

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Táncsport szövetség

Készítette: **Hajdu Cintia**
Neptunkód: **VBNYZW**
Dátum: **2023.12.13.**

Tartalomjegyzék

Feladateleírás	
Az egyedek tulajdonságai	3
1.feladat	
1 a. Az adatbázis ER modellje	4
1 b. Az adatbázis konvertálása XDM modellre	5
1 c. Az XDM modell alapján XML dokumentum készítése	6
1 d. Az XML dokumentum alapján XMLSchema készítése	11
2.feladat	
2 a. DOMRead	16
2 b. DOMWrite	19
2 c. DOMQuery	25
2 d. DOMModify	29

Feladat leírása

A beadandóm a táncgyesületek életéről szól. Az adatbázis az egyesület adatain kívül nyilvántartja a táncosok, a versenyek, a szurkolók és a produkciók adatait is.

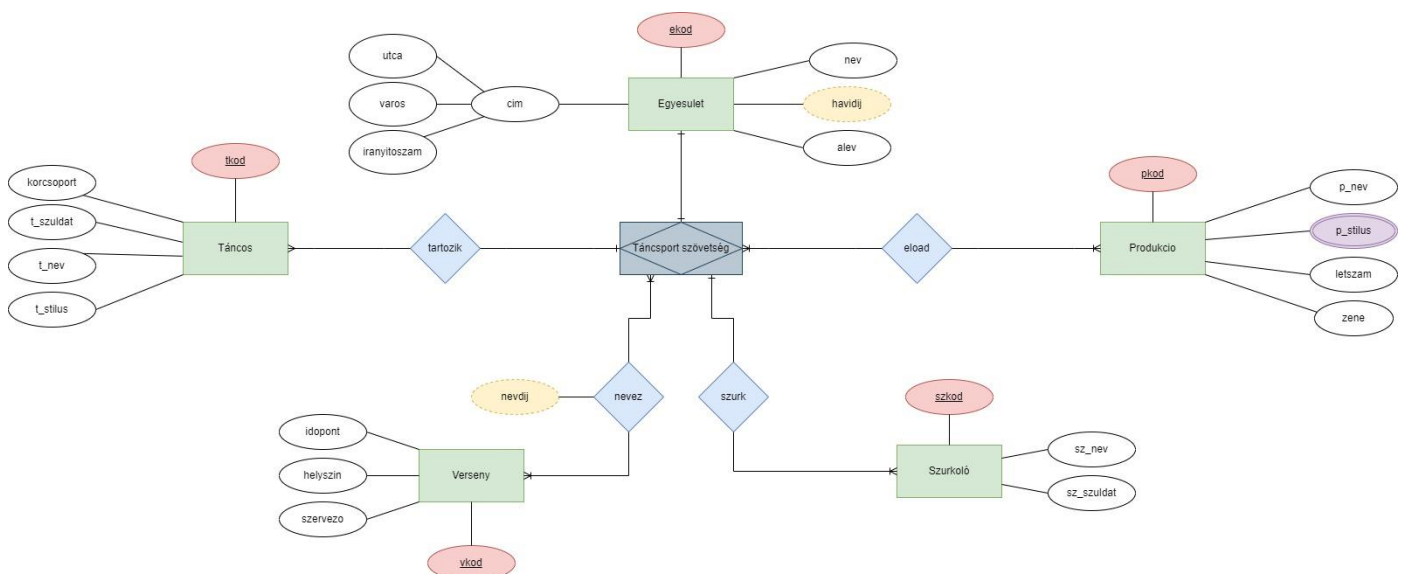
Az egyedek tulajdonságai:

1. Egyesület
 - a. ekod: egyesület elsődleges kulcsa
 - b. havidij: befizetendő havidij
 - c. nev: egyesület neve
 - d. cím : egyesület székhelye
 - i. utca
 - ii. varos
 - iii. irányitoszam
2. Táncos
 - a. tkod: táncos elsődleges kulcsa
 - b. korcsoport: táncos korcsoportja
 - c. t_szuldat: táncos születési dátuma
 - d. t_nev: táncos neve
 - e. t_stilus: többértetű, stílus amiben a táncos táncol
3. Verseny
 - a. vkod: verseny elsődleges kulcsa
 - b. idopont: verseny időpontja
 - c. helyszin: verseny helye
 - d. szervezo: verseny szervezője
4. Szurkoló
 - a. szkod: szurkoló elsődleges kulcsa
 - b. sz_nev: szurkoló neve
 - c. sz_szuldat: szurkoló születési dátuma
5. Produkció
 - a. pkod: produkció elsődleges kulcsa
 - b. zene: zene, amire a produkció készül
 - c. letszam: produkcióba résztvevők száma
 - d. p_nev: produkció neve
 - e. p_stilus: produkció stílusa

1. feladat

a. Az adatbázis ER modellje:

- Az Egyesület nevű egyeddel minden egyed (Táncos, Produkció, Verseny, Szurkoló) kapcsolatban van
- Egyesület – Táncos egy-több kapcsolat van, mivel az Egyesületeknek lehet sok táncosa, de egy Táncos csak egy egyesülethez tartozhat
- Egyesület – Produkció között több-több kapcsolat van, mivel feltételezzük, hogy egy produkciót több egyesület is elő tud adni, egy egyesületnek több produkciója is van
- Egyesület – Verseny több-több kapcsolat, mert egy egyesület több versenyre is nevezhet
- Egyesület – Szurkoló egy-több kapcsolat, mert a szurkoló egy egyesületnek szurkolhat.



1. feladat

b. Az adatbázis konvertálása XDM modellre

Az XDM modellben felhasznált jelölések: rombusz (attribútumok), ellipszis (egyedi entitás) és téglalap (adatok). Az elsődleges kulcsok nevét aláhúztam és kapcsolatait szaggatott vonallal jelöltem.

1. feladat

c. Az XDM modell alapján XML dokumentum készítése:

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

```
<tancegyesulet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XSDvbnyzw.xsd">
```

```
  <!-- Egyesületek -->
```

```
  <!-- eleje -->
```

```
  <egyesulet ekod="1">
```

```
    <havidij>10000</havidij>
```

```
    <nev>Ködmön</nev>
```

```
    <cim>
```

```
      <utca>Majoros</utca>
```

```
      <varos>Miskolc</varos>
```

```
      <iranyitoszam>3524</iranyitoszam>
```

```
    </cim>
```

```
    <alep>1993</alep>
```

```
  </egyesulet>
```

```
  <egyesulet ekod="2">
```

```
    <havidij>12000</havidij>
```

```
    <nev>Lazúr</nev>
```

```
    <cim>
```

```
      <utca>Hóvirág</utca>
```

```
      <varos>Miskolc</varos>
```

```
      <iranyitoszam>3525</iranyitoszam>
```

```
    </cim>
```

```
    <alev>1997</alev>
```

```
  </egyesulet>
```

```
  <egyesulet ekod="3">
```

```
    <havidij>15000</havidij>
```

```
    <nev>Savaria</nev>
```

```
    <cim>
```

```
      <utca>Király</utca>
```

```
      <varos>Szombathely</varos>
```

```

        <iranyitoszam>3800</iranyitoszam>
    </cim>
    <alev>2000</alev>
</egyesulet>
<!-- vége -->

<!-- Táncsoport szövetség -->
<!-- eleje -->
    <tanccsop      egy_idgnkulcs="1"      tanc_idgnkulcs="1"      vers_idgnkulcs="01"
szurk_idgnkulcs="0001" prod_idgkulcs="0001"/>
    <tanccsop      egy_idgnkulcs="1"      tanc_idgnkulcs="2"      vers_idgnkulcs="01"
szurk_idgnkulcs="0002" prod_idgkulcs="0002"/>
    <tanccsop      egy_idgnkulcs="2"      tanc_idgnkulcs="3"      vers_idgnkulcs="02"
szurk_idgnkulcs="0003" prod_idgkulcs="0003"/>
    <tanccsop      egy_idgnkulcs="2"      tanc_idgnkulcs="4"      vers_idgnkulcs="02"
szurk_idgnkulcs="0004" prod_idgkulcs="0004"/>
    <tanccsop      egy_idgnkulcs="3"      tanc_idgnkulcs="5"      vers_idgnkulcs="01"
szurk_idgnkulcs="0005" prod_idgkulcs="0005"/>
    <tanccsop      egy_idgnkulcs="3"      tanc_idgnkulcs="6"      vers_idgnkulcs="02"
szurk_idgnkulcs="0006" prod_idgkulcs="0006"/>
    <!-- vége -->

<!-- Táncosok -->
<!-- eleje -->
<tancos tkod="001">
    <korcsoport>A</korcsoport>
    <t_szuldat>1991.01.01.</t_szuldat>
    <t_nev>Kiss Szimonetta</t_nev>
    <t_stilus>latin</t_stilus>
</tancos>

<tancos tkod="002">
    <korcsoport>B</korcsoport>
    <t_szuldat>1994.04.16.</t_szuldat>
    <t_nev>Fenyves Béla</t_nev>
    <t_stilus>latin</t_stilus>
</tancos>

<tancos tkod="003">
    <korcsoport>C</korcsoport>
    <t_szuldat>1986.05.23.</t_szuldat>
    <t_nev>Fekete Nóra</t_nev>
    <t_stilus>standard</t_stilus>
</tancos>

<tancos tkod="004">
    <korcsoport>C</korcsoport>
    <t_szuldat>1997.08.02.</t_szuldat>
    <t_nev>Kiss Zoltán</t_nev>

