System.out.println("Neve: " + nev);
        System.out.println("Darabszam: " + darabszam);
    }
}

```

```

        } // end if
    } // end for

    for (int i = 0; i < asztalList.getLength(); i++) {

        Node asztalNode = asztalList.item(i);

        System.out.println("\nAktualis elem: " + asztalNode.getNodeName());

        if (asztalNode.getNodeType() == Node.ELEMENT_NODE) {

            Element elem = (Element) asztalNode;
            String aID = elem.getAttribute("aID");

            Node node1 = elem.getElementsByTagName("emelet").item(0);
            String emelet = node1.getTextContent();

            Node node2 = elem.getElementsByTagName("foglalt").item(0);
            String foglalalt = node2.getTextContent();

            Node node3 = elem.getElementsByTagName("ferohely").item(0);
            String ferohely = node3.getTextContent();

            System.out.println("Asztal azonosito: " + aID);
            System.out.println("Emelet: " + emelet);
            System.out.println("Foglalt: " + foglalalt);
            System.out.println("Ferohely: " + ferohely);
        } // end if
    } // end for

    for (int i = 0; i < kAlkList.getLength(); i++) {

        Node kAlkNode = kAlkList.item(i);

        System.out.println("\nAktualis elem: " + kAlkNode.getNodeName());
    }

```

```

        if (kAlkNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) kAlkNode;
            String kID = elem.getAttribute("kID");

            Node node1 = elem.getElementsByTagName("nev").item(0);
            String nev = node1.getTextContent();

            Node node2 = elem.getElementsByTagName("fizetes").item(0);
            String fizetes = node2.getTextContent();

            Node node3 = elem.getElementsByTagName("kor").item(0);
            String kor = node3.getTextContent();

            Node node4 = elem.getElementsByTagName("telefonszam").item(0);
            String telszam = node4.getTextContent();

            System.out.println("Konyhai alkalmazott azonosito: " + kID);
            System.out.println("Nev: " + nev);
            System.out.println("Fizetes: " + fizetes);
            System.out.println("Kor: " + kor);
            System.out.println("Telefonszam: " + telszam);
        } // end if
    } // end for

    for (int i = 0; i < elkeszitList.getLength(); i++) {

        Node elkeszitNode = elkeszitList.item(i);

        System.out.println("\nAktualis elem: " + elkeszitNode.getNodeName());

        if (elkeszitNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem1 = (Element) elkeszitNode;
            String rID = elem1.getAttribute("rID");

```

```

        Element elem2 = (Element) elkeszitNode;
        String kID = elem2.getAttribute("kID");

        Node node1 = elem1.getElementsByTagName("ido").item(0);
        String ido = node1.getTextContent();

        Node node2 = elem1.getElementsByTagName("sutes").item(0);
        String sutes = node2.getTextContent();

        Node node3 = elem1.getElementsByTagName("fozes").item(0);
        String fozes = node3.getTextContent();

        System.out.println("Rendeles azonosito: " + rID);
        System.out.println("Konyhai alkalmazott azonosito: " + kID);
        System.out.println("Elkeszitesi ido: " + ido);
        System.out.println("Sutes: " + sutes);
        System.out.println("Fozes: " + fozes);
    } // end if
} // end for

for (int i = 0; i < tartalmazList.getLength(); i++) {

    Node tartalmazNode = tartalmazList.item(i);

    System.out.println("\nAktualis elem: " + tartalmazNode.getNodeName());

    if (tartalmazNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem1 = (Element) tartalmazNode;
        String rID = elem1.getAttribute("rID");

        Element elem2 = (Element) tartalmazNode;
        String hID = elem2.getAttribute("hID");
    }
}

```

```

        System.out.println("Rendeles azonosito: " + rID);
        System.out.println("Hozzavalo azonosito: " + hID);
    } // end if
} // end for

