

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Készítette: Czíkó Tivadar

Neptunkód: O2IXLB

1. Feladat

Egyedek:

- Vonat
- Mozdonyvezető
- kalauz
- Menetrend
- Útvonal
- Megállót

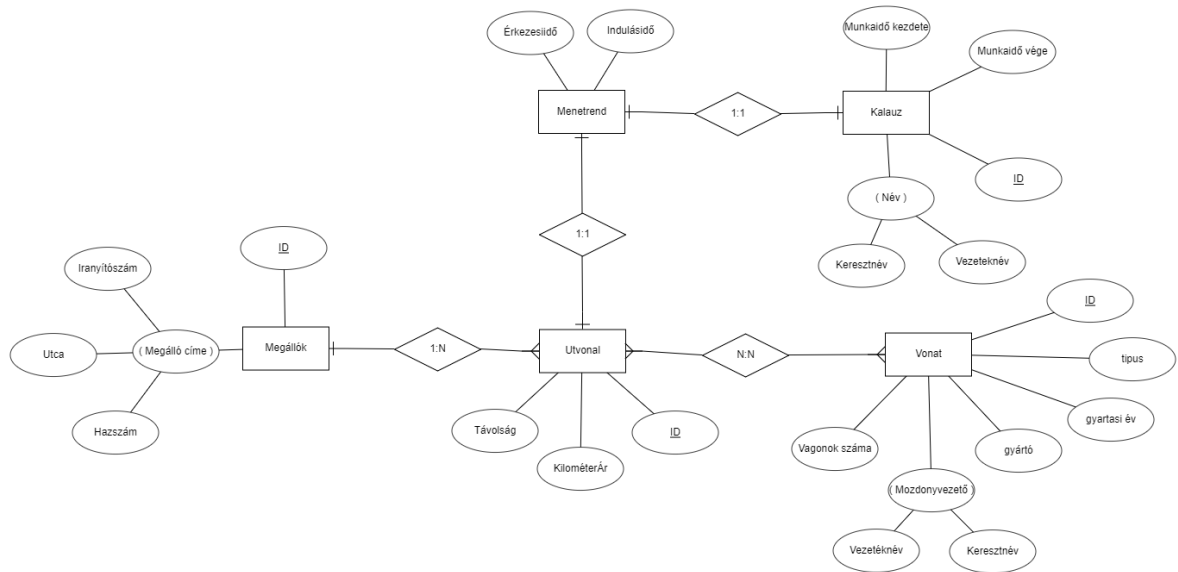
Tulajdonságaik:

- Egy vonat rendelkezik: Vonat azonosítóval, gyártóval, vagonokszámával, gyártási évvel és típussal. A vonat azonosító azonosítja a vonatott. A vonat mozdonyvezetővel is rendelkezik, aminek van neve, amiből leszarmazik a vezetéknév- és keresztnév.
- A kalauz egyed rendelkezik: Névvel, amiből le származik a keresztnév és a vezetéknév. A kalauznak is kell egy azonosító, amivel meg lehessen határozni melyik vonaton tartózkodik.
- A menetrend egyed rendelkezik: Indulási- és érkezési időponttal. A menetrendnek is van egy azonosítója, hogy tudjuk melyik menetrendhez, mely megállók tartoznak és milyen vonatok közlekednek az adott menetrend szerint.
- Az útvonal egyed rendelkezik: Vételdíjjal, távolsággal, kilométer árral és vonalszámmal, ami egy fajta azonosítóként működik.
- A megálló egyed rendelkezik: Megállónévvel és címmel. A megállónév mint azonosító van jelen.

