

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Bolt nyilvántartó redszere

Készítette: **Szivós Ádám**

Nepunkód: **B9ZL3Q**

1. A feladat leírása:

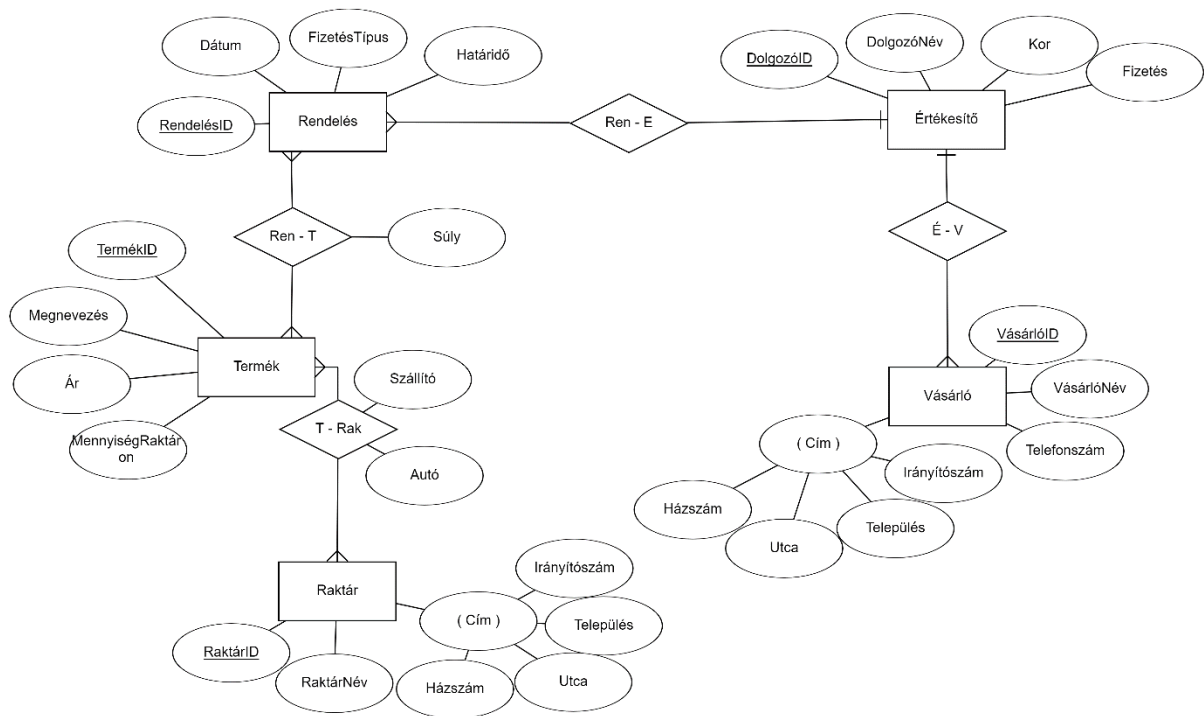
Én egy bolt nyilvántartó rendszeréről készítettem a beadandómat.

5 objektumból áll, köztük a *vásárló*, *értékesítő*, *rendelés*, *termék*, *raktár*. Mind több attribútummal rendelkeznek. A vásárló elemnek van egy neve, telefonszáma és címe, ezek alapján tudja a bolt azonosítani a vevőt. Az értékesítőnek is szerepelnek az alapdatai, amik a cégnek kellenek. Ezek a név, a kor és a fizetés. A vásárlót és az értékesítőt egy 1:N kapcsolat köt össze. A rendelés elemnek van dátuma, teljesítési ideje és egy fizetés típusa attribútuma. A dátumot String-ként tárolom és csak informatív jellegű. Ez is egy 1:N kapcsolattal csatlakozik az értékesítőhöz, ugyanis egy eladónak lehet több rendelése. A fizetés típusa a szállítónak hasznos, hogy a kiszállítás végén számíttania kell-e valamilyen tranzakcióra. Ezt az elemet az értékesítővel szintén egy 1:N kapcsolat köti össze. A terméknek is van neve, azon kívül egy ára és egy darabszáma. Ezt egy N:N kapcsolat köti össze a rendelés elemmel, ami el van látva a bolt által számított súly attribútummal. Végül a raktár elem, aminek van egy neve és egy címe. Ezt is egy N:N kapcsolat köti össze a termékkel, de ennek a kapcsolatnak már 2 attribútuma van: egy autó azonosító és egy szállító név.