} // end main

// A feltöltéshez szükséges függvények létrehozása
private static Node createPincer(Document doc2, String pID, String nev, String fizetes, String kor) {
    Element pincer = doc2.createElement("pincer");

    pincer.setAttribute("pID", pID);
    pincer.appendChild(createPincerElement(doc2, "nev", nev));
    pincer.appendChild(createPincerElement(doc2, "fizetes", fizetes));
    pincer.appendChild(createPincerElement(doc2, "kor", kor));

    return pincer;
} // end createPincer

private static Node createPincerElement(Document doc2, String name, String value) {
    Element node = doc2.createElement(name);
    node.appendChild(doc2.createTextNode(value));

    return node;
} // end createPincerElement

private static Node createRendeles(Document doc2, String rID, String pID, String etelAr, String italAr)
{
    Element rendeles = doc2.createElement("rendeles");

    rendeles.setAttribute("rID", rID);
    rendeles.setAttribute("pID", pID);
    rendeles.appendChild(createRendelesElement(doc2, "etelAr", etelAr));
    rendeles.appendChild(createRendelesElement(doc2, "italAr", italAr));

```

```

        return rendeles;
    } // end createRendeles

    private static Node createRendelesElement(Document doc2, String name, String value) {
        Element node = doc2.createElement(name);
        node.appendChild(doc2.createTextNode(value));

        return node;
    } // end createRendelesElement

    private static Node createHozzavalo(Document doc2, String hID, String nev, String darabszam) {
        Element hozzavalo = doc2.createElement("hozzavalo");

        hozzavalo.setAttribute("hID", hID);
        hozzavalo.appendChild(createHozzavaloElement(doc2, "nev", nev));
        hozzavalo.appendChild(createHozzavaloElement(doc2, "darabszam", darabszam));

        return hozzavalo;
    } // end createHozzavalo

    private static Node createHozzavaloElement(Document doc2, String name, String value) {
        Element node = doc2.createElement(name);
        node.appendChild(doc2.createTextNode(value));

        return node;
    } // end createHozzavaloElement

    private static Node createAsztal(Document doc2, String aID, String emelet, String foglalt, String
ferohely) {
        Element asztal = doc2.createElement("asztal");

        asztal.setAttribute("aID", aID);
        asztal.appendChild(createAsztalElement(doc2, "emelet", emelet));
        asztal.appendChild(createAsztalElement(doc2, "foglalt", foglalt));
        asztal.appendChild(createAsztalElement(doc2, "ferohely", ferohely));
    }

```

```

        return asztal;
    } // end createAsztal

private static Node createAsztalElement(Document doc2, String name, String value) {
    Element node = doc2.createElement(name);
    node.appendChild(doc2.createTextNode(value));

    return node;
} // end createAsztalElement

private static Node createKAlk(Document doc2, String kID, String nev, String fizetes, String kor, String
telszam) {
    Element kAlk = doc2.createElement("kAlk");

    kAlk.setAttribute("kID", kID);
    kAlk.appendChild(createKAlkElement(doc2, "nev", nev));
    kAlk.appendChild(createKAlkElement(doc2, "fizetes", fizetes));
    kAlk.appendChild(createKAlkElement(doc2, "kor", kor));
    kAlk.appendChild(createKAlkElement(doc2, "telefonszam", telszam));

    return kAlk;
} // end createkAlk

private static Node createKAlkElement(Document doc2, String name, String value) {
    Element node = doc2.createElement(name);
    node.appendChild(doc2.createTextNode(value));

    return node;
} // end createKAlkElement

private static Node createElkeszit(Document doc2, String rID, String kID, String ido, String sutes,
String fozes) {
    Element elkeszit = doc2.createElement("elkeszit");

```



```

        elkeszit.setAttribute("rID", rID);
        elkeszit.setAttribute("kID", kID);

        elkeszit.appendChild(createElkeszitElement(doc2, "ido", ido));
        elkeszit.appendChild(createElkeszitElement(doc2, "sutes", sutes));
        elkeszit.appendChild(createElkeszitElement(doc2, "fozes", fozes));

        return elkeszit;
    } // end createElkeszit

    private static Node createElkeszitElement(Document doc2, String name, String value) {
        Element node = doc2.createElement(name);
        node.appendChild(doc2.createTextNode(value));

        return node;
    } // end createElkeszitElement

    private static Node createTartalmaz(Document doc2, String rID, String hID) {
        Element tartalmaz = doc2.createElement("tartalmaz");

        tartalmaz.setAttribute("rID", rID);
        tartalmaz.setAttribute("hID", hID);

        return tartalmaz;
    } // end createElkeszit

} // end class

```

Futási eredmények:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<etterem xmlns="XMLJR9KY7">
  <pincer pID="1">
    <nev>Kis Zoltan</nev>
    <fizetes>240000</fizetes>
    <kor>25</kor>
  </pincer>

