

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat:

Vendéglátás modellezése

Készítette: **Kollár Máté**

Neptunkód: **J004TM**

## A feladat leírása:

A feladatom tárgyaként az vendéglátás általános felépítését vettem alapul, amelyben főként az éttermek és a hozzájuk kapcsolódó személyek szerepét emeltem ki.

A modell középpontjában áll az étterem, amelynek a általam kiemelt tulajdonságai a neve, a címe (felbontva a városra, az utcára és a házszámra), valamint az étterem csillagainak a száma, amely a minőséget garantálja.

Az első személy, akit kiemeltem, a főszakács. Ez az ember vezeti az adott éttermek konyháját, így belőle csak egy lehet egy étteremben, ezért őt és az éttermet egy 1:1 kapcsolattal kötöttem össze. Neki a kiemelt tulajdonságai a neve, az életkora és a végzettsége/végzettségei.

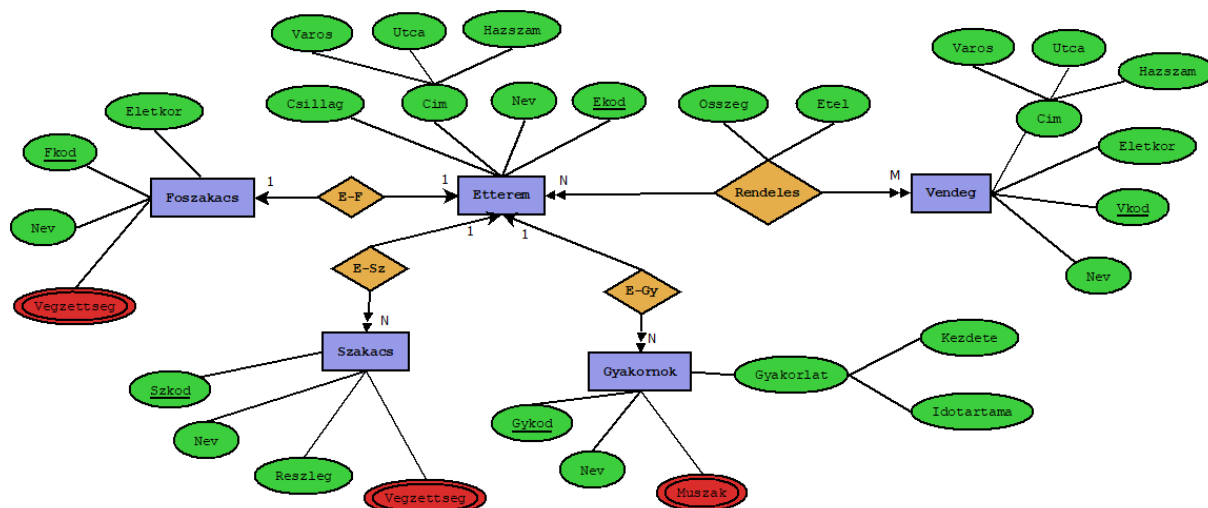
A következő személy a szakács. Ők dolgoznak az éttermek konyháiban, a főszakács irányítása alatt különféle részlegeken. Belőlük több is van egy konyhában, így őket 1:N kapcsolattal kötöttem össze az étteremmel. A kiemelt tulajdonságai a neve, a részleg, ahol dolgozik és a végzettsége/végzettségei.

Ezután következik a gyakornok. Ők, mint betanulók vannak az éttermek konyháiban, segédkeznek az ételek elkészítésének alapjainál, tanulják a szükséges fortélyokat, hogy szépen lassan szakácscsá válhassanak. Az ő kiemelt tulajdonságai a neve, a műszakja/műszakjai és a gyakorlata (felbontva a gyakorlati idejének kezdetére és időtartamára). Belőlük is több lehet egy étteremben, így itt is egy 1:N kapcsolat van.

Az utolsó személy a modellemben a vendég. Ők több étterembe is elmehetnek és többen is lehetnek egy étterembe, így közöttük N:M kapcsolat van, amely saját tulajdonságokkal is rendelkezik, mégpedig a vendég rendelésének összege és a rendelt étel neve. A vendégnek a kiemelt tulajdonságai a neve, az életkora és a címe (felbontva a városra, az utcára és a házszámra).