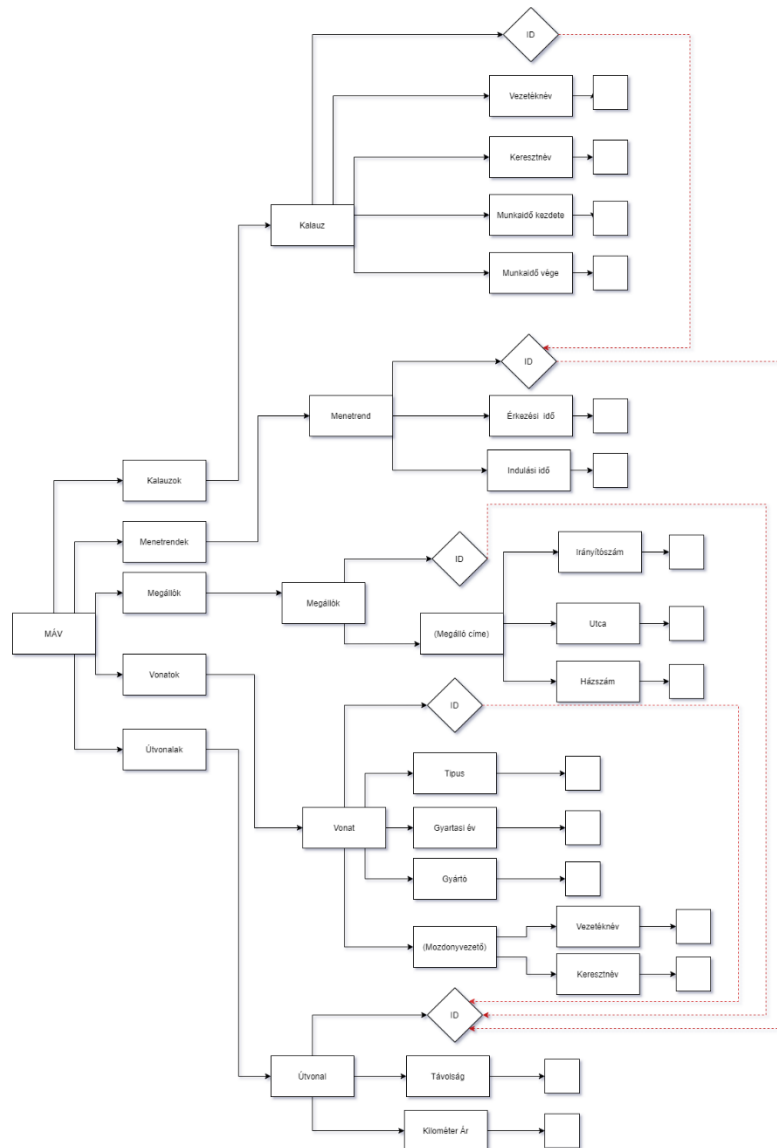
Kapcsolataik:

- Egy vonat mehet több útvonalon is és egy útvonalon mehet több vonat is. (M: N)
- Egy kalauzhoz általában több vonat tartozik, de egy vonathoz nem szokott több kalauz tartozni. (1: N)
- Egy menetrendhez általában egy útvonal tartozik, és egy útvonalhoz is egy menetrend. (1: 1)
- Egy menetrendhez általában több megálló tartozik és egy megállóhoz több menetrend tartozik. (1: N)
- Egy menetrendhez általában egy kalauz tartozik és úgy szintén egy kalauzhoz egy menetrend. (1: 1)

1a) Az adatbázis ER modell



1b) Az adatbázis konvertálása XDM modellre:



1c) Az XDM modell alaján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><mav>  
<mav xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="XSD_02IXLB.xsd"></mav>
```

```
  <vonat id="V01">  
    <vagonokszama>15</vagonokszama>  
    <mozdonyvezeto>  
      <keresztnev>Lazar</keresztnev>  
      <vezeteknev>Kovacs</vezeteknev>  
    </mozdonyvezeto>  
    <gyarto>Siemens Taurus</gyarto>  
    <gyartasiev>2007</gyartasiev>  
    <tipus>Teher</tipus>  
<mVezetokor>35</mVezetokor></vonat>
```

```
  <vonat id="V02">  
    <vagonokszama>6</vagonokszama>  
    <mozdonyvezeto>  
      <keresztnev>Tibor</keresztnev>  
      <vezeteknev>Schaffer</vezeteknev>  
    </mozdonyvezeto>  
    <gyarto>KISS</gyarto>  
    <gyartasiev>2013</gyartasiev>  
    <tipus>Személy vonat</tipus>  
</vonat>
```

```
  <vonat id="V03">  
    <vagonokszama>3</vagonokszama>  
    <mozdonyvezeto>  
      <keresztnev>Meastro</keresztnev>  
      <vezeteknev>Kipembe</vezeteknev>  
    </mozdonyvezeto>  
    <gyarto>FLirt</gyarto>  
    <gyartasiev>1995</gyartasiev>  
    <tipus>EC (Euro City)</tipus>  
</vonat>
```

