

# Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Gyorsétterem felépítése

Készítette: **Simon Dániel**

Neptunkód: **XUE9MH**

Gyakorlati Csoport: **K01** (Szerda 10-12)

**Feladat leírása:** Feladatomként egy Gyros-os gyorsétterem modellezését választottam XML környezetben. Segédprogramként az Eclipse fejlesztőkörnyezetet használtam valamint a „dia” online model szerkesztő programot.

A feladatom entitásainak és azok attribútumainak listája:

1. Gyorsetterem
  - 1.1 ID
  - 1.2 Nev
  - 1.3 Email
  - 1.4 Telefonszam
  - 1.5 Nyitvatartas *(összetett)*
    - 1.5.1 Hetkoznap
    - 1.5.2 Unnepek/Hetvege
- 2 Gyros
  - 2.1 ID
  - 2.2 GyorsetteremFK
  - 2.3 Suly
  - 2.4 Toltelek *(többértékű)*
  - 2.5 Ear *(származtatott)*
3. Vásárló
  - 3.1 ID
  - 3.2 Nev
  - 3.3 Telefonszam
  - 3.4 Cim *(összetett)*
    - 3.4.1 Varos
    - 3.4.2 Utca
    - 3.4.3 Hazszam
4. Bankkártya
  - 4.1 ID
  - 4.2 VasarloFK
  - 4.3 Kartyaszám
  - 4.4 CCV
  - 4.5 Tipus
  - 4.6 Lejarati\_datum
5. Beszállító
  - 5.1 BeszallitoID
  - 5.2 Nev

