

# Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Gyorsétterem

Készítette: **Simon Dániel**

Neptunkód: **XUE9MH**

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

# Tartalomjegyzék:

Jegyzőkönyv .....	1
<b>Feladat leírása .....</b>	<b>3</b>
<b>1. Feladat: .....</b>	<b>4</b>
<b>1a)ER modell:.....</b>	<b>4</b>
<b>1b) XDM modell:.....</b>	<b>4</b>
<b>1c) XML: .....</b>	<b>5</b>
<b>1d)XMLSchema .....</b>	<b>9</b>
<b>2. Feladat: .....</b>	<b>17</b>
<b>2a)Adatolvasás: .....</b>	<b>17</b>
<b>2b)Adatmódosítás:.....</b>	<b>26</b>
<b>2c)Adatlekérdezés:.....</b>	<b>30</b>
<b>2d)Adatírás: .....</b>	<b>33</b>

## Feladat leírása

A feladat egy vagy több Gyros-os gyorsétterem adatalemeinek és kapcsolatainak modellezése és nyilvántartása XML környezetben, validálása valamint korlátozása XSD segítségével majd az adott adatok olvasása, lekérdezése, módosítása és írása DOM programok segítségével.

Helyet kapnak még a feladat megoldásában a gyorsétterem tulajdonosok, beszállítók, gyros-ok, vásárlók és bankkártyaírok.

A megoldás a feladat kiírás szerint tartalmaz legalább 5 egyedet, melyek a következők: Gyorsétterem, Beszállító, Tulaj, Gyros, Vásárló és Bankkártya. Ezek mindegyike tartalmaz minimum 4 tulajdonságot (normál, kulcs, összetett, többértékű). A megoldásban szerepel mindenféle kapcsolat (1:1, 1:N, N:M).

A megoldás témája, ER modellje, XDM modellje egyeztetve lett rövidesen feladatkiírás után Dr. Bednarik László tárgyjegyzővel.

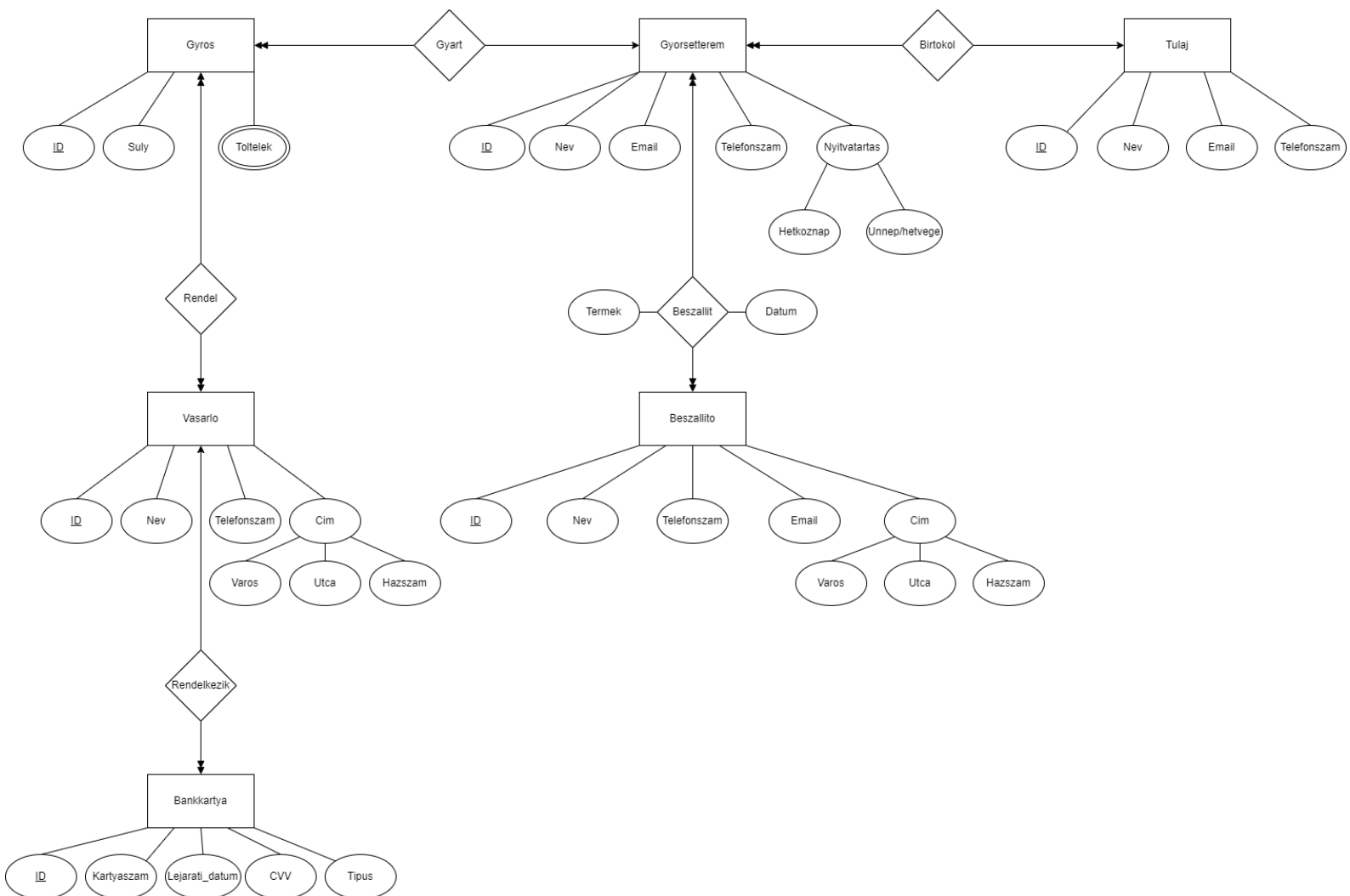
Segédprogramként az Eclipse fejlesztőkörnyezetet használtam valamint a „dia” online model szerkesztő programot.