```
  <kalauz id="k01">  
    <nev>  
      <keresztnev>Joe</keresztnev>  
      <vezeteknev>Johnson</vezeteknev>  
    </nev>  
    <munkaido_kezdetek>07:30</munkaido_kezdetek>
```

```
        <munkaido_vege>15:00</munkaido_vege>
    </kalauz>

    <kalauz id="k02">
        <nev>
            <keresztnev>Erik</keresztnev>
            <vezeteknev>Eriksen</vezeteknev>
        </nev>
        <munkaido_kezdete>08:30</munkaido_kezdete>
        <munkaido_vege>15:30</munkaido_vege>
    </kalauz>

    <kalauz id="k03">
        <nev>
            <keresztnev>Vlad'imir</keresztnev>
            <vezeteknev>Vlad'imirevich</vezeteknev>
        </nev>
        <munkaido_kezdete>07:50</munkaido_kezdete>
        <munkaido_vege>18:00</munkaido_vege>
    </kalauz>

    <megallok id="mk01"> <!--azonosító-->
        <megallo_cime>
            <iranyitoszam>3535</iranyitoszam>
            <hazszam>25/A</hazszam>
            <utca>Kando Kalman</utca>
        </megallo_cime>
    </megallok>

    <megallok id="mk02"> <!--azonosító-->
        <megallo_cime>
            <iranyitoszam>3738</iranyitoszam>
            <hazszam>65/C</hazszam>
            <utca>Petőfi Sándor</utca>
        </megallo_cime>
    </megallok>

    <megallok id="mk03"> <!--azonosító-->
        <megallo_cime>
            <iranyitoszam>3225</iranyitoszam>
            <hazszam>75</hazszam>
            <utca>Petőfi Sandor</utca>
        </megallo_cime>
    </megallok>

    <megallok id="mk04"> <!--azonosító-->
        <megallo_cime>
            <iranyitoszam>3132</iranyitoszam>
            <hazszam>47/E</hazszam>
```

```
        <utca>Berlini</utca>
    </megallo_cime>
</megallok>

<megallok id="mk05"> <!--azonosító-->
    <megallo_cime>
        <iranyitoszam>3354</iranyitoszam>
        <hazszam>46/H</hazszam>
        <utca>Bécsi</utca>
    </megallo_cime>
</megallok>
```

```
<megallok id="mk06"> <!--azonosító-->
    <megallo_cime>
        <iranyitoszam>3925</iranyitoszam>
        <hazszam>87/B</hazszam>
        <utca>Ballassi</utca>
    </megallo_cime>
</megallok>
```

```
<megallok id="mk07"> <!--azonosító-->
    <megallo_cime>
        <iranyitoszam>3865</iranyitoszam>
        <hazszam>17/D</hazszam>
        <utca>Mexikó Völgyi</utca>
    </megallo_cime>
</megallok>
```

```
<megallok id="mk08"> <!--azonosító-->
    <megallo_cime>
        <iranyitoszam>3672</iranyitoszam>
        <hazszam>76</hazszam>
        <utca>Eper</utca>
    </megallo_cime>
</megallok>
```

```
    <utvonali elsomegalló_id="mk1" utolsomegalló_id="mk2" utvonali_id="u01"
vonat_id="V01"> <!--Vonalszám -->
        <tavolság>75</tavolság>
        <kilométerár>200</kilométerár>
    </utvonali>
```

```
    <utvonali elsomegalló_id="mk1" utolsomegalló_id="mk3" utvonali_id="u02"
vonat_id="V02"> <!--Vonalszám -->
        <tavolság>65</tavolság>
```

```
<kilometerAr>170</kilometerAr>
</utvonal>

<utvonal elsomegallo_id="mk2" utolsomegallo_id="mk4" utvonal_id="u03"
vonat_id="V03"> <!--Vonalszam -->
    <tavolsag>55</tavolsag>
    <kilometerAr>120</kilometerAr>
</utvonal>

<utvonal elsomegallo_id="mk8" utolsomegallo_id="mk1" utvonal_id="u04"
vonat_id="V01"> <!--Vonalszam -->
    <tavolsag>175</tavolsag>
    <kilometerAr>1800</kilometerAr>
</utvonal>

<utvonal elsomegallo_id="mk7" utolsomegallo_id="mk6" utvonal_id="u05"
voant_id="V02"> <!--Vonalszam -->
    <tavolsag>125</tavolsag>
    <kilometerAr>1260</kilometerAr>
</utvonal>