```

```

        <t_stilus>standard</t_stilus>
    </tancos>

    <tancos tkod="005">
        <korcsoport>A</korcsoport>
        <t_szuldat>2000.01.22.</t_szuldat>
        <t_nev>Kovács Ádám</t_nev>
        <t_stilus>standard</t_stilus>
    </tancos>

    <tancos tkod="006">
        <korcsoport>C</korcsoport>
        <t_szuldat>1991.12.11.</t_szuldat>
        <t_nev>Mezei Hanga</t_nev>
        <t_stilus>latin</t_stilus>
    </tancos>
    <!-- vége -->

    <!-- Versenyek -->
    <!-- eleje -->
    <verseny vkod="01">
        <idopont>2024.04.27</idopont>
        <helyszin>Tiszaújváros Sportcsarnok</helyszin>
        <szervezo>Malek Andrea</szervezo>
    </verseny>

    <verseny vkod="02">
        <idopont>2024.06.06</idopont>
        <helyszin>Sátoraljaújhely Sportcsarnok</helyszin>
        <szervezo>Zsámboki Marcell</szervezo>
    </verseny>
    <!-- vége -->

    <!-- Nevez -->
    <!-- eleje -->
    <nevez nkod="1" n_v_idgnkulcs="01">
        <nevdij>5000</nevdij>
    </nevez>
    <!-- vége -->

    <!-- Szurkolók -->
    <!-- eleje -->
    <szurkolo szkod="0001">
        <sz_nev>Győri Eszter</sz_nev>
        <sz_szuldat>1978.05.24</sz_szuldat>
    </szurkolo>

    <szurkolo szkod="0002">

```


<sz_nev>Kovács Margit</sz_nev>
<sz_szuldat>1990.04.25</sz_szuldat>
</szurkolo>

<szurkolo szkod="0003">
<sz_nev>Hajdu Péter</sz_nev>
<sz_szuldat>1991.02.07</sz_szuldat>
</szurkolo>

<szurkolo szkod="0004">
<sz_nev>Boros Balázs</sz_nev>
<sz_szuldat>1990.06.17</sz_szuldat>
</szurkolo>

<szurkolo szkod="0005">
<sz_nev>Horváth Gergely</sz_nev>
<sz_szuldat>1967.12.21</sz_szuldat>
</szurkolo>

