

JEGYZŐKÖNYV

Adatkezelés XML
környezetben Féléves
feladat

Készítette:

Balázsfalvi Gergő

Neptunkód:

WCD0QI

1. Feladat leírása

Választott feladatom egy mini online webáruház rendeléseinek tárolása. A fiktív weboldal számítógép eladásra specializálódik, amelyet bárki meghirdethet, és az érdeklődők meg is rendelhetik a terméket.

A számítógépet az eladó rakja össze, egyedi specifikációval, beállíthatja továbbá a rendelés adatait.

A gép-rendelésben megadhatja a rendelni kívánt termék mennyiségét. Ez a rendelés tartalmaz továbbá egy rendelési azonosítót, illetve egy számítógép azonosítót.

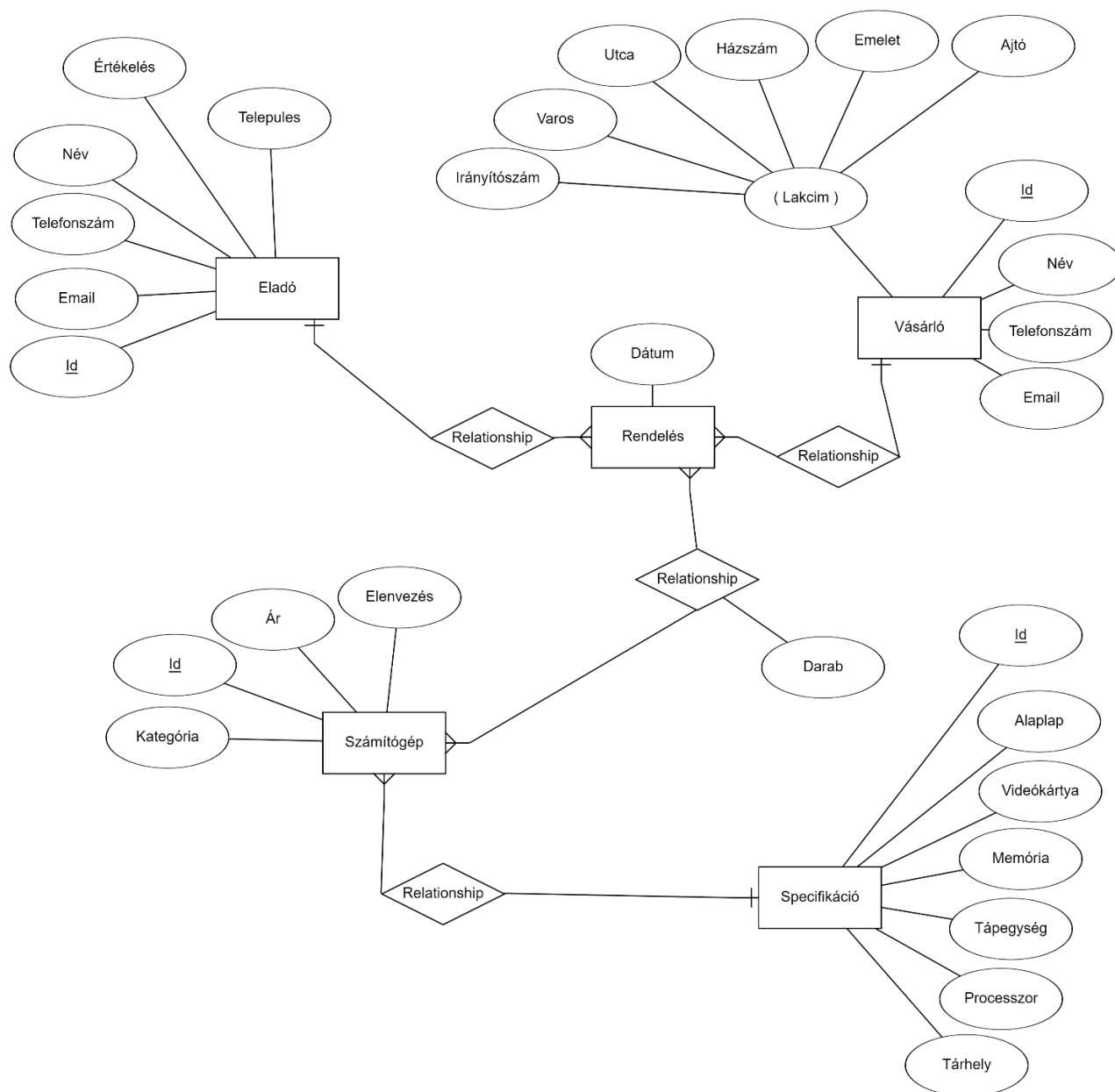
A rendelési adatokban további adatok szerepelnek, mint a rendelés dátuma, illetve egy eladói, illetve vásárlói azonosító is szerepel benne, így tudhatjuk, hogy ki kitől rendelt.

