

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Autós szerviz hálózat

Készítette: Szalai Márton

Neptunkód: DTKUG0

A feladat leírása:

A féléves feladatom egy cég autós szervízhálózatának működését és annak résztvevőit mutatja be.

Maga a hálózat összesen 6 egyedből áll: Cégjegyzék, Cég, Műhely, Raktár, Autó, valamint Tulajdonos. Először is a modell egy Cégjegyzék nevű egyeddel kezdődik, amelynek két tulajdonsága van: egy CégjegyzékID nevű elsődleges kulcsa, valamint egy Bejegyzett cégek tulajdonsága. Ez az egyed egy-több kapcsolatban áll a Céggel, vagyis egy cégjegyzékben több cég is szerepelhet.

Ezután következik a Cég egyed, amely már több különböző tulajdonsággal is rendelkezik. Egy Cégazonosító nevű elsődleges kulccsal, egy Név tulajdonsággal, illetve egy Székhely nevű többértékű tulajdonsággal, ezen belül található a Cím valamint a Telefonszám.

A Cég egyedet a Műhelyvezető kapcsolat köti össze a modell legfontosabb egyedével, a Műhellyel. A két egyed között egy-egy kapcsolat van, hiszen egy cég egy darab műhelyt üzemeltet. A Műhely egyednek az Mkód lesz az elsődleges kulcsa, továbbá rendelkezik egy Javítandó autó tulajdonsággal.

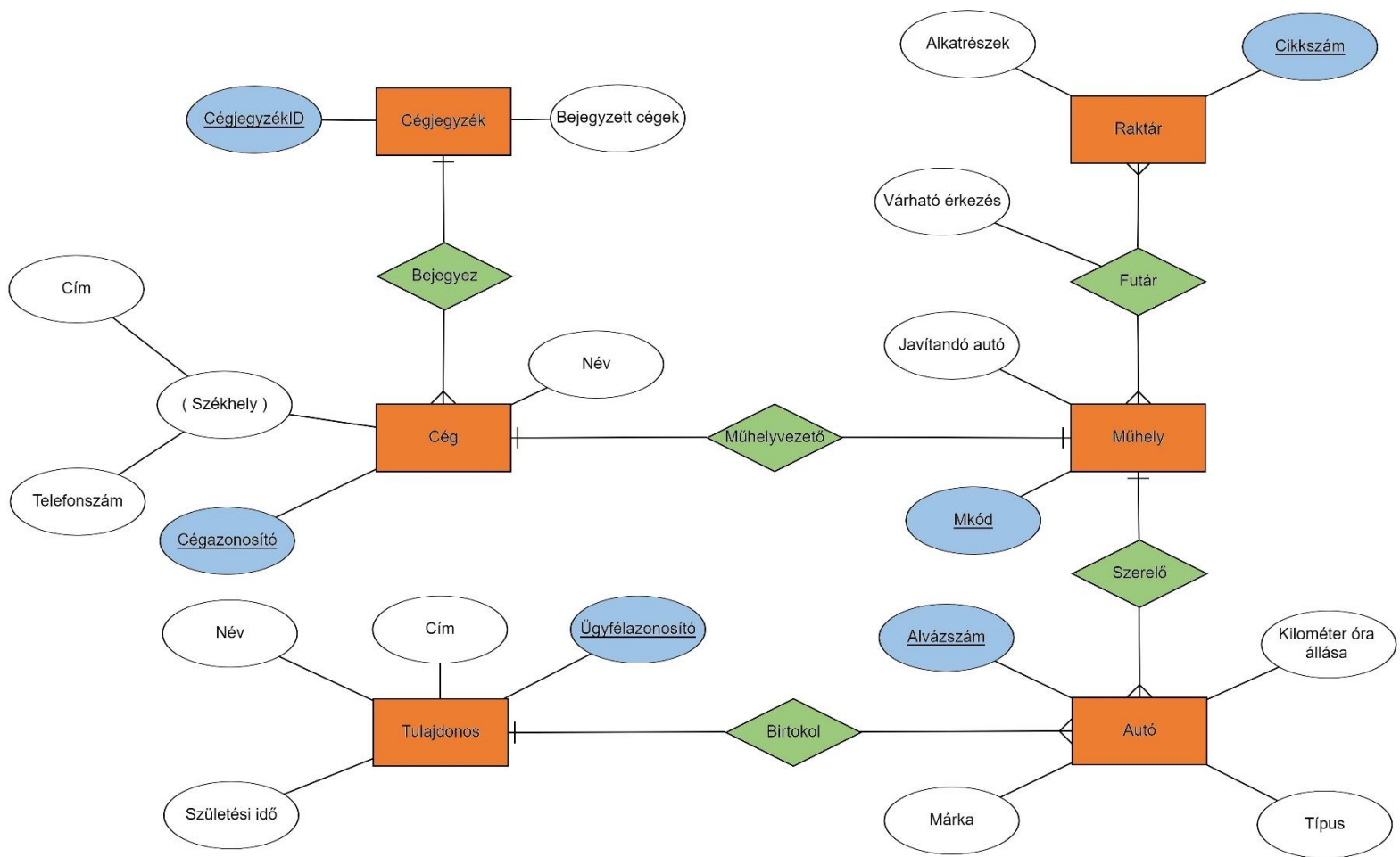
Ez az egyed két továbbival is kapcsolatban van, a Raktárral illetve az Autóval. A raktárnak összesen 2 tulajdonsága van: a Cikkszám, mint elsődleges kulcs, és az Alkatrészek. A Műhely és a Raktár között a Futár tartja a kapcsolatot, melynek van egy Várható érkezés tulajdonsága. A két egyed között több-több kapcsolat van, egy raktárhoz több műhely tartozhat, és ugyanez fordítva.

A Műhely az Autó egyed egy-több, Szerelő nevű kapcsolattal van összekötve. Az Autó egyed összesen 4 tulajdonsággal rendelkezik: egy Alvázszámmal, ez az elsődleges kulcs, valamint egy Kilométer óra állás, Márka és Típus tulajdonságokkal.

