

Jegyzőkönyv
Adatkezelés XML környezetben Féléves feladat
Donga Dániel
PIUMNI

Feladat rövid leírása:

Egy jármű nyilvántartó rendszer elkészítése, amely a járművek részletes adatait tárolja, továbbá a járművek tulajdonosaival, és azok adataival foglalkozik.

Továbbá a feladat foglalkozik a járművek motor típusával, és a gyárak tulajdonosával.

A tulajdonos és a jármű 1:N kapcsolat, mivel egy tulajdonosnak több járműve is lehet.

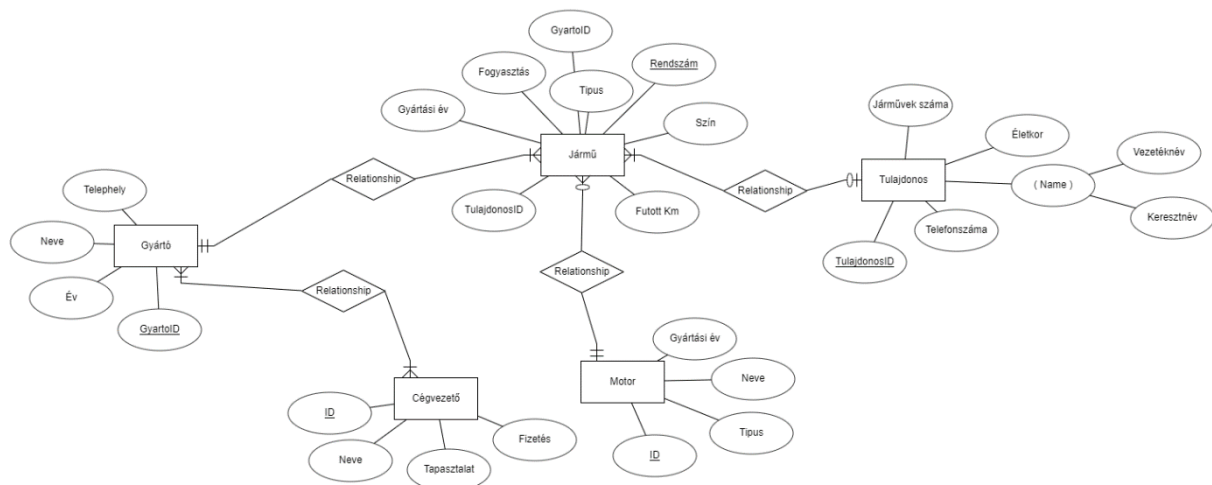
A különböző típusú járműveket gyártó cégeket is nyilvántartjuk, és azok részletes adatait.

Ez egy 1:N kapcsolat, hiszen egy gyártóhoz több jármű is tartozik.

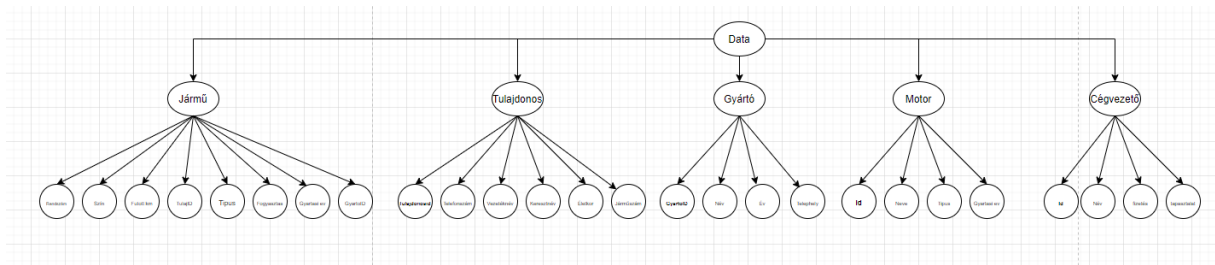
A Gyártó Tulajdonos N:M kapcsolatot valósít meg, hiszen egy gyártónak több tulajdonosnak is lehet, és egy tulajdonosnak több gyára.