Az eladó adatai többek között tartalmazza az értékelést is, így a vásárló a megbízhatóság alapján könnyedén dönthet kitől vásárol.

A számítógép megadásában az ára, kategóriája és elnevezése mellett a hozzárendelt specifikációt is megtekinthetjük.

A rendelés beolvasást követően egyértelműen visszakövethető, akár módosításra is alkalmas lehet. Egy eladó egy specifikációt akár többször is felhasználhat.

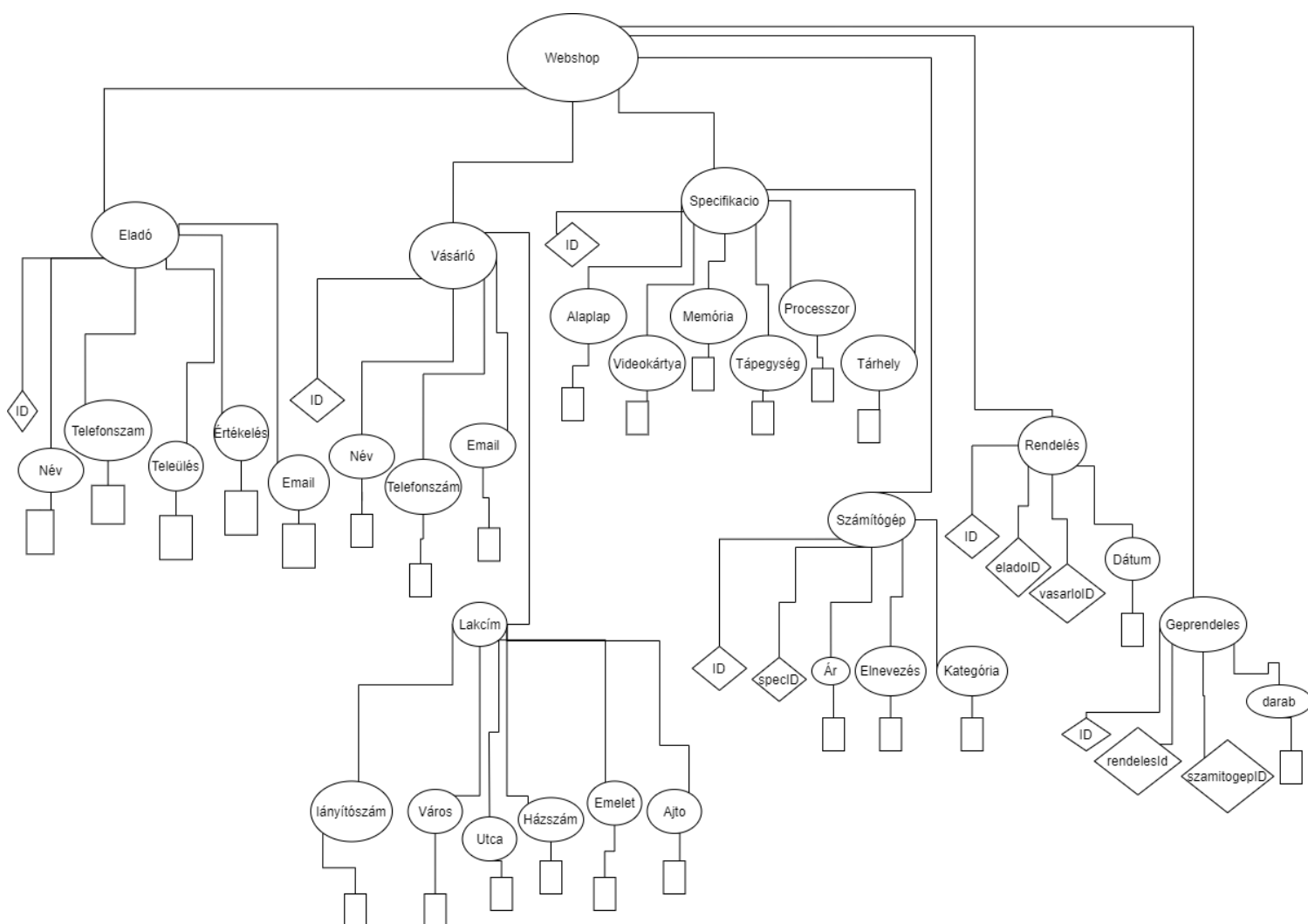
a. Az adatbázis ER modellje



b. Az adatbázis konvertálása XDM modellre

A konvertálást követően idegenkulcsok jönnek létre. A Számítógép objektumban megjelenik így a specId, amely a specifikációra mutató idegenkulcs, illetve a rendelésnél megjelenik az eladold és a vasarlold is.

A több-több kapcsolatból létrejön a Géprendelés objektum, amely tartalmazza a kapcsolat tulajdonságát, illetve további két idegenkulcsot, amely a rendelés azonosítóját, illetve a számítógép azonosítóját jelzi.



c. XML dokumentum

```
d. <?xml version="1.0" encoding="UTF-8" standalone="no"?><webshop xmlns="webshop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="webshop
XMLSchemaWCD0QI.xsd">
  <geprendeles id="1" rendelesId="1" szamitogepId="1">
    <darab>2</darab>
  </geprendeles>
  <geprendeles id="1" rendelesId="2" szamitogepId="1">
    <darab>100</darab>
  </geprendeles>
  <szamitogep id="1" specId="1">
    <ar>75000</ar>
    <elnevezes>Irodai</elnevezes>
    <kategoria>alacsony</kategoria>
  </szamitogep>
  <specifikacio id="1">
    <alaplap>ASUS T4104F</alaplap>
    <videokartya>AMD 8GB</videokartya>
    <memoria>4GB Viper X</memoria>
    <tapegyseg>400W</tapegyseg>
    <processzor>Intel Core i3 3900F</processzor>
    <tarhely>200GB Kingstone HDD</tarhely>
    <egyeb>CD író/olvasó</egyeb>
  </specifikacio>