Végezetül, az ER modell utolsó egyede, a Tulajdonos, egy-több kapcsolattal van összekötve az Autóval, amelynek Birtokol a neve, hiszen egy tulajdonosnak több autója is lehet, viszont egy autónak csak egy tulajdonosa lehet. A Tulajdonos egyednek 5 tulajdonsága van, mégpedig a Név, a Cím, a Születési idő, egy Ügyfélazonosító nevű elsődleges kulcs.

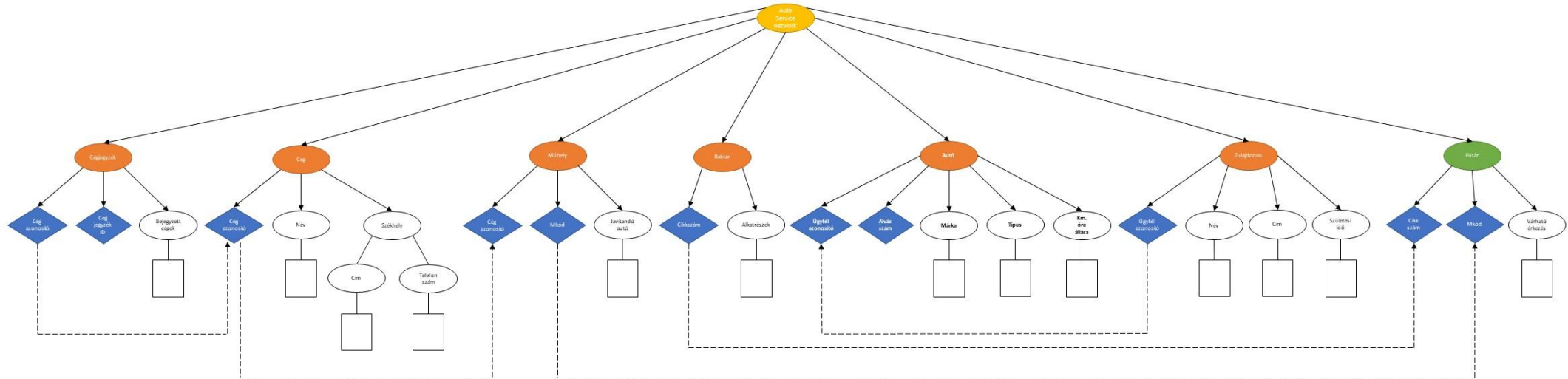
Ezek az egyedek és kapcsolatok építik fel a szervízhálózatot.

Az adatbázis ER modell



Az adatbázis konvertálása XDM modellre

A több-több kapcsolatból létrejön egy futár objektum, amely tartalmazza a Műhely, illetve a Raktár idegen kulcsait, valamint magának a kapcsolatnak a tulajdonságát.



Az XDM modell alapján XML dokumentum készítése

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<auto_nt_service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemadtktug0.xsd">

  <cegjegyzek cegid="1" cegjegyzekid="1">
    <bejegyzett_cegek>Stop'n go</bejegyzett_cegek>
  </cegjegyzek>
  <cegjegyzek cegid="2" cegjegyzekid="2">
    <bejegyzett_cegek>Nonstop service</bejegyzett_cegek>
  </cegjegyzek>
  <cegjegyzek cegid="3" cegjegyzekid="3">
    <bejegyzett_cegek>Bosch Car Service</bejegyzett_cegek>
  </cegjegyzek>

  <ceg cegid="1">
    <nev>Stop'n go</nev>
    <szekhely>
      <cim>Miskolc, Soltész Nagy Kálmán utca 36.</cim>
      <telefonszam>46-343-565</telefonszam>
    </szekhely>
  </ceg>
  <ceg cegid="2">
    <nev>Nonstop service</nev>
    <szekhely>
      <cim>Miskolc, Fonoda utca 36.</cim>
      <telefonszam>46-123-414</telefonszam>
    </szekhely>
  </ceg>
  <ceg cegid="3">
    <nev>Bosch Car Service</nev>
    <szekhely>
      <cim>Karcag, Tőkés utca 25.</cim>
      <telefonszam>53-555-100</telefonszam>
    </szekhely>
  </ceg>

  <muhely cegid="1" mid="1">
    <javitando_auto>Honda Accord</javitando_auto>
  </muhely>
  <muhely cegid="2" mid="2">
    <javitando_auto>Fiat Tipo</javitando_auto>
  </muhely>
  <muhely cegid="3" mid="3">
    <javitando_auto>Honda Civic</javitando_auto>
  </muhely>

  <raktar cikkszam="1">
    <alkatreszek>MAP sensor</alkatreszek>
  </raktar>
  <raktar cikkszam="2">
    <alkatreszek>ECU</alkatreszek>
  </raktar>
```

```
<raktar cikkszam="3">
  <alkatreszek>Lambda sensor</alkatreszek>
</raktar>

<auto alvazszam="JH4KA4660LC006262" ugyfelid="1">
  <marka>Honda</marka>
  <tipus>Accord</tipus>
  <km_ora_allasa>20000</km_ora_allasa>
</auto>
<auto alvazszam="J8DE5B16477903094" ugyfelid="2">
  <marka>Fiat</marka>
  <tipus>Tipo</tipus>
  <km_ora_allasa>98000</km_ora_allasa>
</auto>
<auto alvazszam="JH4KA2550HC005889" ugyfelid="3">
  <marka>Honda</marka>
  <tipus>Civic</tipus>
  <km_ora_allasa>110000</km_ora_allasa>
</auto>

<tulajdonos ugyfelid="1">
  <nev>Kiss Ramóna</nev>
  <cim>5300 Karcag, Déryné utca 10</cim>
  <szulido>1999.01.01</szulido>
</tulajdonos>
<tulajdonos ugyfelid="2">
  <nev>Tóth Fanni</nev>
  <cim>Kiskunfélegyháza, Napsugár út. 50</cim>
  <szulido>1987.10.10</szulido>
</tulajdonos>
<tulajdonos ugyfelid="3">
  <nev>Nagy Béla</nev>
  <cim>Szászhalombatta</cim>
  <szulido>1999.01.01</szulido>
</tulajdonos>