## 1. Feladat:

### 1a)ER modell:

Leírás: Az ER modell megvalósítására először ERDPlus-t használtam azonban a korlátozott eszközök miatt nem tudtam megfelelő minőségű modellt készíteni így később rekreáltam az [app.diagrams.net](http://app.diagrams.net) honlap segítségével. A modell be lett mutatva személyesen, majd javaslatok által kijavítva majd újra bemutatva e-mail segítségével.

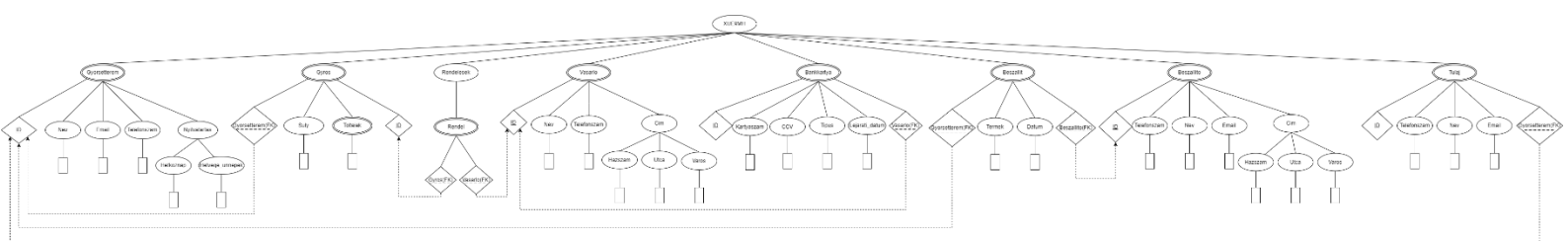
Modell:



### 1b) XDM modell:

Leírás: Korábbi hibámból tanulva ezt a modellt kezdettől az [app.diagrams.net](http://app.diagrams.net) honlap segítségével állítottam elő. Mivel a több-több kapcsolat megvalósításához szükségesek a kapcsolótáblák így azok szerepelnek a megvalósításban a Rendelkezik, Gyart es Birtokol kapcsolatok nem szerepelnek különálló elemként.

Modell:



## 1c) XML:

Leírás: Az XML fájl az XDM modell alapján került megvalósításra. Megvalósítására az Eclipse fejlesztő környezetet használtam. A feladat kiírás szerint minden többszörös előfordulású elem legalább 3 példányban szerepel. Kódban megjegyzések szerepelnek.

Kód:

```
<?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>
  <Nyitvatartas>
    <Hetkoznap>0820</Hetkoznap>
    <Hetvege_unnepek>1116</Hetvege_unnepek>
  </Nyitvatartas>
</Gyorsetterem>
```

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

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

<!-- Bankkartyák -->

```

```
<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 tabla -->
```

```
<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" gyorssetteremFK="3">
    <Nev>Nagy Sanyi</Nev>
    <Email>sanyi.nagy@gmail.com</Email>
    <Telefonszam>0514654254</Telefonszam>
</Tulaj>

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

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



## 1d)XMLSchema

Leírás: Az XSD fájl az XML fájl alapján lett elkészítve. Célja az XML validálása. Tartalmazza a feladat kiírás által elvárt saját típusokat, ref, key, keyref és speciális elemeket. Egyszerűbb olvashatóságért tartalmaz kommenteket is.

