**JEGYZŐKÖNYV**

Adatkezelés XML környezetben

Féléves feladat

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.

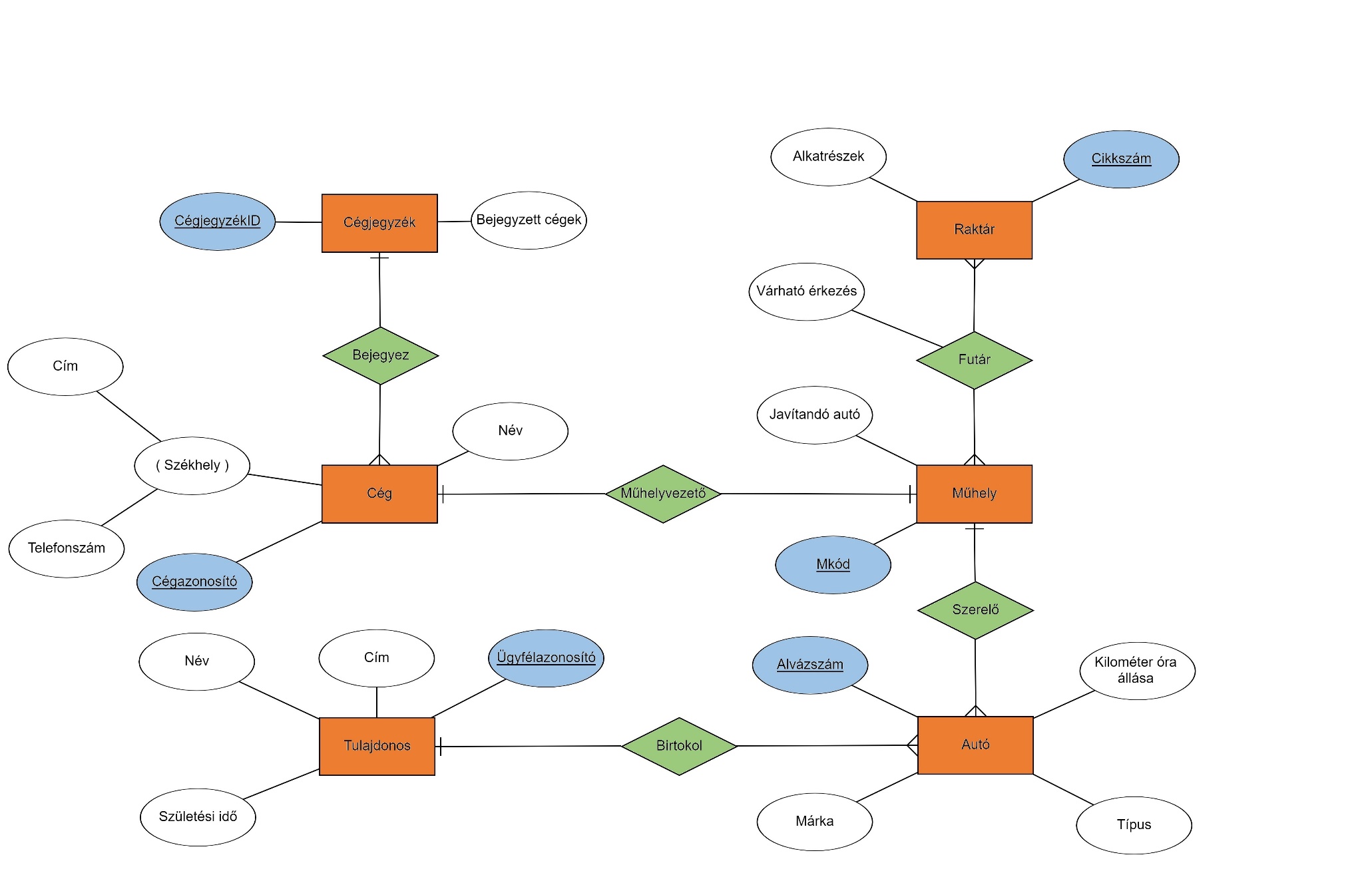
Ez az egyed két továbbival is kapcsolatban van, a Raktárral illetve az Autóval. A raktárnak összesen 2 tulajdonsága van: a Cikkszám, mint elsődleges kulcs, és az Alkatrészek. A Műhely és a Raktár között a Futár tartja a kapcsolatot, melynek van egy Várható érkezés tulajdonsága. A két egyed között több-több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