  <rendeles eladoId="1" id="1" vasarloId="1">
    <datum>2014.01.12</datum>
  </rendeles>
  <elado id="1">
    <nev>Kocsis Sándor</nev>
    <email>kocsis@sandor.hu</email>
    <telszam>+36702552432</telszam>
    <telepules>Szeged</telepules>
    <ertekeles>5</ertekeles>
  </elado>
  <vasarlo id="1">
    <nev>Varga Bela</nev>
    <telefonszam>+3652412414</telefonszam>
    <email>bela@varga.hu</email>
    <lackim>
      <irsz>4200</irsz>
      <varos>Szeged</varos>
      <utca>Arany Janos</utca>
      <hazszam>10</hazszam>
      <emelet>1</emelet>
      <ajto>3</ajto>
    </lackim>
  </vasarlo>
  <geprendeles id="2" rendelesId="2" szamitogepId="2">
    <darab>4</darab>
  </geprendeles>
  <szamitogep id="2" specId="2">
    <ar>500000</ar>
    <elnevezes>Gamer</elnevezes>
    <kategoria>csúcs</kategoria>
  </szamitogep>
  <specifikacio id="2">
    <alaplap>ASUS T9104F</alaplap>
    <videokartya>AMD 16GB</videokartya>
```

```

    <memoria>32GB Viper X</memoria>
    <tapegyseg>600W</tapegyseg>
    <processzor>Intel Core i9 9400F</processzor>
    <tarhely>2TB Kingstone SSD</tarhely>
    <egyeb>CD író/olvasó, processzor vízűtő</egyeb>
  </specifikacio>
  <rendeles eladoId="1" id="2" vasarloId="1">
    <datum>2015.01.12</datum>
  </rendeles>
  <elado id="2">
    <nev>Kocsis Sándorkakas</nev>
    <email>kocsis@sandor.hu</email>
    <telszam>+36702552432</telszam>
    <telepules>Miskolc</telepules>
    <ertekeles>4</ertekeles>
  </elado>
  <vasarlo id="2">
    <nev>Varga Andras</nev>
    <telefonszam>+3652412414</telefonszam>
    <email>andras@varga.hu</email>
    <lackim>
      <irsz>3900</irsz>
      <varos>Szerencs</varos>
      <utca>Kovacs Janos</utca>
      <hazszam>50</hazszam>
      <emelet>11</emelet>
      <ajto>23</ajto>
    </lackim>
  </vasarlo>
</webshop>