Kód:

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

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

    <!-- Altalanos saját típusok -->

    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Email" type="xs:string"/>
    <xs:element name="Telefonszam" type="telefonszamTipus"/>

    <!-- Gyorsetterem saját típusok -->

    <xs:element name="Nyitvatartas" type="nyitvatartasType"/>

    <!-- Gyros saját típusok -->

    <xs:element name="Suly" type="xs:integer"/>
    <xs:element name="Toltelek" type="toltelekTipus"/>

    <!-- Vasarlo saját típusok -->

    <xs:element name="Cim" type="cimType"/>

    <!-- Bankkartya saját típusok -->

    <xs:element name="Kartyaszam" type="kartyaszamTipus"/>
    <xs:element name="CCV" type="ccvTipus"/>
    <xs:element name="Tipus" type="bankkartySzolgáltatoTipus"/>
    <xs:element name="Lejarati_datum" type="xs:string"/>
```

<!-- Beszallit kapcsolotabla saját típusok -->

<xs:element name="Termek" type="xs:string"/>

<xs:element name="Datum" type="xs:date"/>

<!-- Egyszeru típusok -->

<xs:simpleType name="bankkartySzolgáltatoTípus">

<xs:restriction base="xs:string">

<xs:enumeration value="Visa"/>

<xs:enumeration value="Master Card"/>

<xs:enumeration value="American Express"/>

<xs:enumeration value="Discover"/>

<xs:enumeration value="Maestro"/>

<xs:enumeration value="UnionPay"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="idTípus">

<xs:restriction base="xs:integer">

<xs:minInclusive value="1"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="oraTípus">

<xs:restriction base="xs:string">

<xs:pattern value="\d{4}"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="telefonszamTípus">

<xs:restriction base="xs:string">

<xs:pattern value="\d{10}"/>

</xs:restriction>

</xs:simpleType>

```

<xs:simpleType name="toltelekTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="hus"/>
    <xs:enumeration value="zoldseg"/>
    <xs:enumeration value="pikans"/>
    <xs:enumeration value="szosz"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="kartyaszamTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{16}"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ccvTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{3}"/>
  </xs:restriction>
</xs:simpleType>

<!-- Gyokerelem összetett tipusa -->
<xs:complexType name="rootType">
  <xs:sequence>
    <xs:element name="Gyorsetterem" type="gyorsetteremType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Gyros" type="gyrosType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Rendelesek" type="rendelesekType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Vasarlo" type="vasarloType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Bankkartya" type="bankkartyaType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Beszallit" type="beszallitType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Beszallito" type="beszallitoType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Tulaj" type="tulajType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>

```

```
</xs:complexType>
```

```
<!-- Egyebbb osszetett tipusok -->
```

```
<xs:complexType name="gyorsetteremType">
```

```
  <xs:sequence>
```

```
    <xs:element ref="Nev"/>
```

```
    <xs:element ref="Email"/>
```

```
    <xs:element ref="Telefonszam"/>
```

```
    <xs:element ref="Nytivatartas"/>
```

```
  </xs:sequence>
```

```
  <xs:attribute name="id" type="idTipus" use="required"/>
```

```
</xs:complexType>
```

```
<xs:complexType name="gyrosType">
```

```
  <xs:sequence>
```

```
    <xs:element ref="Suly"/>
```

```
    <xs:element ref="Toltelek" maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
  <xs:attribute name="id" type="idTipus" use="required"/>
```

```
  <xs:attribute name="gyorsetteremFK" type="idTipus" use="required"/>
```

```
</xs:complexType>
```

```
<xs:complexType name="nyitvatartasType">
```

```
  <xs:sequence>
```

```
    <xs:element name="Hetkoznap" type="oraTipus"/>
```

```
    <xs:element name="Hetvege_unnepek" type="oraTipus"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

```
<xs:complexType name="rendelesekType">
```

```
  <xs:sequence>
```

```
    <xs:element name="Rendel" type="rendelType" maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