A motor jármű pedig 1:N kapcsolat, mivel egyfajta motor több járműbe is szerelhetnek.

1.Er model :



2. XDM modellre történő konvertálás :



3. Xml dokumentum elkészítése

```
<data xsi:schemaLocation="XMLSchemaPIUMNI.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="data"><tulajdonosok><tulajdonos id="tulaj1">
```

```
<nev>
```

```
<vezeteknev>Nagy</vezeteknev>
```

```
<keresztnev>Pista</keresztnev>
```

```
</nev>
```

```
<eletkora>36</eletkora>
```

```
<jarmuvekszama>2</jarmuvekszama>
```

```
<telefonszama>3630457645</telefonszama>
```

```
</tulajdonos>
```

```
<tulajdonos id="tulaj2">
```

```
<nev>
```

```
<vezeteknev>Kis</vezeteknev>
```

```
<keresztnev>Bence</keresztnev>
```

```
</nev>
```

```
<eletkora>29</eletkora>
```

```
<jarmuvekszama>1</jarmuvekszama>
```

```
<telefonszama>062057535</telefonszama>
```

```
</tulajdonos><tulajdonos id="tulaj3">
```

```
<nev>
```

```
<vezeteknev>Varga</vezeteknev>
```

```
<keresztnev>János</keresztnev>
```

</nev>
<eletkora>44</eletkora>
<jarmuvekszama>2</jarmuvekszama>
<telefonszama>46124624</telefonszama>
</tulajdonos>
</tulajdonosok>
<jarmuvek><jarmu tulajid="tulaj2" gyartoid="gyarto2" rsz="ETR-235">
<fogyasztas>8l/100</fogyasztas>
<tipus>AudiR8</tipus>
<szine>Fehér</szine>
<futottkm>80,000km</futottkm>
<gyartasiev>2012</gyartasiev>
</jarmu>
<jarmu tulajid="tulaj1" gyartoid="gyarto3" rsz="RTT-745">
<fogyasztas>17l/100</fogyasztas>
<tipus>Chevrolette Camaro</tipus>
<szine>Sárga</szine>
<futottkm>65,000km</futottkm>
<gyartasiev>2014</gyartasiev>
</jarmu><jarmu tulajid="tulaj1" gyartoid="gyarto1" rsz="LJS-732">
<fogyasztas>15l/100</fogyasztas>
<tipus>Lamborghini hurrican</tipus>
<szine>Zöld</szine>
<futottkm>40,000km</futottkm>
<gyartasiev>2017</gyartasiev>
</jarmu>
<jarmu tulajid="tulaj3" gyartoid="gyarto1" rsz="FDG-345">
<fogyasztas>14l/100</fogyasztas>
<tipus>Lamborghini aventador</tipus>
<szine>Fehér</szine>
<futottkm>80,000km</futottkm>