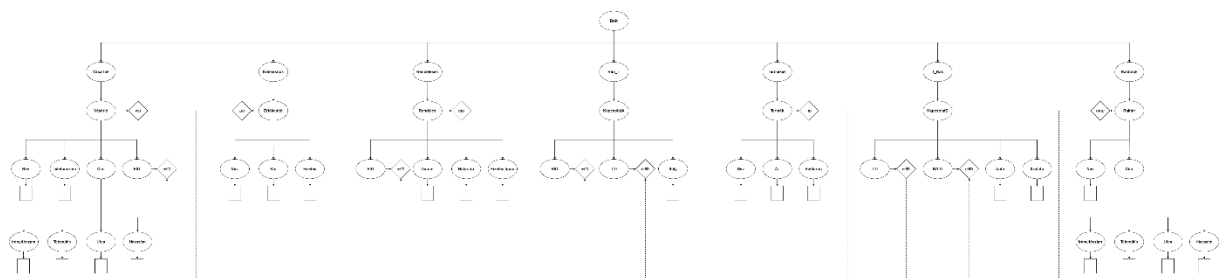
Soron lehet követni egy tranzakció lefolyását a vásárlótól a kiszállításig.

1. feladat

1a, Az adatbázis ER modell:



1b, Az adatbázis konvertálása XDM modellre:



1c, Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-model href="XMLSchemaB9ZL3Q.xsd" type="application/xml"
schematypens="http://www.w3.org/2001/XMLSchema"?>

<bolt>
  <vasarlok>
    <vasarlo vid="v1">
      <vnev>Vas Alex</vnev>
      <telefonszam>06201234566</telefonszam>
      <lakcim>
        <iranyitoszam>3333</iranyitoszam>
        <telepules>Abaujteleki</telepules>
        <utca>Petofi</utca>
        <hazszam>2</hazszam>
      </lakcim>
      <eid refID="e1"/>
    </vasarlo>

    <vasarlo vid="v2">
      <vnev>Kis Bela</vnev>
      <telefonszam>06704446667</telefonszam>
      <lakcim>
        <iranyitoszam>1463</iranyitoszam>
        <telepules>Peterujfalva</telepules>
        <utca>Fo</utca>
        <hazszam>33</hazszam>
      </lakcim>
      <eid refID="e2"/>
    </vasarlo>

    <vasarlo vid="v3">
      <vnev>Kovacs Nora</vnev>
      <telefonszam>06307653486</telefonszam>
      <lakcim>
        <iranyitoszam>3325</iranyitoszam>
        <telepules>Felsőignac</telepules>
        <utca>Pesti</utca>
        <hazszam>13</hazszam>
      </lakcim>
      <eid refID="e3"/>
    </vasarlo>
  </vasarlok>

  <ertekesitok>
    <ertekesito eid="e1">
      <enev>Petting Elek</enev>
```

```
<kor>34</kor>
<fizetes>21000</fizetes>
</ertekesito>

<ertekesito eid="e2">
  <enev>Kiss Tivadar</enev>
  <kor>55</kor>
  <fizetes>27000</fizetes>
</ertekesito>

<ertekesito eid="e3">
  <enev>Laszlo Evelin</enev>
  <kor>21</kor>
  <fizetes>19000</fizetes>
</ertekesito>
</ertekesitok>

<rendelesek>
  <rendeles rid="r1">
    <datum>2021.10.12.</datum>
    <hatarido>2022.02.01.</hatarido>
    <fiztip>KP</fiztip>
    <eid refID="e1"/>
  </rendeles>

  <rendeles rid="r2">
    <datum>2021.12.15.</datum>
    <hatarido>2022.01.08.</hatarido>
    <fiztip>utalas</fiztip>
    <eid refID="e2"/>
  </rendeles>

  <rendeles rid="r3">
    <datum>2021.09.01.</datum>
    <hatarido>2022.12.23.</hatarido>
    <fiztip>kartya</fiztip>
    <eid refID="e3"/>
  </rendeles>
</rendelesek>