```
<xs:complexType name="rendelType">
```

```

        <xs:attribute name="gyrosFK" type="idTipus" use="required"/>
        <xs:attribute name="vasarloFK" type="idTipus" use="required"/>
    </xs:complexType>

```

```

<xs:complexType name="vasarloType">
    <xs:sequence>
        <xs:element ref="Nev"/>
        <xs:element ref="Telefonszam"/>
        <xs:element ref="Cim"/>
    </xs:sequence>
    <xs:attribute name="id" type="idTipus" use="required"/>
</xs:complexType>

```

```

<xs:complexType name="cimType">
    <xs:sequence>
        <xs:element name="Hatszam" type="xs:integer"/>
        <xs:element name="Utca" type="xs:string"/>
        <xs:element name="Varos" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="bankkartyaType">
    <xs:sequence>
        <xs:element ref="Kartyaszam"/>
        <xs:element ref="CCV"/>
        <xs:element ref="Tipus"/>
        <xs:element ref="Lejarati_datum"/>
    </xs:sequence>
    <xs:attribute name="id" type="idTipus" use="required"/>
    <xs:attribute name="vasarloFK" type="idTipus" use="required"/>
</xs:complexType>

```

```

<xs:complexType name="beszallitType">
    <xs:sequence>
        <xs:element ref="Termek"/>

```

```

        <xs:element ref="Datum"/>

    </xs:sequence>

    <xs:attribute name="gyorsetteremFK" type="idTipus" use="required"/>

    <xs:attribute name="beszallitoFK" type="idTipus" use="required"/>
</xs:complexType>

<xs:complexType name="beszallitoType">
    <xs:sequence>
        <xs:element ref="Nev"/>
        <xs:element ref="Email"/>
        <xs:element ref="Telefonszam"/>
        <xs:element ref="Cim"/>
    </xs:sequence>
    <xs:attribute name="id" type="idTipus" use="required"/>
</xs:complexType>

<xs:complexType name="tulajType">
    <xs:sequence>
        <xs:element ref="Nev"/>
        <xs:element ref="Email"/>
        <xs:element ref="Telefonszam"/>
    </xs:sequence>
    <xs:attribute name="id" type="idTipus" use="required"/>
    <xs:attribute name="gyorsetteremFK" type="idTipus" use="required"/>
</xs:complexType>

<!-- kulcsok -->

<xs:element name="XUE9MH" type="rootType">
    <!-- Elsodleges kulcsok -->
    <xs:key name="GyorsetteremPK">
        <xs:selector xpath="Gyorsetterem" />
        <xs:field xpath="@id" />
    </xs:key>
    <xs:key name="GyrosPK">

```

```

        <xs:selector xpath="Gyros" />
        <xs:field xpath="@id" />
    </xs:key>
    <xs:key name="VasarloPK">
        <xs:selector xpath="Vasarlo" />
        <xs:field xpath="@id" />
    </xs:key>
    <xs:key name="BankkartyaPK">
        <xs:selector xpath="Bankkartya" />
        <xs:field xpath="@id" />
    </xs:key>
    <xs:key name="BeszallitoPK">
        <xs:selector xpath="Beszallito" />
        <xs:field xpath="@id" />
    </xs:key>
    <xs:key name="TulajPK">
        <xs:selector xpath="Tulaj" />
        <xs:field xpath="@id" />
    </xs:key>
    <!-- Idegen kulcsok -->
    <xs:keyref refer="GyorsetteremPK" name="GyrosArusitojaFK">
        <xs:selector xpath="Gyros" />
        <xs:field xpath="@gyorsetteremFK" />
    </xs:keyref>
    <xs:keyref refer="GyrosPK" name="vasarloGyrosaFK">
        <xs:selector xpath="Rendel" />
        <xs:field xpath="@gyrosFK" />
    </xs:keyref>
    <xs:keyref refer="VasarloPK" name="gyrosVasarlojaFK">
        <xs:selector xpath="Rendel" />
        <xs:field xpath="@vasarloFK" />
    </xs:keyref>
    <xs:keyref refer="VasarloPK" name="BankkartyaTulajFK">
        <xs:selector xpath="Bankkartya" />
        <xs:field xpath="@vasarloFK" />
    </xs:keyref>

```

```

</xs:keyref>

<xs:keyref refer="BeszallitoPK" name="gyrososBeszallitottolFK">
    <xs:selector xpath="Beszallit" />
    <xs:field xpath="@beszallitoFK" />
</xs:keyref>