<gyartasiev>2011</gyartasiev>

</jarmu>

<jarmu_tulajid="tulaj3" gyartoid="gyarto2" rsz="XCV-842">

<fogyasztas>11l/100</fogyasztas>

<tipus>Audi Rs6</tipus>

<szine>Fekete</szine>

<futottkm>120,000km</futottkm>

<gyartasiev>2018</gyartasiev>

</jarmu>

</jarmuvek>

<gyartok id="gyarto1">

<gyarto>

<nev>Lamborghini</nev>

<ev>1949</ev>

<telephely>U.K</telephely>

</gyarto>

<gyarto id="gyarto2">

<nev>Audi</nev>

<ev>1936</ev>

<telephely>Germany</telephely>

</gyarto>

<gyarto id="gyarto3">

<nev>Chevrolette</nev>

<ev>1966</ev>

<telephely>Usa</telephely>

</gyarto>

</gyartok>

<motorok>

<motor id="1">

```
<nev>Siemens</nev>
<gyartasiev>2002</gyartasiev>
<tipusa>Dizel</tipusa>
</motor>
<motor id="2">
<nev>Volvo</nev>
<gyartasiev>1988</gyartasiev>
<tipusa>Benzin</tipusa>
</motor>
<motor id="3">
<nev>Tesla</nev>
<gyartasiev>2011</gyartasiev>
<tipusa>Elektromos</tipusa></motor></motorok>
<cegvezetok>
<cegvezeto id="1"><nev>Béla</nev>
<tapasztalat>8 év</tapasztalat>
<fizetés>1,200,000 ft</fizetés>
</cegvezeto>
<cegvezeto id="2">
<nev>Pista</nev>
<tapasztalat>16 év</tapasztalat>
<fizetés>1,800,000 ft</fizetés>
</cegvezeto><cegvezeto id="3">
<nev>Béla</nev>
<tapasztalat>12 év</tapasztalat>
<fizetés>1,130,000 ft</fizetés>
</cegvezeto>
</cegvezetok>

</data>
```

3. XDM modell Készítése.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" targetNamespace="data" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="data">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="tulajdonosok">
          <xs:complexType>
            <xs:sequence>
<xs:element name="tulajdonos" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nev">
                <xs:complexType>
                  <xs:sequence>
<xs:element type="xs:string" name="vezeteknev"/>

                <xs:element type="xs:string" name="keresztnev"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>

              <xs:element type="xs:byte" name="eletkora"/>
              <xs:element type="xs:byte" name="jarmuvekszama"/>
              <xs:element type="xs:long" name="telefonszama"/>
            </xs:sequence>
            <xs:attribute type="xs:string" name="id" use="optional"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="jarmuvek">
    <xs:complexType>
      <xs:sequence>
<xs:element name="jarmu" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="fogyasztas"/>
              <xs:element type="xs:string" name="tipus"/>
              <xs:element type="xs:string" name="szine"/>
              <xs:element type="xs:string" name="f
              <xs:element type="xs:short" name="gyartasiev"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:attribute type="xs:string" name="rsz" use="optional"/>
        <xs:attribute type="xs:string" name="gyartoid" use="optional">
<xs:attribute type="xs:string" name="tulajid" use="optional"/>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="gyartok">
    <xs:complexType>
        <xs:sequence>
<xs:element name="gyarto" maxOccurs="unbounded" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element type="xs:string" name="nev"/>
                    <xs:element type="xs:short" name="ev"/>
                    <xs:element type="xs:string" name="telephely"/>
                </xs:sequence>
                <xs:attribute type="xs:string" name="id" use="optional"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute type="xs:string" name="id"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

4 Dom készítése (Olvasás) :

```

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;

```

```

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

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

public class DomReadPIUMNI {

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

        try {

            File xmlFile = new File("XMLPIUMNI.xml");

            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

            DocumentBuilder dBuilder = factory.newDocumentBuilder();

            Document doc = dBuilder.parse(xmlFile);

            doc.getDocumentElement().normalize();

            System.out.println("Donga Daniel beadando");

            Action(doc);

        } catch (ParserConfigurationException pce) {

            pce.printStackTrace();

        } catch (TransformerException tfe) {

            tfe.printStackTrace();

        } catch (IOException ioe) {

            ioe.printStackTrace();

        } catch (SAXException sae) {

            sae.printStackTrace();

        }

    }

}

```



```
public static void Action(Document doc) throws TransformerException {
```

```
    System.out.println("Press 1 to read data");
```

```
    System.out.println("Press 2 to edit data");
```

```
    int action = ReadCategory();
```

```
    switch (action) {
```

```
        case 1:
```

```
            Read(doc);
```

```
            break;
```

```
        case 2:
```

```
            Update(doc);
```

```
            break;
```

```
        default:
```

```
            Action(doc);
```

```
            break;
```

```
    }
```

```
}
```

```
public static int ReadCategory() {
```