<futar cikkszam="1" mid="1">
  <varhato_erk>15:15</varhato_erk>
</futar>
<futar cikkszam="2" mid="2">
  <varhato_erk>10:00</varhato_erk>
</futar>
<futar cikkszam="3" mid="3">
  <varhato_erk>6:00</varhato_erk>
</futar>

</auto_nt_service>
```

Az XML dokumentum alapján XMLSchema készítése

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

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

  <xs:element name="auto_nt_service">
    <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="cegjegyzek" type="cegjegyzek_type"/>
          <xs:element name="ceg" type="ceg_type"/>
          <xs:element name="muhely" type="muhely_type"/>
          <xs:element name="raktar" type="raktar_type"/>
          <xs:element name="auto" type="auto_type"/>
          <xs:element name="tulajdonos" type="tulajdonos_type"/>
          <xs:element name="futar" type="futar_type"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>

    <xs:key name="cegjegyzekid_primaryKey">
      <xs:selector xpath="cegjegyzek"/>
      <xs:field xpath="@cegjegyzekid"/>
    </xs:key>
    <xs:keyref name="cegid_foreignKey" refer="cegid_primaryKey">
      <xs:selector xpath="cegjegyzek"/>
      <xs:field xpath="@cegid"/>
    </xs:keyref>
    <xs:key name="cegid_primaryKey">
      <xs:selector xpath="ceg"/>
      <xs:field xpath="@cegid"/>
    </xs:key>
    <xs:key name="mid_primaryKey">
      <xs:selector xpath="muhely"/>
      <xs:field xpath="@mid"/></xs:field>
    </xs:key>
    <xs:keyref name="cegid2_foreignKey" refer="cegid_primaryKey">
      <xs:selector xpath="muhely"/>
      <xs:field xpath="@cegid"/>
    </xs:keyref>
    <xs:key name="cikkszam_primaryKey">
      <xs:selector xpath="raktar"/>
      <xs:field xpath="@cikkszam"/>
    </xs:key>
    <xs:key name="alvazszam_primaryKey">
      <xs:selector xpath="auto"/>
      <xs:field xpath="@alvazszam"/>
    </xs:key>
    <xs:keyref name="ugyfelid_foreignKey" refer="ugyfelid_primaryKey">
      <xs:selector xpath="auto"/>
      <xs:field xpath="@ugyfelid"/>
    </xs:keyref>

  </xs:schema>
```

```

    <xs:key name="ugyfelid_primaryKey">
        <xs:selector xpath="tulajdonos"/>
        <xs:field xpath="@ugyfelid"/>
    </xs:key>
    <xs:keyref name="mid_foreignKey" refer="mid_primaryKey">
        <xs:selector xpath="futar"/>
        <xs:field xpath="@mid"/>
    </xs:keyref>
    <xs:keyref name="cikkszam_foreignKey" refer="cikkszam_primaryKey">
        <xs:selector xpath="futar"/>
        <xs:field xpath="@cikkszam"/>
    </xs:keyref>
</xs:element>

<xs:complexType name="cegjegyzek_type">
    <xs:sequence>
        <xs:element name="bejegyzett_cegek" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="cegid" type="xs:unsignedByte" use="required"/>
    <xs:attribute name="cegjegyzekid" type="xs:unsignedByte"
use="required"/>
</xs:complexType>
<xs:complexType name="ceg_type">
    <xs:sequence>
        <xs:element name="nev" type="xs:string"/>
        <xs:element name="szekhely" type="szekhely_type"/>
    </xs:sequence>
    <xs:attribute name="cegid" type="xs:unsignedByte" use="required"/>
</xs:complexType>
<xs:complexType name="muhely_type">
    <xs:sequence>
        <xs:element name="javitando_auto" type="javitando_auto_type"/>
    </xs:sequence>
    <xs:attribute name="cegid" type="xs:unsignedByte" use="required"/>
    <xs:attribute name="mid" type="xs:unsignedByte" use="required"/>
</xs:complexType>
<xs:complexType name="raktar_type">
    <xs:sequence>
        <xs:element name="alkatreszek" type="alkatreszek_type"/>
    </xs:sequence>
    <xs:attribute name="cikkszam" type="xs:unsignedByte" use="required"/>
</xs:complexType>
<xs:complexType name="auto_type">
    <xs:sequence>
        <xs:element name="marka" type="xs:string"/>
        <xs:element name="tipus" type="xs:string"/>
        <xs:element name="km_ora_allasa" type="km_ora_allasa_type"/>
    </xs:sequence>
    <xs:attribute name="ugyfelid" type="xs:unsignedByte" use="required"/>
    <xs:attribute name="alvazszam" type="xs:string" use="required"/>
</xs:complexType>

```



```

<xs:complexType name="tulajdonos_type">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="cim" type="xs:string"/>
    <xs:element name="szulido" type="szulido_type"/>
  </xs:sequence>
  <xs:attribute name="ugyfelid" type="xs:unsignedByte" use="required"/>
</xs:complexType>
<xs:complexType name="futar_type">
  <xs:sequence>
    <xs:element name="varhato_erk" type="varhato_erk_type"/>
  </xs:sequence>
  <xs:attribute name="mid" type="xs:unsignedByte" use="required"/>
  <xs:attribute name="cikkszam" type="xs:unsignedByte" use="required"/>
</xs:complexType>
<xs:complexType name="szekhely_type">
  <xs:sequence>
    <xs:element name="cim" type="xs:string"/>
    <xs:element name="telefonszam" type="telefonszam_type"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="javitando_auto_type">
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-zA-Z]+\s[a-zA-Z]+"\> <!-- "string string" -->
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="alkatreszek_type">
  <xs:restriction base="xs:string">
    <xs:minLength value="3" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="km_or_a_allasa_type">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="szulido_type">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{4}\.(0[1-9]|1[012])\.(0[1-9]|1[12])[0-9]/3[01]"/> <!-- YYYY.MM.DD -->
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="varhato_erk_type">
  <xs:restriction base="xs:string">
    <xs:pattern value="([0-9]|0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]"/> <!-- HH.MM -->
  </xs:restriction>
</xs:simpleType>

