JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: Szabó Zsófia Viola

Neptunkód: LFM4UL

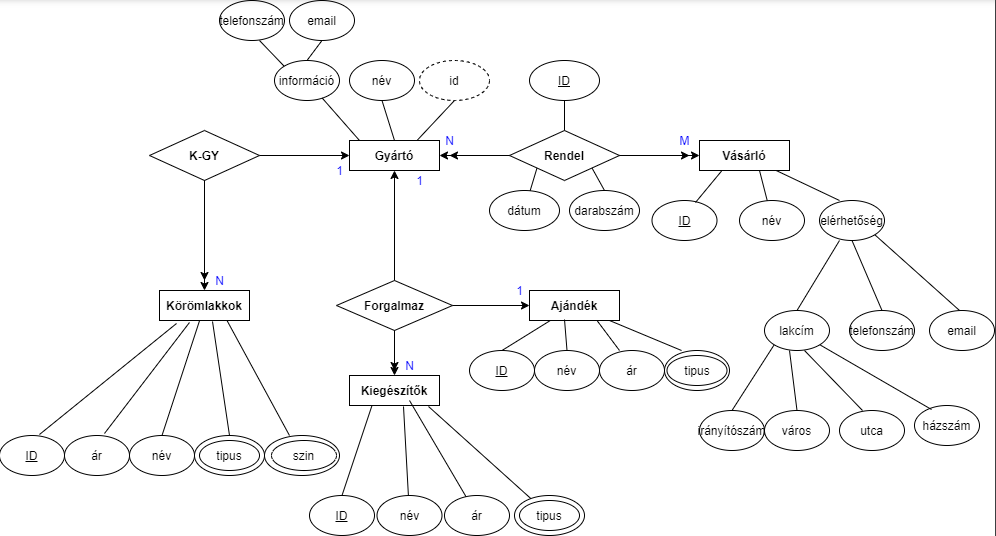
**A feladat leírása:**

A feladatban a CandyNails körömlakkgyártó cég, ennek vásárlói és megrendelések ER modelljét készítettem el az alábbi jellemzőkkel:

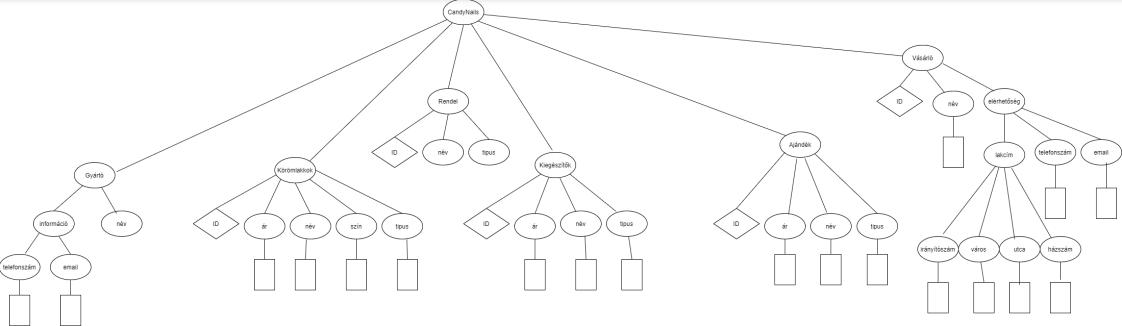
A *Vásárló*egyed rendelkezik egy kulcstulajdonsággal, egy egyszerű tulajdonsággal és egy többértékű tulajdonsággal, melyen belül található még egy többértékű tulajdonság. Ez az egyed N:M típusú kapcsolatban áll a *Gyártó* egyeddel, melyek közt a *Rendel* nevű kapcsolat található. Ez a kapcsolat két egyszerű tulajdonságot és egy kulcstulajdonságot tartalmaz. A *Gyártó* egyed rendelkezik egy egyszerű tulajdonsággal, egy többértékű tulajdonsággal és egy származtatott tulajdonsággal. A *Gyártó* egyed 1:N típusú kapcsolatban van a *Körömlakkok* egyeddel, köztük a *K-GY* nevű kapcsolat található. A *Körömlakkok* egyednek van egy kulcstulajdonsága, két egyszerű tulajdonsága és két összetett tulajdonsága. A *Gyártó* egyed 1:N kapcsolatban van a *Kiegészítők* egyeddel, a köztük lévő kapcsolat neve: *Forgalmaz*. A *Kiegészítők* egyednek van egy kulcstulajdonsága, két egyszerű tulajdonsága és egy összetett tulajdonsága. A *Gyártó* és az *Ajándékok* között 1:1 típusú kapcsolat van, a kapcsolat neve: *Forgalmaz*.