## 1. feladat

### 1a) Az adatbázis ER modell:



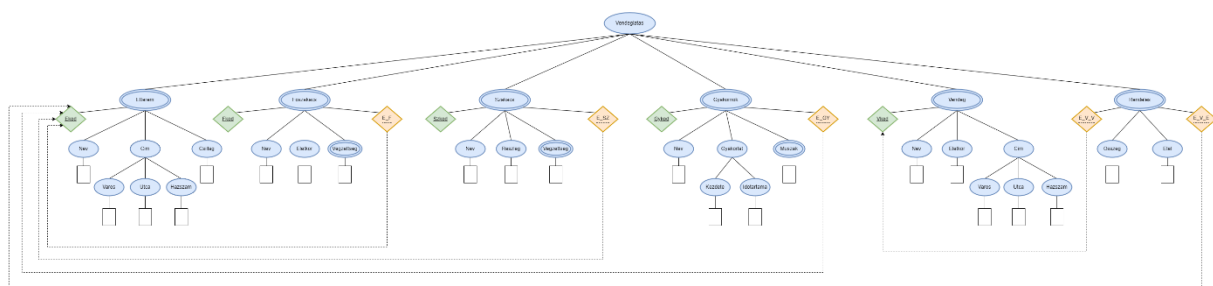
### 1b) Az adatbázis konvertálása XDM modellre:

A konvertálás következtében az egyedeimből elemek lesznek, amelyeknek az azonosítójuk attribútumává válnak, amelyeken keresztül megvalósulnak az adott elemek közötti kapcsolatok.

Az étterem és vendég közti N:M kapcsolatból létrejön egy rendelés nevű kapcsoló elem, amelynek a két attribútuma mutat a két elem attribútumára, valamint ez tartalmazza gyerekelemként a kapcsolat tulajdonságait.

A 1:N kapcsolatok úgy valósulnak meg, hogy az étteremben dolgozó személyek (akik az N végén vannak a kapcsolatnak) tartalmazzák a saját azonosítójukon kívül egy plusz attribútumot, amely az étterem azonosítójára mutat.

Az 1:1 kapcsolat ugyanígy valósul meg, a különbség csak a feladat későbbi részében fog meglátszódni.