<szurkolo szkod="0006">
<sz_nev>Németh Márk</sz_nev>
<sz_szuldat>1974.08.19</sz_szuldat>
</szurkolo>
<!-- vége -->

<!-- Produkciók -->
<!-- eleje -->

<produkcio pkod="0001">
<zene>Breakaway</zene>
<letszam>2</letszam>
<p_nev>Karácsony</p_nev>
<p_stilus>keringő</p_stilus>
</produkcio>

<produkcio pkod="0002">
<zene>Zumba track</zene>
<letszam>16</letszam>
<p_nev>Karibi tánc</p_nev>
<p_stilus>latin</p_stilus>
<p_stilus>zumba</p_stilus>
</produkcio>

<produkcio pkod="0003">
<zene>Paso soundtrack</zene>
<letszam>6</letszam>
<p_nev>Torreádor</p_nev>
<p_stilus>pasodoble</p_stilus>
</produkcio>

<produkcio pkod="0004">

```
<zene>Moulin Rouge</zene>
<letszam>2</letszam>
<p_nev>Roxanne</p_nev>
<p_stilus>tango</p_stilus>
</produkcio>
```

```
<produkcio pkod="0005">
  <zene>Fall in love</zene>
  <letszam>1</letszam>
  <p_nev>Szerelem</p_nev>
  <p_stilus>rumba</p_stilus>
</produkcio>
```

```
<produkcio pkod="0006">
  <zene>Mika Grace Kelly</zene>
  <letszam>10</letszam>
  <p_nev>Lollipop</p_nev>
  <p_stilus>cha cha cha</p_stilus>
</produkcio>
<!-- vége -->
```

```
</tancegyesulet>
```

1. feladat

d. Az XML dokumentum alapján XMLSchema készítése

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

  <!-- Elemek -->

  <xs:element name="havidij" type="xs:integer"/>
  <xs:element name="nev" type="xs:string"/>
  <xs:element name="alev" type="xs:integer"/>
  <xs:element name="utca" type="xs:integer"/>
  <xs:element name="varos" type="xs:integer"/>
  <xs:element name="iranyitoszam" type="xs:integer"/>
  <xs:element name="korcsoport" type="xs:string"/>
  <xs:element name="t_nev" type="xs:string"/>
  <xs:element name="t_stilus" type="xs:string"/>
  <xs:element name="helyszin" type="xs:string"/>
  <xs:element name="szervezo" type="xs:string"/>
  <xs:element name="nevdijsz" type="xs:integer"/>
  <xs:element name="sz_nev" type="xs:string"/>
  <xs:element name="zene" type="xs:string"/>
  <xs:element name="letszam" type="xs:integer"/>
  <xs:element name="p_nev" type="xs:string"/>
  <xs:element name="p_stilus" type="xs:string"/>
  <xs:element name="nevez" type="xs:string"/>
  <xs:element name="ev" type="xs:integer"/>
  <xs:element name="honap" type="xs:integer"/>
  <xs:element name="nap" type="xs:integer"/>

  <!-- Attribútumok -->

  <xs:attribute name="ekod" type="xs:string"/>
  <xs:attribute name="egy_idgkulcs" type="xs:string"/>
  <xs:attribute name="prod_idgkulcs" type="xs:string"/>
  <xs:attribute name="szurk_idgkulcs" type="xs:string"/>
  <xs:attribute name="vers_idgkulcs" type="xs:string"/>
  <xs:attribute name="tanc_idgkulcs" type="xs:string"/>
  <xs:attribute name="n_v_idgkulcs" type="xs:string"/>
  <xs:attribute name="t_kod" type="xs:string"/>
  <xs:attribute name="v_kod" type="xs:string"/>
  <xs:attribute name="sz_kod" type="xs:string"/>
  <xs:attribute name="n_kod" type="xs:string"/>
  <xs:attribute name="p_kod" type="xs:string"/>
```