<xs:keyref refer="GyorsetteremPK" name="beszallitoGyrososnakFK">
    <xs:selector xpath="Beszallit" />
    <xs:field xpath="@gyorsetteremFK" />
</xs:keyref>

<xs:keyref refer="GyorsetteremPK" name="tulajGyorsettermeFK">
    <xs:selector xpath="Tulaj" />
    <xs:field xpath="@gyorsetteremFK" />
</xs:keyref>

</xs:element>

</xs:schema>

```



## 2. Feladat:

### 2a)Adatolvasás:

Leírás: A program célja az előre megadott nevű XML fájl olvasása majd kiírása a konzolra strukturált formában. A megvalósításnál használtam a File, DocumentBuilderFactory, DocumentBuilder, Document, NodeList, Node és Element osztályokat. Az első három az xml file bejárására szolgált és a DOM fa megépítésére. A Document és a NodeList az elemek példányainak egy listába gyűjtésére lettek használva. Végül a Node és az Element az elemek értékeinek/adatainak lekérdezésére lettek használva.

Kód:

```
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-fa létrehozása az XML dokumentum eléréséhez
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(inputFile);
            doc.getDocumentElement().normalize();
            System.out.println("<" + doc.getDocumentElement().getNodeName() + ">");

            //Elemek beolvasása és listázása metódusokkal
            NodeList nList = doc.getElementsByTagName("Gyorsetterem");
```

```
gyorsetteremRead(nList);
```

```
NodeList nList2 = doc.getElementsByTagName("Gyros");  
gyrosRead(nList2);
```

```
NodeList nList3 = doc.getElementsByTagName("Rendel");  
rendelesRead(nList3);
```

```
NodeList nList4 = doc.getElementsByTagName("Vasarlo");  
vasarloRead(nList4);
```

```
NodeList nList5 = doc.getElementsByTagName("Bankkartya");  
bankkartyaRead(nList5);
```

```
NodeList nList6 = doc.getElementsByTagName("Beszallit");  
beszallitRead(nList6);
```

```
NodeList nList7 = doc.getElementsByTagName("Beszallito");  
beszallitoRead(nList7);
```

```
NodeList nList8 = doc.getElementsByTagName("Tulaj");  
tulajRead(nList8);
```

```
System.out.println("</" + doc.getDocumentElement().getNodeName() + ">");
```

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

```
//Metodus a gyerekelemek kiirasara
```

```
public static void getElement(Element eElement, String elementOut, String elementName) {
```

```
    System.out.println("\t\t<" + elementOut + ">" + eElement.getElementsByTagName(elementName).item(0).getTextContent() +  
    "</" + elementOut + ">");
```

```

}

public static void getSubElement(Element eElement, String elementOut, String elementName) {

    System.out.println("\t\t\t<" + elementOut + ">" + eElement.getElementsByTagName(elementName).item(0).getTextContent() +
"</" + elementOut + ">");

}

//beolvaso metodusok

public static void gyorsetteremRead(NodeList nList) {

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

        Node nNode = nList.item(temp);

        if(temp==0) {

            System.out.println("\t<" + nNode.getNodeName() + " id:" + ((Element) nNode).getAttribute("id") + ">");

        } else {

            System.out.println("\n\t<" + nNode.getNodeName() + " id:" + ((Element) nNode).getAttribute("id") + ">");

        }

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

            Element eElement = (Element) nNode;

            getElement(eElement, "Nev", "Nev");

            getElement(eElement, "Email", "Email");

            getElement(eElement, "Telefonszam", "Telefonszam");

            System.out.println("\t\t<Nyitvatartas>");

            getSubElement(eElement, "Hetkoznap", "Hetkoznap");

            getSubElement(eElement, "Hetvege_unnepek", "Hetvege_unnepek");

            System.out.println("\t\t</Nyitvatartas>");

        }

        System.out.println("\t</" + nNode.getNodeName() + ">");

    }

}

public static void gyrosRead(NodeList nList) {

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

        Node nNode = nList.item(temp);

        System.out.println("\n\t<" + nNode.getNodeName() + " id:" + ((Element) nNode).getAttribute("id") + " gyorsetteremFK:" +
((Element) nNode).getAttribute("gyorsetteremFK") + ">");

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

            Element eElement = (Element) nNode;


```

```

        getElement(eElement, "Suly", "Suly");

        NodeList toltelekList = eElement.getElementsByTagName("Toltelek");
        toltelekRead(toltelekList);
    }
    System.out.println("\t</" + nNode.getNodeName() + ">");
}