```

```
<xs:simpleType name="telefonszam_type">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{2}-\d{3}-\d{2}-\d{2}"/>
    <xs:pattern value="\d{2}-\d{3}-\d{3}"/>
  </xs:restriction>
</xs:simpleType>

</xs:schema>
```

DOM adatolvasás

Ezen osztály segítségével történik meg az XML dokumentum tartalmának olvasása és listázása a console-ra.

```
package hu.domparsedtkug0;

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 DOMReadDTKUG0 {

    public static void main(String[] args) {

        try {

            // XML fájl bekerese
            File xmlFile = new File("XMLdtkug0.xml");

            // Objektumfak eloallitasa a dokumentumbol
            DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();

            // XML dokumentum atadasa es egy DOM Document objektum
            Document doc = dBuilder.parse(xmlFile);

            // szomszedos es ures text node-ok eltavolitasara szolgal
            doc.getDocumentElement().normalize();

            ReadCegjegyzek(doc);
            ReadAuto(doc);
        } catch (ParserConfigurationException pce) {
            pce.printStackTrace();
        } catch (IOException ioe) {
            ioe.printStackTrace();
        } catch (SAXException sae) {
            sae.printStackTrace();
        }

    }

    // a fa struktura lehetove teszi hogy id alapjan kerdezzunk le
    // a legtöbb objektum rendelkezik leszarmazottal
    public static void ReadCegjegyzek(Document doc) {

        // cegjegyzek tag-el rendelkezo elemek lekerese
        NodeList nList = doc.getElementsByTagName("cegjegyzek");

        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i); // lista aktualis elemeinek lekerese
            Element element = (Element) node; // konvertalas elementekke
            // attributomok lekerese majd a definialt metodusok meghivasa
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                String cegid = element.getAttribute("cegid");
                String cegjegyzekid =
element.getAttribute("cegjegyzekid");
            }
        }
    }
}
```

```

        String bejegyzettCegek =
element.getElementsByTagName("bejegyzett_cegek").item(0).getTextContent();

        System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
        System.out.println("-----");
        System.out.println("Current Element : " +
node.getNodeName());

        System.out.println("Ceg azonosito : " + cegid);
        System.out.println("Cegjegyzek ID : " + cegjegyzekid);
        System.out.println("Bejegyzett cegek - " + (i + 1) +
".ceg: " + bejegyzettCegek);

        ReadCegById(doc, cegid);
    }
}

public static void ReadCegById(Document doc, String id) {
    // ceg tag-el rendelkezo elemek lekerese
    NodeList nList = doc.getElementsByTagName("ceg");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("cegid").equals(id)) {
                String cegid = element.getAttribute("cegid");
                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();
                String cim =
element.getElementsByTagName("cim").item(0).getTextContent();
                String tel =
element.getElementsByTagName("telefonszam").item(0).getTextContent();

                System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
                System.out.println("-----");
                System.out.println("Current Element : " +
node.getNodeName());

                System.out.println("Ceg azonosito : " + cegid);
                System.out.println("Cegnev : " + nev);
                System.out.println("Szekhely : " + cim +
"\nTelefonszam: " + tel);

                ReadMuhelyById(doc, cegid);
            }
        }
    }
}

// Erre a metodusra azert van szukseg, hogy amikor a DOMQuery osztaly meghivja
a
// ReadCeg-et, akkor csak a ceg adatai jelenjenek meg
public static void ReadCegByIdWithoutMuhely(Document doc, String id) {
    // ceg tag-el rendelkezo elemek lekerese
    NodeList nList = doc.getElementsByTagName("ceg");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("cegid").equals(id)) {
                String cegid = element.getAttribute("cegid");

```

```

        String nev =
element.getElementsByTagName("nev").item(0).getTextContent();
        String cim =
element.getElementsByTagName("cim").item(0).getTextContent();
        String tel =
element.getElementsByTagName("telefonszam").item(0).getTextContent();

        System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
        System.out.println("-----");
        System.out.println("Current Element : " +
node.getNodeName());

        System.out.println("Ceg azonosito : " + cegid);
        System.out.println("Cegnev : " + nev);
        System.out.println("Szekhely : " + cim +
"\nTelefonszam: " + tel);
    }
}

public static void ReadMuhelyById(Document doc, String id) {
    // muhely tag-el rendelkezo elemek lekerese
    NodeList nList = doc.getElementsByTagName("muhely");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("cegid").equals(id)) {
                String cegid = element.getAttribute("cegid");
                String mid = element.getAttribute("mid");
                String javitandoAuto =
element.getElementsByTagName("javitando_auto").item(0).getTextContent();

                System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
                System.out.println("-----");
                System.out.println("Current Element : " +
node.getNodeName());

                System.out.println("Ceg azonosito : " + cegid);
                System.out.println("MuhelyID : " + mid);
                System.out.println("Javitando auto : " +
javitandoAuto);

                ReadFutarById(doc, mid);
            }
        }
    }
}

// Ezt a metodust a DomModify hasznalja. Azert van ra szukseg, hogy CSAK a
// muhely adatait listazzuk a modositashoz
public static void ReadMuhelyByIdWithoutCeg(Document doc, String id) {
    // muhely tag-el rendelkezo elemek lekerese, a ceg olvasasa nelkul
    NodeList nList = doc.getElementsByTagName("muhely");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("cegid").equals(id)) {
                String cegid = element.getAttribute("cegid");