<termek>
  <termek tid="t1">
    <tnev>Zold 3-lyuku feltegl</tnev>
    <raktaron>166</raktaron>
    <ar>11000</ar>
  </termek>

  <termek tid="t2">
    <tnev>Piros 8-lyuku feltegl</tnev>
```

```
<raktaron>300</raktaron>
<ar>553</ar>
</termek>

<termek tid="t3">
  <tnev>Ekru 4-lyuku felteglá</tnev>
  <raktaron>29</raktaron>
  <ar>99999</ar>
</termek>
</termekek>

<raktarak>
  <raktar raid="ra1">
    <rnev>Zsolcai telephely</rnev>
    <cim>
      <iranyitoszam>2263</iranyitoszam>
      <telepules>Alszolca</telepules>
      <utca>Miskolci</utca>
      <hazszam>43</hazszam>
    </cim>
  </raktar>

  <raktar raid="ra2">
    <rnev>Kapolnai raktar</rnev>
    <cim>
      <iranyitoszam>4325</iranyitoszam>
      <telepules>Kapolna</telepules>
      <utca>Ersek</utca>
      <hazszam>1</hazszam>
    </cim>
  </raktar>

  <raktar raid="ra3">
    <rnev>Pesti telephely</rnev>
    <cim>
      <iranyitoszam>1111</iranyitoszam>
      <telepules>Budapest</telepules>
      <utca>Ipari park 2</utca>
      <hazszam>2</hazszam>
    </cim>
  </raktar>
</raktarak>

<ren_t>
  <kapcsolatA rentid="ka1">
    <idrendeles refID="r1"/>
    <idtermek refID="t1"/>
    <suly>75638</suly>
  </kapcsolatA>
```

```
<kapcsolatA rentid="ka2">
  <idrendeles refID="r2"/>
  <idtermek refID="t2"/>
  <suly>537366</suly>
</kapcsolatA>

<kapcsolatA rentid="ka3">
  <idrendeles refID="r3"/>
  <idtermek refID="t3"/>
  <suly>275399</suly>
</kapcsolatA>
</ren_t>

<t_rak>
  <kapcsolatB trakid="ka1">
    <idtermek refID="t1"/>
    <idraktar refID="ra1"/>
    <auto>rg-324</auto>
    <szallito>Kis Tivadar</szallito>
  </kapcsolatB>

  <kapcsolatB trakid="ka2">
    <idtermek refID="t2"/>
    <idraktar refID="ra2"/>
    <auto>frg-123</auto>
    <szallito>Nagy Peter</szallito>
  </kapcsolatB>

  <kapcsolatB trakid="ka3">
    <idtermek refID="t3"/>
    <idraktar refID="ra3"/>
    <auto>mnz-734</auto>
    <szallito>Toth Laszlo</szallito>
  </kapcsolatB>
</t_rak>

