

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Csabai Tibor**

Neptunkód: **CIEYIE**

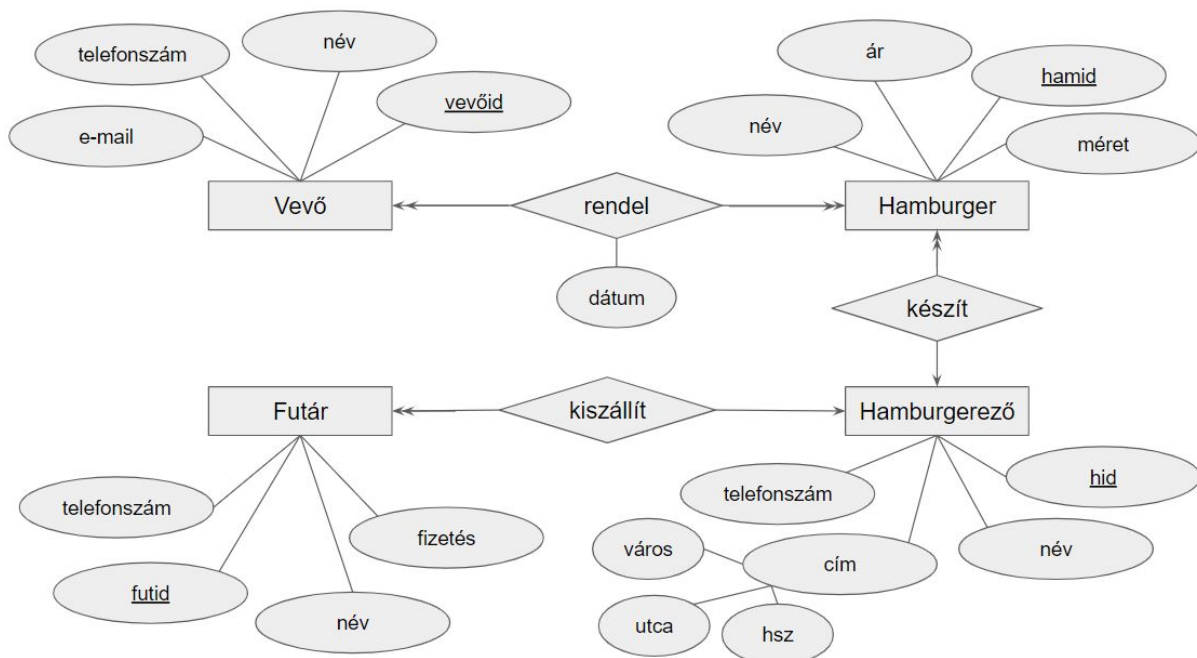
A feladat leírása:

Az általam elkészített feladat hamburgerezők rendeléseit szemlélteti. A vevő tud hamburgereket rendelni, melyeket egy hamburgerező készít el a megrendelésre. A hamburgerezőkhöz tartoznak futárok is, akik a rendeléseket kiszállítják a rendelőnek. A vevő tulajdonságai: telefonszám, e-mail, név, vevőid(azonosító). A hamburger tulajdonságai: név, ár, méret, hamid(azonosító). A hamburgerező tulajdonságai: telefonszám, név, cím(város, utca, hsz), hid(azonosító). A futár tulajdonságai: név, fizetés, telefonszám, futid(azonosító). Ezek között pedig három kapcsolat áll fenn.