```

d. XMLSchema készítése

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="webshop" xmlns="webshop">
  <xs:element name="webshop">
    <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="geprendeles" type="geprendelesType"/>
          <xs:element name="szamitogep" type="szamitogepType"/>
          <xs:element name="specifikacio" type="specifikacioType"/>
          <xs:element name="rendeles" type="rendelesType"/>
          <xs:element name="elado" type="eladoType"/>
          <xs:element name="vasarlo" type="vasarloType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
    <xs:key name="geprendelesID_PK">
      <xs:selector xpath="geprendeles"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="szamitogepID_PK">
      <xs:selector xpath="szamitogep"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="specifikacioID_PK">
      <xs:selector xpath="specifikacio"/>
    </xs:key>

```

```

        <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="rendelesID_PK">
        <xs:selector xpath="rendeles"/>
        <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="eladoID_PK">
        <xs:selector xpath="elado"/>
        <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="vasarloID_PK">
        <xs:selector xpath="vasarlo"/>
        <xs:field xpath="@id"/>
    </xs:key>
    <xs:keyref name="szamitogepID_FK" refer="szamitogepID_PK">
        <xs:selector xpath="geprendeles"/>
        <xs:field xpath="@szamitogepId"/>
    </xs:keyref>
    <xs:keyref name="rendelesID_FK" refer="rendelesID_PK">
        <xs:selector xpath="geprendeles"/>
        <xs:field xpath="@rendelesId"/>
    </xs:keyref>
    <xs:keyref name="specID_FK" refer="specifikacioID_PK">
        <xs:selector xpath="szamitogep"/>
        <xs:field xpath="@specId"/>
    </xs:keyref>
    <xs:keyref name="eladoId_FK" refer="eladoID_PK">
        <xs:selector xpath="rendeles"/>
        <xs:field xpath="@eladoId"/>
    </xs:keyref>
    <xs:keyref name="vasarloId_FK" refer="vasarloID_PK">
        <xs:selector xpath="rendeles"/>
        <xs:field xpath="@vasarloId"/>
    </xs:keyref>
</xs:element>

<xs:complexType name="telepulesType">
    <xs:sequence>
        <xs:element name="irsz" type="xs:unsignedShort" />
        <xs:element name="varos" type="xs:string" />
        <xs:element name="utca" type="xs:string" />
        <xs:element name="hazszam" type="xs:unsignedByte" />
        <xs:element name="emelet" type="xs:unsignedByte" />
        <xs:element name="ajto" type="xs:unsignedByte" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="geprendelesType">
    <xs:sequence>
        <xs:element name="darab" type="xs:unsignedByte" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte" use="required" />
    <xs:attribute name="szamitogepId" type="xs:unsignedByte" use="required" />
    <xs:attribute name="rendelesId" type="xs:unsignedByte" use="required" />
</xs:complexType>

<xs:complexType name="szamitogepType">
    <xs:sequence>
        <xs:element name="ar" type="xs:unsignedInt" />
        <xs:element name="elnevezes" type="xs:string" />
        <xs:element name="kategoria" type="xs:string" />
    </xs:sequence>

```