### 5.3 Email

### 5.4 Telefonszam

### 5.5 Cim (összetett)

#### 5.5.1 Varos

#### 5.5.2 Utca

#### 5.5.3 Hazszam

## 3. Tulaj

### 3.1 ID

### 3.2 GyorsetteremFK

### 3.3 Nev

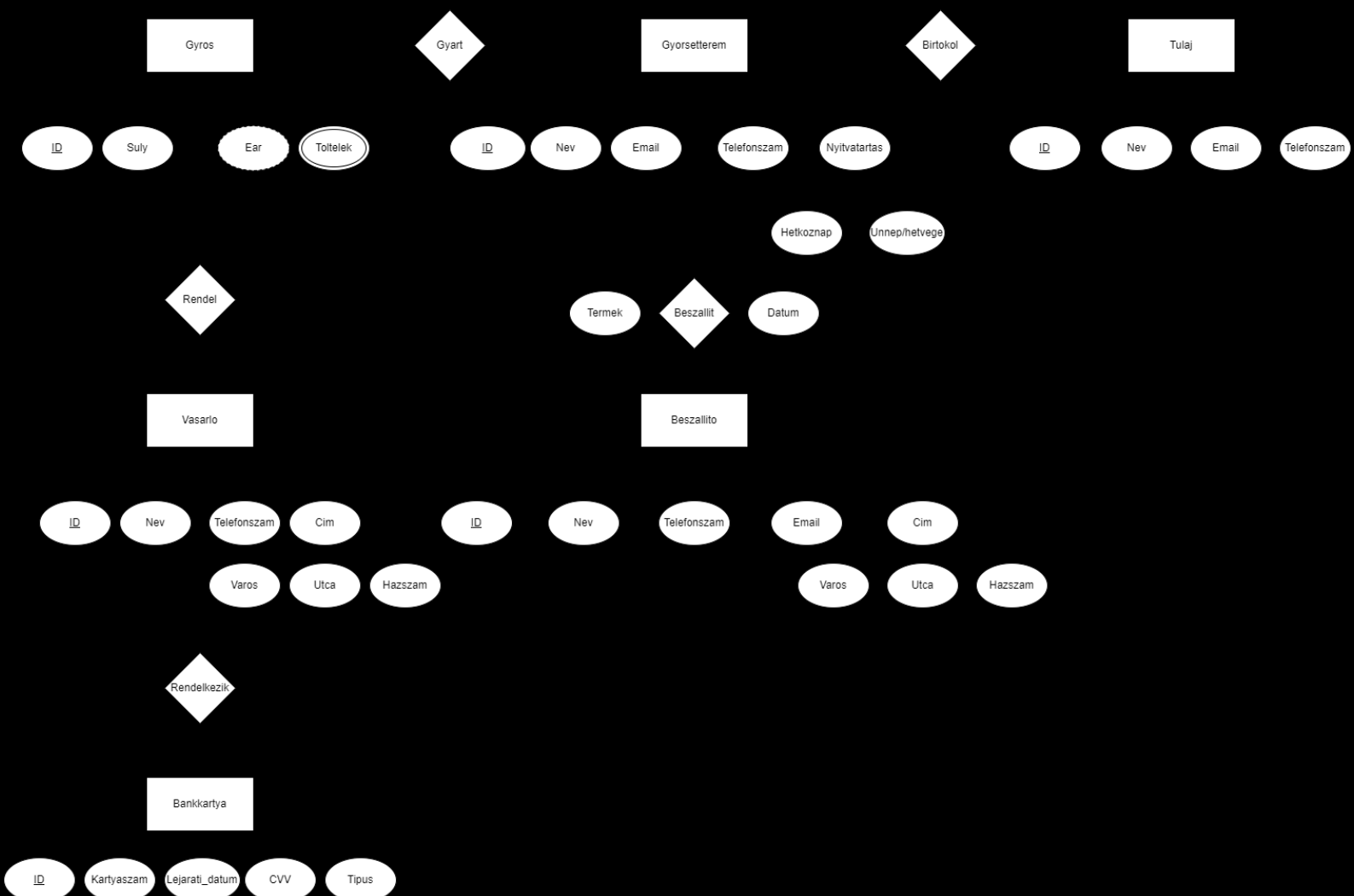
### 3.4 Email

### 3.5 Telefonszam

Több féle kapcsolat is található a modellben például: Gyorsetterem : Gyros (1:N),  
Gyros-os : Beszállító (N:M),

## 1. Feladat:

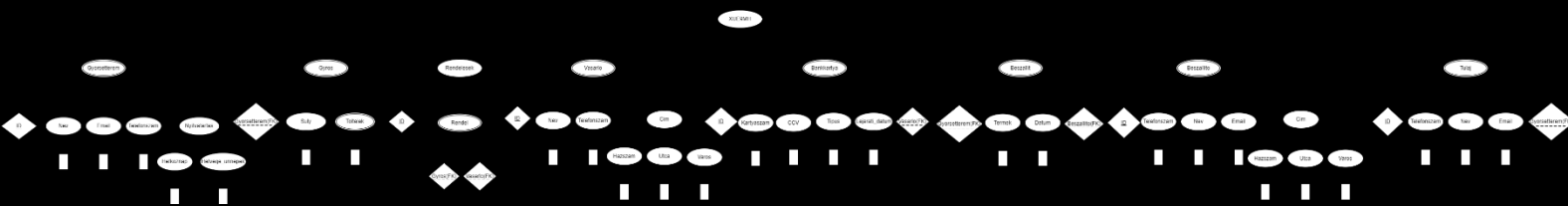
### ER modell:



Leírás:

A gyorséttermünk **Gyorsetterem** típusú, nevét gyerek elemként tárolja. Egy Tulaj több Gyorsettermet is birtokolhat viszont egy Gyorsetterem csak egy Tulajdonoshoz tartozik. Egy Gyorsetteremnek több Beszállítója is lehet valamint egy Beszállító több Gyorsetteremnek is szállíthat, ennek a kapcsolatnak van Datum es Termek adattagja. Egy Gyorsetterem gyárt többféle Gyros-t is, ennek az egyednek van származtatott és többértékű attribútuma is. Több vásárló rendelhet egy féle Gyros-t, valamint egy vásárló többféle Gyros-t is rendelhet. Vásárlók rendelkezhetnek több Bankkártyával, viszont egy Bankkártyához csak egy Vásárló tartozik.

## ER modell konvertálása XDM modellre:



Dupla-körvonalú elipszissel jelöltem a többértékű attribútumokat.