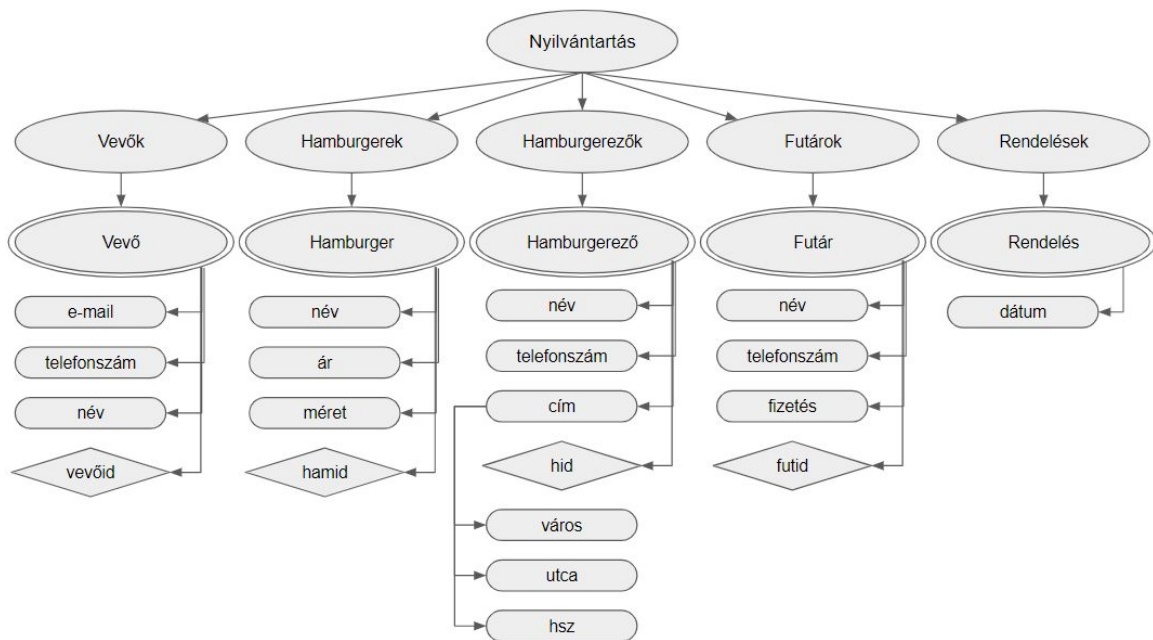
- rendel: (Vevő <-> Hamburger) N:M kapcsolat, dátum tulajdonsággal.
- készít: (Hamburger <-> Hamburgerező) 1:N kapcsolat.
- kiszállít (Hamburgerező <-> Futár) 1:N kapcsolat.

1.feladat

1a) ER modell:



1b) XDM modell:



1c) XML kód(validált):

```

<?xml version="1.0" encoding="UTF-8" ?>
<nyilvantartas xmlns="beadando"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="beadando XMLSchemeCIEYIE.xsd">
  <vevok>
    <vevo vevoid="0">
      <nev>XMLTeszt Elek</nev>
      <email>t.elek@gmail.com</email>
      <telefonszam>066666666666</telefonszam>
    </vevo>
    <vevo vevoid="1">
      <nev>Totál Levente</nev>
      <email>t.levente@gmail.com</email>
      <telefonszam>066636666666</telefonszam>
    </vevo>
    <vevo vevoid="2">
      <nev>Lopós Lajos</nev>
      <email>l.lajcsi@gmail.com</email>
      <telefonszam>066616666666</telefonszam>
    </vevo>
    <vevo vevoid="3">
      <nev>Oltatlan Péter</nev>
      <email>o.petya@gmail.com</email>
      <telefonszam>066356666666</telefonszam>
    </vevo>
  </vevok>
  <hamburgerek>