```

```

        String mid = element.getAttribute("mid");
        String javitandoAuto =
element.getElementsByTagName("javitando_auto").item(0).getTextContent();

        System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
        System.out.println("-----");
        System.out.println("Current Element : " +
node.getNodeName());

        System.out.println("Ceg azonosito : " + cegid);
        System.out.println("MuhelyID : " + mid);
        System.out.println("Javitando auto : " +
javitandoAuto);
    }
}

public static void ReadFutarById(Document doc, String id) {
    // futar tag-el rendelkezo elemek lekerese
    NodeList nList = doc.getElementsByTagName("futar");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("mid").equals(id)) {

                String mid = element.getAttribute("mid");
                String cikkszam =
element.getAttribute("cikkszam");

                String varhatoErkezes =
element.getElementsByTagName("varhato_erk").item(0).getTextContent();

                System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
                System.out.println("-----");
                System.out.println("Current Element : " +
node.getNodeName());

                System.out.println("Cikkszam : " + cikkszam);
                System.out.println("MuhelyID : " + mid);
                System.out.println("Varhato erkezes : " +
varhatoErkezes);

                ReadRaktarById(doc, cikkszam);
            }
        }
    }

    public static void ReadRaktarById(Document doc, String id) {
        // raktar tag-el rendelkezo elemek lekerese
        NodeList nList = doc.getElementsByTagName("raktar");

        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i); // lista aktualis elemeinek lekerese
            Element element = (Element) node; // konvertalas elementekke

            if (node.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("cikkszam").equals(id)) {
                    String cikkszam =
element.getAttribute("cikkszam");

                    String alkatreszek =
element.getElementsByTagName("alkatreszek").item(0).getTextContent();

```

```

        System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
        System.out.println("-----");
        System.out.println("Current Element : " +
node.getNodeName());

        System.out.println("Cikkszam : " + cikkszam);
        System.out.println("Alkatresz : " + alkatreszek);
    }
}

public static void ReadAuto(Document doc) {

    // auto tag-el rendelkezo elemek lekerese
    NodeList nList = doc.getElementsByTagName("auto");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

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

            String ugyfelid = element.getAttribute("ugyfelid");
            String alvazszam = element.getAttribute("alvazszam");
            String marka =
element.getElementsByTagName("marka").item(0).getTextContent();
            String tipus =
element.getElementsByTagName("tipus").item(0).getTextContent();
            String km =
element.getElementsByTagName("km_ora_allasa").item(0).getTextContent();

            System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
            System.out.println("-----");
            System.out.println("Current Element : " +
node.getNodeName());

            System.out.println("UgyfelID : " + ugyfelid);
            System.out.println("Alvazszam : " + alvazszam);
            System.out.println("Marka : " + marka);
            System.out.println("Tipus : " + tipus);
            System.out.println("Km ora allasa : " + km);

            ReadTulajdonosById(doc, ugyfelid);

        }
    }

}

// Ezt a metodust a DomModify hasznalja. Azert van ra szukseg, hogy CSAK az
// auto adatait listazzuk a modositashoz
public static void ReadAutoWithoutTulajdonos(Document doc) {

    // auto tag-el rendelkezo elemek lekerese, tulajdonos nelkul
    NodeList nList = doc.getElementsByTagName("auto");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

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

            String ugyfelid = element.getAttribute("ugyfelid");
            String alvazszam = element.getAttribute("alvazszam");
            String marka =
element.getElementsByTagName("marka").item(0).getTextContent();

```

```

        String tipus =
element.getElementsByTagName("tipus").item(0).getTextContent();
        String km =
element.getElementsByTagName("km_ora_allasa").item(0).getTextContent();

        System.out.println(i + 1 + ". auto : ");
        System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
        System.out.println("-----");
        System.out.println("Current Element : " +
node.getNodeName());

        System.out.println("UgyfelID : " + ugyfelid);
        System.out.println("Alvazszam : " + alvazszam);
        System.out.println("Marka : " + marka);
        System.out.println("Tipus : " + tipus);
        System.out.println("Km ora allasa : " + km);
    }
}

public static void ReadTulajdonosById(Document doc, String id) {
    // tulajdonos tag-el rendelkezo elemek lekerese
    NodeList nList = doc.getElementsByTagName("tulajdonos");

    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i); // lista aktualis elemeinek lekerese
        Element element = (Element) node; // konvertalas elementekke

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("ugyfelid").equals(id)) {
                String ugyfelid =
element.getAttribute("ugyfelid");
                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();
                String cim =
element.getElementsByTagName("cim").item(0).getTextContent();
                String szulido =
element.getElementsByTagName("szulido").item(0).getTextContent();

                System.out.println("\nRoot Element : " +
doc.getDocumentElement().getNodeName());
                System.out.println("-----");
                System.out.println("Current Element : " +
node.getNodeName());

                System.out.println("UgyfelID : " + ugyfelid);
                System.out.println("Nev : " + nev);
                System.out.println("Cim : " + cim);
                System.out.println("Szuletesi ido : " + szulido);
            }
        }
    }
}
}