## 1c) Az XDM modell alapján XML dokumentum készítése:

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

```
<vendeglatas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaJ004TM.xsd">
```

```
<!-- Ettermek -->
<etterem ekod="e1">
  <nev>Tapassio</nev>
  <cim>
    <varos>Budapest</varos>
    <utca>Herczegprimás</utca>
    <hazszam>13</hazszam>
  </cim>
  <csillag>5</csillag>
</etterem>

<etterem ekod="e2">
  <nev>Arany Kaviár</nev>
  <cim>
    <varos>Budapest</varos>
    <utca>Ostrom</utca>
    <hazszam>19</hazszam>
  </cim>
  <csillag>5</csillag>
</etterem>

<!-- Foszakacsok -->
<foszakacs fkod="f1" e_f="e1">
  <nev>Havas Henrik</nev>
  <eletkor>45</eletkor>
  <vegzettseg>Szakközépiskola</vegzettseg>
  <vegzettseg>Institut Paul Bocuse</vegzettseg>
</foszakacs>

<foszakacs fkod="f2" e_f="e2">
  <nev>Nagy Dániel</nev>
  <eletkor>30</eletkor>
  <vegzettseg>Le Cordon Bleu</vegzettseg>
</foszakacs>

<!-- Szakacsok -->
<szakacs szkod="sz1" e_sz="e1">
  <nev>Ötlet Elek</nev>
  <reszleg>Saucier</reszleg>
  <vegzettseg>Szakközépiskola</vegzettseg>
  <vegzettseg>Le Cordon Bleu</vegzettseg>
  <vegzettseg>Institut Paul Bocuse</vegzettseg>
</szakacs>

<szakacs szkod="sz2" e_sz="e2">
  <nev>Kocsis Tibor</nev>
  <reszleg>Entremetier</reszleg>
  <vegzettseg>Szakközépiskola</vegzettseg>
  <vegzettseg>Le Cordon Bleu</vegzettseg>
</szakacs>

<szakacs szkod="sz3" e_sz="e2">
  <nev>Tálas Levente</nev>
  <reszleg>Gardemanger</reszleg>
  <vegzettseg>Szakközépiskola</vegzettseg>
</szakacs>

<szakacs szkod="sz4" e_sz="e1">
  <nev>Szűcs Judit</nev>
  <reszleg>Entremetier</reszleg>
  <vegzettseg>Le Cordon Bleu</vegzettseg>
  <vegzettseg>Institut Paul Bocuse</vegzettseg>
</szakacs>
```

```
<!-- Gyakorlatok -->
<gyakorlat gykod="gy1" e_gy="e1">
  <nev>Bordás Dávid</nev>
  <gyakorlat>
    <kezdet>2021.08.20.</kezdet>
    <időtartam>Két hónap</időtartam>
  </gyakorlat>
  <muszak>Délelőtt</muszak>
  <muszak>Este</muszak>
</gyakorlat>

<gyakorlat gykod="gy2" e_gy="e2">
  <nev>Szilágyi Marcell</nev>
  <gyakorlat>
    <kezdet>2021.06.01.</kezdet>
    <időtartam>Négy hónap</időtartam>
  </gyakorlat>
  <muszak>Délután</muszak>
</gyakorlat>

<gyakorlat gykod="gy3" e_gy="e1">
  <nev>Szentpétery Mózes</nev>
  <gyakorlat>
    <kezdet>2021.01.01.</kezdet>
    <időtartam>Tíz hónap</időtartam>
  </gyakorlat>
  <muszak>Délelőtt</muszak>
  <muszak>Délután</muszak>
</gyakorlat>

<!-- Vendégek -->
<vendeg vkod="v1">
  <nev>Kollár Máté</nev>
  <életkor>21</életkor>
  <cím>
    <varos>Annót</varos>
    <utca>Pázsit</utca>
    <hazszam>74</hazszam>
  </cím>
</vendeg>

<vendeg vkod="v2">
  <nev>Szentpétery Áron</nev>
  <életkor>15</életkor>
  <cím>
    <varos>Miskolc</varos>
    <utca>Örs</utca>
    <hazszam>24</hazszam>
  </cím>
</vendeg>

<vendeg vkod="v3">
  <nev>Elon Musk</nev>
  <életkor>26</életkor>
  <cím>
    <varos>Miskolc</varos>
    <utca>Szilvás</utca>
    <hazszam>4</hazszam>
  </cím>
</vendeg>

<!-- Rendelesek -->
<rendeles e_v_e="e1" e_v_v="v1">
  <összeg>9000</összeg>
  <étel>Steak</étel>
</rendeles>

<rendeles e_v_e="e2" e_v_v="v2">
  <összeg>8000</összeg>
  <étel>Kaviár</étel>
</rendeles>

<rendeles e_v_e="e1" e_v_v="v3">
  <összeg>20000</összeg>
  <étel>Zsíros kenyér</étel>
</rendeles>

<rendeles e_v_e="e2" e_v_v="v3">
  <összeg>300</összeg>
  <étel>Mackó sajt</étel>
</rendeles>

</vendéglista>
```

## 1d) Az XML dokumentum alapján XMLSchema készítése:

```
<?xml version="1.0" encoding="UTF-8"?>|
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://www.example.org/XMLJ004TM"
elementFormDefault="qualified">
```

```
<!-- Felepités -->
<xs:element name="vendeglatas">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="etterem" type="etterem_tipus" maxOccurs="unbounded"/>
      <xs:element name="foszakacs" type="foszakacs_tipus" maxOccurs="unbounded"/>
      <xs:element name="szakacs" type="szakacs_tipus" maxOccurs="unbounded"/>
      <xs:element name="gyakornok" type="gyakornok_tipus" maxOccurs="unbounded"/>
      <xs:element name="vendeg" type="vendeg_tipus" maxOccurs="unbounded"/>
      <xs:element name="rendeles" type="rendeles_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

```
<!-- Kulcsok -->
<xs:key name="etterem_kulcs">
  <xs:selector xpath="etterem" />
  <xs:field xpath="@ekod" />
</xs:key>
```

```
<xs:key name="foszakacs_kulcs">
  <xs:selector xpath="foszakacs" />
  <xs:field xpath="@fkod" />
</xs:key>
```