```

```
<hamburger hamid="0" hid="0">
  <nev>Legjobb Hamburger</nev>
  <ar>123456789</ar>
  <meret>Óriási</meret>
</hamburger>
<hamburger hamid="1" hid="0">
  <nev>Olcsóburger</nev>
  <ar>12</ar>
  <meret>Kicsi</meret>
</hamburger>
<hamburger hamid="2" hid="1">
  <nev>Finomburi</nev>
  <ar>1200</ar>
  <meret>Nagy</meret>
</hamburger>
<hamburger hamid="3" hid="2">
  <nev>Kreatívburger</nev>
  <ar>1100</ar>
  <meret>Kicsi</meret>
</hamburger>
<hamburger hamid="4" hid="3">
  <nev>Viktorburger</nev>
  <ar>50</ar>
  <meret>Kicsi</meret>
</hamburger>
<hamburger hamid="5" hid="3">
  <nev>Macsóburger</nev>
  <ar>5000</ar>
  <meret>Óriási</meret>
</hamburger>
</hamburgerek>
<hamburgerezok>
  <hamburgerezo hid="0">
    <nev>UTRYZÓ</nev>
    <telefonszam>06666666667</telefonszam>
    <cim>
      <varos>Miskolc</varos>
      <utca>Teszt út</utca>
      <hsz>43</hsz>
    </cim>
  </hamburgerezo>
  <hamburgerezo hid="1">
    <nev>DERIVÁLÓ HAMBIZÓ</nev>
    <telefonszam>06666666660</telefonszam>
    <cim>
      <varos>Budapest</varos>
      <utca>Tesztelők útja</utca>
      <hsz>41</hsz>
    </cim>
  </hamburgerezo>
</hamburgerezok>
```

```

        </cim>
    </hamburgerezo>
    <hamburgerezo hid="2">
        <nev>FALD FEL EDELÉNYT</nev>
        <telefonszam>06666626660</telefonszam>
        <cim>
            <varos>Edelény</varos>
            <utca>Tesla utca</utca>
            <hsz>1</hsz>
        </cim>
    </hamburgerezo>
    <hamburgerezo hid="3">
        <nev>MÁTÉKASOHANEMJÖNSEHOVA BURGEREZŐ</nev>
        <telefonszam>06646626660</telefonszam>
        <cim>
            <varos>Edelény</varos>
            <utca>Tél út</utca>
            <hsz>4</hsz>
        </cim>
    </hamburgerezo>
</hamburgerezok>

<futarok>
    <futar futid="0" hid="0">
        <nev>Téglás Máté</nev>
        <telefonszam>06666666661</telefonszam>
        <fizetes>940000</fizetes>
    </futar>
    <futar futid="1" hid="1">
        <nev>Oláh Renátó</nev>
        <telefonszam>06266666661</telefonszam>
        <fizetes>40000</fizetes>
    </futar>
    <futar futid="2" hid="2">
        <nev>Bubenkó Zsolt</nev>
        <telefonszam>06266663661</telefonszam>
        <fizetes>140000</fizetes>
    </futar>
    <futar futid="3" hid="3">
        <nev>Lakatos Trump</nev>
        <telefonszam>06269663661</telefonszam>
        <fizetes>1400000</fizetes>
    </futar>
</futarok>
<rendelesek>
    <rendeles vevoid="0" hamid="0">
        <datum>2019</datum>
    </rendeles>

```

```

    <rendeles vevoid="0" hamid="1">
      <datum>2020</datum>
    </rendeles>
    <rendeles vevoid="1" hamid="3">
      <datum>2013</datum>
    </rendeles>
    <rendeles vevoid="1" hamid="5">
      <datum>2014</datum>
    </rendeles>
    <rendeles vevoid="1" hamid="1">
      <datum>2015</datum>
    </rendeles>
    <rendeles vevoid="2" hamid="3">
      <datum>2019</datum>
    </rendeles>
    <rendeles vevoid="3" hamid="2">
      <datum>2020</datum>
    </rendeles>
    <rendeles vevoid="3" hamid="1">
      <datum>2013</datum>
    </rendeles>
  </rendelesek>
</nyilvantartas>

```

1d) XMSchema kód(validált):

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="beadando" xmlns="beadando"
elementFormDefault="qualified">
  <xs:element name="nyilvantartas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="vevok">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded"
name="vevo">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="nev"
type="xs:string"></xs:element>
                    <xs:element name="email"
type="xs:string"></xs:element>
                    <xs:element
name="telefonszam" type="xs:unsignedLong"></xs:element>
                  </xs:sequence>

```