**1. feladat**

**1a) adatbázis ER modell**



**1b) Az adatbázis konvertálása XDM modellre:**



**1c) Az XDM modell alapján XML dokumentum készítése:**

<?xml version="1.0" encoding="utf-8" ?>

<koromlakkgyarto xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="XMLSchemaLFM4UL.xsd">

<gyarto>

<nev>CandyNails</nev>

<informacio>

<telefonszam>06309876543</telefonszam>

<email>candynails@gmail.com</email>

</informacio>

</gyarto>

<koromlakkok ID="1234">

<ar>2600</ar>

<nev>Magnet Illusion</nev>

<szin>marvany</szin>

<tipus>magneses koromlakk</tipus>

</koromlakkok>

<koromlakkok ID="1235">

<ar>2400</ar>

<nev>Digital Illusion</nev>

<szin>feher</szin>

<tipus>csillogos koromlakk</tipus>

</koromlakkok>

<rendel ID="011">

<datum>2021-11-20</datum>

<darabszam>2</darabszam>

</rendel>

<vasarlo ID="111">

<nev>Kiss Anna</nev>

<elerhetoseg>

<lakcim>

<iranyitoszam>3423</iranyitoszam>

<varos>Tibolddaroc</varos>

<utca>Kacsi</utca>

<hazszam>11</hazszam>

</lakcim>

<telefonszam>06301234567</telefonszam>

<email>kissanna@gmail.com</email>

</elerhetoseg>

</vasarlo>

<vasarlo ID="222">

<nev>Nagy Niki</nev>

<elerhetoseg>

<lakcim>

<iranyitoszam>3423</iranyitoszam>

<varos>Tibolddaroc</varos>

<utca>Abranyi</utca>

<hazszam>45</hazszam>

</lakcim>

<telefonszam>06301237863</telefonszam>

<email>nagyniki@gmail.com</email>

</elerhetoseg>

</vasarlo>

<kiegeszitok ID="2345">

<ar>1600</ar>

<nev>magnes</nev>

<szin>fekete</szin>

<tipus>magnes</tipus>

</kiegeszitok>

<kiegeszitok ID="2346">

<ar>2000</ar>

<nev>matricak</nev>

<szin>mindenfele</szin>

<tipus>matrica</tipus>

</kiegeszitok>

<ajandek ID="26">

<ar>0</ar>

<nev>Sprarkling Crystals</nev>

<szin>szinvalto</szin>

<tipus>csillam</tipus>

</ajandek>

<ajandek ID="27">

<ar>0</ar>

<nev>Koromdiszito kovek</nev>

<szin>kek</szin>

<tipus>diszkovek</tipus>

</ajandek>

</koromlakkgyarto>

**1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok):**

<?xml version="1.0" encoding="utf-8" ?>

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

<xs:element name="koromlakkgyarto">

<xs:complexType>

<xs:sequence>

<xs:element name="gyarto" type="gyartoTipus" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="koromlakkok" type="koromlakkokTipus" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="rendel" type="rendelTipus" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="vasarlo" type="vasarloTipus" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="kiegeszitok" type="kiegeszitokTipus" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="ajandek" type="ajandekTipus" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:complexType name="gyartoTipus">

<xs:sequence>

<xs:element name="nev" type="xs:string"/>

<xs:element name="informacio" type="informacioTipus"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="informacioTipus">

<xs:sequence>

<xs:element name="telefonszam" type="xs:positiveInteger"/>

<xs:element name="email" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="koromlakkokTipus">

<xs:sequence>

<xs:element name="ar" type="xs:double"/>

<xs:element name="nev" type="xs:string"/>

<xs:element name="szin" type="xs:string"/>

<xs:element name="tipus" type="xs:string" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="ID" type="xs:integer"/>