```
<xs:key name="szakacs_kulcs">
  <xs:selector xpath="szakacs" />
  <xs:field xpath="@szkod" />
</xs:key>
```

```
<xs:key name="gyakornok_kulcs">
  <xs:selector xpath="gyakornok" />
  <xs:field xpath="@gykod" />
</xs:key>
```

```
<xs:key name="vendeg_kulcs">
  <xs:selector xpath="vendeg" />
  <xs:field xpath="@vkod" />
</xs:key>
```

```
<!-- Idegen kulcsok -->
<xs:keyref refer="etterem_kulcs" name="etterem_idegen_kulcs">
  <xs:selector xpath="rendeles" />
  <xs:field xpath="@e_v_e" />
</xs:keyref>
<xs:keyref refer="vendeg_kulcs" name="vendeg_idegen_kulcs">
  <xs:selector xpath="rendeles" />
  <xs:field xpath="@e_v_v" />
</xs:keyref>

<xs:keyref refer="etterem_kulcs" name="etterem_foszakacs_idegen_kulcs">
  <xs:selector xpath="foszakacs" />
  <xs:field xpath="@e_f" />
</xs:keyref>

<xs:keyref refer="etterem_kulcs" name="etterem_szakacs_idegen_kulcs">
  <xs:selector xpath="szakacs" />
  <xs:field xpath="@e_sz" />
</xs:keyref>

<xs:keyref refer="etterem_kulcs" name="etterem_gyakornok_idegen_kulcs">
  <xs:selector xpath="gyakornok" />
  <xs:field xpath="@e_gy" />
</xs:keyref>
```

```
<!-- Az 1:1 kapcsolat megvalósításához -->
<xs:unique name="unique_foszakacs">
  <xs:selector xpath="foszakacs"/>
  <xs:field xpath="@e_f"/>
</xs:unique>
```

```
</xs:element>
```

```

<!-- Tipusok -->
<xs:complexType name="etterem_tipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="cim" type="cim_tipus"/>
    <xs:element name="csillag" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="ekod" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="cim_tipus">
  <xs:sequence>
    <xs:element name="varos" type="xs:string"/>
    <xs:element name="utca" type="xs:string"/>
    <xs:element name="hazszam" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="foszakacs_tipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="eletkor" type="xs:integer"/>
    <xs:element name="vegzettseg" type="xs:string" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="fkod" type="xs:string" use="required"/>
  <xs:attribute name="e_f" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="szakacs_tipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="reszleg" type="xs:string"/>
    <xs:element name="vegzettseg" type="xs:string" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="szkod" type="xs:string" use="required"/>
  <xs:attribute name="e_sz" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="gyakornok_tipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="gyakorlat" type="gyakorlat_tipus"/>
    <xs:element name="muszak" type="xs:string" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="gykod" type="xs:string" use="required"/>
  <xs:attribute name="e_gy" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="gyakorlat_tipus">
  <xs:sequence>
    <xs:element name="kezdetek" type="xs:string"/>
    <xs:element name="idotartama" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="vendeg_tipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="eletkor" type="xs:integer"/>
    <xs:element name="cim" type="cim_tipus"/>
  </xs:sequence>
  <xs:attribute name="vkod" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="rendeles_tipus">
  <xs:sequence>
    <xs:element name="osszeg" type="xs:integer"/>
    <xs:element name="etel" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="e_v_e" type="xs:string" use="required"/>
  <xs:attribute name="e_v_v" type="xs:string" use="required"/>
</xs:complexType>

</xs:schema>

```

## 2. feladat

### 2a) DOM adatolvasás:

```
package hu.domparsing.j004tm;

import java.io.File;

public class DomReadJ004TM {

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

        //Ebből a fájlból olvas
        File xmlFile = new File("src/XMLJ004TM.xml");

        //A DocumentBuilderFactory-ből megkapjuk a DocumentBuildert
        //A DocumentBuilder tartalmazza az API-t a DOM-dokumentum példányok XML-dokumentumból való beszerzéséhez
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        //A parse() metódus elemzi az XML fájlt és lekérlem a documentbe
        Document doc = dBuilder.parse(xmlFile);