```

<xs:attribute name="id" type="xs:unsignedByte" use="required" />
<xs:attribute name="specId" type="xs:unsignedByte" use="required" />
</xs:complexType>
<xs:complexType name="specifikacioType">
  <xs:sequence>
    <xs:element name="alaplap" type="xs:string" />
    <xs:element name="videokartya" type="xs:string" />
    <xs:element name="memoria" type="xs:string" />
    <xs:element name="tapegyseg" type="xs:string" />
    <xs:element name="processzor" type="xs:string" />
    <xs:element name="tarhely" type="xs:string" />
    <xs:element name="egyeb" type="xs:string" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte" use="required" />
</xs:complexType>
<xs:complexType name="rendelesType">
  <xs:sequence>
    <xs:element name="datum" type="xs:string" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte" use="required" />
  <xs:attribute name="eladoId" type="xs:unsignedByte" use="required" />
  <xs:attribute name="vasarloId" type="xs:unsignedByte" use="required" />
</xs:complexType>
<xs:complexType name="eladoType">
  <xs:sequence>
    <xs:element name="nev" type="xs:string" />
    <xs:element name="email" type="xs:string" />
    <xs:element name="telszam" type="xs:long" />
    <xs:element name="telepules" type="xs:string" />
    <xs:element name="ertekeles" type="xs:unsignedByte" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte" use="required" />
</xs:complexType>
<xs:complexType name="vasarloType">
  <xs:sequence>
    <xs:element name="nev" type="xs:string" />
    <xs:element name="telefonszam" type="xs:long" />
    <xs:element name="email" type="xs:string" />
    <xs:element name="lackim" type="telepulesType">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte" use="required" />
  </xs:complexType>
</xs:schema>

```

2. feladat

a. DOM JAVA olvasás kommentekkel

```

package hu.domparse.wcd0qi;

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

public class DOMReadWCD0QI {

    public static void main(String[] args) {

        try {
            File xmlFile = new File("src/hu/domparse/wcd0qi/XMLWCD0QI.xml"); //fájl, amiből olvasunk
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //XML dokumentumból DOM objektum - lehetővé tétele
            DocumentBuilder dBuilder = factory.newDocumentBuilder(); //XML fájl, Document lekéréséhez
            Document doc = dBuilder.parse(xmlFile); //dokument lekérése
            doc.getDocumentElement().normalize();
            System.out.println("Számítógép rendelések lekérése");
            Read(doc); //fő metódus, meghívódik a Read
        } catch (ParserConfigurationException pce) {
            pce.printStackTrace();
        } catch (IOException ioe) {
            ioe.printStackTrace();
        } catch (SAXException sae) {
            sae.printStackTrace();
        }
    }

    public static void Read(Document doc) {

        NodeList nList = doc.getElementsByTagName("geprendeles"); //géprendelés taggal rendelkező elemek lekérése listába

        for (int i = 0; i < nList.getLength(); i++) { //listán végigmegyünk
            Node nNode = nList.item(i); //lekérjük a lista aktuális elemét, Elementé konvertáljuk
            Element element = (Element) nNode;
            //Lekérjük az attribútumokat, majd azok segítségével meghívjuk a definiált metódusokat
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                String darab = element.getElementsByTagName("darab").item(0).getTextContent(); //darabszám lekérdezése
                String szamitogepId = element.getAttribute("szamitogepId");
                String rendelesId = element.getAttribute("rendelesId");
                System.out.println("\n-----" + (i + 1) + ". rendelés-----");
                System.out.println("\n\tDarab:\t" + darab);
                ReadSzamitogepById(doc, szamitogepId);
                ReadRendelesById(doc, rendelesId);
            }
        }
    }

    //fa struktúra miatt az attribútumban megadott id alapján kérdezzük le az egyes rendeléshez tartozó elemeket
    // A legtöbb objektum rendelkezik leszármazottal, amelyet egy újabb metódus kérdez le, az attribútumban megadott ID alapján
    public static void ReadEladoById(Document doc, String id) {

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

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

            Node nNode = nList.item(i);
            Element element = (Element) nNode;