```


DOM adatlekérdezés

Ez az osztály listázza azon cégjegyzékben szereplő cégnek az adatait – úgymint cég azonosító, cégnév, székhely és telefonszám – amelynek a sorszámát megadjuk.

```
package hu.domparsedtkug0;

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

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

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 DOMQueryDTKUG0 {

    public static void main(String[] args)
        throws ParserConfigurationException, IOException, SAXException,
        TransformerException {
        File xmlFile = new File("XMLdtkug0.xml"); // XML fájl bekerese
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        // olvasás lehetővé tetele
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        // XML dokumentum atadása és egy DOM Document objektum létrehozása
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        System.out.println("Root element: " +
            doc.getDocumentElement().getNodeName());
        System.out.println("Cégjegyzék listázása: ");
        LoadCégjegyzékQuery(doc);
    }

    // van értékelő ellenőrzés, vagyis ha rossz számot adunk meg, vagy esetleg nem
    // számot, akkor hibát kapunk
    private static int readCégSorszám() {
        Scanner input = new Scanner(System.in);
        int cégSorszám;
        System.out.print("Írja be annak a cégnek a sorszámát, amelynek az
adatait meg szeretne jeleníteni: ");
        while (!input.hasNextInt()) {
            System.out.println("Nem számmal adott meg!\nAdjon meg új
sorszámot:");
            input.next();
        }
        cégSorszám = input.nextInt();
        return cégSorszám;
    }

    public static void LoadCégjegyzékQuery(Document doc) throws
        TransformerException {
        NodeList nodeList = doc.getElementsByTagName("cégjegyzék"); //
Cégjegyzék elemeinek listázása
        int cég = 0; // cég sorszámainak létrehozott változó
        Element element = null;
        Node nNode = null;
```

```

        int countCegekSzama = 0; // Cegek megszamolasa, hogy ellenorizni
        lehessen hany ceg van osszesen -> csak a
                                                                    // megfelelo
        sorszamot fogadjuk el
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            String bejegyzettCeg =
        element.getElementsByTagName("bejegyzett_cegek").item(0).getTextContent();
            System.out.println((i + 1) + ". ceg: " + bejegyzettCeg);
            countCegekSzama++;
        }
        // Ceg kivlasztasa, sorszam ellenorzott beolvasasa
        ceg = readCegSorszam();
        if (ceg < 1 || ceg > countCegekSzama) {
            System.out.println("Nem megfelelo sorszam!");
        }

        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                switch (ceg) {
                    case 1: {
                        LoadCegQuery(doc, "1");
                        break;
                    }
                    case 2: {
                        LoadCegQuery(doc, "2");
                        break;
                    }
                    case 3: {
                        LoadCegQuery(doc, "3");
                        break;
                    }
                }
            }
        }

        // Kivalasztott ceg adatai
        public static void LoadCegQuery(Document doc, String id) throws
        TransformerException {
            NodeList nodeList = doc.getElementsByTagName("ceg");

            for (int i = 0; i < nodeList.getLength(); i++) {
                Node nNode = nodeList.item(i);
                Element element = (Element) nNode;
                String cegid = element.getAttribute("cegid");
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                    if (id.equals(cegid)) {
                        System.out.println("\nA valasztott ceg adatai:

");
                        DOMReadDTKUG0.ReadCegByIdWithoutMuhely(doc, id);
                    }
                }
            }
        }
    }
}

```

DOM adatmódosítás

Ez az osztály lehetővé teszi a Műhely, Raktár, Autó és Tulajdonos egyeden belül megadott adatok módosítását, a választott ID alapján.

```
package hu.domparsedtkug0;

import java.io.File;
import java.io.IOException;
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.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMModifyDTKUG0 {

    public static void main(String[] args)
        throws ParserConfigurationException, IOException, SAXException,
        TransformerException {
        File xmlFile = new File("XMLdtkug0.xml"); // XML fájl bekerese
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        // olvasas lehetove tetele
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();
        System.out.println("XML fájl modositasa");
        System.out.println("Adja meg mit szeretne modositani: ");
        System.out.println("1. -> Muhely modositasa");
        System.out.println("2. -> Raktar modositasa");
        System.out.println("3. -> Auto modositasa");
        System.out.println("4. -> Tulajdonos modositasa");
        Modify(doc);
    }

    public static void ModifyXML(Document doc) throws TransformerException {
        // Transformer objektumot hozunk létre amely lehetove teszi a
        dokumentum
        // modositast
        TransformerFactory transformerFactory =
        TransformerFactory.newInstance();
        // létrehoz egy masolatot a source-bol a result-ba
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(new File("XMLdtkug0.xml"));
        // létrehoz egy masolatot a source-bol a result-ba
        transformer.transform(source, result);
    }

    public static void Modify(Document doc) throws TransformerException {
        int muhelyekSzama = doc.getElementsByTagName("muhely").getLength();
        int raktarakSzama = doc.getElementsByTagName("raktar").getLength();
        int autokSzama = doc.getElementsByTagName("auto").getLength();
        int tulajdonosokSzama =
        doc.getElementsByTagName("tulajdonos").getLength();
    }
}
```

```

Scanner scanner = new Scanner(System.in);
System.out.print("Adja meg a sorszámot: ");
int readCategory = scanner.nextInt();
switch (readCategory) {
case 1:
    ModifyMuhely(doc, muhelyekSzama);
    break;
case 2:
    ModifyRaktar(doc, raktarakSzama);
    break;
case 3:
    ModifyAuto(doc, autokSzama);
    break;
case 4:
    ModifyTulajdonos(doc, tulajdonosokSzama);
    break;
}
}

public static String ReadId() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("\nAdja meg az ID-t :");
    String id = scanner.nextLine();
    return id;
}

private static void ModifyMuhely(Document doc, int muhelyekszama) throws
TransformerException {

    System.out.println(muhelyekszama + " db muhely erhete el. Melyik muhely
adatait szeretne modositani?");
    for (int i = 1; i < muhelyekszama + 1; i++) {
        System.out.println();
        System.out.println(i + ". muhely:");
        DOMReadDTKUG0.ReadMuhelyByIdWithoutCeg(doc, String.valueOf(i));
        System.out.print("_____ \n");
    }
    String id = ReadId();
    // Adat bekerese
    Scanner scan = new Scanner(System.in);
    System.out.print("Javitando auto: ");
    String masikJavitandoAuto = scan.nextLine();

    NodeList nodeList = doc.getElementsByTagName("muhely");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;
            String sid = element.getAttribute("mid");
            if (sid.equals(id)) {
                Node node1 =
element.getElementsByTagName("javitando_auto").item(0);
                node1.setTextContent(masikJavitandoAuto);

                System.out.println("Sikeres modositas!");
            }
        }
    }
    ModifyXML(doc); // XML letrehozasa
}

private static void ModifyRaktar(Document doc, int raktarakszama) throws
TransformerException {

    System.out.println(raktarakszama + " db raktar erheto el. Melyik raktar
adatait szeretne modositani?");