        //A dokumentum normalizálása segít a helyes eredmények elérésében
        doc.getDocumentElement().normalize();

        //Kiíratjuk a dokumentum gyökerelemét
        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());

        //A getElementsByTagName() metódus segítségével megkapjuk az etterem elem NodeListjét
        NodeList nList = doc.getElementsByTagName("etterem");

        //Végigmegyünk rajta egy for ciklussal
        for(int i = 0; i < nList.getLength(); i++) {

            //Lekérjük a lista aktuális elemét
            Node nNode = nList.item(i);

            System.out.println("\nCurrent Element: " + nNode.getNodeName());

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

                //Elementté konvertáljuk az aktuális elemet
                Element elem = (Element) nNode;

                //Lekérjük az aktuális elem attribútumának tartalmát
                String id = elem.getAttribute("ekod");

                //Lekérjük az aktuális elem gyerekelemeinek tartalmát
                Node node1 = elem.getElementsByTagName("nev").item(0);
                String name = node1.getTextContent();

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

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

                Node node4 = elem.getElementsByTagName("hazszam").item(0);
                String number = node4.getTextContent();

                Node node5 = elem.getElementsByTagName("csillag").item(0);
                String stars = node5.getTextContent();

                //A cím gyerekelemeinek összerakása egy stringbe
                String adr = city + ", " + street + " utca " + number + ".";

                //Formázva kiíratjuk a lekért információkat az adott elemről
                System.out.println("Étterem ID: " + id);
                System.out.println("Név: " + name);
                System.out.println("Cím: " + adr);
                System.out.println("Csillag: " + stars);

            }

        }

    }

}
```



```

nList = doc.getElementsByTagName("foszakacs");

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

    Node nNode = nList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        String id = elem.getAttribute("fkod");
        String eid = elem.getAttribute("e_f");

        String work = "Ez a főszakács a(z) " + eid + " azonosítójú étteremben dolgozik." ;

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

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

        Node node3;
        String edu = "";

        //Több végzettsége is lehet egy szakácsnak, így ezt a formázott kiíratást csináltam,
        //hogyan helyesen írassa ki a végzettségek számától függetlenül
        for(int j=0;j<elem.getElementsByTagName("vegzettseg").getLength();j++) {
            node3 = elem.getElementsByTagName("vegzettseg").item(j);
            if(j == elem.getElementsByTagName("vegzettseg").getLength()-1) {
                edu += node3.getTextContent();
            }else {
                edu += node3.getTextContent() + ", ";
            }
        }

        System.out.println("Főszakács ID: " + id);
        System.out.println("Név: " + name);
        System.out.println("Kor: " + age);
        System.out.println("Végzettségek: " + edu);
        System.out.println(work);
    }
}

nList = doc.getElementsByTagName("szakacs");

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

    Node nNode = nList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        String id = elem.getAttribute("szkod");
        String eid = elem.getAttribute("e_sz");

        String work = "Ez a szakács a(z) " + eid + " azonosítójú étteremben dolgozik." ;

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

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

        Node node3;
        String edu = "";
    }
}

```

```

        for(int j=0;j<elem.getElementsByTagName("vegzettseg").getLength();j++) {
            node3 = elem.getElementsByTagName("vegzettseg").item(j);
            if(j == elem.getElementsByTagName("vegzettseg").getLength()-1) {
                edu += node3.getTextContent();
            }else {
                edu += node3.getTextContent() + ", ";
            }
        }

        System.out.println("Szakács ID: " + id);
        System.out.println("Név: " + name);
        System.out.println("Részleg: " + department);
        System.out.println("Végzettségek: " + edu);
        System.out.println(work);
    }
}

nList = doc.getElementsByTagName("gyakornok");

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

    Node nNode = nList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        String id = elem.getAttribute("gykod");
        String eid = elem.getAttribute("e_gy");

        String work = "Ez a gyakornok a(z) " + eid + " azonosítójú étteremben dolgozik." ;

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

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

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

        String practical = "kezdete: " + start + ", időtartama: " + duration;

        Node node4;
        String shift = "";