```

<!-- Egyszerű típusok -->

<xs:simpleType name="letszamType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{4}" />
  </xs:restriction>
</xs:simpleType>

<!-- Összetett típusok -->

<xs:complexType name="cimType">
  <xs:sequence>
    <xs:element ref="utca"/>
    <xs:element ref="varos"/>
    <xs:element ref="iranyitoszam"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="datumType">
  <xs:sequence>
    <xs:element ref="ev"/>
    <xs:element ref="honap"/>
    <xs:element ref="iranyitoszam"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="egyesuletType">
  <xs:sequence>
    <xs:element ref="havidij"/>
    <xs:element ref="nev"/>
    <xs:element name="cim" type="cimType"/>
    <xs:element ref="ev"/>
  </xs:sequence>
  <xs:attribute ref="ekod" use="required"/>
</xs:complexType>

<xs:complexType name="tancosType">
  <xs:sequence>
    <xs:element ref="korcsoport"/>
    <xs:element name="t_szuldat" type="datumType"/>
    <xs:element ref="t_nev"/>
    <xs:element ref="t_stilus"/>
  </xs:sequence>
  <xs:attribute ref="t_kod" use="required"/>
</xs:complexType>

<xs:complexType name="versenyType">
  <xs:sequence>
    <xs:element name="idopont" type="datumType"/>
    <xs:element ref="helyszin"/>
  </xs:sequence>
</xs:complexType>

```

```

    <xs:element ref="szervezo"/>
  </xs:sequence>
  <xs:attribute ref="v_kod" use="required"/>
</xs:complexType>

<xs:complexType name="szurkoloType">
  <xs:sequence>
    <xs:element ref="sz_nev"/>
    <xs:element name="sz_szuldat" type="datumType"/>
  </xs:sequence>
  <xs:attribute ref="sz_kod" use="required"/>
</xs:complexType>

<xs:complexType name="produkcioType">
  <xs:sequence>
    <xs:element ref="zene"/>
    <xs:element name="letszam" type="letszamType"/>
    <xs:element ref="p_nev"/>
    <xs:element ref="p_stilus"/>
  </xs:sequence>
  <xs:attribute ref="p_kod" use="required"/>
</xs:complexType>

<!-- N:M kapcsolat-->

<xs:complexType name="nevezType">
  <xs:sequence>
    <xs:element ref="nevdi"/>
  </xs:sequence>
  <xs:attribute ref="vkod" use="required"/>
  <xs:attribute ref="n_v_idgkulcs" use="required"/>
</xs:complexType>

<xs:complexType name="tancsportegyesuletType">
  <xs:attribute ref="egy_idgkulcs" use="required"/>
  <xs:attribute ref="prod_idgkulcs" use="optional"/>
  <xs:attribute ref="szurk_idgkulcs" use="optional"/>
  <xs:attribute ref="vers_idgkulcs" use="optional"/>
  <xs:attribute ref="tanc_idgkulcs" use="optional"/>
</xs:complexType>

<!-- Táncegyesület -->

<xs:element name="tancegyesulet">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="egyesulet" type="egyesuletType" maxOccurs="unbounded"/>
      <xs:element name="tancos" type="tancosType" maxOccurs="unbounded"/>
      <xs:element name="verseny" type="versenyType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

```

    <xs:element name="nevez" type="nevezType" maxOccurs="unbounded"/>
    <xs:element name="szurkolo" type="szurkoloType" maxOccurs="unbounded"/>
    <xs:element name="produkcio" type="produkcioType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- Elsődleges kulcsok -->

<xs:unique name="ekod">
  <xs:selector xpath="egyesulet"/>
  <xs:field xpath="@ekod"/>
</xs:unique>

<xs:unique name="tkod">
  <xs:selector xpath="tancos"/>
  <xs:field xpath="@tkod"/>
</xs:unique>

<xs:unique name="vkod">
  <xs:selector xpath="verseny"/>
  <xs:field xpath="@vkod"/>
</xs:unique>

<xs:unique name="nkod">
  <xs:selector xpath="nevez"/>
  <xs:field xpath="@nkod"/>
</xs:unique>

<xs:unique name="szkod">
  <xs:selector xpath="szurkolo"/>
  <xs:field xpath="@szkod"/>
</xs:unique>

<xs:unique name="pkod">
  <xs:selector xpath="produkcio"/>
  <xs:field xpath="@pkod"/>
</xs:unique>

<!-- Idegenkulcsok -->

<xs:keyref name="egy_idgkulcsFK" refer="ekod">
  <xs:selector xpath="egyesulet"/>
  <xs:field xpath="@egy_idgkulcs"/>
</xs:keyref>

<xs:keyref name="prod_idgkulcsFK" refer="pkod">
  <xs:selector xpath="produkcio"/>
  <xs:field xpath="@prod_idgkulcs"/>
</xs:keyref>