```

```

        for (int i = 1; i < raktarakszama + 1; i++) {
            System.out.println();
            System.out.println(i + ". raktar:");
            DOMReadDTKUG0.ReadRaktarById(doc, String.valueOf(i));
            System.out.print("_____ \n");
        }
        String id = ReadId();
        // Az uj adat bekerese
        Scanner scanner = new Scanner(System.in);
        System.out.print("Alkatresz : ");
        String masikAlkatresz = scanner.nextLine();

        NodeList nodeList = doc.getElementsByTagName("raktar");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node nNode = nodeList.item(i);
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) nNode;
                String sid = element.getAttribute("cikkszam");
                if (sid.equals(id)) {
                    Node node1 =
element.getElementsByTagName("alkatreszek").item(0);
                    node1.setTextContent(masikAlkatresz);

                    System.out.println("Sikeres modositas");
                }
            }
        }
        ModifyXML(doc);
    }

    private static void ModifyAuto(Document doc, int autokszama) throws
TransformerException {

        System.out.println(autokszama + " db auto erheto el. Melyik auto
adatait szeretne modositani?");

        DOMReadDTKUG0.ReadAutoWithoutTulajdonos(doc);
        System.out.print("_____ \n");

        String id = ReadId();
        // Adat bekerese
        Scanner scanner = new Scanner(System.in);
        System.out.print("Marka: ");
        String masikMarka = scanner.nextLine();
        System.out.print("Tipus: ");
        String masikTipus = scanner.nextLine();
        System.out.print("Km ora allasa: ");
        String masikKmOra = scanner.nextLine();

        NodeList nodeList = doc.getElementsByTagName("auto");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node nNode = nodeList.item(i);
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) nNode;
                String sid = element.getAttribute("ugyfelid");
                if (sid.equals(id)) {
                    Node node1 =
element.getElementsByTagName("marka").item(0);
                    node1.setTextContent(masikMarka);
                    Node node2 =
element.getElementsByTagName("tipus").item(0);
                    node2.setTextContent(masikTipus);
                    Node node3 =
element.getElementsByTagName("km_ora_allasa").item(0);
                    node3.setTextContent(masikKmOra);

                    System.out.println("Sikeres modositas");
                }
            }
        }
    }
}

```

```

    }
    }
    ModifyXML(doc);
}

private static void ModifyTulajdonos(Document doc, int tulajokszama) throws
TransformerException {

    System.out.println(tulajokszama + " db tulajdonos erhető el. Melyik
tulajdonos adatait szeretné módosítani?");
    for (int i = 1; i < tulajokszama + 1; i++) {
        System.out.println();
        System.out.println(i + ". tulajdonos:");
        DOMReadDTKUG0.ReadTulajdonosById(doc, String.valueOf(i));
        System.out.print("_____\\n");
    }
    String id = ReadId();
    // Bekerjük az új adatot
    Scanner scanner = new Scanner(System.in);
    System.out.print("Név: ");
    String masikNev = scanner.nextLine();
    System.out.print("Cím: ");
    String masikCim = scanner.nextLine();
    System.out.print("Születési idő: ");
    String masikSzulIdo = scanner.nextLine();

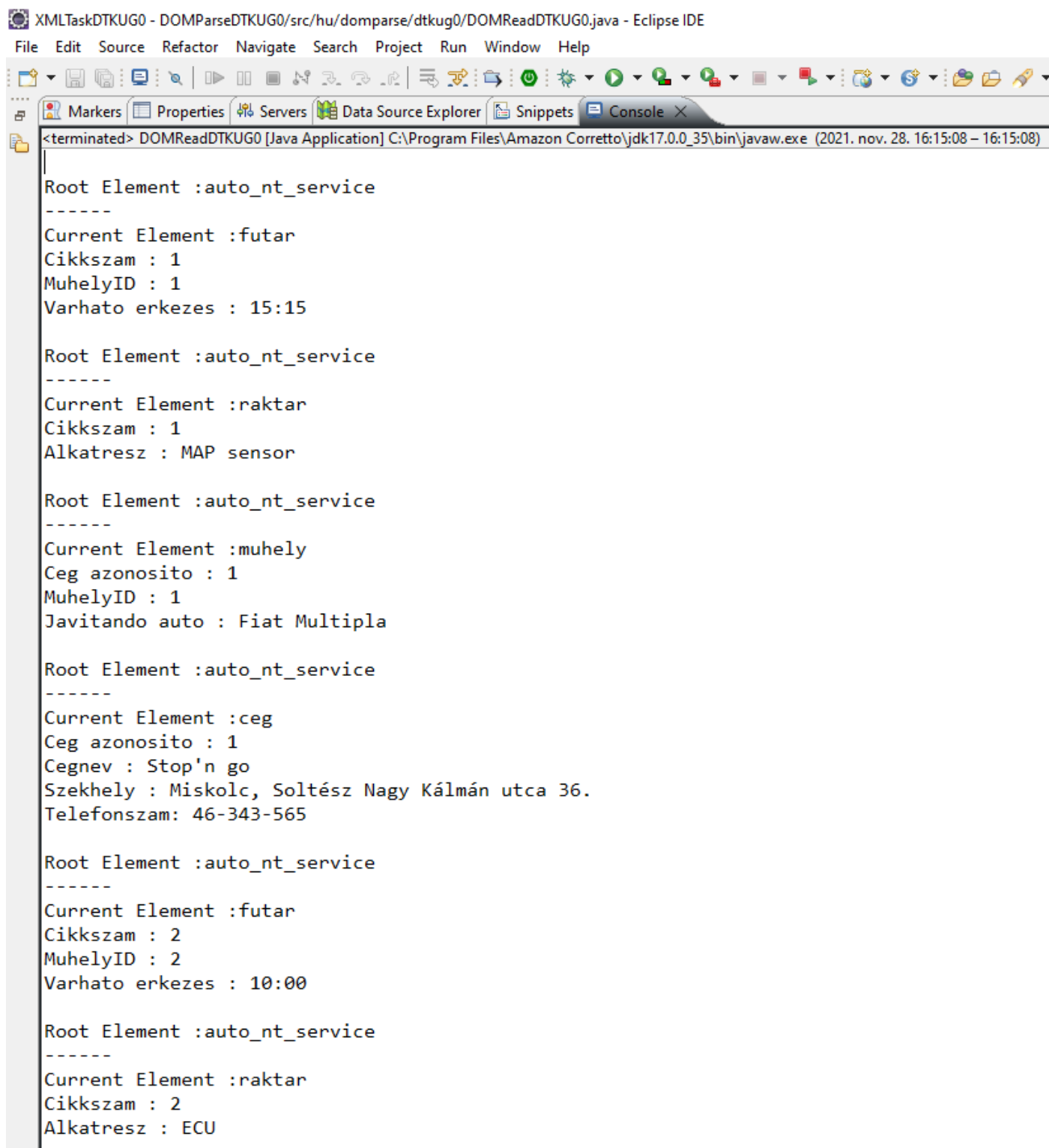
    NodeList nodeList = doc.getElementsByTagName("tulajdonos");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;
            String sid = element.getAttribute("ugyfeliid");
            if (sid.equals(id)) {
                Node node1 =
element.getElementsByTagName("nev").item(0);
                node1.setTextContent(masikNev);
                Node node2 =
element.getElementsByTagName("cim").item(0);
                node2.setTextContent(masikCim);
                Node node3 =
element.getElementsByTagName("szulido").item(0);
                node3.setTextContent(masikSzulIdo);

                System.out.println("Sikeres módosítás");
            }
        }
    }
    ModifyXML(doc); // XML letrehozása
}
}
}