</bolt>
```

1d, Az XML dokumentum alapján XMLSchema készítése:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"    elementFormDefault="qualified">

    <xs:element name="bolt">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="vasarlok" type="vasarloktype"
maxOccurs="1"/>
                <xs:element name="ertekesitok" type="ertekesitoktype"
maxOccurs="1"/>
                <xs:element name="rendelesek" type="rendelesektype"
maxOccurs="1"/>
                <xs:element name="termek" type="termektype"
maxOccurs="1"/>
                <xs:element name="raktarak" type="raktaraktype"
maxOccurs="1"/>
                <xs:element name="ren_t" type="ren_ttype" maxOccurs="1"/>
                <xs:element name="t_rak" type="t_raktype" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>

        <xs:key name="vid">
            <xs:selector xpath="vasarlo"/>
            <xs:field xpath="@vid"/>
        </xs:key>
        <xs:key name="eid">
            <xs:selector xpath="ertekesito"/>
            <xs:field xpath="@eid"/>
        </xs:key>
        <xs:key name="rid">
            <xs:selector xpath="rendeles"/>
            <xs:field xpath="@rid"/>
        </xs:key>
        <xs:key name="tid">
            <xs:selector xpath="termek"/>
            <xs:field xpath="@tid"/>
        </xs:key>
        <xs:key name="raid">
            <xs:selector xpath="raktar"/>
            <xs:field xpath="@raid"/>
        </xs:key>
        <xs:keyref name="refeid" refer="eid">
            <xs:selector xpath="rendeles/eid"/>
            <xs:field xpath="@refID"/>
        </xs:keyref>
        <xs:keyref name="refeid" refer="eid">
```



```

        <xs:selector xpath="vasarlo/eid"/>
        <xs:field xpath="@refID"/>
    </xs:keyref>
    <xs:keyref name="refeid" refer="eid">
        <xs:selector xpath="rendeles/eid"/>
        <xs:field xpath="@refID"/>
    </xs:keyref>
    <xs:keyref name="refidtermek" refer="refidtermek">
        <xs:selector xpath="ren_t/refidtermek"/>
        <xs:field xpath="@refID"/>
    </xs:keyref>
    <xs:keyref name="refidrendeles" refer="refidrendeles">
        <xs:selector xpath="ren_t/refidrendeles"/>
        <xs:field xpath="@refID"/>
    </xs:keyref>
    <xs:keyref name="refidtermek" refer="idtermek">
        <xs:selector xpath="t_rak/idtermek"/>
        <xs:field xpath="@refID"/>
    </xs:keyref>
    <xs:keyref name="refidraktar" refer="idraktar">
        <xs:selector xpath="t_rak/idraktar"/>
        <xs:field xpath="@refID"/>
    </xs:keyref>
</xs:element>

<xs:complexType name="vasarloktype">
    <xs:sequence>
        <xs:element name="vasarlo" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="vnev" type="xs:string"/>
                    <xs:element name="telefonszam" type="xs:string"/>
                    <xs:element name="lakcim" type="cimtype"/>
                    <xs:element name="eid">
                        <xs:complexType>
                            <xs:attribute name="refID" type="xs:IDREF"/>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="vid" type="xs:ID" use="required"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="ertekesitoktype">
    <xs:sequence>
        <xs:element name="ertekesito" maxOccurs="unbounded">
            <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="nev" type="xs:string"/>
            <xs:element name="kor" type="xs:integer"/>
            <xs:element name="fizetes" type="xs:integer"/>
        </xs:sequence>
        <xs:attribute name="eid" type="xs:ID" use="required"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="rendelesektype">
    <xs:sequence>
        <xs:element name="rendeles" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="datum" type="xs:string"/>
                    <xs:element name="hatarido" type="xs:string"/>
                    <xs:element name="fiztip" type="xs:string"/>
                    <xs:element name="eid">
                        <xs:complexType>
                            <xs:attribute name="refID" type="xs:IDREF"/>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="rid" type="xs:ID" use="required"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="termekektype">
    <xs:sequence>
        <xs:element name="termek" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="tnev" type="xs:string"/>
                    <xs:element name="raktaron" type="xs:integer"/>
                    <xs:element name="ar" type="xs:integer"/>
                </xs:sequence>
                <xs:attribute name="tid" type="xs:ID" use="required"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="raktaraktype">
    <xs:sequence>
        <xs:element name="raktar" maxOccurs="unbounded">

```



```
        <xs:complexType>
            <xs:attribute name="refID" type="xs:IDREF"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="idraktar">
        <xs:complexType>
            <xs:attribute name="refID" type="xs:IDREF"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="autoid" type="xs:string"/>
    <xs:element name="szallito" type="xs:string"/>
</xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

2. feladat

2a, Adatolvasás:

Beolvassa és kiírja az xml elemeit.

```
package hu.domparse.b9zl3q;

import java.io.*;
import javax.xml.parsers.*;
import org.xml.sax.SAXException;
import org.w3c.dom.*;

public class DOMReadB9ZL3Q {

    public static void main(String[] args) {
        // DOMBuilder-rel DOM letrheozasa
        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(new
File("XMLB9ZL3Q.xml"));
            document.getDocumentElement().normalize();
            //root elem megadasa
            Node root = document.getDocumentElement();
            listAll(root);
        } catch (ParserConfigurationException | SAXException | IOException e)
{
            e.printStackTrace();
        }
    }

    public static void listAll(Node root) {
        //a child elemek megadasa
        NodeList nodeList = root.getChildNodes();
        String vasarlo = "Vasarlok\n\n";
        String ertekesito = "Ertekesitok\n\n";
        String rendeles = "Rendelesek\n\n";
        String termek = "Termek\n\n";
        String raktar = "Raktarak\n\n";
        String kapcsolatA = "ren_t\n\n";
        String kapcsolatB = "t_rak\n\n";
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                //a child elemek child-jainak kivalassztasa
                NodeList subNodeList = nodeList.item(i).getChildNodes();
```