```

                                <xs:attribute name="vevoid"
type="xs:unsignedByte" use="required"></xs:attribute>
                                </xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                <xs:element name="hamburgerek">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element maxOccurs="unbounded"
name="hamburger">
                                                <xs:complexType>
                                                    <xs:sequence>
                                                        <xs:element name="nev"
type="xs:string"></xs:element>
                                                        <xs:element name="ar"
type="xs:unsignedInt"></xs:element>
                                                        <xs:element name="meret"
type="xs:string"></xs:element>
                                                    </xs:sequence>
                                                    <xs:attribute name="hamid"
type="xs:unsignedByte" use="required"></xs:attribute>
                                                    <xs:attribute name="hid"
type="xs:unsignedByte" use="required"></xs:attribute>
                                                </xs:complexType>
                                            </xs:element>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                                <xs:element name="hamburgerezok">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element maxOccurs="unbounded"
name="hamburgerezo">
                                                <xs:complexType>
                                                    <xs:sequence>
                                                        <xs:element name="nev"
type="xs:string"></xs:element>
                                                        <xs:element
name="telefonszam" type="xs:unsignedLong"></xs:element>
                                                        <xs:element name="cim">
                                                            <xs:complexType>
                                                                <xs:sequence>
                                                                    <xs:element
name="varos" type="xs:string"></xs:element>
                                                                    <xs:element
name="utca" type="xs:string"></xs:element>
                                                                </xs:sequence>
                                                            </xs:complexType>
                                                        </xs:element>
                                                    </xs:sequence>
                                                </xs:complexType>
                                            </xs:element>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```

```

<xs:element
name="hsz" type="xs:unsignedByte"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="hid"
type="xs:unsignedByte" use="required"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="futarok">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded"
name="futar">
<xs:complexType>
<xs:sequence>
<xs:element name="nev"
type="xs:string"></xs:element>
<xs:element
name="telefonszam" type="xs:unsignedLong"></xs:element>
<xs:element name="fizetes"
type="xs:unsignedInt"></xs:element>
</xs:sequence>
<xs:attribute name="futid"
type="xs:unsignedByte" use="required"></xs:attribute>
<xs:attribute name="hid"
type="xs:unsignedByte" use="required"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rendelesek">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded"
name="rendeles">
<xs:complexType>
<xs:sequence>
<xs:element name="datum"
type="xs:unsignedShort"></xs:element>
</xs:sequence>
<xs:attribute name="vevoid"
type="xs:unsignedByte" use="required"></xs:attribute>

```



```

                                <xs:attribute name="hamid"
type="xs:unsignedByte" use="required"></xs:attribute>
                                </xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
</xs:schema>

```

2.feladat

2a) Adatolvasás

```

package hu.domparse.cieyie;

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.*;
import org.xml.sax.SAXException;

public class DOMReadCIEYIE {
    public static void main(String[] args) {
        try{
            DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document doc = db.parse(new File("XMLCIEYIE.xml")); //Fájl
kiválasztása, hogy mit szeretnénk olvasni
            doc.getDocumentElement().normalize();

            Element rootElement = doc.getDocumentElement(); //Kiválasztjuk a
gyökér elemet
            System.out.println("Gyökér elem: " + rootElement.getNodeName());
            NodeList childNodes = rootElement.getChildNodes(); //A gyökér
elemnek a gyerekeit eltároljuk

```

```

        for(int i=0; i<childNodes.getLength(); i++) { //Végigjárjuk az eltárolt
csomópontokat
            Node node = childNodes.item(i); //Kiválasztjuk az i.
csomópontot
            if(node.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element)node;
                Node actualNode = element.getFirstChild();
                while(actualNode != null) {
                    if(actualNode.getNodeType() ==
Node.ELEMENT_NODE) {
                        Element actualElement =
(Element)actualNode;
                        System.out.println(" " +
actualElement.getNodeName() + ": " + actualElement.getTextContent()); //Kíírjuk az aktuális
elem adatait
                    }
                    actualNode=actualNode.getNextSibling();
                }
                System.out.println();
            }
        }

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