## XDM alapján elkészített XML:

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

```
<XUE9MH xsi:noNamespaceSchemaLocation="XMLSchemaXUE9MH.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<!-- Gyorsettermek -->
```

```
<Gyorsetterem id="1">
```

```
<Nev>Hammas</Nev>
```

```
<Email>hammas@hesbollah.allah</Email>
```

```
<Telefonszam>0608449860</Telefonszam>
```

```
<Nytivatartas>
```

```
<Hetkoznap>0820</Hetkoznap>
```

```
<Hetvege_unnepek>1116</Hetvege_unnepek>
```

```
</Nytivatartas>
```

```
</Gyorsetterem>
```

```
<Gyorsetterem id="2">
```

```
<Nev>DonnerKebab</Nev>
```

```
<Email>donnerk@gmail.com</Email>
```

```
<Telefonszam>0608456321</Telefonszam>
```

```
<Nytivatartas>
```

```
<Hetkoznap>1422</Hetkoznap>
```

```
<Hetvege_unnepek>1116</Hetvege_unnepek>
```

```

        </Nytvatartas>
</Gyorsetterem>

<Gyorsetterem id="3">
    <Nev>GyrosKing</Nev>
    <Email>info@gyrosking.de</Email>
    <Telefonszam>0608449532</Telefonszam>
    <Nytvatartas>
        <Hetkoznap>0618</Hetkoznap>
        <Hetvege_unnepek>1120</Hetvege_unnepek>
    </Nytvatartas>
</Gyorsetterem>

<!-- Gyros-ok -->

<Gyros id="1" gyorsetteremFK="1">
    <Suly>250</Suly>
    <Toltelek>hus</Toltelek>
    <Toltelek>zoldseg</Toltelek>
    <Toltelek>szosz</Toltelek>
</Gyros>

<Gyros id="2" gyorsetteremFK="2">
    <Suly>500</Suly>
    <Toltelek>hus</Toltelek>
    <Toltelek>zoldseg</Toltelek>
    <Toltelek>pikans</Toltelek>
    <Toltelek>szosz</Toltelek>
</Gyros>

<Gyros id="3" gyorsetteremFK="2">
    <Suly>350</Suly>
    <Toltelek>zoldseg</Toltelek>
    <Toltelek>szosz</Toltelek>
    <Toltelek>pikans</Toltelek>
</Gyros>

<!-- Rendelesek kapcsoló tábla -->

<Rendelesek>
    <Rendel gyrosFK="1" vasarloFK="3"/>
    <Rendel gyrosFK="2" vasarloFK="1"/>
    <Rendel gyrosFK="3" vasarloFK="2"/>
</Rendelesek>

<!-- Vasarlok -->

<Vasarlo id="1">
    <Nev>Simon Daniel</Nev>
    <Telefonszam>0655510430</Telefonszam>
    <Cim>
        <Hatszam>23</Hatszam>
        <Utca>Egyetem ut</Utca>
        <Varos>Miskolc</Varos>
    </Cim>
</Vasarlo>

```

```

<Vasarlo id="2">
  <Nev>Veres Lipot</Nev>
  <Telefonszam>0655156831</Telefonszam>
  <Cim>
    <Hatszam>29</Hatszam>
    <Utca>Nyar utca</Utca>
    <Varos>Salgotarjan</Varos>
  </Cim>
</Vasarlo>

<Vasarlo id="3">
  <Nev>Seres Adam</Nev>
  <Telefonszam>0655813448</Telefonszam>
  <Cim>
    <Hatszam>49</Hatszam>
    <Utca>Rakoczi ut</Utca>
    <Varos>Budapest</Varos>
  </Cim>
</Vasarlo>

<!-- Bankkartyak -->

<Bankkartya id="1" vasarloFK="1">
  <Kartyaszam>1234432112344321</Kartyaszam>
  <CCV>653</CCV>
  <Tipus>Visa</Tipus>
  <Lejarati_datum>11/24</Lejarati_datum>
</Bankkartya>

<Bankkartya id="2" vasarloFK="2">
  <Kartyaszam>7456654774566547</Kartyaszam>
  <CCV>534</CCV>
  <Tipus>Master Card</Tipus>
  <Lejarati_datum>06/25</Lejarati_datum>
</Bankkartya>

<Bankkartya id="3" vasarloFK="3">
  <Kartyaszam>5419298043384654</Kartyaszam>
  <CCV>052</CCV>
  <Tipus>Master Card</Tipus>
  <Lejarati_datum>10/27</Lejarati_datum>
</Bankkartya>

<!-- Beszallit Kapcsoló tábla -->

<Beszallit gyorsetteremFK="3" beszallitoFK="1">
  <Termek>Hus</Termek>
  <Datum>2023-12-05</Datum>
</Beszallit>

<Beszallit gyorsetteremFK="3" beszallitoFK="1">
  <Termek>Zoldseg</Termek>
  <Datum>2023-11-28</Datum>
</Beszallit>