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

```
    System.out.println("Adja meg a sorszamot:");
```

```
    int readCategory = scan.nextInt();
```

```
    return readCategory;
```

```
}
```

```
public static void Update(Document doc) throws TransformerException {
```

```
    System.out.println(" XML Editing.. \n");
```

```
    System.out.println("What do you want to edit ? ");
```

```
    System.out.println("1. Car\n2. Manufacture \n3. owner ");
```

```
int category = 0;
category = ReadCategory();
ShowElementUpdates(category, doc);
}
```

```
public static void Read(Document doc) {
    System.out.println("XML reading.. \n");
    System.out.println("What do you want to read? ");
    System.out.println("1. Car\\n2. Manufacture \\n3. owner");
    int category = 0;
    category = ReadCategory();
    ShowCategoryElements(category, doc);
}
```

```
public static void ShowCategoryElements(int category, Document doc) {
    switch (category) {
        case 1:
            ReadCar(doc);
            break;
        case 2:
            ReadGyar(doc);
            break;
        case 3:
            Readtulajdonos(doc);
            break;
        default:
            int newCategory = ReadCategory();
            ShowCategoryElements(newCategory, doc);
            break;
    }
}
```

```

    public static void ShowElementUpdates(int category, Document doc) throws
TransformerException {
        switch (category) {
        case 1:
            DOMModifyPIUMNI.UpdateCar(doc);
            break;
        case 2:
            DOMModifyPIUMNI.UpdateGyar(doc);
            break;
        case 3:
            DOMModifyPIUMNI.UpdateTulajdonos(doc);
            break;
        }
    }
}

```

```

public static void Readtulajdonos(Document doc) {

    NodeList nList = doc.getElementsByTagName("tulajdonos");

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

        Node nNode = nList.item(i);

        NodeList cList = nList.item(i).getChildNodes();

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

            Element element = (Element) nNode;

            String tulajid = element.getAttribute("id");

            Node node1 = element.getElementsByTagName("eletkora").item(0);

```

```

        String age = node1.getTextContent();

        Node node2 =
element.getElementsByTagName("jarmuvekszama").item(0);

        String jarmuvekszama = node2.getTextContent();

        Node node4 =
element.getElementsByTagName("telefonszama").item(0);

        String phonenumber = node4.getTextContent();

        String firstname = "";
        String lastname = "";

        for (int j = 0; j < cList.getLength(); j++) {

            Node cnode1 =
element.getElementsByTagName("vezeteknev").item(0);

            firstname = cnode1.getTextContent();

            Node cnode2 =
element.getElementsByTagName("keresztnev").item(0);

            lastname = cnode2.getTextContent();

        }

        System.out.println("Tulajdonos id:" + tulajid + "\tEletkor: " + age +
"\tJarmuvek szama: " + jarmuvekszama
                                + "\tTelefonszam : " + phonenumber + "\t Firstname
= : " + firstname + " lastname = " + lastname

                                );

    }

}

}

```

```

public static void ReadGyar(Document doc) {

    NodeList nList = doc.getElementsByTagName("gyarto");

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

        Node nNode = nList.item(i);

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

            Element element = (Element) nNode;

            String gyartoid = element.getAttribute("id");

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

            Node node2 = element.getElementsByTagName("ev").item(0);
            String year = node2.getTextContent();

            Node node3 = element.getElementsByTagName("telephely").item(0);
            String telephely = node3.getTextContent();

            System.out.println("Gyarto id:" + gyartoid + "\tNev: " + name + "\tÉv: " + year
                               + "\tTelephely: " + telephely);

        }

    }

}

public static void ReadCar(Document doc) {