```

<pincer pID="2">
 <nev>Nagy Albert</nev>
 <fizetes>245000</fizetes>
 <kor>24</kor>
</pincer>
<pincer pID="3">
 <nev>Toth Balazs</nev>
 <fizetes>228000</fizetes>
 <kor>22</kor>
</pincer>
<rendeles pID="1" rID="1">
 <etelAr>5600</etelAr>
 <italAr>2500</italAr>
</rendeles>
<rendeles pID="2" rID="2">
 <etelAr>4100</etelAr>
 <italAr>1500</italAr>
</rendeles>
<rendeles pID="3" rID="3">
 <etelAr>5900</etelAr>
 <italAr>3500</italAr>
</rendeles>
<hozzavalo hID="1">
 <nev>Csirkehus</nev>
 <darabszam>2</darabszam>
</hozzavalo>
<hozzavalo hID="2">
 <nev>Marhahus</nev>
 <darabszam>3</darabszam>
</hozzavalo>
<hozzavalo hID="3">
 <nev>Borjuhus</nev>
 <darabszam>2</darabszam>
</hozzavalo>
<asztal aID="1">

<emelet>1</emelet>
<foglalt>igen</foglalt>
<ferohely>2</ferohely>
</asztal>
<asztal aID="2">
<emelet>0</emelet>
<foglalt>igen</foglalt>
<ferohely>2</ferohely>
</asztal>
<asztal aID="3">
<emelet>0</emelet>
<foglalt>igen</foglalt>
<ferohely>4</ferohely>
</asztal>
<kAlk kID="1">
<nev>Kiss Anita</nev>
<fizetes>290000</fizetes>
<kor>28</kor>
<telefonszam>06703579415</telefonszam>
</kAlk>
<kAlk kID="2">
<nev>Nagy Anita</nev>
<fizetes>320000</fizetes>
<kor>32</kor>
<telefonszam>06200252585</telefonszam>
</kAlk>
<kAlk kID="3">
<nev>Toth Anita</nev>
<fizetes>210000</fizetes>
<kor>21</kor>
<telefonszam>06305548644</telefonszam>
</kAlk>
<elkeszit kID="1" rID="1">
<ido>40</ido>
<sutes>igen</sutes>

```
<fozes>nem</fozes>
</elkeszit>
<elkeszit kID="2" rID="2">
  <ido>70</ido>
  <sutes>igen</sutes>
  <fozes>igen</fozes>
</elkeszit>
<elkeszit kID="3" rID="3">
  <ido>45</ido>
  <sutes>igen</sutes>
  <fozes>nem</fozes>
</elkeszit>
<tartalmaz hID="1" rID="1"/>
<tartalmaz hID="2" rID="2"/>
<tartalmaz hID="3" rID="3"/>
</etterem>
```