```

```

<xs:keyref name="szurk_idgkulcsFK" refer="szkod">
  <xs:selector xpath="szurkolo"/>
  <xs:field xpath="szurk_idgkulcs"/>
</xs:keyref>

<xs:keyref name="vers_idgkulcsFK" refer="nkod">
  <xs:selector xpath="nevez"/>
  <xs:field xpath="vers_idgkulcs"/>
</xs:keyref>

<xs:keyref name="tanc_idgkulcsFK" refer="tkod">
  <xs:selector xpath="tancos"/>
  <xs:field xpath="tanc_idgkulcs"/>
</xs:keyref>

<xs:keyref name="n_v_idgkulcsFK" refer="vkod">
  <xs:selector xpath="verseny"/>
  <xs:field xpath="n_v_idgkulcsFK"/>
</xs:keyref>

</xs:element>

</xs:schema>

```

2. feladat

DOM programok Java nyelven

a. DOMRead elkészítése

```
import org.w3c.dom.*;

import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.io.IOException;

import org.xml.sax.SAXException;

public class DomReadVBNYZW {

    /*A metódus betölt egy fájlt, elemzi és hibát küld, ha nem találja */
    public static void main(String[] args) {
        File xmlFile = new File("XMLvbnyzw.xml");
        Document doc = introduceFile(xmlFile);

        if (doc == null) {
            System.out.println("A dokumentum nem található");
            System.exit(-1);
        } else {
            processDocument(doc);
        }
    }

    /*Formátum elkészítése*/
    private static void processDocument(Document doc) {
        doc.getDocumentElement().normalize();
        System.out.println("<?xml version=\"1.0\" encoding=\"utf-8\"?>");
        System.out.println("<" + doc.getDocumentElement().getNodeName() +
            "
            xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\"XSDvbnyzw.xsd\">");

        NodeList nodeList = doc.getDocumentElement().getChildNodes();
        String indent = "";
        listData(nodeList, indent);
        System.out.println("</" + doc.getDocumentElement().getNodeName() + ">");

        /*Dokumentum fájlba mentése.*/
        saveDocumentToFile(doc, "XMLvbnyzwReadThisFile.xml");
    }
}
```



```

private static void saveDocumentToFile(Document doc, String fileName) {
    /*A TransformerFactory elemzés helyett a fájlba ír*/
    try {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        transformer.setOutputProperty(OutputKeys.INDENT, "no");

        transformer.transform(new DOMSource(doc), new StreamResult(fileName));

    } catch (Exception e) {
        e.printStackTrace();
    }
}

/*A DocumentBuilderFactory az XML fájlt olvassa és átalakítja*/
public static Document introduceFile(File xmlFile) {
    Document doc = null;

    try {
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
        doc = dbBuilder.parse(xmlFile);
    } catch (ParserConfigurationException | SAXException | IOException e) {
        e.printStackTrace();
    }
    return doc;
}

/*Képernyőre kiírás*/
private static void listData(NodeList nodeList, String indent) {

    indent += "\t";

    if (nodeList != null) {
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE &&
!node.getTextContent().trim().isEmpty()) {
                processElementNode((Element) node, indent);
            } else if (node instanceof Text) {
                String value = node.getNodeValue().trim();
                if (!value.isEmpty()) {
                    System.out.println(indent + node.getTextContent());
                }
            }
        }
    }
}

/*Elemek feldolgozása és írásra készen állása */
private static void processElementNode(Element element, String indent) {