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("id").equals(id)) {
                    String name = element.getElementsByTagName("nev").item(0).getTextContent();
                    String email = element.getElementsByTagName("email").item(0).getTextContent();
                    String telszam = element.getElementsByTagName("telszam").item(0).getTextContent();
                    String telepules = element.getElementsByTagName("telepules").item(0).getTextContent();
                    String ertekeles = element.getElementsByTagName("ertekeles").item(0).getTextContent();
                    System.out.println("Eladó adatok: \n\tNév:\t" + name + "\n\tEmail:\t" + email + "\n\tTelefonszám:\t" + telszam
                        + "\n\tTelepülés:\t" + telepules + "\n\tElégedettségi érték:\t" + ertekeles);
                }
            }
        }
    }
}

```

```

}

public static void ReadSzamitogepById(Document doc, String id) {
    NodeList nList = doc.getElementsByTagName("szamitogep");

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

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

            if (element.getAttribute("id").equals(id)) {
                String ar = element.getElementsByTagName("ar").item(0).getTextContent();
                String elnevezes = element.getElementsByTagName("elnevezes").item(0).getTextContent();
                String kategoria = element.getElementsByTagName("kategoria").item(0).getTextContent();
                String specId = element.getAttribute("specId");
                System.out.println("Számítógép adatok: \n\tÁr:\t" + ar + "\n\tElnevezes:\t" + elnevezes + "\n\tKategoria:\t" + kategoria);
                ReadSpecifikacioById(doc, specId);
            }
        }
    }
}
}

```

```

public static void ReadRendelesById(Document doc, String id) {

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

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

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

            if (element.getAttribute("id").equals(id)) {
                String ar = element.getElementsByTagName("datum").item(0).getTextContent();

                String eladoId = element.getAttribute("eladoId");
                String vasarloId = element.getAttribute("vasarloId");

                System.out.println("Rendelés adatok: \n\tDátum:\t" + ar);
                ReadEladoById(doc, eladoId);
                ReadVasarloById(doc, vasarloId);
            }
        }
    }
}
}

```

```

public static void ReadVasarloById(Document doc, String id) {

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

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

        Node nNode = nList.item(i);
        Element element = (Element) nNode;

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

            if (element.getAttribute("id").equals(id)) {
                String name = element.getElementsByTagName("nev").item(0).getTextContent();
                String email = element.getElementsByTagName("email").item(0).getTextContent();
                String telszam = element.getElementsByTagName("telefonszam").item(0).getTextContent();
                String irsz = element.getElementsByTagName("irsz").item(0).getTextContent();
            }
        }
    }
}

```

```

        String varos = element.getElementsByTagName("varos").item(0).getTextContent();
        String utca = element.getElementsByTagName("utca").item(0).getTextContent();
        String hazszam = element.getElementsByTagName("hazszam").item(0).getTextContent();
        String emelet = element.getElementsByTagName("emelet").item(0).getTextContent();
        String ajto = element.getElementsByTagName("ajto").item(0).getTextContent();

        System.out.println("Eladó adatok: \n\tNév:\t" + name + "\n\tEmail:\t" + email + "\n\tTelefonszám:\t" + telszam
            + "\n\tLakcím:\t" + irsz + ',' + varos + ',' + utca + " utca," + hazszam + "(emelet:" + emelet + " ajto:" + ajto + ")");
    }
}
}
}

public static void ReadSpecifikacioById(Document doc, String id) {

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

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

        Node nNode = nList.item(i);
        Element element = (Element) nNode;

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

            if (element.getAttribute("id").equals(id)) {
                String alaplap = element.getElementsByTagName("alaplap").item(0).getTextContent();
                String videokartya = element.getElementsByTagName("videokartya").item(0).getTextContent();
                String memoria = element.getElementsByTagName("memoria").item(0).getTextContent();
                String tapegyseg = element.getElementsByTagName("tapegyseg").item(0).getTextContent();
                String processzor = element.getElementsByTagName("processzor").item(0).getTextContent();
                String tarhely = element.getElementsByTagName("tarhely").item(0).getTextContent();
                String egyeb = element.getElementsByTagName("egyeb").item(0).getTextContent();