<utvonal elsomegallo_id="mk3" utolsomegallo_id="mk5" utvonal_id="u06"
vonat_id="V03"> <!--Vonalszam -->
    <tavolsag>200</tavolsag>
    <kilometerAr>2000</kilometerAr>
</utvonal>

<menetrend kalauz_id="K01" utvonal_id="u01">
    <indulasiido>07:40</indulasiido>
    <erkezesiido>08:50</erkezesiido>
</menetrend>

<menetrend kalauz_id="K01" utvonal_id="u02">
    <indulasiido>08:40</indulasiido>
    <erkezesiido>10:40</erkezesiido>
</menetrend>

<menetrend kalauz_id="K01" utvonal_id="u03">
    <indulasiido>08:10</indulasiido>
    <erkezesiido>08:50</erkezesiido>
</menetrend>

<menetrend kalauz_id="K01" utvonal_id="u04">
    <indulasiido>11:40</indulasiido>
    <erkezesiido>13:50</erkezesiido>
</menetrend>

<menetrend kalauz_id="K01" utvonal_id="u05">
```

```

        <indulasiido>12:40</indulasiido>
        <erkezesiido>13:55</erkezesiido>
    </menetrend>

    <menetrend kalauz_id="K01" utvonal_id="u06">
        <indulasiido>15:40</indulasiido>
        <erkezesiido>17:00</erkezesiido>
    </menetrend>
</mav>

```

1d) Az XML dokumentum alapján XML schema készítése:

```

<?xml version="1.0" encoding="utf-8"?>
<!-- MAV's xml Schema -->
<xs:schema attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="mav">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded"
                    name="vonat">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="vagonokszama"
                                type="xs:unsignedByte" />
                            <xs:element name="mozdonyvezeto">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="keresztnev"
                                            type="xs:string" />
                                        <xs:element name="vezeteknev"
                                            type="xs:string" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="gyarto"
                                type="xs:string" />
                            <xs:element name="gyartasiev"
                                type="xs:unsignedShort" />
                            <xs:element name="tipus"
                                type="xs:string" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:attribute name="id"

```



```

                type="xs:string"
                use="required" />
            </xs:complexType>
        </xs:element>
        <xs:element maxOccurs="unbounded"
            name="kalauz">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="nev">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="keresztnev"
                                    type="xs:string" />
                                <xs:element name="vezeteknev"
                                    type="xs:string" />
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="munkaido_kezdetek"
                        type="xs:string" />
                    <xs:element name="munkaido_vege"
                        type="xs:string" />
                </xs:sequence>
                <xs:attribute name="id"
                    type="xs:string"
                    use="required" />
            </xs:complexType>
        </xs:element>
        <xs:element maxOccurs="unbounded"
            name="megallok">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="megallo_cime">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="iranyitoszam"
                                    type="xs:unsignedShort" />
                                <xs:element name="hazszam"
                                    type="xs:string" />
                                <xs:element name="utca"
                                    type="xs:string" />
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="id"
                    type="xs:string"
                    use="required" />
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="id"
        type="xs:string"
        use="required" />
</xs:complexType>

```

```

</xs:element>
<xs:element maxOccurs="unbounded"
            name="utvonal">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="tavolsag"
                  type="xs:unsignedByte" />
      <xs:element name="kilometerAr"
                  type="xs:unsignedShort" />
    </xs:sequence>
    <xs:attribute name="utvonal_id"
                  type="xs:string"
                  use="required" />
    <xs:attribute name="elsomegallo_id"
                  type="xs:string"
                  use="required" />
    <xs:attribute name="utolsomegallo"
                  type="xs:string"
                  use="required" />
    <xs:attribute name="vonat_id"
                  type="xs:string"
                  use="optional" />
    <xs:attribute name="voant_id"
                  type="xs:string"
                  use="optional" />
  </xs:complexType>
</xs:element>
<xs:element maxOccurs="unbounded"
            name="menetrend">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="indulasiido"
                  type="xs:string" />
      <xs:element name="erkezesiido"
                  type="xs:string" />
    </xs:sequence>
    <xs:attribute name="utvonal_id"
                  type="xs:string"
                  use="required" />
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<!--ID-k meg szoritasa -->
<xs:unique name="vonat_id">
  <xs:selector xpath="."/>
  <xs:field xpath="@vonat_id" />
</xs:unique>

```