```

```

<Beszallit gyorsetteremFK="2" beszallitoFK="2">
  <Termek>Szosz</Termek>
  <Datum>2023-06-23</Datum>
</Beszallit>

<!-- Beszallito -->

<Beszallito id="1">
  <Nev>Mucsen-ducsen</Nev>
  <Email>muchen@duchen.ak</Email>
  <Telefonszam>0756354789</Telefonszam>
  <Cim>
    <Hatszam>12</Hatszam>
    <Utca>Hajnal utca</Utca>
    <Varos>Debrecen</Varos>
  </Cim>
</Beszallito>

<Beszallito id="2">
  <Nev>Dipmaster</Nev>
  <Email>dipmaster@dmmail.eu</Email>
  <Telefonszam>0655395894</Telefonszam>
  <Cim>
    <Hatszam>24</Hatszam>
    <Utca>Seres utca</Utca>
    <Varos>Szombathely</Varos>
  </Cim>
</Beszallito>

<!-- Tulajok -->

<Tulaj id="1" gyorsetteremFK="3">
  <Nev>Nagy Sanyi</Nev>
  <Email>sanyi.nagy@gmail.com</Email>
  <Telefonszam>0514654254</Telefonszam>
</Tulaj>

<Tulaj id="2" gyorsetteremFK="2">
  <Nev>Fejes Kristof</Nev>
  <Email>FejesKristof@rhyta.com</Email>
  <Telefonszam>0655355327</Telefonszam>
</Tulaj>

<Tulaj id="3" gyorsetteremFK="1">
  <Nev>Jakab Geza</Nev>
  <Email>JakabGeza@armyspy.com</Email>
  <Telefonszam>0655317182</Telefonszam>
</Tulaj>
</XUE9MH>

```

## XMLSchema az XML alapján

```

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

  <!-- Altalanos saját típusok -->

```













Az XMLSchema elkészítésénél saját típusokat és az Eclipse fejlesztőkörnyezetet használtam.

Mind a xml mind az xsd fájlokat validáltam.

## 2. Feladat:

### 2a)Adatolvasás:

Itt iratom ki a dom segítségével az xml fájl adatai. Külön metódusokba rendeztem a kiíratást hogy ezeket a további részfeladatokban is felhasználhassam.

```
package hu.domparse.xue9mh;

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

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

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

public class DomReadXUE9MH {

    public static void main(String[] args) {
        try {
            //xml file megadása
            File inputFile = new File("XML_XUE9MH.xml");

            //Dom-dokumentum létrehozása az XML dokumentum eléréséhez
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(inputFile);
            doc.getDocumentElement().normalize();
            System.out.println("Root element : " +
doc.getDocumentElement().getNodeName());
            System.out.println("-----");

            //Elemek beolvasása és listázása metódusokkal
            NodeList nList = doc.getElementsByTagName("Gyorsetterem");
            gyorsetteremRead(nList);
            System.out.println("-----");

            NodeList nList2 = doc.getElementsByTagName("Gyros");
            gyrosRead(nList2);
            System.out.println("-----");

            NodeList nList3 = doc.getElementsByTagName("Rendel");
            rendelesekRead(nList3);
            System.out.println("-----");

            NodeList nList4 = doc.getElementsByTagName("Vasarlo");
            vasarloRead(nList4);
            System.out.println("-----");
```

```

        NodeList nList5 = doc.getElementsByTagName("Bankkartya");
        bankkartyaRead(nList5);
        System.out.println("-----");

        NodeList nList6 = doc.getElementsByTagName("Beszallit");
        beszallitRead(nList6);
        System.out.println("-----");

        NodeList nList7 = doc.getElementsByTagName("Beszallito");
        beszallitoRead(nList7);
        System.out.println("-----");

        NodeList nList8 = doc.getElementsByTagName("Tulaj");
        tulajRead(nList8);
        System.out.println("-----");

    } catch (Exception e) {
        e.printStackTrace();
    }
}
//Metodus a gyerekelemek kiirasara
public static void getElement(Element eElement, String elementOut, String
elementName) {

    System.out.println(elementOut + " : " +
eElement.getElementsByTagName(elementName).item(0).getTextContent());
}
//kiiro metodusok

public static void gyorsetteremRead(NodeList nList) {
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);
        System.out.println("\nRoot element : " + nNode.getNodeName());

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

            System.out.println("ID : " + eElement.getAttribute("id"));
            getElement(eElement, "Nev", "Nev");
            getElement(eElement, "Email", "Email");
            getElement(eElement, "Telefonszam", "Telefonszam");
            getElement(eElement, "Hetkoznap", "Hetkoznap");
            getElement(eElement, "Hetvege_unnepek", "Hetvege_unnepek");

        }
    }
}

public static void gyrosRead(NodeList nList) {
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);
        System.out.println("\nRoot element : " + nNode.getNodeName());

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

            System.out.println("ID : " + eElement.getAttribute("id"));