```

2b) Adatmódosítás

```

package hu.domparse.cieyie;

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 DOMModifyCIEYIE {
    public static void main(String[] args) {
        try{
            DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document doc = db.parse(new File("XMLCIEYIE.xml")); //Fájl
kiválasztása, hogy mit szeretnénk olvasni

            //FUTÁR NEVEK, FIZETÉS MEGVÁLTOZTATÁSA
            Node futarok = doc.getElementsByTagName("futarok").item(0);
//Kiválasztjuk a futárok csomópontot
            NodeList flist = futarok.getChildNodes();
            System.out.println("-----[ FUTÁROK ]-----");
            for (int i = 0; i < flist.getLength(); i++) {
                Node node = flist.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) node;
                    System.out.println("Név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());

                    if(eElement.getElementsByTagName("nev").item(0).getTextContent().equals("Téglás Máté"))
                    {

                        eElement.getElementsByTagName("nev").item(0).setTextContent("Répás Réka");
                        System.out.println("-->Új név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
                    }

                    if(eElement.getElementsByTagName("nev").item(0).getTextContent().equals("Bubenkó
Zsolt")) {

                        eElement.getElementsByTagName("nev").item(0).setTextContent("Új Zsolt");
                        System.out.println("-->Új név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
                    }

                    if(eElement.getElementsByTagName("fizetes").item(0).getTextContent().equals("40000")) {

                        eElement.getElementsByTagName("fizetes").item(0).setTextContent("400000");
                        System.out.println("-->Új fizetés : " +
eElement.getElementsByTagName("fizetes").item(0).getTextContent()+"", nála:
"+eElement.getElementsByTagName("nev").item(0).getTextContent());
                    }
                }
            }
        }
    }
}

```

```

        System.out.println("-----[ FUTÁROK ]-----");
        //VEVŐ NÉV, EMAIL MEGVÁLTOZTATÁS
        System.out.println("-----[ VEVŐK ]-----");
        Node vevok = doc.getElementsByTagName("vevok").item(0);
//Kiválasztjuk a vevők csomópontot
        NodeList vlist = vevok.getChildNodes();

        for (int i = 0; i < vlist.getLength(); i++) {
            Node node = vlist.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element eElement = (Element) node;
                System.out.println("Név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());

                if(eElement.getElementsByTagName("nev").item(0).getTextContent().equals("XMLTeszt
Elek")) {

                    eElement.getElementsByTagName("nev").item(0).setTextContent("Hiteles Henrik");
                    System.out.println("-->Új név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
                }

                if(eElement.getElementsByTagName("email").item(0).getTextContent().equals("l.lajcsi@gma
il.com")) {

                    eElement.getElementsByTagName("email").item(0).setTextContent("l.lajoska32523@gmail.
com");
                    System.out.println("-->Új email : " +
eElement.getElementsByTagName("email").item(0).getTextContent());
                }
            }
        }

        System.out.println("-----[ VEVŐK ]-----");
        Element rootElement = doc.getDocumentElement(); //Kiválasztjuk a
gyökér elemet
        System.out.println("Gyökér elem: " + rootElement.getNodeName());
        NodeList childNodes = rootElement.getChildNodes(); //A gyökér
elemnek a gyerekeit eltároljuk

        for(int i=0; i<childNodes.getLength(); i++) { //Végigjárjuk az eltárolt
csomópontokat
            Node node = childNodes.item(i); //Kiválasztjuk az i.
csomópontot
            if(node.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element)node;
                Node actualNode = element.getFirstChild();
                while(actualNode != null) {

```

```
        if(actualNode.getNodeType() ==  
Node.ELEMENT_NODE) {  
  
            Element actualElement =  
(Element)actualNode;  
  
            System.out.println(" " +  
actualElement.getNodeName() + ": " + actualElement.getTextContent()); //Kiírjuk az aktuális  
elem adatait  
  
        }  
        actualNode=actualNode.getNextSibling();  
    }  
    System.out.println();  
}  
}  
}  
  
} catch(ParserConfigurationException e){ //Hibakezelők  
    e.printStackTrace();  
} catch (SAXException e) {  
    e.printStackTrace();  
} catch (IOException e) {  
    e.printStackTrace();  
}  
}  
}
```