```

    <xs:keyref name="vona_idref" refer="vonat_id">
        <xs:selector xpath="//vonat" />
        <xs:field xpath="@id" />
    </xs:keyref>

    <xs:key name="megallok_id">
        <xs:selector xpath="//utvonat" />
        <xs:field xpath="@megallok_id" />
    </xs:key>
    <xs:keyref name="megallok_idref" refer="megallok_id">
        <xs:selector xpath="//megallok" />
        <xs:field xpath="@id" />
    </xs:keyref>

    <xs:key name="menetrend_id">
        <xs:selector xpath="//utvonat" />
        <xs:field xpath="@menetrend_id" />
    </xs:key>
    <xs:keyref name="menetrend_idref" refer="menetrend_id">
        <xs:selector xpath="//menetrend" />
        <xs:field xpath="@id" />
    </xs:keyref>

    <xs:key name="kalauz_id">
        <xs:selector xpath="//menetrend" />
        <xs:field xpath="@kalauz_Id" />
    </xs:key>
    <xs:keyref name="kalauz_idref" refer="kalauz_id">
        <xs:selector xpath="//kalauz" />
        <xs:field xpath="@id" />
    </xs:keyref>
</xs:element>

<!--Egyedi komplextipusok -->

<xs:complexType name="kalauz">
    <xs:sequence>
        <xs:element name="vezeteknev" type="xs:string" />
        <xs:element name="keresztnev" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:int" use="required" />
</xs:complexType>

<xs:complexType name="megallok">
    <xs:sequence>
        <xs:element name="iranyitoszam" type="xs:int"/>
        <xs:element name="utca" type="xs:string" />
        <xs:element name="hazszam" type="xs:string" />
    </xs:sequence>

```

```

    <xs:attribute name="id" type="xs:int" use="required" />
</xs:complexType>

<xs:complexType name="utvonal">
  <xs:sequence>
    <xs:element name="tavlosag" type="xs:int" />
    <xs:element name="kilometerAr" type="xs:decimal" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required" />
  <xs:attribute name="meneterndId" type="xs:int" use="required" />
</xs:complexType>

</xs:schema>

```

2. Feladat

A feladatban egy DOM program Készítése az XML dokumentum adatainak adminisztrálása alapján:

2a) Adatolvasás

```

package hu.domparse.o2ixlb;

import java.io.File;

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

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

public class DOMReadO2IXLB {

    // Forras: https://www.javatpoint.com/how-to-read-xml-file-in-java

    public static void main(String[] args) {
        try {
            File file = new File(
                "D:\\UniversityOfMiskolc\\University\\2021_22_1\\XML\\O2IXLB_FelevesBeadando\\O2IXLBvonat.xml"); // XML

            // file

```

```

// forrasa
DocumentBuilder documentBuilder =
DocumentBuilderFactory.newInstance().newDocumentBuilder();
Document document = documentBuilder.parse(file);
System.out.println("Root element: " +
document.getDocumentElement().getNodeName());
    if (document.hasChildNodes()) {
        printNodeList(document.getChildNodes());
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}

private static void printNodeList(NodeList nodeList) {
    for (int count = 0; count < nodeList.getLength(); count++) {
        Node elemNode = nodeList.item(count);
        if (elemNode.getNodeType() == Node.ELEMENT_NODE) {
// get node name and value
            System.out.println("\nNode Name =" + elemNode.getNodeName() +
" [OPEN]");
            System.out.println("Node Content =" +
elemNode.getTextContent());
            if (elemNode.hasAttributes()) {
                NamedNodeMap nodeMap = elemNode.getAttributes();
                for (int i = 0; i < nodeMap.getLength(); i++) {
                    Node node = nodeMap.item(i);
                    System.out.println("attr name : " +
node.getNodeName());
                    System.out.println("attr value : " +
node.getNodeValue());
                }
            }
            if (elemNode.hasChildNodes()) {
//recursive call if the node has child nodes
                printNodeList(elemNode.getChildNodes());
            }
            System.out.println("Node Name =" + elemNode.getNodeName() + "
[CLOSE]");
        }
    }
}
}
}

```

2b) Adatmódosítás

```
package hu.domparse.o2ixlb;