```

```

        System.out.println("GyorsetteremFK : " +
eElement.getAttribute("gyorsetteremFK"));
        getElement(eElement, "Suly", "Suly");

        NodeList toltelekList =
eElement.getElementsByTagName("Toltelek");
        toltelekRead(toltelekList);
    }
}

private static void toltelekRead(NodeList toltelekList) {
    String temp = "Toltelekek: ";
    for (int i = 0; i < toltelekList.getLength(); i++) {
        Node toltelekNode = toltelekList.item(i);
        if (toltelekNode.getNodeType() == Node.ELEMENT_NODE) {
            Element toltelekElement = (Element) toltelekNode;
            temp += toltelekElement.getTextContent() + ", ";
        }
    }
    temp = temp.substring(0, temp.length()-2);
    System.out.println(temp);
}

public static void rendelesekRead(NodeList nList) {
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);
        System.out.println("\nRoot element : " + nNode.getNodeName());

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

            System.out.println("gyrosFK : " +
eElement.getAttribute("gyrosFK"));
            System.out.println("vasarloFK : " +
eElement.getAttribute("vasarloFK"));
        }
    }

    public static void vasarloRead(NodeList nList) {
        for (int temp = 0; temp < nList.getLength(); temp++) {
            Node nNode = nList.item(temp);
            System.out.println("\nRoot element : " + nNode.getNodeName());

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

                System.out.println("ID : " + eElement.getAttribute("id"));
                getElement(eElement, "Nev", "Nev");
                getElement(eElement, "Telefonszam", "Telefonszam");
                getElement(eElement, "Hazzsam", "Hazzsam");
                getElement(eElement, "Utca", "Utca");
                getElement(eElement, "Varos", "Varos");
            }
        }
    }
}

```

```

    }

    public static void bankkartyaRead(NodeList nList) {
        for (int temp = 0; temp < nList.getLength(); temp++) {
            Node nNode = nList.item(temp);
            System.out.println("\nRoot element : " + nNode.getNodeName());

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

                System.out.println("ID : " + eElement.getAttribute("id"));
                getElement(eElement, "Kartyaszam", "Kartyaszam");
                getElement(eElement, "CCV", "CCV");
                getElement(eElement, "Tipus", "Tipus");
                getElement(eElement, "Lejarati_datum", "Lejarati_datum");
                System.out.println("VasarloFK : " +
eElement.getAttribute("vasarloFK"));
            }
        }
    }

    public static void beszallitRead(NodeList nList) {
        for (int temp = 0; temp < nList.getLength(); temp++) {
            Node nNode = nList.item(temp);
            System.out.println("\nRoot element : " + nNode.getNodeName());

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

                System.out.println("GyorsetteremFK : " +
eElement.getAttribute("gyorsetteremFK"));
                System.out.println("BeszallitoFK : " +
eElement.getAttribute("beszallitoFK"));
                getElement(eElement, "Termek", "Termek");
                getElement(eElement, "Datum", "Datum");
            }
        }
    }

    public static void beszallitoRead(NodeList nList) {
        for (int temp = 0; temp < nList.getLength(); temp++) {
            Node nNode = nList.item(temp);
            System.out.println("\nRoot element : " + nNode.getNodeName());

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

                System.out.println("ID : " + eElement.getAttribute("id"));
                getElement(eElement, "Nev", "Nev");
                getElement(eElement, "Email", "Email");
                getElement(eElement, "Telefonszam", "Telefonszam");
                getElement(eElement, "Hazzsam", "Hazzsam");
                getElement(eElement, "Utca", "Utca");
                getElement(eElement, "Varos", "Varos");
            }
        }
    }