```

        for (int j = 0; j < subNodeList.getLength(); j++) {
            Node subNode = subNodeList.item(j);
            //kigyujtes a definialt string-ekbe
            if(subNode.getNodeType() == Node.ELEMENT_NODE) {
                vasarlo += oneNode(subNode,"vasarlo");
                ertekesito += oneNode(subNode,"ertekesito");
                rendeles += oneNode(subNode,"rendeles");
                termek += oneNode(subNode,"termek");
                raktar += oneNode(subNode,"raktar");
                kapcsolatA += oneNode(subNode,"ren_t");
                kapcsolatB += oneNode(subNode,"t_rak");
            }
        }
    }
    //kiiratas
    System.out.println(vasarlo);
    System.out.println(ertekesito);
    System.out.println(rendeles);
    System.out.println(termek);
    System.out.println(raktar);
    System.out.println(kapcsolatA);
    System.out.println(kapcsolatB);
}

public static String oneNode(Node subNode, String x) {
    String out = "";
    if(subNode.getNodeName().equals(x)) {
        if(subNode.getAttributes().getLength()>0)
            //ha a node megegyezik x-el és van attribute-ja, akkor kiiruk
            {
                out += x + " ID : " +
subNode.getAttributes().item(0).getTextContent()+"\n";
            }
        NodeList subSubNodeList = subNode.getChildNodes();
        for (int k = 0; k < subSubNodeList.getLength(); k++) {
            Node subSubNode = subSubNodeList.item(k);
            if(subSubNode.getNodeType() == Node.ELEMENT_NODE) {
                //id alapjan elkuloniti
                switch(subSubNode.getNodeName()) {
                    case "vid":
                    case "eid":
                    case "rid":
                    case "tid":
                    case "raid":
                    //case "refID":
                        out += subSubNode.getNodeName() + " : " +
subSubNode.getAttributes().item(0).getTextContent() + "\n";
                        break;
                }
            }
        }
    }
}

```