```

```
NodeList nList = doc.getElementsByTagName("jarmu");

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

    Node nNode = nList.item(i);

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

        Element element = (Element) nNode;

        String rsz = element.getAttribute("rsz");
        String tulajid = element.getAttribute("tulajid");
        String gyartoid = element.getAttribute("gyartoid");

        Node node1 =
element.getElementsByTagName("fogyasztas").item(0);
        String fogyasztas = node1.getTextContent();

        Node node2 = element.getElementsByTagName("tipus").item(0);
        String type = node2.getTextContent();

        Node node3 = element.getElementsByTagName("szine").item(0);
        String color = node3.getTextContent();

        Node node4 = element.getElementsByTagName("futottkm").item(0);
        String futottkm = node4.getTextContent();

        Node node5 =
element.getElementsByTagName("gyartasiev").item(0);
        String gyartasiev = node5.getTextContent();
```

```

        System.out.println(" Rendszam : " + rsz + "\tTulajd id: " + tulajid + "\t"
gyarto id id: " + gyartoid
        + "\tfogyasztas: " + fogyasztas + "\ttipus: " + type +
"\tSzine: "
        + color + "\tFutott km : " + futottkm + "\tGyartasi ev:
" + gyartasiev);

    }

}

}

}

```

4.Dom készítése (Szerkeztés):

```

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

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

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;

public class DOMModifyPIUMNI {

    public static String ReadId() {
        Scanner sc = new Scanner(System.in);
        System.out.print("\nid:");
        String id = sc.nextLine();
        return id;
    }

    public static void CreateXML(Document doc) throws TransformerException {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(new File("XMLPIUMNI.updated.xml"));
        transformer.transform(source, result);
    }

    public static void UpdateCar(Document doc) throws TransformerException {

        System.out.println("\nMelyik járművet szeretn mdostani?\n");

        DomReadPIUMNI.ReadCar(doc);

        String id = ReadId();

        Scanner sc = new Scanner(System.in);
        System.out.print("Jármű fogyasztása: ");
        String fogyasztas = sc.nextLine();
        System.out.print("Jármű típusa: ");
    }
}

```



```

String type = sc.nextLine();
System.out.print("Jármű színe: ");
String color = sc.nextLine();
System.out.print("Jármű futott km: ");
String futottkm = sc.nextLine();
System.out.print("Gyártási éve : ");
String year = sc.nextLine();

UpdateCarById(doc, id, fogyasztas, type, color, futottkm, year);
}

```

```

public static void UpdateCarById(Document doc, String id, String fogyasztas, String type,
String color, String futottkm, String gyartasiev) throws TransformerException {

```

```

    NodeList nList = doc.getElementsByTagName("jarmu");

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

        Node nNode = nList.item(i);

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

            Element element = (Element) nNode;

            String rsz = element.getAttribute("rsz");

            if (rsz.equals(id)) {

                String tulajid = element.getAttribute("tulajid");
                String gyartoid = element.getAttribute("gyartoid");

                Node node1 =
element.getElementsByTagName("fogyasztas").item(0);

```

```

        node1.setTextContent(fogyasztas);

        Node node2 =
element.getElementsByTagName("tipus").item(0);
        node2.setTextContent(type);

        Node node3 =
element.getElementsByTagName("szine").item(0);
        node3.setTextContent(color);

        Node node4 =
element.getElementsByTagName("futottkm").item(0);
        node4.setTextContent(futottkm);

        Node node5 =
element.getElementsByTagName("gyartasiev").item(0);
        node5.setTextContent(gyartasiev);

        System.out.println(" Rendszam :" + rsz + "\tTulajID id:" +
tulajid + "\tGyarto id:"
                                + gyartoid + "\tFogyasztas: " +
node1.getTextContent() + "\tTipus: " + node2.getTextContent()
                                + "\tSzine: " + node3.getTextContent() +
"\tfutottkm:" + node4.getTextContent() + "\tgyartasi ev: " + node5.getTextContent());

        System.out.println("A módosítás sikeresen megtörtént \n");

    }

}

CreateXML(doc);

}

public static void UpdateGyar(Document doc) throws TransformerException {