```



```

    }
}

public static void tulajRead(NodeList nList) {
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);
        System.out.println("\nRoot element :" + nNode.getNodeName());

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

            System.out.println("ID : " + eElement.getAttribute("id"));
            System.out.println("GyorsetteremFK : " +
eElement.getAttribute("gyorsetteremFK"));
            getElement(eElement, "Nev", "Nev");
            getElement(eElement, "Email", "Email");
            getElement(eElement, "Telefonszam", "Telefonszam");
        }
    }
}
}

```

## Output:

```
Root element :XUE9MH
-----

Root element :Gyorsetterem
ID : 1
Nev : Hammas
Email : hammas@hesbollah.allah
Telefonszam : 0608449860
Hetkoznap : 0820
Hetvege_unnepek : 1116

Root element :Gyorsetterem
ID : 2
Nev : DonnerKebab
Email : donnerk@gmail.com
Telefonszam : 0608456321
Hetkoznap : 1422
Hetvege_unnepek : 1116

Root element :Gyorsetterem
ID : 3
Nev : GyrosKing
Email : info@gyrosking.de
Telefonszam : 0608449532
Hetkoznap : 0618
Hetvege_unnepek : 1120
-----

Root element :Gyros
ID : 1
GyorsetteremFK : 1
Suly : 250
Tolelekek: hus, zoldseg, szosz

Root element :Gyros
ID : 2
GyorsetteremFK : 2
Suly : 500
Tolelekek: hus, zoldseg, pikans, szosz
```

```
-----

Root element :Rendel
gyrosFK : 1
vasarloFK : 3

Root element :Rendel
gyrosFK : 2
vasarloFK : 1

Root element :Rendel
gyrosFK : 3
vasarloFK : 2
-----

Root element :Vasarlo
ID : 1
Nev : Simon Daniel
Telefonszam : 0655510430
Hazszam : 23
Utca : Egyetem ut
Varos : Miskolc

Root element :Vasarlo
ID : 2
Nev : Veres Lipot
Telefonszam : 0655156831
Hazszam : 29
Utca : Nyar utca
Varos : Salgotarjan

Root element :Vasarlo
ID : 3
Nev : Seres Adam
Telefonszam : 0655813448
Hazszam : 49
Utca : Rakoczi ut
Varos : Budapest
-----
```