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

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

public class DOMModifyO2IXLB {

    // Forras:
    // https://examples.javacodegeeks.com/core-
    java/xml/parsers/documentbuilderfactory/modify-xml-file-in-java-using-dom-
    parser-example/

    public static final String xmlFilePath =
"D:\\UniversityOfMiskolc\\University\\2021_22_1\\XML\\O2IXLB_FelevesBeadando\\
O2IXLBvonat.xml";

    public static void main(String argv[]) {

        try {

            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();

            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

            Document document = documentBuilder.parse(xmlFilePath);

            // Get employee by tag name
            // use item(0) to get the first node with tage name "vonat"
```

```

Node employee = document.getElementsByTagName("vonat").item(0);

// update employee , set the id to 10
NamedNodeMap attribute = employee.getAttributes();
Node nodeAttr = attribute.getNamedItem("id");
nodeAttr.setTextContent("VV1");

// append a new node to the first vonat
Element age = document.createElement("mVezetokor");

age.appendChild(document.createTextNode("35"));

employee.appendChild(age);

// loop the vonat node and update vagonokszama, and delete a node
NodeList nodes = employee.getChildNodes();

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

    Node element = nodes.item(i);

    if ("vagonokszama".equals(element.getNodeName())) {
        element.setTextContent("15");
    }

    // remove keresztnev
    if ("keresztnev".equals(element.getNodeName())) {
        employee.removeChild(element);
    }

}

// write the DOM object to the file
TransformerFactory transformerFactory =
TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();
DOMSource domSource = new DOMSource(document);

StreamResult streamResult = new StreamResult(new
File(xmlFilePath));
transformer.transform(domSource, streamResult);

System.out.println("The XML File was ");

} catch (ParserConfigurationException pce) {
    pce.printStackTrace();
} catch (TransformerException tfe) {
    tfe.printStackTrace();
}

```

```

        } catch (IOException ioe) {
            ioe.printStackTrace();
        } catch (SAXException sae) {
            sae.printStackTrace();
        }
    }
}

```

2c) Adatlegkérés

```

package hu.domparse.o2ixlb;

import java.io.File;

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

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

public class DOMQueryO2IXLB {

    // Forras:
    https://www.tutorialspoint.com/java\_xml/java\_dom\_query\_document.htm

    public static void main(String argv[]) {

        try {
            File inputFile = new File(
                "D:\\UniversityOfMiskolc\\University\\2021_22_1\\XML\\O2IXLB_FelevesBeadando\\O2IXLBvonat.xml");
            DocumentBuilderFactory dbFactory =
                DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(inputFile);
            doc.getDocumentElement().normalize();
            System.out.print("Root element: ");
            System.out.println(doc.getDocumentElement().getNodeName());
            NodeList nList = doc.getElementsByTagName("vonat");
            NodeList mList = doc.getElementsByTagName("megallok");
            System.out.println("-----");

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

```



```

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

if (nNode.getNodeType() == Node.ELEMENT_NODE) {
    Element eElement = (Element) nNode;
    System.out.print(" ID : ");
    System.out.println(eElement.getAttribute("id"));
    NodeList vagonokszamaelement =
eElement.getElementsByTagName("vagonokszama");

    for (int count = 0; count <
vagonokszamaelement.getLength(); count++) {
        Node node1 = vagonokszamaelement.item(count);

        if (node1.getNodeType() == Node.ELEMENT_NODE) {
            Element vonatVagon = (Element) node1;
            System.out.print("Vagonok szama : ");
            System.out.println(vonatVagon.getTextContent());
            System.out.print(vonatVagon.getAttribute("type"));
        }
    }
}
if (nNode.getNodeType() == Node.ELEMENT_NODE) {
    Element eElement = (Element) nNode;
    eElement.getAttribute("id");
    NodeList nodeElement =
eElement.getElementsByTagName("keresztnev");

    for (int count = 0; count < nodeElement.getLength();
count++) {
        Node node1 = nodeElement.item(count);

        if (node1.getNodeType() == Node.ELEMENT_NODE) {
            Element keresztnevelement = (Element) node1;
            System.out.print("Mozdonyvezet :\n\tKeresztnev :
");

            System.out.println(keresztnevelement.getTextConen
t());

            System.out.print(keresztnevelement.getAttribute("t
ype"));
        }
    }
}
if (nNode.getNodeType() == Node.ELEMENT_NODE) {
    Element eElement = (Element) nNode;
    eElement.getAttribute("id");
    NodeList nodeElement =
eElement.getElementsByTagName("vezeteknev");

```