Gyokerelem: etterem

Aktualis elem: pincer

Pincer azonosito: 1

Nev: Kiss László

Fizetes: 230000

Aktualis elem: pincer

Pincer azonosito: 2

Nev: Szabó Norbert

Fizetes: 280000

Aktualis elem: pincer

Pincer azonosito: 3

Nev: Almási Péter

Fizetes: 280000

Aktualis elem: rendeles

Rendeles azonosító: 1

Pincer azonosító: 1

Etel(ek) ára: 6000

Ital(ok) ára: 2500

Aktuális elem: rendeles

Rendeles azonosító: 2

Pincer azonosító: 2

Etel(ek) ára: 5000

Ital(ok) ára: 2000

Aktuális elem: rendeles

Rendeles azonosító: 3

Pincer azonosító: 3

Etel(ek) ára: 18000

Ital(ok) ára: 6500

Aktuális elem: hozzavaló

Hozzavaló azonosító: 1

Neve: Burgonya

Darabszám: 4

Aktuális elem: hozzavaló

Hozzavaló azonosító: 2

Neve: Csirkehús

Darabszám: 2

Aktuális elem: hozzavaló

Hozzavaló azonosító: 3

Neve: Marhahús

Darabszám: 3

Aktuális elem: asztal

Asztal azonosító: 1

Emelet: 1

Foglalt: igen

Ferohely: 2

Aktualis elem: asztal

Asztal azonosito: 2

Emelet: 0

Foglalt: igen

Ferohely: 2

Aktualis elem: asztal

Asztal azonosito: 3

Emelet: 1

Foglalt: igen

Ferohely: 2

Aktualis elem: kAlk

Konyhai alkalmazott azonosito: 1

Nev: Nagy Sándor

Fizetes: 300000

Kor: 26

Telefonszam: 06204793472

Aktualis elem: kAlk

Konyhai alkalmazott azonosito: 2

Nev: Tóth József

Fizetes: 290000

Kor: 26

Telefonszam: 06207355921

Aktualis elem: kAlk

Konyhai alkalmazott azonosito: 3

Nev: Szilva Béla

Fizetes: 310000

Kor: 30

Telefonszam: 06206395629

Aktualis elem: elkeszit

Rendeles azonosito: 1

Konyhai alkalmazott azonosito: 1

Elkeszitesi ido: 40

Sutes: igen

Fozes: igen

Aktualis elem: elkeszit

Rendeles azonosito: 2

Konyhai alkalmazott azonosito: 2

Elkeszitesi ido: 45

Sutes: igen

Fozes: nem

Aktualis elem: elkeszit

Rendeles azonosito: 3

Konyhai alkalmazott azonosito: 3

Elkeszitesi ido: 70

Sutes: igen

Fozes: nem

Aktualis elem: tartalmaz

Rendeles azonosito: 1

Hozzavalo azonosito: 1

Aktualis elem: tartalmaz

Rendeles azonosito: 2

Hozzavalo azonosito: 2

Aktualis elem: tartalmaz

Rendeles azonosito: 3

Hozzavalo azonosito: 3

Adatok módosítása (Java DOM)

Elsőként a megfelelő csomagokat importáltam a Java osztályba, majd megnyitottam a szükséges fájlokat, amikkel dolgoznom kellett. Létrehoztam a buildereket és meghatároztam a gyökérelemet. Minden módosításhoz egy ciklus készítése, amiben maga a vizsgálás és módosítás zajlik. Végül a módosított elemek kiírása a konzolra.

```
public class DOMModifyJR9KY7 {

    public static void main(String[] args)

        throws IOException, ParserConfigurationException, TransformerException,
SAXException {

        try {

            // File megnyitása

            File inputFile = new File("XMLJR9KY7.xml");

            // Dokumentum builderek létrehozása

            DocumentBuilderFactory docfactory = DocumentBuilderFactory.newInstance();

            DocumentBuilder docBuilder = docfactory.newDocumentBuilder();

            // Dokumentum beállítása

            Document doc = docBuilder.parse(inputFile);

            // Elemek megkeresése tag név alapján

            Node pincer = doc.getElementsByTagName("pincer").item(0);

            Node rendeles = doc.getElementsByTagName("rendeles").item(0);

            Node hozzavalo = doc.getElementsByTagName("hozzavalo").item(0);

            Node asztal = doc.getElementsByTagName("asztal").item(0);

            Node kAlk = doc.getElementsByTagName("kAlk").item(0);

            NodeList pincerList = pincer.getChildNodes();
```



```

NodeList rendelesList = rendeles.getChildNodes();

NodeList hozzavaloList = hozzavalo.getChildNodes();

NodeList asztalList = asztal.getChildNodes();

NodeList kAlkList = kAlk.getChildNodes();


// Pincér elemenen történő módosítás

for (int i = 0; i < pincerList.getLength(); i++) {

    Node node = pincerList.item(i);


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

        Element element = (Element) node;


        if ("nev".equals(element.getNodeName())) {

            if ("Kiss László".equals(element.getTextContent())) {

                element.setTextContent("Kovacs Gabor");

            }

        }

    }

}


// Rendelés elemen történő módosítás

for (int i = 0; i < rendelesList.getLength(); i++) {

    Node node = rendelesList.item(i);


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

        Element element = (Element) node;


        if ("etelAr".equals(element.getNodeName())) {

            if ("6000".equals(element.getTextContent())) {

```

```

        element.setTextContent("10000");
    }
}

}

}

// Hozzávaló elemen történő módosítás

for (int i = 0; i < hozzavaloList.getLength(); i++) {

    Node node = hozzavaloList.item(i);

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

        Element element = (Element) node;

        if ("nev".equals(element.getNodeName())) {

            if ("Burgonya".equals(element.getTextContent())) {

                element.setTextContent("Krumpli");

            }

        }

    }

}

// Asztal elemen történő módosítás

for (int i = 0; i < asztalList.getLength(); i++) {

    Node node = asztalList.item(i);

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

        Element element = (Element) node;

        if ("emelet".equals(element.getNodeName())) {