```

```

System.out.print(indent + "<" + element.getNodeName());
if (element.hasAttributes()) {
    processAttributes(element);
}
System.out.println(">");

NodeList nodeList_new = element.getChildNodes();
listData(nodeList_new, indent);
System.out.println(indent + "</" + element.getNodeName() + ">");
}
/*Attribútumok feldolgozása és írásra készen állása */
private static void processAttributes(Element element) {
    NamedNodeMap attributes = element.getAttributes();
    for (int k = 0; k < attributes.getLength(); k++) {
        Node attribute = attributes.item(k);
        System.out.print(" " + attribute.getNodeName() + "=\"" + attribute.getNodeValue() +
"\");
    }
}
}

```

2. feladat

b. DOMWrite elkészítése

```
import org.w3c.dom.Document;
import org.w3c.dom.Element;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.io.FileWriter;
import java.util.Arrays;
import java.util.List;

public class DomWriteVBNYZW {

    public static void main(String[] args) {
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document document = builder.newDocument();
            Element rootElement = document.createElement("tancegyesulet");
            rootElement.setAttribute("xmlns:xsi", "http://www.w3.org/2001/XMLSchema-
instance");
            rootElement.setAttribute("xsi:noNamespaceSchemaLocation",
"XSDvbnyzw.xsd");
            document.appendChild(rootElement);

            /*Elemek feltöltése, feldolgozása*/
            /*Egyesület*/
            processEgyesulet(document, rootElement, "1", 10000, "Ködmön", "Majoros",
"Miskolc", "3524", 1993);
            processEgyesulet(document, rootElement, "2", 12000, "Lazúr", "Hóvirág",
"Miskolc", "3525", 1997);
            processEgyesulet(document, rootElement, "3", 15000, "Savaria", "Király",
"Szombathely", "3800", 2000);
            /*Táncos*/
            processTancos(document, rootElement, "001", "A", "1991.01.01", "Kiss
Szimonetta", "latin");
            processTancos(document, rootElement, "002", "B", "1994.04.16", "Fenyves
Béla", "latin");
            processTancos(document, rootElement, "003", "C", "1986.05.23", "Fekete
Nóra", "standard");
            processTancos(document, rootElement, "004", "C", "1997.08.02", "Kiss
Zoltán", "standard");
```

```

        processTancos(document, rootElement, "005", "A", "2000.01.22", "Kovács
        Ádám", "standard");
        processTancos(document, rootElement, "006", "A", "1991.12.11", "Mezei
        Hanga", "latin");
        /*Verseny*/
        processVerseny(document, rootElement, "01", "2024.04.27", "Tiszaújváros
        Sportcsarnok", "Malek Andrea");
        processVerseny(document, rootElement, "02", "2024.06.06", "Sátoraljaújhely
        Sportcsarnok", "Zsámboki Marcell");
        /*Szurkoló*/
        processSzurkolo(document, rootElement, "0001", "Győri Eszter",
        "1978.05.24");
        processSzurkolo(document, rootElement, "0002", "Kovács Margit",
        "1990.04.25");
        processSzurkolo(document, rootElement, "0003", "Hajdu Péter", "1991.02.07");
        processSzurkolo(document, rootElement, "0004", "Boros Balázs",
        "1990.06.17");
        processSzurkolo(document, rootElement, "0005", "Horváth Gergely",
        "1967.12.21");
        processSzurkolo(document, rootElement, "0006", "Németh Márk",
        "1974.08.19");
        /*Produkció*/
        processProdukcio(document, rootElement, "0001", "Breakaway", 2,
        "Karácsony", Arrays.asList("keringő"));
        processProdukcio(document, rootElement, "0002", "Zumba", 16, "Karibi tánc",
        Arrays.asList("latin", "zumba"));
        processProdukcio(document, rootElement, "0003", "Paso soundtrac", 6,
        "Torreádor", Arrays.asList("pasodoble"));
        processProdukcio(document, rootElement, "0004", "Moulin Rouge", 2,
        "Roxanne", Arrays.asList("tango"));
        processProdukcio(document, rootElement, "0005", "Fall in love", 1,
        "Szerelem", Arrays.asList("rumba"));
        processProdukcio(document, rootElement, "0006", "Mika Grace Kelly", 10,
        "Lollipop", Arrays.asList("cha cha cha"));

        printDocument(document);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/* Deklaráció, azokat az elemeket tartalmazza, amiket a programnak el kell
készítenie*/
/*Egyesület*/
private static void processEgyesulet(Document document, Element rootElement,
String ekod,
    int havidij, String nev, String utca, String varos, String irányitoszam, int alev) {
    Element egy = document.createElement("egyesulet");
    egy.setAttribute("ekod", ekod);