}

private static void toltelekRead(NodeList toltelekList) {
    for (int i = 0; i < toltelekList.getLength(); i++) {
        Node toltelekNode = toltelekList.item(i);
        if (toltelekNode.getNodeType() == Node.ELEMENT_NODE) {
            Element toltelekElement = (Element) toltelekNode;
            System.out.println("\t\t<Toltelek>" + toltelekElement.getTextContent() + "</Toltelek>");
        }
    }
}

public static void rendelesekRead(NodeList nList) {
    System.out.println("\n\t<Rendelesek>");
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) nNode;
            System.out.println("\t\t<" + nNode.getNodeName() + " gyrosFK : " + eElement.getAttribute("gyrosFK") + " vasarloFK : " + eElement.getAttribute("vasarloFK") + "/>");
        }
    }
    System.out.println("\t</Rendelesek>");
}

public static void vasarloRead(NodeList nList) {
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);
        System.out.println("\n\t<" + nNode.getNodeName() + " id:" + ((Element) nNode).getAttribute("id") + ">");

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

```

```

        getElement(eElement, "Nev", "Nev");

        getElement(eElement, "Telefonszam", "Telefonszam");

        System.out.println("\t\t<Cim>");

        getSubElement(eElement, "Hatszam", "Hatszam");

        getSubElement(eElement, "Utca", "Utca");

        getSubElement(eElement, "Varos", "Varos");

        System.out.println("\t\t</Cim>");

    }

    System.out.println("\t</>" + nNode.getNodeName() + ">");

}

}

public static void bankkartyaRead(NodeList nList) {

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

        Node nNode = nList.item(temp);

        System.out.println("\n\t<" + nNode.getNodeName() + " id: " + ((Element) nNode).getAttribute("id") + " vasarloFK : " +
        ((Element) nNode).getAttribute("vasarloFK") + ">");

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

            Element eElement = (Element) nNode;

            getElement(eElement, "Kartyaszam", "Kartyaszam");

            getElement(eElement, "CCV", "CCV");

            getElement(eElement, "Tipus", "Tipus");

            getElement(eElement, "Lejarati_datum", "Lejarati_datum");

            System.out.println("\t</>" + nNode.getNodeName() + ">");

        }

    }

}

public static void beszallitRead(NodeList nList) {

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

        Node nNode = nList.item(temp);

        System.out.println("\n\t<" + nNode.getNodeName() + " gyorsetteremFK : " + ((Element)
nNode).getAttribute("gyorsetteremFK") + " beszallitoFK : " + ((Element) nNode).getAttribute("beszallitoFK") + ">");

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

            Element eElement = (Element) nNode;

            getElement(eElement, "Termek", "Termek");

            getElement(eElement, "Datum", "Datum");


```

```

        System.out.println("\t</" + nNode.getNodeName() + ">");
    }
}

public static void beszallitoRead(NodeList nList) {
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);

        System.out.println("\n\t<" + nNode.getNodeName() + " id:" + ((Element) nNode).getAttribute("id") + ">");

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

            getElement(eElement, "Nev", "Nev");
            getElement(eElement, "Email", "Email");
            getElement(eElement, "Telefonszam", "Telefonszam");

            System.out.println("\t\t<Cim>");

            getSubElement(eElement, "Hatszam", "Hatszam");
            getSubElement(eElement, "Utca", "Utca");
            getSubElement(eElement, "Varos", "Varos");

            System.out.println("\t\t</Cim>");

            System.out.println("\t</" + nNode.getNodeName() + ">");
        }
    }
}

public static void tulajRead(NodeList nList) {
    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);

        System.out.println("\n\t<" + nNode.getNodeName() + " id:" + ((Element) nNode).getAttribute("id") + " gyorssetteremFK : " +
        ((Element) nNode).getAttribute("gyorssetteremFK") + ">");

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

            getElement(eElement, "Nev", "Nev");
            getElement(eElement, "Email", "Email");
            getElement(eElement, "Telefonszam", "Telefonszam");

            System.out.println("\t</" + nNode.getNodeName() + ">");
        }
    }
}
}

```

## Output:

```
<XUE9MH>
  <Gyorsetterem id:1>
    <Nev>Hammas</Nev>
    <Email>hammas@hesbollah.allah</Email>
    <Telefonszam>0608449860</Telefonszam>
    <Nyitvatartas>
      <Hetkoznap>0820</Hetkoznap>
      <Hetvege_unnepek>1116</Hetvege_unnepek>
    </Nyitvatartas>
  </Gyorsetterem>