```

```

        if ("1".equals(element.getTextContent())) {

            element.setTextContent("0");

        }

    }

}

// Konyhai alkalmazott elemen történő módosítás
for (int i = 0; i < kAlkList.getLength(); i++) {

    Node node = kAlkList.item(i);

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

        Element element = (Element) node;

        if ("nev".equals(element.getNodeName())) {

            if ("Nagy Sándor".equals(element.getTextContent())) {

                element.setTextContent("Arany Norbert");

            }

        }

    }

}

// Valtoztatasok kiirasa

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transf = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

System.out.println("Modositott fajl: ");

StreamResult consoleResult = new StreamResult(System.out);

```

```

        transf.transform(source, consoleResult);

    } catch (IOException e) {

        e.printStackTrace();

    } catch (ParserConfigurationException e) {

        e.printStackTrace();

    } catch (TransformerException e) {

        e.printStackTrace();

    } catch (SAXException e) {

        e.printStackTrace();

    }

} // end main

} // end class

```

Futási eredmények:

Modosított fájl:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><etterem>
```

```
<!-- 1. feltöltés -->
```

```
<pincer pID="1">
```

```
<nev>Kovacs Gabor</nev>
```

```
<fizetes>230000</fizetes>
```

```
<kor>23</kor>
```

```
</pincer>
```

```
<rendeles pID="1" rID="1">
```

```
<etelAr>10000</etelAr>
```

```
<italAr>2500</italAr>
```

</rendeles>

<hozzavalo hID="1">

<nev>Krumpli</nev>

<darabszam>4</darabszam>

</hozzavalo>

<asztal aID="1">

<emelet>0</emelet>

<foglalt>igen</foglalt>

<ferohely>2</ferohely>

</asztal>

<kAlk kID="1">

<nev>Arany Norbert</nev>

<fizetes>300000</fizetes>

<kor>26</kor>

<telefonszam>06204793472</telefonszam>

</kAlk>

<elkeszit kID="1" rID="1">

<ido>40</ido>

<sutes>igen</sutes>

<fozes>igen</fozes>

</elkeszit>

<tartalmaz hID="1" rID="1">

</tartalmaz>

<!-- 2. feltöltés -->

<pincer pID="2">

<nev>Szabó Norbert</nev>

<fizetes>280000</fizetes>

<kor>28</kor>

</pincer>

<rendeles pID="2" rID="2">

<etelAr>5000</etelAr>

<italAr>2000</italAr>

</rendeles>

<hozzavalo hID="2">

<nev>CsirkeHús</nev>

<darabszam>2</darabszam>

</hozzavalo>

<asztal aID="2">

<emelet>0</emelet>

<foglalt>igen</foglalt>

<ferohely>2</ferohely>

</asztal>

<kAlk kID="2">

<nev>Tóth József</nev>

<fizetes>290000</fizetes>

<kor>26</kor>

<telefonszam>06207355921</telefonszam>

</kAlk>

<elkeszit kID="2" rID="2">

<ido>45</ido>

<sutes>igen</sutes>

<fozes>nem</fozes>

</elkeszit>

<tartalmaz hID="2" rID="2">

</tartalmaz>

<!-- 3. feltöltés -->

<pincer pID="3">

<nev>AlmÄsi PÄter</nev>

<fizetes>280000</fizetes>

<kor>25</kor>

</pincer>

<rendeles pID="3" rID="3">

<etelAr>18000</etelAr>

<italAr>6500</italAr>

</rendeles>

<hozzavalo hID="3">

<nev>MarhahÄss</nev>

<darabszam>3</darabszam>

</hozzavalo>

<asztal aID="3">

<emelet>1</emelet>

```
<foglalt>igen</foglalt>

<ferohely>2</ferohely>

</asztal>

<kAlk kID="3">

    <nev>Szilva B  la</nev>

    <fizetes>310000</fizetes>

    <kor>30</kor>

    <telefonszam>06206395629</telefonszam>

</kAlk>

<elkeszit kID="3" rID="3">

    <ido>70</ido>

    <sutes>igen</sutes>

    <fozes>nem</fozes>

</elkeszit>

<tartalmaz hID="3" rID="3">

</tartalmaz>

</etterem>
```

Adatok lek rdez se (Java DOM)

El sz r a megfelel  csomagok import l sa, f jlok megnyit sa, l trehoz sa, majd a builderek fel p t se, a gy k relem kezel se. Ezt k vet en a gyermekelemek hozz ad sa a gy k relemhez, majd minden lek rdez shez 1-1 ciklus l trehoz sa, majd a megfelel  elemek lek rdez se ut n form zott ki rat sa a konzolra.