```

Képernyőképek a futtatásról

1. Adatolvasás – DomReadDTKUG0



The screenshot shows the Eclipse IDE interface with the 'Console' tab active. The title bar indicates the project is 'XMLTaskDTKUG0 - DOMParseDTKUG0/src/hu/domparse/dtkug0/DOMReadDTKUG0.java - Eclipse IDE'. The console output shows the results of parsing XML data, with each entry starting with 'Root Element :auto_nt_service' and 'Current Element :futar'. The data is organized into five distinct blocks, each representing a different vehicle or location entry.

```
<terminated> DOMReadDTKUG0 [Java Application] C:\Program Files\Amazon Corretto\jdk17.0.0_35\bin\javaw.exe (2021. nov. 28. 16:15:08 – 16:15:08)

Root Element :auto_nt_service
-----
Current Element :futar
Cikkszam : 1
MuhelyID : 1
Varhato erkezes : 15:15

Root Element :auto_nt_service
-----
Current Element :raktar
Cikkszam : 1
Alkatresz : MAP sensor

Root Element :auto_nt_service
-----
Current Element :muhely
Ceg azonosito : 1
MuhelyID : 1
Javitando auto : Fiat Multipla

Root Element :auto_nt_service
-----
Current Element :ceg
Ceg azonosito : 1
Cegnev : Stop'n go
Szekhely : Miskolc, Soltész Nagy Kálmán utca 36.
Telefonszam: 46-343-565

Root Element :auto_nt_service
-----
Current Element :futar
Cikkszam : 2
MuhelyID : 2
Varhato erkezes : 10:00

Root Element :auto_nt_service
-----
Current Element :raktar
Cikkszam : 2
Alkatresz : ECU
```

2. Adatlekérdezés – DomQueryDTKUG0

```
XMLTaskDTKUG0 - DOMParseDTKUG0/src/hu/domparse/dtkug0/DOMQueryDTKUG0.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

<terminated> DOMQueryDTKUG0 [Java Application] C:\Program Files\Amazon Corretto\jdk17.0.0_35\bin\javaw.exe (2021. nov. 28. 16:16:17 – 16:16:21)
Root element: auto_nt_service

Cegjegyzek listazasa:
1. ceg: Stop'n go
2. ceg: Nonstop service
3. ceg: Bosch Car Service
Irja be annak a cegnek a sorszamat, amelynek az adatait meg szeretne jeleniteni: 1
|
A valasztott ceg adatai:

Root Element :auto_nt_service
-----
Current Element :ceg
Ceg azonosito : 1
Cegnev : Stop'n go
Szekhely : Miskolc, Soltész Nagy Kálmán utca 36.
Telefonszam: 46-343-565
```


3. Adatmódosítás – DomModifyDTKUG0

```
XMLTaskDTKUG0 - DOMParseDTKUG0/src/hu/domparse/dtkug0/DOMModifyDTKUG0.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console X
<terminated> DOMModifyDTKUG0 [Java Application] C:\Program Files\Amazon Corretto\jdk17.0.0_35\bin\javaw.exe (2021. nov. 28. 16:17:00 – 16:17:44)
XML fajl modositasa
Adja meg mit szeretne modositani:
1. -> Muhely modositasa
2. -> Raktar modositasa
3. -> Auto modositasa
4. -> Tulajdonos modositasa
Adja meg a sorszamot: 4
3 db tulajdonos erhető el. Melyik tulajdonos adatait szeretne modositani?

1. tulajdonos:

Root Element :auto_nt_service
-----
Current Element :tulajdonos
UgyfelID : 1
Nev : Kiss Ramóna
Cim : 5300 Karcag, Déryné utca 10
Szuletesi ido : 1999.01.01
-----

2. tulajdonos:

Root Element :auto_nt_service
-----
Current Element :tulajdonos
UgyfelID : 2
Nev : Tóth Fanni
Cim : Kiskunfélegyháza, Napsugár út. 50
Szuletesi ido : 1987.10.10
-----

3. tulajdonos:

Root Element :auto_nt_service
-----
Current Element :tulajdonos
UgyfelID : 3
Nev : Szepe Timea
Cim : 3530 Miskolc, Szentgyörgy út 43.
Szuletesi ido : 1999.04.04
-----

Adja meg az ID-t :3
Nev: Kiss Tamás
Cim: 3434 Malyi, Rakoczi ut 14.
Szuletesi ido: 1999.01.01
Sikereres modositas
```

Módosítás az XML dokumentumban:

```
70
71 <tulajdonos ügyfelid="1">
72     <nev>Kiss Ramóna</nev>
73     <cim>5300 Karcag, Déryné utca 10</cim>
74     <szulido>1999.01.01</szulido>
75 </tulajdonos>
76 <tulajdonos ügyfelid="2">
77     <nev>Tóth Fanni</nev>
78     <cim>Kiskunfélegyháza, Napsugár út. 50</cim>
79     <szulido>1987.10.10</szulido>
80 </tulajdonos>
81 <tulajdonos ügyfelid="3">
82     <nev>Kiss Tamás</nev>
83     <cim>3434 Malvi, Rakoczi ut 14.</cim>
84     <szulido>1999.01.01</szulido>
85 </tulajdonos>
```