## 2b)Adatmódosítás:

Ebben a programban megkérdezem a felhasználótól hogy milyen típusú elem, melyik példányának (ID) mely adattagját szeretné és mire módosítani. Miután ezt beolvastam az xml fájlból, módosítom azt és kiírom a módosított adatokat a konzol kimenetre.

```
package hu.domparse.xue9mh;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Scanner;

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

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

public class DomModifyXUE9MH {
    public static void main(String[] args) throws TransformerException {
        //Dom-dokumentum letrehozasa az XML dokumentum elereséhez
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        try (InputStream is = new FileInputStream("XML_XUE9MH.xml")) {
            DocumentBuilder db = dbf.newDocumentBuilder();

            Document doc = db.parse(is);
            // Bekerjuk a modositani kivant elemet
            Scanner rootElem = new Scanner(System.in);
            System.out.println("Which element do you want to modify?");
            String element = rootElem.nextLine();

            Scanner elemID = new Scanner(System.in);
            System.out.println("Please provide the ID of said element.");
            String childElementID = elemID.nextLine();

            NodeList listOfElements = doc.getElementsByTagName(element);

            Scanner elem = new Scanner(System.in);
            System.out.println("Which child element do you want to modify?");
            String childElement = elem.nextLine();

            System.out.println("What should its new value be?");
            Scanner ujElem = new Scanner(System.in);
            String childElementElementNew = ujElem.nextLine();

            //megkeressuk a kivant id-vel rendelkezo elemet
            for (int i = 0; i < listOfElements.getLength(); i++) {
```

```

        Node staff = listOfElements.item(i);
        if (staff.getNodeType() == Node.ELEMENT_NODE) {
            String id =
staff.getAttributes().getNamedItem("id").getTextContent();

                if (childElementID.equals(id.trim())) {

                    NodeList childNodes = staff.getChildNodes();
                    //beazonositjuk a gyerek elemet aminek modositjuk
az erteket
                    for (int j = 0; j < childNodes.getLength(); j++) {
                        Node item = childNodes.item(j);
                        if (item.getNodeType() == Node.ELEMENT_NODE)
{
                            if
(childElement.equalsIgnoreCase(item.getNodeName())) {
                                item.setTextContent(childElementElementNew);
                            }
                        }
                    }
                }
            }
        }

        // Leegyszerusitem a kiirast egy TransformerFactory, egy DOMSource
objektum es egy StreamResult letrehozasaal
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        System.out.println("-----After Modification-----");
        transformer.transform(new DOMSource(doc), new
StreamResult(System.out));

    } catch (ParserConfigurationException | SAXException | IOException e) {
        e.printStackTrace();
    }
}
}

```

## OutPut:

```
Which element do you want to modify?
Gyorsetterem
Please provide the ID of said element.
1
Which child element do you want to modify?
Nev
What should its new value be?
HammasHesbollah
-----After Modification-----
<?xml version="1.0" encoding="UTF-8" standalone="no"?><XUE9MH xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaXUE9MH.xsd">

  <!-- Gyorsettermek -->

  <Gyorsetterem id="1">
    <Nev>HammasHesbollah</Nev>
    <Email>hammas@hesbollah.allah</Email>
    <Telefonszam>0608449860</Telefonszam>
    <Nytivatartas>
      <Hetkoznap>0820</Hetkoznap>
      <Hetvege_unnepek>1116</Hetvege_unnepek>
    </Nytivatartas>
  </Gyorsetterem>

  <Gyorsetterem id="2">
    <Nev>DonnerKebab</Nev>
    <Email>donnerk@gmail.com</Email>
    <Telefonszam>0608456321</Telefonszam>
    <Nytivatartas>
      <Hetkoznap>1422</Hetkoznap>
      <Hetvege_unnepek>1116</Hetvege_unnepek>
    </Nytivatartas>
  </Gyorsetterem>

  <Gyorsetterem id="3">
    <Nev>GyrosKing</Nev>
    <Email>info@gyrosking.de</Email>
    <Telefonszam>0608449532</Telefonszam>
    <Nytivatartas>
      <Hetkoznap>0618</Hetkoznap>
      <Hetvege_unnepek>1120</Hetvege_unnepek>
    </Nytivatartas>
  </Gyorsetterem>

  <!-- Gyros-ok -->
```

## 2c)Adatlekérdezés:

Ebben a programban egy felhasználó által kiválasztott elem minden példányát és azon példányok adatait kiíratom ki a konzolra.

```
package hu.domparse.xue9mh;

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

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

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

//DomRead leszarmazottjakent hozom ezt létre hogy a read metodusokat ne kelljen ujra
irni
public class DomQueryXUE9MH extends DomReadXUE9MH {
    public static void main(String[] args) {
        try {
            //xml file nev megadása
            File inputFile = new File("XML_XUE9MH.xml");

            ///Dom-dokumentum létrehozása az XML dokumentum eléréséhez
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(inputFile);
            doc.getDocumentElement().normalize();
            System.out.println("Root element : " +
doc.getDocumentElement().getNodeName());
            System.out.println("-----");

            //Beolvasás
            Scanner myObj = new Scanner(System.in);
            System.out.println("Which element would you like to see?");

            System.out.println("Opciók:Gyros\tGyorsetterem\tRendel\tVasarlo\tBeszallit\tBesz
allito\tTulaj\tBankkartya");
            String element = myObj.nextLine();

            NodeList nList = doc.getElementsByTagName(element);
            switch(element) {
                case "Gyros":
                    gyrosRead(nList);
                    break;
                case "Gyorsetterem":
                    gyorsetteremRead(nList);
                    break;
                case "Rendel":
                    rendelesekRead(nList);
                    break;
                case "Vasarlo":
                    vasarloRead(nList);
                    break;
                case "Beszallit":
                    beszallitRead(nList);
            }
        }
    }
}
```

```

        break;
    case "Beszallito":
        beszallitoRead(nList);
        break;
    case "Tulaj":
        tulajRead(nList);
        break;
    case "Bankkartya":
        bankkartyaRead(nList);
        break;
    default:
        System.out.println("No such element!");
    }
    myObj.close();
} catch (Exception e) {
    e.printStackTrace();
}
}