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

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

  <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>
  <Rendel gyrosFK : 1 vasarloFK : 3/>
  <Rendel gyrosFK : 2 vasarloFK : 1/>
  <Rendel gyrosFK : 3 vasarloFK : 2/>
</Rendelesek>

<Vasarlo id:1>
  <Nev>Simon Daniel</Nev>
  <Telefonszam>0655510430</Telefonszam>
  <Cim>
    <Hazzsam>23</Hazzsam>
    <Utca>Egyetem ut</Utca>
    <Varos>Miskolc</Varos>
  </Cim>
</Vasarlo>

<Vasarlo id:2>
  <Nev>Veres Lipot</Nev>
  <Telefonszam>0655156831</Telefonszam>
  <Cim>
    <Hazzsam>29</Hazzsam>
    <Utca>Nyar utca</Utca>
    <Varos>Salgotarjan</Varos>
  </Cim>
</Vasarlo>

```

```

<Vasarlo id:3>
  <Nev>Seres Adam</Nev>
  <Telefonszam>0655813448</Telefonszam>
  <Cim>
    <Hazzsam>49</Hazzsam>
    <Utca>Rakoczi ut</Utca>
    <Varos>Budapest</Varos>
  </Cim>
</Vasarlo>

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

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

<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>
```

## 2b)Adatmódosítás:

Leírás: A program célja egy előre megadott XML fájl beolvasása, módosítása majd kiírása konzolra. A megvalósításnál a DocumentBuilderFactory, InputStream, Documentbuilder és Document osztályokat használtam a DOM fa felépítésére az XML fájl olvasásakor. Ezután a Scanner, NodeList és Node osztályokat használtam az a felhasznalo inputjanak bekeresere, elemek közti navigálásra valamint később a módosításra is. Végül a TransformerFactory és Transformer osztályok segítségével kiírtam konzolra a teljes módosított xml fájlt.

Kód:

```
package hu.domparsing.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.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-fa létrehozása az XML dokumentum eléréséhez

        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        try (InputStream is = new FileInputStream("XML_XUE9MH.xml")) {

            DocumentBuilder db = dbf.newDocumentBuilder();


            Document doc = db.parse(is);
```

```

doc.getDocumentElement().normalize();

// 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.getAttribute().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) {

                    //modositjuk a tartalmat

                    if (childElement.equalsIgnoreCase(item.getNodeName())) {

                        item.setTextContent(childElementElementNew);
                    }
                }
            }
        }
    }
}

```

```

        } else if(childElement.equalsIgnoreCase("Hazszam") ||
childElement.equalsIgnoreCase("Utca") || childElement.equalsIgnoreCase("Varos") || childElement.equalsIgnoreCase("Hetkoznap") ||
childElement.equalsIgnoreCase("Hetvege_unnepek")) {

        NodeList subchildren = item.getChildNodes();

        for(int k = 0; k < subchildren.getLength();k++) {

            Node subchild = subchildren.item(k);

            if

(childElement.equalsIgnoreCase(subchild.getNodeName())) {

                subchild.setTextContent(childElementElementNew);

            }

        }

    }

}

// kiirjuk a konzolra a modositott xml-t

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();

}

}

}

```

OutPut1:

```
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="ht

    <!-- 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>
```

Output2:

```
Which element do you want to modify?
Tulaj
Please provide the ID of said element.
1
Which child element do you want to modify?
Nev
What should its new value be?
Nagy Sandor
```

```
<!-- Tulajok -->
```

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

Output3:

```
Which element do you want to modify?
Bankkartya
Please provide the ID of said element.
2
Which child element do you want to modify?
CCV
What should its new value be?
543
<Bankkartya id="2" vasarloFK="2">
    <Kartyaszam>7456654774566547</Kartyaszam>
    <CCV>543</CCV>
    <Tipus>Master Card</Tipus>
    <Lejarati_datum>06/25</Lejarati_datum>
</Bankkartya>
```

Output4:

```
Which element do you want to modify?
Gyorsetterem
Please provide the ID of said element.
3
Which child element do you want to modify?
Hetvege_unnepek
What should its new value be?
1020
<Gyorsetterem id="3">
    <Nev>GyrosKing</Nev>
    <Email>info@gyrosking.de</Email>
    <Telefonszam>0608449532</Telefonszam>
    <Nytivatartas>
        <Hetkoznap>0618</Hetkoznap>
        <Hetvege_unnepek>1020</Hetvege_unnepek>
    </Nytivatartas>
</Gyorsetterem>
```

Output5:

```
Which element do you want to modify?
Vasarlo
Please provide the ID of said element.
3
Which child element do you want to modify?
Varos
What should its new value be?
Gyor
```

```
<Vasarlo id="3">
    <Nev>Seres Adam</Nev>
    <Telefonszam>0655813448</Telefonszam>
    <Cim>
        <Hazzsam>49</Hazzsam>
        <Utca>Rakoczi ut</Utca>
        <Varos>Gyor</Varos>
    </Cim>
</Vasarlo>
```

## 2c)Adatlekérdezés:

Leírás: Ennek a programnak a DOMQuery osztálya leszármazottja a DomRead osztálynak. File, DocumentBuilderFactory, DocumentBuilder, Document osztályok segítségével beolvastam az XML fájlt és létrehoztam a Dom fát. A Scanner osztály segítségével bekértem a felhasználótól a kívánt elem nevét. Ezután egy NodeList és az űs osztály metódusai segítségével kiírom a konzolra strukturáltan a választott elem példányait.

Kód:

```
package hu.domparsing.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 leszármazottjaként hozom ezt létre hogy a read metódusokat ne kelljen újra írni
```

```
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();
```

```
            //Keresett elem nevének bekérése
```

```
            Scanner myObj = new Scanner(System.in);
```

```
            System.out.println("Which element would you like to see?");
```

```
            System.out.println("Opciók:Gyros/Gyorsettem/Rendel/Vasarlo/Beszallit/Beszallito/Tulaj/Bankkartya");
```

```
            String element = myObj.nextLine();
```

```
            //Keresett elem összes példányának kiírása
```

```

        System.out.println("<" + doc.getDocumentElement().getNodeName() + ">");

        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();

        System.out.println("</" + doc.getDocumentElement().getNodeName() + ">");

    } catch (Exception e) {

        e.printStackTrace();

    }

}

}

```

## Output1:

```
Which element would you like to see?
Opciók:Gyros/Gyorsetterem/Rendel/Vasarlo/Beszallit/Beszallito/Tulaj/Bankkartya
Tulaj
<XUE9MH>

  <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>
```

## Output2:

```
Which element would you like to see?
Opciók:Gyros/Gyorsetterem/Rendel/Vasarlo/Beszallit/Beszallito/Tulaj/Bankkartya
Gyros
<XUE9MH>

  <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>
</XUE9MH>
```

## Output3:

```
Which element would you like to see?
Opciók:Gyros/Gyorsetterem/Rendel/Vasarlo/Beszallit/Beszallito/Tulaj/Bankkartya
Rendel
<XUE9MH>

  <Rendelesek>
    <Rendel gyrosFK : 1 vasarloFK : 3/>
    <Rendel gyrosFK : 2 vasarloFK : 1/>
    <Rendel gyrosFK : 3 vasarloFK : 2/>
  </Rendelesek>
</XUE9MH>
```

## Output4:

```
Which element would you like to see?
Opciók:Gyros/Gyorsetterem/Rendel/Vasarlo/Beszallit/Beszallito/Tulaj/Bankkartya
Beszallit
<XUE9MH>

  <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>
</XUE9MH>
```

## Output5:

```
Which element would you like to see?
Opciók:Gyros/Gyorsetterem/Rendel/Vasarlo/Beszallit/Beszallito/Tulaj/Bankkartya
Beszallito
<XUE9MH>

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

  <Beszallito id:2>
    <Nev>Dipmaster</Nev>
    <Email>dipmaster@dmmail.eu</Email>
    <Telefonszam>0655395894</Telefonszam>
    <Cim>
      <Haszam>24</Haszam>
      <Utca>Seres utca</Utca>
      <Varos>Szombathely</Varos>
    </Cim>
  </Beszallito>
</XUE9MH>
```



## 2d)Adatírás:

Leírás: Ez a program strukturáltan kiírja a konzolra egy xml fájl tartalmát majd készít egy kópia xml fájlt. File, DocumentBuilderFactory, DocumentBuilder, Document osztályok segítségével beolvastam az XML fájlt és létrehoztam a Dom fát. Majd TransformerFactory és Transformer osztályokat felhasználva kirtam a konzolra és egy új fájlba.

Kód:

```
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 uj fajlba
    File outputFile = new File("XML_XUE9MH1.xml");
    writeDocument(document, outputFile);
} 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>Hamas</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 -->
```