```

        for (int count = 0; count < nodeElement.getLength();
count++) {
            Node node1 = nodeElement.item(count);

            if (node1.getNodeType() == Node.ELEMENT_NODE) {
                Element vezetekenvelement = (Element) node1;
                System.out.print("\tVezeteknev : ");
                System.out.println(vezetekenvelement.getTextConte
nt());

                System.out.print(vezetekenvelement.getAttribute("
type"));
            }
        }
    }

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

    for (int temp = 0; temp < mList.getLength(); temp++) {
        Node nNode = mList.item(temp);
        System.out.println("\nCurrent Element :");
        System.out.print(nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) nNode;
            System.out.print(" ID : ");
            System.out.println(eElement.getAttribute("id"));
            NodeList iranyoszanelement =
eElement.getElementsByTagName("iranyitoszam");

            for (int count = 0; count < iranyoszanelement.getLength();
count++) {
                Node node1 = iranyoszanelement.item(count);

                if (node1.getNodeType() == node1.ELEMENT_NODE) {
                    Element iranyitoszamelement = (Element) node1;
                    System.out.print("Iranyitoszam : ");
                    System.out.println(iranyitoszamelement.getTextCont
ent());

                    System.out.print(iranyitoszamelement.getAttribute(
"type"));
                }
            }
        }
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) nNode;

```

```

        eElement.getAttribute("id");
        NodeList nodeElement =
eElement.getElementsByTagName("hazszam");

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

            Node node1 = nodeElement.item(count);

            if (node1.getNodeType() == Node.ELEMENT_NODE) {
                Element hazszamelement = (Element) node1;
                System.out.print("Hazszam : ");
                System.out.println(hazszamelement.getTextContent()
);

                System.out.print(hazszamelement.getAttribute("type
"));

            }

        }
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) nNode;
            eElement.getAttribute("id");
            NodeList nodeElement =
eElement.getElementsByTagName("utca");

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

                Node node1 = nodeElement.item(count);

                if (node1.getNodeType() == Node.ELEMENT_NODE) {
                    Element utcaelement = (Element) node1;
                    System.out.print("Utca : ");
                    System.out.println(utcaelement.getTextContent());
                    System.out.print(utcaelement.getAttribute("type"))
;

                }

            }

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

```

Outputok:

DOMmodify:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><mav>
  <vonat id="V01">
    <vonokszama>20</vonokszama>
    <mozdonyvezeto>
      <keresztnev>Lazar</keresztnev>
      <vezeteknev>Kovacs</vezeteknev>
    </mozdonyvezeto>
    <gyarto>Siemens Taurus</gyarto>
    <gyartasiev>2007</gyartasiev>
    <tipus>Teher</tipus>
  </vonat>

  <vonat id="V02">
    <vonokszama>6</vonokszama>
    <mozdonyvezeto>
      <keresztnev>Tibor</keresztnev>
      <vezeteknev>Schaffer</vezeteknev>
    </mozdonyvezeto>
    <gyarto>KISS</gyarto>
    <gyartasiev>2013</gyartasiev>
    <tipus>Személy vonat</tipus>
  </vonat>

  <vonat id="V03">
    <vonokszama>3</vonokszama>
    <mozdonyvezeto>
      <keresztnev>Meastro</keresztnev>
      <vezeteknev>Kipembe</vezeteknev>
    </mozdonyvezeto>
  </vonat>
</mav>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><mav>
  <vonat id="V01">
    <vonokszama>120</vonokszama>
    <mozdonyvezeto>
      <keresztnev>Lazar</keresztnev>
      <vezeteknev>Kovacs</vezeteknev>
    </mozdonyvezeto>
    <gyarto>Siemens Taurus</gyarto>
    <gyartasiev>2007</gyartasiev>
    <tipus>Teher</tipus>
  </vonat>

  <vonat id="V02">
    <vonokszama>6</vonokszama>
    <mozdonyvezeto>
      <keresztnev>Tibor</keresztnev>
      <vezeteknev>Schaffer</vezeteknev>
    </mozdonyvezeto>
    <gyarto>KISS</gyarto>
    <gyartasiev>2013</gyartasiev>
    <tipus>Személy vonat</tipus>
  </vonat>

  <vonat id="V03">
    <vonokszama>3</vonokszama>
    <mozdonyvezeto>
      <keresztnev>Meastro</keresztnev>
    </mozdonyvezeto>
  </vonat>
</mav>
```