```
public class DOMQueryJR9KY7 {
```



```

    public static void main(String[] args) throws IOException, ParserConfigurationException,
    SAXException {

        try {

            // XML file kiválasztása

            File inputFile = new File("XMLJR9KY7.xml");

            // Dokumentum builderek létrehozása

            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

            // Dokumentumok beállítása

            Document doc = dBuilder.parse(inputFile);

            doc.getDocumentElement().normalize();

            // Gyökérelem meghatározása, konzolon feltüntetése

            System.out.println("-----");

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

            System.out.println(doc.getDocumentElement().getNodeName());

            // 1. Lekérdezés fő részének meghatározása

            NodeList nList1 = doc.getElementsByTagName("pincer");

            System.out.println("-----");

            System.out.println("\n-----");

            System.out.println("1. Pincerek adatainak lekerdezese:");

            System.out.println("-----");

            // Iterálás az elemeken és adott elemek kiírása a konzolra

```

```

for (int temp = 0; temp < nList1.getLength(); temp++) {

    Node nNode = nList1.item(temp);

    System.out.println("\nAktualis elem :");

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element element = (Element) nNode;

        System.out.print("Pincer ID: ");

        System.out.println(element.getAttribute("pID"));

        NodeList pincerNameList =
element.getElementsByTagName("nev");

        NodeList pincerFizList =
element.getElementsByTagName("fizetes");

        NodeList pincerKorList = element.getElementsByTagName("kor");

        for (int count = 0; count < pincerNameList.getLength(); count++) {

            Node node1 = pincerNameList.item(count);

            Node node2 = pincerFizList.item(count);

            Node node3 = pincerKorList.item(count);

            Element pincerNev = (Element) node1;

            System.out.print("Pincer neve: ");

            System.out.println(pincerNev.getTextContent());

            Element pincerFiz = (Element) node2;

            System.out.print("Pincer fizetese: ");

            System.out.println(pincerFiz.getTextContent());

            Element pincerKor = (Element) node3;

            System.out.print("Pincer kora: ");

            System.out.println(pincerKor.getTextContent());

```

```

    }

}

}

// 2. Lekérdezés fő részének meghatározása

NodeList nList2 = doc.getElementsByTagName("kAlk");

System.out.println("\n-----");

System.out.println("2. Konyhai alkalmazottak neve es telefonszamanak
lekerdezese:");

System.out.println("-----");

// Iterálás az elemeken és adott elemek kiírása a konzolra
for (int temp = 0; temp < nList2.getLength(); temp++) {

    Node nNode = nList2.item(temp);

    System.out.println("\nAktualis elem :");

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element element = (Element) nNode;

        System.out.print("Konyhai alkalmazott ID: ");

        System.out.println(element.getAttribute("kID"));

        NodeList kAlkNameList = element.getElementsByTagName("nev");

        NodeList kAlkTelszamList =
element.getElementsByTagName("telefonszam");

        for (int count = 0; count < kAlkNameList.getLength(); count++) {

            Node node1 = kAlkNameList.item(count);

            Node node2 = kAlkTelszamList.item(count);

            Element kAlkNev = (Element) node1;

            System.out.print("Konyhai alkalmazott neve: ");

```

```

        System.out.println(kAlkNev.getTextContent());

        Element kAlkTelszam = (Element) node2;

        System.out.print("Konyhai alkalmazott telefonszama: ");

        System.out.println(kAlkTelszam.getTextContent());

    }

}

}

// 3. Lekérdezés fő részének meghatározása
NodeList nList3 = doc.getElementsByTagName("rendeles");

System.out.println("\n-----");

System.out.println("3. Rendelesek vegosszege:");

System.out.println("-----");

// Iterálás az elemeken és adott elemek kiírása a konzolra
for (int temp = 0; temp < nList3.getLength(); temp++) {

    Node nNode = nList3.item(temp);

    System.out.println("\nAktualis elem :");

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element element = (Element) nNode;

        System.out.print("Rendeles ID: ");

        System.out.println(element.getAttribute("rID"));