                System.out.println("Specifikacio adatok: \n\tAlaplap:\t" + alaplap + "\n\tVideokartya:\t" + videokartya + "\n\tMemoria:\t" +
                    memoria
                        + "\n\tTapegyseg:\t" + tapegyseg + "\n\tProcesszor:\t" + processzor + "\n\tTarhely:\t" + tarhely + "\n\tEgyéb:\t" + egyeb);
            }
        }
    }
}
}
}
}

```

b. DOM JAVA módosítás kód kommentekkel

```

package hu.domparse.wcd0qi;

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.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.io.IOException;
import java.util.Scanner;

```

```

public class DOMModifyWCD0QI {

    public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException, TransformerException {
        File xmlFile = new File("src/hu/domparse/wcd0qi/XMLWCD0QI.xml"); //xml fájl bekérése
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //olvasás lehetővé tétele
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        System.out.println("XML Módosítása");
        System.out.println("Adja meg mit szeretne módosítani: ");
        System.out.println("1 Specifikacio adatok módosítása\n2 Szamitogep alap adatok módosítása\n3 Geprendelés darabszámának módosítása\n4 Vasarlo adatainak módosítása");
        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("src/hu/domparse/wcd0qi/XMLWCD0QI.xml"));
        transformer.transform(source, result);
    }
    //Modify-ban a felhasználót megkérdezzük, hogy a rendelés mely adatát kívánja módosítani
    public static void Modify(Document doc) throws TransformerException {
        int rendelesekSzama = doc.getElementsByTagName("geprendeles").getLength(); //geprendelesek számának lekérdezése
        int gepekSzama = doc.getElementsByTagName("szamitogep").getLength(); //szamitogepek számának lekérdezése
        int vasarlokSzama = doc.getElementsByTagName("vasarlo").getLength(); //vasarlok számának lekérdezése
        int specifikaciokSzama = doc.getElementsByTagName("specifikacio").getLength(); //specifikacio számának lekérdezése
        Scanner scan = new Scanner(System.in);
        System.out.println("Adja meg a sorszámot: ");
        int readCategory = scan.nextInt();

        switch (readCategory) {
            case 1:
                ModifySpecification(doc, specifikaciokSzama);
                break;
            case 2:
                ModifySzamitogep(doc, gepekSzama);
                break;
            case 3:
                Geprendeles(doc, rendelesekSzama);
                break;
            case 4:
                ModifyVasarlo(doc, vasarlokSzama);
                break;
        }
    }

    private static void ModifyVasarlo(Document doc, int rendelesszam) throws TransformerException {
        //Kiirattuk a jelenlegi vásárlókat, majd lekérdezzük melyiket kívánja módosítani.
        System.out.println("Melyik vásárló adatait szeretné módosítani?");
        for (int i = 1; i < rendelesszam+1; i++) {
            System.out.println(i + ". vásárló");
            DOMReadWCD0QI.ReadVasarloById(doc, String.valueOf(i));
            System.out.println("-----");
        }
        String id = ReadId();
        //Bekérjük az új adatokat
        Scanner sc = new Scanner(System.in);
        System.out.print("Név ");
        String nev = sc.nextLine();
        System.out.print("Telefonszám: ");
    }
}

```

```

String telefonszam = sc.nextLine();
System.out.print("E-mail cím: ");
String email = sc.nextLine();
System.out.print("Írányítószám: ");
String irsz = sc.nextLine();
System.out.print("Város: ");
String varos = sc.nextLine();
System.out.print("Utca: ");
String utca = sc.nextLine();
System.out.print("Házszám: ");
String hazszam = sc.nextLine();
System.out.print("Emelet: ");
String emelet = sc.nextLine();
System.out.print("Ajtó: ");
String ajto = sc.nextLine();
//lekérdezzük az Elementeket, majd setTextColor-el módosítjuk.
NodeList nodeList = doc.getElementsByTagName("vasarlo");
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("id");
        if (sid.equals(id)) {
            Node node1 = element.getElementsByTagName("nev").item(0);
            node1.setTextNode(nev);