```

### Output:

```

Root element :XUE9MH
-----
Which element would you like to see?
Opciók:Gyros      Gyorsetterem      Rendel  Vasarlo Beszallit      Beszallito      Tulaj      Bankkartya
Gyorsetterem

Root element :Gyorsetterem
ID : 1
Nev : Hammas
Email : hammas@hesbollah.allah
Telefonszam : 0608449860
Hetkoznap : 0820
Hetvege_unnepek : 1116

Root element :Gyorsetterem
ID : 2
Nev : DonnerKebab
Email : donnerk@gmail.com
Telefonszam : 0608456321
Hetkoznap : 1422
Hetvege_unnepek : 1116

Root element :Gyorsetterem
ID : 3
Nev : GyrosKing
Email : info@gyrosking.de
Telefonszam : 0608449532
Hetkoznap : 0618
Hetvege_unnepek : 1120

```

## 2d)Adatírás:

Ebben a programban az xml fájlt kiírom konzolra majd majd létrehozok egy új xml fájlt és kiírom abba is.  
package hu.domparse.xue9mh;

```
import javax.xml.parsers.*;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import java.io.*;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class DomWriteXUE9MH {
    public static void main(String[] args) {
        try {
            // Beolvasom az xml file tartalmat
            File inputFile = new File("XML_XUE9MH.xml");
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document document = builder.parse(inputFile);

            // Kiíratom az xml tartalmat konzolra
            printDocument(document);

            // Kiirom a dokumentumot egy új fájlba
            File outputFile = new File("XML_XUE9MH1.xml");
            writeDocument(document, outputFile);

            System.out.println("XML content written to 'output.xml'.");

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



```
// XML tartalmat konzolra kiíró metódus
private static void printDocument(Document document) {
    try {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        StringWriter writer = new StringWriter();
        transformer.transform(new DOMSource(document), new StreamResult(writer));
        System.out.println(writer.toString());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

// XML tartalmat új fájlba kiíró metódus
private static void writeDocument(Document document, File outputFile) {
    try {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        transformer.setOutputProperty(OutputKeys.INDENT, "yes");
        transformer.transform(new DOMSource(document), new StreamResult(outputFile));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

Console Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><XUE9MH xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaXUE9MH.xsd">

  <!-- Gyorsettermek -->

  <Gyorsetterem id="1">
    <Nev>Hammas</Nev>
    <Email>hammas@hesbollah.allah</Email>
    <Telefonszam>0608449860</Telefonszam>
    <Nytivatartas>
      <Hetkoznap>0820</Hetkoznap>
      <Hetvege_unnepek>1116</Hetvege_unnepek>
    </Nytivatartas>
  </Gyorsetterem>

  <Gyorsetterem id="2">
    <Nev>DonnerKebab</Nev>
    <Email>donnerk@gmail.com</Email>
    <Telefonszam>0608456321</Telefonszam>
    <Nytivatartas>
      <Hetkoznap>1422</Hetkoznap>
      <Hetvege_unnepek>1116</Hetvege_unnepek>
    </Nytivatartas>
  </Gyorsetterem>

  <Gyorsetterem id="3">
    <Nev>GyrosKing</Nev>
    <Email>info@gyrosking.de</Email>
    <Telefonszam>0608449532</Telefonszam>
    <Nytivatartas>
      <Hetkoznap>0618</Hetkoznap>
      <Hetvege_unnepek>1120</Hetvege_unnepek>
    </Nytivatartas>
  </Gyorsetterem>

  <!-- Gyros-ok -->
```