        NodeList etelArList = element.getElementsByTagName("etelAr");

        NodeList italArList = element.getElementsByTagName("italAr");

        for (int count = 0; count < etelArList.getLength(); count++) {

            Node node1 = etelArList.item(count);

```

```

        Node node2 = italArList.item(count);

        Element rEtelAr = (Element) node1;

        Element rItalAr = (Element) node2;

        System.out.print("Rendeles vegosszege: ");

        System.out.println(Integer.parseInt(rEtelAr.getTextContent())

                                                                    +
        Integer.parseInt(rItalAr.getTextContent()));

    }

}

// 4. Lekérdezés fő részének meghatározása

NodeList nList4 = doc.getElementsByTagName("elkeszit");

System.out.println("\n-----");

System.out.println("4. Rendelesek azonositoja, ahol fozes szukseges:");

System.out.println("-----");

// Iterálás az elemeken és adott elemek kiírása a konzolra

for (int temp = 0; temp < nList4.getLength(); temp++) {

    Node nNode = nList4.item(temp);

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element element = (Element) nNode;

        NodeList fozesList = element.getElementsByTagName("fozes");

        for (int count = 0; count < fozesList.getLength(); count++) {

            Node node = fozesList.item(count);

```

```

        Element fozes = (Element) node;

        if (fozes.getTextContent().equalsIgnoreCase("igen")) {

            System.out.println("\nID-k: " +
element.getAttribute("rID"));

        }

    }

}

// 5. Lekérdezés fő részének meghatározása

NodeList nList5 = doc.getElementsByTagName("pincer");

System.out.println("\n-----");

System.out.println("5. Pincerek, akik 24 evnel idosebbek:");

System.out.println("-----");

// Iterálás az elemeken és adott elemek kiírása a konzolra

for (int temp = 0; temp < nList5.getLength(); temp++) {

    Node nNode = nList5.item(temp);

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element element = (Element) nNode;

        NodeList pincerNameList =
element.getElementsByTagName("nev");

        NodeList pincerKorList = element.getElementsByTagName("kor");

        for (int count = 0; count < pincerNameList.getLength(); count++) {

            Node node1 = pincerNameList.item(count);

            Node node2 = pincerKorList.item(count);

```

```

        Element pincerNev = (Element) node1;

        Element pincerKor = (Element) node2;

        if (Integer.parseInt(pincerKor.getTextContent()) > 24) {

            System.out.println("\nPincer neve: " +
pincerNev.getTextContent());

        }

    }

}

} catch (IOException e) {

    e.printStackTrace();

} catch (ParserConfigurationException e) {

    e.printStackTrace();

} catch (SAXException e) {

    e.printStackTrace();

}

} // end main

} // end class

```

Futási eredmények:

Gyokerelem: etterem

1. Pincerek adatainak lekerdezese:

Aktualis elem :

Pincer ID: 1

Pincer neve: Kiss László

Pincer fizetese: 230000

Pincer kora: 23

Aktualis elem :

Pincer ID: 2

Pincer neve: Szabó Norbert

Pincer fizetese: 280000

Pincer kora: 28

Aktualis elem :

Pincer ID: 3

Pincer neve: Almási Péter

Pincer fizetese: 280000

Pincer kora: 25

2. Konyhai alkalmazottak neve es telefonszamanak lekerdezese:

Aktualis elem :

Konyhai alkalmazott ID: 1

Konyhai alkalmazott neve: Nagy Sándor

Konyhai alkalmazott telefonszama: 06204793472

Aktualis elem :

Konyhai alkalmazott ID: 2

Konyhai alkalmazott neve: Tóth József

Konyhai alkalmazott telefonszama: 06207355921

Aktualis elem :

Konyhai alkalmazott ID: 3

Konyhai alkalmazott neve: Szilva Béla

Konyhai alkalmazott telefonszama: 06206395629

3. Rendelesek vegosszege:

Aktualis elem :

Rendeles ID: 1

Rendeles vegosszege: 8500

Aktualis elem :

Rendeles ID: 2

Rendeles vegosszege: 7000

Aktualis elem :

Rendeles ID: 3

Rendeles vegosszege: 24500

4. Rendelesek azonositoja, ahol fozes szukseges:

ID-k: 1

5. Pincerek, akik 24 evnel idosebbek:

Pincer neve: Szabó Norbert

Pincer neve: Almási Péter