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

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

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

            Node node5 = element.getElementsByTagName("varos").item(0);
            node5.setTextNode(varos);
            Node node6 = element.getElementsByTagName("utca").item(0);
            node6.setTextNode(utca);
            Node node7 = element.getElementsByTagName("hazszam").item(0);
            node7.setTextNode(hazszam);
            Node node8 = element.getElementsByTagName("emelet").item(0);
            node8.setTextNode(emelet);
            Node node9 = element.getElementsByTagName("ajto").item(0);
            node9.setTextNode(ajto);

            System.out.println("Sikeres módosítás");
        }
    }
}
ModifyXML(doc); //Létrehozzuk az XML-t
}

private static void Geprendeles(Document doc, int rendleaszam) throws TransformerException {
    System.out.println("Melyik rendelés darabszámát kívánja módosítani?");
    DOMReadWCD0QI.Read(doc);

    String id = ReadId();

    Scanner sc = new Scanner(System.in);
    System.out.print("Darabszám: ");
    String db = sc.nextLine();

```

```

NodeList nodeList = doc.getElementsByTagName("geprendeles");
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("id");
        if (sid.equals(id)) {
            Node node1 = element.getElementsByTagName("darab").item(0);
            node1.setTextContent(db);
            System.out.println("Sikeres módosítás");
        }
    }
}
ModifyXML(doc);
}

private static void ModifySzamitogep(Document doc, int rendeleesszam) throws TransformerException {
    System.out.println("Melyik számítógépet szeretné módosítani?");
    for (int i = 1; i < rendeleesszam+1; i++) {
        System.out.println(i + ". számítógép");
        DOMReadWCD0QI.ReadSzamitogepById(doc, String.valueOf(i));
        System.out.println("-----");
    }
    String id = ReadId();

    Scanner sc = new Scanner(System.in);
    System.out.print("Ár");
    String ar = sc.nextLine();
    System.out.print("Elnevezés: ");
    String elnevezes = sc.nextLine();
    System.out.print("Kategória: ");
    String kategoria = sc.nextLine();

    NodeList nodeList = doc.getElementsByTagName("szamitogep");
    for (int i = 0; i < rendeleesszam; i++) {
        Node nNode = nodeList.item(i);

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;

            String sid = element.getAttribute("id");
            if (sid.equals(id)) {
                Node node1 = element.getElementsByTagName("ar").item(0);
                node1.setTextContent(ar);

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

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

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

public static String ReadId() {
    Scanner sc = new Scanner(System.in);

```

```

        System.out.println("\nid:");
        String id = sc.nextLine();
        return id;
    }

    private static void ModifySpecification(Document doc, int rendelesSzam) throws TransformerException {
        System.out.println("Melyik specifikációt szeretné módosítani?");
        for (int i = 0; i <= rendelesSzam; i++) {
            System.out.println(i + ". specifikacio");
            DOMReadWCD0QI.ReadSpecifikacioById(doc, String.valueOf(i));
            System.out.println("-----");
        }
        String id = ReadId();

        Scanner sc = new Scanner(System.in);
        System.out.print("Alaplap ");
        String alaplap = sc.nextLine();
        System.out.print("Videókártya ");
        String videokoartya = sc.nextLine();
        System.out.print("Memória: ");
        String memoria = sc.nextLine();
        System.out.print("Tápegység: ");
        String tapegyseg = sc.nextLine();
        System.out.print("Processzor: ");
        String processzor = sc.nextLine();
        System.out.print("Tárhely: ");
        String tarhely = sc.nextLine();
        System.out.print("Egyeb: ");
        String egyeb = sc.nextLine();

        NodeList nodeList = doc.getElementsByTagName("specifikacio");
        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("id");
                if (sid.equals(id)) {
                    Node node1 = element.getElementsByTagName("alaplap").item(0);
                    node1.setTextContent(alaplap);

                    Node node2 = element.getElementsByTagName("videokartya").item(0);
                    node2.setTextContent(videokoartya);

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

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

                    Node node5 = element.getElementsByTagName("processzor").item(0);
                    node5.setTextContent(processzor);
                    Node node6 = element.getElementsByTagName("tarhely").item(0);
                    node6.setTextContent(tarhely);
                    Node node7 = element.getElementsByTagName("egyeb").item(0);
                    node7.setTextContent(egyeb);

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

```

}