        //Több műszakban is dolgozhat egy gyakornok, így ezt a formázott kiíratást csináltam,
        //hogyan helyesen írassa ki a műszakok számától függetlenül
        for(int j=0;j<elem.getElementsByTagName("muszak").getLength();j++) {
            node4 = elem.getElementsByTagName("muszak").item(j);
            if(j == elem.getElementsByTagName("muszak").getLength()-1) {
                shift += node4.getTextContent();
            }else {
                shift += node4.getTextContent() + ", ";
            }
        }

        System.out.println("Gyakornok ID: " + id);
        System.out.println("Név: " + name);
        System.out.println("Gyakorlat " + practical);
        System.out.println("Műszak: " + shift);
        System.out.println(work);
    }
}

nList = doc.getElementsByTagName("vendeg");

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

    Node nNode = nList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

```

```

        if(nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) nNode;

            String id = elem.getAttribute("vkod");

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

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

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

            Node node4 = elem.getElementsByTagName("utca").item(0);
            String street = node4.getTextContent();

            Node node5 = elem.getElementsByTagName("hazszam").item(0);
            String number = node5.getTextContent();

            String adr = city + " " + street + " utca " + number + ".";

            System.out.println("Vendég ID: " + id);
            System.out.println("Név: " + name);
            System.out.println("Kor: " + age);
            System.out.println("Cím: " + adr);
        }
    }

    nList = doc.getElementsByTagName("rendeles");

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

        Node nNode = nList.item(i);

        System.out.println("\nCurrent Element: " + nNode.getNodeName());

        if(nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) nNode;

            String eid = elem.getAttribute("e_v_e");
            String vid = elem.getAttribute("e_v_v");

            String dinner = "A(z) " + eid + " azonosítóval rendelkező étteremből rendelt a(z) " + vid + " azonosítójú vendég.";

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

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

            System.out.println(dinner);
            System.out.println("Összeg: " + price);
            System.out.println("étel: " + food);
        }
    }
}
}
}

```

## 2b) DOM adatmódosítás:

```
package hu.domparse.j004tm;

import java.io.File;

public class DOMModifyJ004TM {

    public static void main(String[] args){

        try {
            File xmlFile = new File("src/XMLJ004TM.xml");

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

            Document doc = dBuilder.parse(xmlFile);

            doc.getDocumentElement().normalize();

            //Végigmegy a vendég elemeken for ciklussal és a feltételben megadott
            //nevű és értékű adatokat átírja a kért tartalommal
            NodeList nList = doc.getElementsByTagName("vendeg");

            for(int i = 0; i < nList.getLength(); i++) {
                Node vendeg = doc.getElementsByTagName("vendeg").item(i);

                NodeList list = vendeg.getChildNodes();

                for (int temp = 0; temp < list.getLength(); temp++) {
                    Node node = list.item(temp);
                    if (node.getNodeType() == Node.ELEMENT_NODE) {
                        Element eElement = (Element) node;
                        if ("eletkor".equals(eElement.getNodeName())) {
                            if("21".equals(eElement.getTextContent())) {
                                eElement.setTextContent("22");
                            }
                        }
                        if ("nev".equals(eElement.getNodeName())) {
                            if("Elon Musk".equals(eElement.getTextContent())) {
                                eElement.setTextContent("Mr. Elon Musk");
                            }
                        }
                    }
                }
            }

            // Átírja az adott gyakornok attribútumát a megadott értékre
            Node gyakornok = doc.getElementsByTagName("gyakornok").item(2);

            NamedNodeMap attr = gyakornok.getAttributes();
            Node nodeAttr = attr.getNamedItem("e_gy");
            nodeAttr.setTextContent("e2");

            //Kitörli az összes rendelés elemet az xml fájlból
            Node vendeglatas = doc.getFirstChild();
            NodeList childNodes = vendeglatas.getChildNodes();

            for(int count = 0; count < childNodes.getLength(); count++) {
                Node node = childNodes.item(count);

                if("rendeles".equals(node.getNodeName()))
                    vendeglatas.removeChild(node);
            }
        }
    }
}
```

```
//Kiiratja a módosított xml fájlt a konzolra
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMSource source = new DOMSource(doc);
System.out.println("-----Modified File-----");
StreamResult consoleResult = new StreamResult(System.out);
transformer.transform(source, consoleResult);