</xs:complexType>

<xs:complexType name="rendelTipus">

<xs:sequence>

<xs:element name="datum" type="xs:date"/>

<xs:element name="darabszam" type="xs:integer"/>

</xs:sequence>

<xs:attribute name="ID" type="xs:integer"/>

</xs:complexType>

<xs:complexType name="vasarloTipus">

<xs:sequence>

<xs:element name="nev" type="xs:string"/>

<xs:element name="elerhetoseg" type="elerhetosegTipus"/>

</xs:sequence>

<xs:attribute name="ID" type="xs:integer"/>

</xs:complexType>

<xs:complexType name="elerhetosegTipus">

<xs:sequence>

<xs:element name="lakcim" type="lakcimTipus"/>

<xs:element name="telefonszam" type="xs:positiveInteger"/>

<xs:element name="email" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="lakcimTipus">

<xs:sequence>

<xs:element name="iranyitoszam" type="xs:integer"/>

<xs:element name="varos" type="xs:string"/>

<xs:element name="utca" type="xs:string"/>

<xs:element name="hazszam" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="kiegeszitokTipus">

<xs:sequence>

<xs:element name="ar" type="xs:double"/>

<xs:element name="nev" type="xs:string"/>

<xs:element name="szin" type="xs:string"/>

<xs:element name="tipus" type="xs:string" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="ID" type="xs:integer"/>

</xs:complexType>

<xs:complexType name="ajandekTipus">

<xs:sequence>

<xs:element name="ar" type="xs:double"/>

<xs:element name="nev" type="xs:string"/>

<xs:element name="szin" type="xs:string"/>

<xs:element name="tipus" type="xs:string" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="ID" type="xs:integer"/>

</xs:complexType>

</xs:schema>

**2. feladat**

**2a) adatolvasás**

package hu.domparse.lfm4ul;

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 DomReadLFM4UL {

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

try {

//megadom a file nevet

File file = new File("xmlLFM4UL.xml");

//DOM létrehozása

//DocumentBuilderFactorybol megkapjuk a DocumentBuildert

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = factory.newDocumentBuilder();

Document doc = dBuilder.parse(file); //file elemzes

doc.getDocumentElement().normalize();

System.out.println("Root element: " + doc.getDocumentElement().getNodeName()); //megkapjuk, gyokerelem kiiratasa

NodeList nList = (NodeList) doc.getDocumentElement(); //gyerekelemek mentjuk listaba

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

Node nNode = nList.item(i);

System.out.println("Aktualis elem:"+ nNode.getNodeName());

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element elem = (Element)nNode;

//ID kiiratás

String ID = elem.getAttribute("ID"); //elem attributumot megkapjuk

System.out.println("ID: " + ID);

//tulajdonságok kiiratása (gyerekelemek)

String nodeContent = " ";

NodeList childNodes = elem.getChildNodes();

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

if (childNodes.item(j).getTextContent().trim() != "") {

nodeContent = childNodes.item(j).getTextContent().trim();

System.out.println(childNodes.item(j).getNodeName() + " : " + nodeContent);

}

}

}

System.out.println();

}

//hibakezelés

}catch (SAXException | IOException | ParserConfigurationException e) {

System.out.println("Hiba" + e.getMessage());

e.printStackTrace();

}

}

}

**2b) adatmódosítás**

package hu.domparse.lfm4ul;

import java.io.File;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.parsers.SAXParser;

import javax.xml.parsers.SAXParserFactory;

import javax.xml.xpath.XPath;

import javax.xml.xpath.XPathConstants;

import javax.xml.xpath.XPathExpressionException;

import javax.xml.xpath.XPathFactory;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.Attributes;

import org.xml.sax.SAXException;

import org.xml.sax.helpers.DefaultHandler;

public class DomQueryLFM4UL {

public static void main(String[]args) {

try {

//DOM objektum létrehozása

File xmlFile = new File("xmlLFM4UL.xml");

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = factory.newDocumentBuilder();

Document document = dBuilder.parse(xmlFile); //parszolas

document.getDocumentElement().normalize();

//XPath obejktum létrehozás

XPath xPath = XPathFactory.newInstance().newXPath();

//lekérdezések

//adatbázis lekérdezés (összes adat)

//String expression = "/koromlakkgyarto/\*"; //megadjuk az eleresi utat és a csomópont listát

//gyártó adatai

String expression = "koromlakkgyarto/gyarto";