```

        //a child elemek komplex eleme
        case "cim":
        case "lakcim":
            NodeList subSubSubNode = subSubNode.getChildNodes();
            for (int l = 0; l < subSubSubNode.getLength(); l++) {
                if (subSubSubNode.item(l).getNodeType() ==
Node.ELEMENT_NODE) {
                    out += subSubNode.getNodeName() + "-" +
subSubSubNode.item(l).getNodeName() + " : " +
subSubSubNode.item(l).getTextContent()+"\n";
                }
            }
            break;
        default:
            //az elem nevet es a kontextust String-be irja
            out += subSubNode.getNodeName() + " : " +
subSubNode.getTextContent()+"\n";
        }
    }
    out += "\n";
}
return out;
}
}

```

Output:

Vasarlok

vasarlo ID : v1

vnev : Vas Alex

telefonszam : 06201234566

lakcim-iranyitoszam : 3333

lakcim-telepules : Abaujteleki

lakcim-utca : Petofi

lakcim-hazszam : 2

eid : e1

vasarlo ID : v2

vnev : Kis Bela

telefonszam : 06704446667

lakcim-iranyitoszam : 1463

lakcim-telepules : Peterujfalva

lakcim-utca : Fo

lakcim-hazszam : 33

eid : e2

vasarlo ID : v3

vnev : Kovacs Nora

telefonszam : 06307653486

lakcim-iranyitoszam : 3325

lakcim-telepules : Felsőignac

lakcim-utca : Pesti

lakcim-hazszam : 13

eid : e3

Ertekesitok

ertekesito ID : e1

enev : Petting Elek

kor : 34

fizetes : 210000

ertekesito ID : e2

enev : Kiss Tivadar

kor : 55

fizetes : 270000

ertekesito ID : e3

enev : Laszlo Evelin

kor : 21

fizetes : 190000

Rendelesek

rendeles ID : r1

datum : 2021.10.12.

hatarido : 2022.02.01.

fiztip : KP

eid : e1

rendeles ID : r2

datum : 2021.12.15.

hatarido : 2022.01.08.

fiztip : utalas

eid : e2

rendeles ID : r3

datum : 2021.09.01.

hatarido : 2022.12.23.

fiztip : kartya

eid : e3

Termekek

termek ID : t1

tnev : Zold 3-lyuku feltegl

raktaron : 166

ar : 11000

termek ID : t2

tnev : Piros 8-lyuku feltegl

raktaron : 300

ar : 553

termek ID : t3

tnev : Ekru 4-lyuku feltegl

raktaron : 29

ar : 99999

Raktarak

raktar ID : ra1

rnev : Zsolcai telephely

cim-iranyitoszam : 2263

cim-telepules : Alsozsolca

cim-utca : Miskolci

cim-hazszam : 43

raktar ID : ra2

rnev : Kapolnai raktar

cim-iranyitoszam : 4325

cim-telepules : Kapolna

cim-utca : Ersek

cim-hazszam : 1

raktar ID : ra3

rnev : Pesti telephely

cim-iranyitoszam : 1111

cim-telepules : Budapest

cim-utca : Ipari park 2

cim-hazszam : 2

ren_t

suly : 75638

suly : 537366

suly : 275399

t_rak

auto : rgh-324

szallito : Kis Tivadar

auto : frg-123

szallito : Nagy Peter

auto : mnz-734

szallito : Toth Laszlo

2b, Adatmódosítás:

Itt beolvasom a módosítani kívánt elemeket, majd kiíratom a XMLB9ZL3Q.out.xml fájlba. A kóddal módosítható elemek előre megadott ID alapján:

- termékek ára
- vásárló telefonszáma
- raktár neve

```
package hu.domparse.b9zl3q;

import java.io.*;
import java.util.*;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.*;
import org.xml.sax.SAXException;

public class DOMModifyB9ZL3Q {

    public static void main(String[] args){
        // DOMBuilder-rel DOM letrheozasa
        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(new
File("XMLB9ZL3Q.xml"));
            //normalizalas
            document.getDocumentElement().normalize();
            Scanner scanner = new Scanner(System.in);
            boolean exit=true;
            while(exit) {
                //switch case segitsegevel menu
                System.out.println("0: Kilepes es vegrehajtas\n1: Termek ar
mododsitas\n2: Vasarlo telefonszam mododsitas\n3: Raktar nev mododsitas");
                switch(scanner.nextInt()) {
                    case 1 : termAr(document, scanner);break;
                    case 2 : vasTel(document, scanner);break;
                    case 3 : rakNev(document, scanner);break;
                    case 0 : exit=false;
                    default : exit=false;
                }
            }
            scanner.close();
            //kiirja az uj doksiba
            writeToXml(document);
        }
    }
}
```

```

        } catch (ParserConfigurationException | SAXException | IOException |
TransformerException | InputMismatchException e) {
            e.printStackTrace();
        }
    }

    public static void termAr(Document document, Scanner scanner) {
        //kikeresi tag alapjan
        NodeList nodeList = document.getElementsByTagName("termek");
        System.out.println("Termek ID: ");
        String input = scanner.next();
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                //kikeresi id alapjan
                if
(node.getAttributes().getNamedItem("tid").getTextContent().equals(input)) {
                    //lekeri a child node-okat
                    NodeList subNodeList = node.getChildNodes();
                    for (int j = 0; j < subNodeList.getLength(); j++) {
                        Node subNode = subNodeList.item(j);
                        if (subNode.getNodeName().equals("ar")) {
                            System.out.println("Uj ar: ");
                            subNode.setTextContent(scanner.next());
                        }
                    }
                }
            }
        }
    }

    public static void vasTel(Document document, Scanner scanner) {
        //kikeresi tag alapjan
        NodeList nodeList = document.getElementsByTagName("vasarlo");
        System.out.println("Vasarlo ID: ");
        String input = scanner.next();
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                //kikeresi id alapjan
                if
(node.getAttributes().getNamedItem("vid").getTextContent().equals(input)) {
                    //lekeri a child node-okatv
                    NodeList subNodeList = node.getChildNodes();
                    for (int j = 0; j < subNodeList.getLength(); j++) {
                        Node subNode = subNodeList.item(j);
                        if (subNode.getNodeName().equals("telefonszam")) {
                            System.out.println("Uj telefonszam: ");
                            subNode.setTextContent(scanner.next());
                        }
                    }
                }
            }
        }
    }

```