```

```
System.out.println("\nMelyik gyártót szeretné módosítani?\n");
```

```
DomReadPIUMNI.ReadGyar(doc);
```

```
String id = ReadId();
```

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

```
System.out.print("Név: ");
```

```
String name = sc.nextLine();
```

```
System.out.print("Év: ");
```

```
String year = sc.nextLine();
```

```
System.out.print("Telephely: ");
```

```
String telephely = sc.nextLine();
```

```
UpdategyarById(doc, id, name, year, telephely);
```

```
}
```

```
public static void UpdategyarById(Document doc, String id, String name, String year, String  
telephely) throws TransformerException {
```

```
    NodeList nList = doc.getElementsByTagName("gyarto");
```

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

```
        Node nNode = nList.item(i);
```

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

```
            Element element = (Element) nNode;
```

```
            String gyartoid = element.getAttribute("id");
```

```

        if (gyartoid.equals(id)) {

            Node node1 =
element.getElementsByTagName("nev").item(0);

            node1.setTextContent(name);

            Node node2 =
element.getElementsByTagName("ev").item(0);

            node2.setTextContent(year);

            Node node3 =
element.getElementsByTagName("telephely").item(0);

            node3.setTextContent(telephely);

            System.out.println("Szponzor id:" + gyartoid + "\tNev: " +
node1.getTextContent() + "\tev: " + node2.getTextContent()
                                + "\ttelephely: " + node3.getTextContent());

            System.out.println("\nModositas sikeresen megtortent\n");

        }
    }

    CreateXML(doc);
}

public static void UpdateTulajdonos(Document doc) throws TransformerException {

    System.out.println("\nMelyik tulajdonost szeretne modositani ?\n");

    DomReadPIUMNI.Readtulajdonos(doc);

    String id = ReadId();

```

```

Scanner sc = new Scanner(System.in);
System.out.print("Age : ");
String age = sc.nextLine();
System.out.print("jarmuvek szama : ");
String vehiclenumber = sc.nextLine();
System.out.print("Telefonszam : ");
String phonenumber = sc.nextLine();
System.out.print("Firstname: ");
String firstname = sc.nextLine();
System.out.print("Lastname: ");
String lastname = sc.nextLine();

UpdateTulajdonosById(doc, id, age, vehiclenumber, phonenumber, firstname,
lastname);

}

public static void UpdateTulajdonosById(Document doc, String id, String age, String
vehiclenumber, String phonenumber, String firstname, String lastname) throws TransformerException
{

    NodeList nList = doc.getElementsByTagName("tulajdonos");

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

        Node nNode = nList.item(i);

        NodeList cList = nList.item(i).getChildNodes();

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

            Element element = (Element) nNode;

```

```

        String tulajid = element.getAttribute("id");

        if (tulajid.equals(id)) {

            Node node1 =
element.getElementsByTagName("eletkora").item(0);

            node1.setTextContent(age);

            Node node2 =
element.getElementsByTagName("jarmuvekszama").item(0);

            node2.setTextContent(vehiclenumber);

            Node node3 =
element.getElementsByTagName("telefonszama").item(0);

            node3.setTextContent(phonenumber);

            for (int j = 0; j < cList.getLength(); j++) {

                Node cnode1 =
element.getElementsByTagName("vezeteknev").item(0);

                cnode1.setTextContent(firstname);

                Node cnode2 =
element.getElementsByTagName("keresztnev").item(0);

                cnode2.setTextContent(lastname);

            }

            System.out.println("Tulajdonos id:" + id + "\tEletkor: " +
node1.getTextContent() + "\tJarmuvek szama: " + node2.getTextContent()
+ "\ttelefonszam: " + node3.getTextContent()
+ "\tfirstname: " + firstname + " lastname = " + lastname);

```

```
System.out.println("\nModositas sikeresen megtortent \n");
```

```
}
```

```
}
```

```
}
```

```
CreateXML(doc);
```

```
}
```

```
}
```