Problems Javadoc Declaration Console x

terminated> DOMModify02IXLB [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (2021, dec. 7, 13:00:22 – 13:00:26)

The XML File was

DOMquery:

```
Problems Javadoc Declaration Declaration
<terminated> DOMQuery02IXLB [Java Application]
Root element: mav
-----
Current Element :
vonat ID : V01
Vagonok száma : 120
Mozdonyvezető :
  Keresztnev : Lazar
  Vezeteknev : Kovacs

Current Element :
vonat ID : V02
Vagonok száma : 6
Mozdonyvezető :
  Keresztnev : Tibor
  Vezeteknev : Schaffer

Current Element :
vonat ID : V03
Vagonok száma : 3
Mozdonyvezető :
  Keresztnev : Meastro
  Vezeteknev : Kipembe
-----
Current Element :
megallok ID : mk01
Iranyitoszam : 3535
Hazzsam : 25/A
Utca : Kando Kalman

Current Element :
megallok ID : mk02
Iranyitoszam : 3738
Hazzsam : 65/C
Utca : Petőfi Sándor
<

Problems Javadoc Declaration Declaration
<terminated> DOMQuery02IXLB [Java Application]
Utca : Petőfi Sándor

Current Element :
megallok ID : mk03
Iranyitoszam : 3225
Hazzsam : 75
Utca : Petőfi Sándor

Current Element :
megallok ID : mk04
Iranyitoszam : 3132
Hazzsam : 47/E
Utca : Berlini

Current Element :
megallok ID : mk05
Iranyitoszam : 3354
Hazzsam : 46/H
Utca : Bécsi

Current Element :
megallok ID : mk06
Iranyitoszam : 3925
Hazzsam : 87/B
Utca : Ballassi

Current Element :
megallok ID : mk07
Iranyitoszam : 3865
Hazzsam : 17/D
Utca : Mexikó Völgyi

Current Element :
megallok ID : mk08
Iranyitoszam : 3672
Hazzsam : 76
Utca : Eper
```

DOMread:

```
Root element: mav
Node Name =mav [OPEN]
Node Content =
    120
        Lazar
        Kovacs
    Siemens Taurus
    2007
    Teher
>35
    6
        Tibor
        Schaffer
    KISS
    2013
    Személy vonat
    3
        Meastro
        Kipembe
    FLirt
    1995
    EC (Euro City)
```

```
>35
attr name : id
attr value : WV1
Node Name =vagonokszama [OPEN]
Node Content =120
Node Name =vagonokszama [CLOSE]
Node Name =mozdonyvezeto [OPEN]
Node Content =
    Lazar
    Kovacs
Node Name =keresztnev [OPEN]
Node Content =Lazar
Node Name =keresztnev [CLOSE]
Node Name =vezeteknev [OPEN]
Node Content =Kovacs
Node Name =vezeteknev [CLOSE]
Node Name =mozdonyvezeto [CLOSE]
Node Name =gyarto [OPEN]
Node Content =Siemens Taurus
Node Name =gyarto [CLOSE]
Node Name =gyartasiev [OPEN]
Node Content =2007
Node Name =gyartasiev [CLOSE]
Node Name =tipus [OPEN]
Node Content =Teher
Node Name =tipus [CLOSE]
Node Name =mVezetokor [OPEN]
Node Content =35
Node Name =mVezetokor [CLOSE]
```

```
Node Content =
    12:40
    13:55
attr name : kalauz_id
attr value : K01
attr name : utvonal_id
attr value : u05
Node Name =indulasiido [OPEN]
Node Content =12:40
Node Name =indulasiido [CLOSE]
Node Name =erkezesiido [OPEN]
Node Content =13:55
Node Name =erkezesiido [CLOSE]
Node Name =menetrend [CLOSE]
Node Name =menetrend [OPEN]
Node Content =
    15:40
    17:00
attr name : kalauz_id
attr value : K01
attr name : utvonal_id
attr value : u06
Node Name =indulasiido [OPEN]
Node Content =15:40
Node Name =indulasiido [CLOSE]
Node Name =erkezesiido [OPEN]
Node Content =17:00
Node Name =erkezesiido [CLOSE]
Node Name =menetrend [CLOSE]
Node Name =mav [CLOSE]
```