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

### 3c) DOM adatlekérdezés:

```
package hu.domparsing.j004tm;

import java.io.File;

public class DOMQueryJ004TM {

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

        File xmlFile = new File("src/XMLJ004TM.xml");

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

        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        System.out.println("Root element: " + doc.getDocumentElement().getNodeName() + "\n");

        //Kiíratom azokat a szakácsokat, akiknek a végzettségeik között van szakközépiskola
        System.out.println("Azok a szakácsok, akiknek a végzettségeik között van szakközépiskola:\n");

        NodeList nodeList = doc.getElementsByTagName("szakacs");

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

            Node nNode = nodeList.item(i);

            if(nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) nNode;

                Node node3;

                //Végignézem a végzettségeit, hogy van-e közt szakközépiskola
                for(int j=0;j<elem.getElementsByTagName("vegzettseg").getLength();j++) {
                    node3 = elem.getElementsByTagName("vegzettseg").item(j);
                    String edu1 = node3.getTextContent();
                    //Ha igen, akkor kiíratom az adatait
                    if("Szakközépiskola".equals(edu1)) {
                        String id = elem.getAttribute("szkod");
                        String eid = elem.getAttribute("e_sz");

                        String work = "Ez a szakács a(z) " + eid + " azonosítójú étteremben dolgozik." ;

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

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

                        String edu2 = "";

                        for(int k=0;k<elem.getElementsByTagName("vegzettseg").getLength();k++) {
                            node3 = elem.getElementsByTagName("vegzettseg").item(k);
                            if(k == elem.getElementsByTagName("vegzettseg").getLength()-1) {
                                edu2 += node3.getTextContent();
                            }else {
                                edu2 += node3.getTextContent() + ", ";
                            }
                        }

                        System.out.println("Szakács ID: " + id);
                        System.out.println("Név: " + name);
                        System.out.println("Részleg: " + department);
                        System.out.println("Végzettségek: " + edu2);
                        System.out.println(work + "\n");
                    }
                }
            }
        }
    }
}
```

```

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

//Kiíratom azokat az éttermeket, amik öt csillagosak
System.out.println("Azok az éttermek, amik öt csillagosak:\n");

nodeList = doc.getElementsByTagName("etterm");

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

    Node nNode = nodeList.item(i);

    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        Node node5 = elem.getElementsByTagName("csillag").item(0);
        String stars = node5.getTextContent();

        if("5".equals(stars)) {
            String id = elem.getAttribute("ekod");

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

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

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

            Node node4 = elem.getElementsByTagName("hazszam").item(0);
            String number = node4.getTextContent();

            String adr = city + ", " + street + " utca " + number + ".";

            System.out.println("Étterm ID: " + id);
            System.out.println("Név: " + name);
            System.out.println("Cím: " + adr);
            System.out.println("Csillag: " + stars + "\n");
        }
    }
}

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

//Kiíratom azokat a gyakor nokokat, akik be vannak osztva délutánra
System.out.println("Azok a gyakor nokok, akik be vannak osztva délutánra:\n");

nodeList = doc.getElementsByTagName("gyakor nok");

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

    Node nNode = nodeList.item(i);

    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        Node node4;

        for(int j=0;j<elem.getElementsByTagName("muszak").getLength();j++) {
            node4 = elem.getElementsByTagName("muszak").item(j);
            String shift1 = node4.getTextContent();
            if("Délután".equals(shift1)) {
                String id = elem.getAttribute("gykod");
                String eid = elem.getAttribute("e_gy");

                String work = "Ez a gyakor nok a(z) " + eid + " azonosítójú étteremben dolgozik. ";

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

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

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

                String practical = "kezdet: " + start + ", időtartama: " + duration;
            }
        }
    }
}

```

```
String shift2 = "";

for(int k=0;k<elem.getElementsByTagName("muszak").getLength();k++) {
    node4 = elem.getElementsByTagName("muszak").item(k);
    if(k == elem.getElementsByTagName("muszak").getLength()-1) {
        shift2 += node4.getTextContent();
    }else {
        shift2 += node4.getTextContent() + ", ";
    }
}

System.out.println("Gyakornok ID: " + id);
System.out.println("Név: " + name);
System.out.println("Gyakorlat " + practical);
System.out.println("Műszak: " + shift2);
System.out.println(work + "\n");
}
}
}
}
}
```