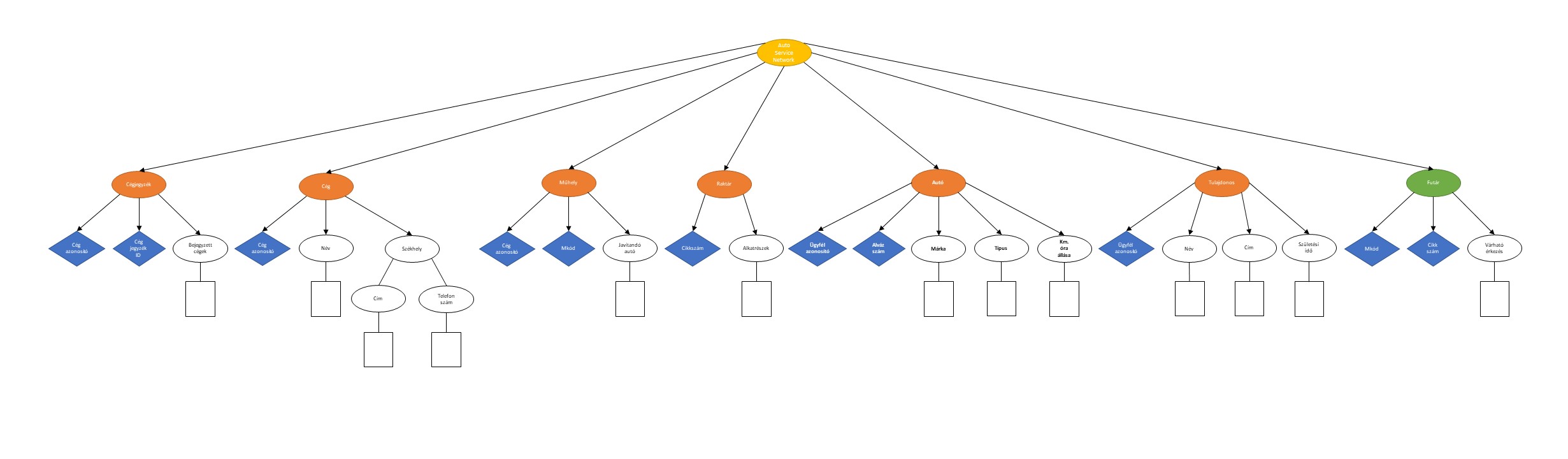
Végezetül, az ER modell utolsó egyede, a Tulajdonos, egy-több kapcsolattal van összekötve az Autóval, amelynek Birtokol a neve, hiszen egy tulajdonosnak több autója is lehet, viszont egy autónak csak egy tulajdonosa lehet. A Tulajdonos egyednek 5 tulajdonsága van, mégpedig a Név, a Cím, a Születési idő, 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=*"XMLSchemadtkug0.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: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\_ora\_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]|[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**

package hu.domparse.dtkug0;

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 fajl 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 letrehozasa

Document doc = dBuilder.parse(xmlFile);

// szomszedos es ures text node-ok eltavolitasara szolgal

doc.getDocumentElement().normalize();

Read(doc);

ReadCegjegyzek(doc);

ReadAuto(doc);

} catch (ParserConfigurationException pce) {

pce.printStackTrace();

} catch (IOException ioe) {

ioe.printStackTrace();

} catch (SAXException sae) {

sae.printStackTrace();

}

}

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 lekérése majd a definiált metódusok meghívása

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);

}

}

}

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);

}

}

}

}

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);

ReadCegById(doc, cegid);

}

}

}

}

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 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);

}

}

}

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 + ". aut� : ");

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);

}

}

}

}

public static void Read(Document doc) {

// 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) {

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);

ReadMuhelyById(doc, mid);

}

}

}

}

**DOM adatlekérdezés**

package hu.domparse.dtkug0;

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 fajl bekerese

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); // olvasas lehetove tetele

DocumentBuilder dBuilder = factory.newDocumentBuilder();

// XML dokumentum atadasa es egy DOM Document objektum letrehozasa

Document doc = dBuilder.parse(xmlFile);

doc.getDocumentElement().normalize();

System.out.println("Root element: " + doc.getDocumentElement().getNodeName());

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Cegjegyzek listazasa: ");

LoadCegjegyzekQuery(doc);

}

private static int readCegSorszam() {

Scanner input = new Scanner(System.in);

int cegSorszam;

System.out.print("Irja be annak a cegnek a sorszamat, amelynek az adatait meg szeretne jeleniteni: ");

while (!input.hasNextInt()) {

System.out.println("Nem szamott adott meg!\nAdjon meg uj sorszamot:");

input.next();

}

cegSorszam = input.nextInt();

return cegSorszam;

}

public static void LoadCegjegyzekQuery(Document doc) throws TransformerException {

NodeList nodeList = doc.getElementsByTagName("cegjegyzek"); // Cegjegyzek elemeinek listazasa

int ceg = 0; // ceg sorszamainak letrehozott valtozo

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 kivalasztasa, 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;

}

}

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.ReadCegById(doc, id);

}

}

}

}

}

**DOM adatmódosítás**

package hu.domparse.dtkug0;

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 fajl bekerese

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); // olvasas lehetove tetele

DocumentBuilder dBuilder = factory.newDocumentBuilder();

Document doc = dBuilder.parse(xmlFile);

doc.getDocumentElement().normalize();

System.out.println("XML fajl 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 {

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

StreamResult result = new StreamResult(new File("XMLdtkug0.xml"));

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 sorszamot: ");

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 erheto el. Melyik tulajdonos adatait szeretne modositani?");

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();

// Bekerjuk az uj adatot

Scanner scanner = new Scanner(System.in);

System.out.print("Nev: ");

String masikNev = scanner.nextLine();

System.out.print("Cim: ");

String masikCim = scanner.nextLine();

System.out.print("Szuletesi ido: ");

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("ugyfelid");

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 modositas");

}

}

}

ModifyXML(doc); // XML letrehozasa

}

}