```

    }
    }
    }
}

public static void rakNev(Document document, Scanner scanner) {
    //kikeresi tag alapjan
    NodeList nodeList = document.getElementsByTagName("raktar");
    System.out.println("Raktar ID: ");
    String input = scanner.next();
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            //kikeresi id alapjan
            if
(node.getAttributes().getNamedItem("raid").getTextContent().equals(input)) {
                //lekeri a child node-okat
                NodeList subNodeList = node.getChildNodes();
                for (int j = 0; j < subNodeList.getLength(); j++) {
                    Node subNode = subNodeList.item(j);
                    if (subNode.getNodeName().equals("rnev")) {
                        System.out.println("Uj nev: ");
                        String name = scanner.next();
                        name += scanner.nextLine();
                        subNode.setTextContent(name);
                    }
                }
            }
        }
    }
}

//kiiras xml-be
public static void writeToXml(Document document) throws
TransformerException, UnsupportedEncodingException {
    TransformerFactory transformerFactory =
TransformerFactory.newInstance();
    Transformer transf = transformerFactory.newTransformer();
    DOMSource source = new DOMSource(document);
    StreamResult console = new StreamResult(System.out);
    StreamResult file = new StreamResult(new File("XMLB9ZL3Q.out.xml"));
    transf.transform(source, console);
    transf.transform(source, file);
}
}

```

Működése:

Kapunk egy menüt:

0: Kilepes es vegrehajtas

1: Termek ar mododsitas

2: Vasarlo telefonszam mododsitas

3: Raktar nev mododsitas

Ezek után a bevitel alapján folytatódik a futás.

1

Termek ID:

t1

Uj ar:

3333

Eredmény:

Régi:

```
<termek tid="t1">
  <tnev>Zold 3-lyuku felteгла</tnev>
  <raktaron>166</raktaron>
  <ar>11000</ar>
</termek>
```

Új:

```
<termek tid="t1">
  <tnev>Zold 3-lyuku felteгла</tnev>
  <raktaron>166</raktaron>
  <ar>3333</ar>
</termek>
```

2c, Adat lekérdezés:

Kiírja az általunk feltétellel megadott elemeket. Nekem ezek a:

- azon termékek amelyeknek az ára nagyobb, mint a felhasználó által megadott szám
- a pontosan egyező határidővel ellátott rendelések
- az adott kornál idősebb értékesítők

```
package hu.domparse.b9zl3q;

import java.io.*;
import java.util.*;

import javax.xml.parsers.*;
import org.w3c.dom.*;
import org.xml.sax.SAXException;