//111. számú vásárló lekérdezés

//String expression = "//vasarlo[@id='111']";

//2500ft feletti kiegészítő

//String expression = "/koromlakkgyarto/kiegeszitok[ar>2500]";

//utolso ajandek

//String expression = "/koromlakkgyarto/ajandek[last()]";

//lista készités, XPath kifejezés megirása, kiértékelés

NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(document, XPathConstants.NODESET);

//végigmegyünk a csomópontokon

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

Node node = nodeList.item(i);

Node nNode = nodeList.item(i);

System.out.println("Aktualis elem:" + node.getNodeName());

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element elem = (Element)nNode;

// ID kiírása

String ID = elem.getAttribute("ID");

System.out.println("ID: " + ID);

//tulajondságok kiiratása (gyerekelemek)

String nodeContent = " ";

NodeList childNodes = elem.getChildNodes();

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

if (childNodes.item(j).getTextContent().trim() != "") {

nodeContent = childNodes.item(j).getTextContent().trim();

System.out.println(childNodes.item(j).getNodeName() + " : " + nodeContent);

}

}

}

}

//hibakezelés

}catch(ParserConfigurationException e) {

e.printStackTrace();

} catch(SAXException e) {

e.printStackTrace();

}

catch(IOException e) {

e.printStackTrace();

} catch(XPathExpressionException e) {

e.printStackTrace();

}

}

}

**2c) adatlekérdezés**

package hu.domparse.lfm4ul;

import java.io.File;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.parsers.SAXParser;

import javax.xml.parsers.SAXParserFactory;

import javax.xml.transform.OutputKeys;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerConfigurationException;

import javax.xml.transform.TransformerException;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.NamedNodeMap;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.Attributes;

import org.xml.sax.SAXException;

import org.xml.sax.helpers.DefaultHandler;

public class DomModifyLFM4UL {

public static void main(String argv[]) {

try {

File inputFile = new File("xmlLFM4UL.xml"); //megadom a file nevet

DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

Document doc = docBuilder.parse(inputFile); //file elemzes

// ID:1234 körömlakk ár módosítás 2400 ft-ra

NodeList nodes = doc.getElementsByTagName("koromlakkok"); //megkapjuk a koromlakkok NodeListet a dokumentumban

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

Node node = nodes.item(temp);

if (node.getNodeType() == Node.ELEMENT\_NODE) {

if (node.getAttributes().getNamedItem("ID").getTextContent().equals("1234")) {

NodeList childNodes = node.getChildNodes();

for (int temp2 = 0; temp2 < childNodes.getLength(); temp2++) {

Node childNode = childNodes.item(temp2);

if (childNode.getNodeName().equals("ar")) {

childNode.setTextContent("2400");

}

}

}

}

}

// ID:111 vásárló név módosítás

nodes = doc.getElementsByTagName("vasarlo");

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

Node node = nodes.item(temp);

if (node.getNodeType() == Node.ELEMENT\_NODE) {

if (node.getAttributes().getNamedItem("ID").getTextContent().equals("111")) {

NodeList childNodes = node.getChildNodes();

for (int temp2 = 0; temp2 < childNodes.getLength(); temp2++) {

Node childNode = childNodes.item(temp2);

if (childNode.getNodeName().equals("nev")) {

childNode.setTextContent("Nagyne Kiss Anna");

}

}

}

}

}

//Ajándékok törlése

Node koromlakkgyarto = doc.getFirstChild();

NodeList childNodes = koromlakkgyarto.getChildNodes();

for(int temp3 = 0; temp3 < childNodes.getLength(); temp3++) {

Node node = childNodes.item(temp3);

if("ajandek".equals(node.getNodeName()))

koromlakkgyarto.removeChild(node);

}

write(doc);

} catch (Exception e) {

e.printStackTrace();

}

}

//kiirató fgv

private static void write(Document doc) throws TransformerException {

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

System.out.println("-Modified File-");

transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");

transformer.setOutputProperty(OutputKeys.INDENT, "yes");

transformer.setOutputProperty("{http://xml.apache.org/xslt}indent-amunt", "2");

DOMSource source = new DOMSource(doc);

StreamResult console = new StreamResult(System.out); //kiiratás konzolra

transformer.transform(source, console);

}

}