public class DOMQueryB9ZL3Q {

    public static void main(String[] args){
        // DOMBuilder-rel DOM letrheozasa
        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(new
File("XMLB9ZL3Q.xml"));
            //normalizalas
            document.getDocumentElement().normalize();
            Scanner scanner = new Scanner(System.in);
            boolean exit=true;
            while(exit) {
                //switch case segitsegevel menu
                System.out.println("0: Kilepes\n1: Bizonyos ar feletti termek
\n2: Rendeles egyezo hatarido\n3: Adott kornal idosebb elado ");
                switch(scanner.nextInt()) {
                    case 1 : termAr(document, scanner);break;
                    case 2 : rendHat(document, scanner);break;
                    case 3 : ertKor(document, scanner);break;
                    case 0 : exit=false;
                    default : exit=false;
                }
            }
            scanner.close();
        } catch (ParserConfigurationException | SAXException | IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

```

}

//termek aranak megvaltoztatasa
public static void termAr(Document document,Scanner scanner) {
    //kikeresi tag alapjan
    NodeList nodeList = document.getElementsByTagName("termek");
    System.out.println("Adja meg minel nagyobb: ");
    Long input = scanner.nextLong();
    System.out.println("\n");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if(node.getNodeType() == Node.ELEMENT_NODE) {
            //lekeri a child node-okat
            NodeList subNodeList = node.getChildNodes();
            for (int j = 0; j < subNodeList.getLength(); j++) {
                Node subNode = subNodeList.item(j);
                if(subNode.getNodeType() == Node.ELEMENT_NODE) {
                    if (subNode.getNodeName().equals("ar") )
                        if (Long.parseLong(subNode.getTextContent()) > input)
                            listSub(node, "Termek");
                }
            }
        }
    }
}

public static void rendHat(Document document,Scanner scanner) {
    //kikeresi tag alapjan
    NodeList nodeList = document.getElementsByTagName("rendeles");
    System.out.println("Adja meg a hataridot: ");
    String input = scanner.next();
    System.out.println("\n");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if(node.getNodeType() == Node.ELEMENT_NODE) {
            //lekeri a child node-okat
            NodeList subNodeList = node.getChildNodes();
            for (int j = 0; j < subNodeList.getLength(); j++) {
                Node subNode = subNodeList.item(j);
                if(subNode.getNodeType() == Node.ELEMENT_NODE) {
                    if (subNode.getNodeName().equals("hatarido") )
                        if (subNode.getTextContent().equals(input))
                            listSub(node, "Rendeles");
                }
            }
        }
    }
}
}

```



```

public static void ertKor(Document document, Scanner scanner) {
    //kikeresi tag alapjan
    NodeList nodeList = document.getElementsByTagName("ertesito");
    System.out.println("Adja meg a kort: ");
    int input = scanner.nextInt();
    System.out.println("\n");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            //lekeri a child node-okat
            NodeList subNodeList = node.getChildNodes();
            for (int j = 0; j < subNodeList.getLength(); j++) {
                Node subNode = subNodeList.item(j);
                if (subNode.getNodeType() == Node.ELEMENT_NODE) {
                    if (subNode.getNodeName().equals("kor") )
                        if (Integer.parseInt(subNode.getTextContent()) >
input)
                            listSub(node, "Ertesito");
                }
            }
        }
    }
}

private static void listSub(Node node, String tag) {
    //id alapjan az elemek kilistazasa
    String out="";
    out +=tag + " ID : " +
node.getAttribute().item(0).getTextContent()+"\n";
    Element element = (Element) node;
    switch (tag) {
        case "Termek":
            out +=
element.getElementsByTagName("tnév").item(0).getNodeName() + " : "
+element.getElementsByTagName("tnév").item(0).getTextContent()+"\n";
            out +=
element.getElementsByTagName("ár").item(0).getNodeName() + " : "
+element.getElementsByTagName("ár").item(0).getTextContent()+"\n";
            break;
        case "Rendelés":
            out +=
element.getElementsByTagName("datum").item(0).getNodeName() + " : "
+element.getElementsByTagName("datum").item(0).getTextContent()+"\n";
            out +=
element.getElementsByTagName("hatarido").item(0).getNodeName() + " : "
+element.getElementsByTagName("hatarido").item(0).getTextContent()+"\n";
            break;
        case "Ertesito":

```

```

        out +=
element.getElementsByTagName("enev").item(0).getNodeName() + " : "
+element.getElementsByTagName("enev").item(0).getTextContent()+"\n";
        out +=
element.getElementsByTagName("kor").item(0).getNodeName() + " : "
+element.getElementsByTagName("kor").item(0).getTextContent()+"\n";
        break;
    }

    System.out.println(out);
}
}

```

Működése:

Kapunk egy menüt:

0: Kilepes

1: Bizonyos ar feletti termek

2: Rendeles egyezo hatarido

3: Adott kornal idosebb elado

Ezek után a bevitel alapján folytatódik a futás.

1

Adja meg minel nagyobb:

33

Termek ID : t1

tnev : Zold 3-lyuku felteglá

ar : 11000

Termek ID : t2

tnev : Piros 8-lyuku felteglá

ar : 553

Termek ID : t3

tnev : Ekru 4-lyuku feltegla

ar : 99999