```

```

    Element havidijE = createElement(document, "havidij", String.valueOf(havidij));
    egy.appendChild(havidijE);

    Element nevE = createElement(document, "nev", nev);
    egy.appendChild(nevE);

    Element cimE = document.createElement("cim");
    Element utcaE = createElement(document, "utca", utca);
    Element varosE = createElement(document, "varos", varos);
    Element irányitoszamE = createElement(document, "irányitoszam",
    irányitoszam);

    cimE.appendChild(utcaE);
    cimE.appendChild(varosE);
    cimE.appendChild(irányitoszamE);

    egy.appendChild(cimE);
    rootElement.appendChild(egy);
}

/*Táncos*/

private static void processTancos(Document document, Element rootElement, String
tkod, String korcsoport, String t_szuldat, String t_nev, String t_stilus) {
    Element tanc = document.createElement("tancos");
    tanc.setAttribute("tkod", tkod);

    Element korcsoportE = createElement(document, "korcsoport", korcsoport);
    tanc.appendChild(korcsoportE);

    Element t_szuldatE = createElement(document, "t_szuldat", t_szuldat);
    tanc.appendChild(t_szuldatE);

    Element t_nevE = createElement(document, "t_nev", t_nev);
    tanc.appendChild(t_nevE);

    Element t_stilusE = createElement(document, "t_stilus", t_stilus);
    tanc.appendChild(t_stilusE);

    rootElement.appendChild(tanc);
}

/*Verseny*/

private static void processVerseny(Document document, Element rootElement,
String vkod, String idopont, String helyszin, String szervezo) {
    Element vers = document.createElement("verseny");
    vers.setAttribute("vkod", vkod);

```

```

    Element idopontE = createElement(document, "idopont", idopont);
    vers.appendChild(idopontE);

    Element helyszinE = createElement(document, "helyszin", helyszin);
    vers.appendChild(helyszinE);

    Element szervezoE = createElement(document, "szervezo", szervezo);
    vers.appendChild(szervezoE);

    rootElement.appendChild(vers);
}

/*Szurkoló*/

private static void processSzurkolo(Document document, Element rootElement,
String szkod, String sz_nev, String sz_szuldat) {
    Element szurk = document.createElement("szurkolo");
    szurk.setAttribute("szkod", szkod);

    Element sz_nevE = createElement(document, "sz_nev", sz_nev);
    szurk.appendChild(sz_nevE);

    Element sz_szuldatE = createElement(document, "sz_szuldat", sz_szuldat);
    szurk.appendChild(sz_szuldatE);

    rootElement.appendChild(szurk);
}

/*Produkció*/

private static void processProdukcio(Document document, Element rootElement,
String pkod, String zene, int letszam, String p_nev, List<String> p_stiluslista) {
    Element prod = document.createElement("produkcio");
    prod.setAttribute("pkod", pkod);

    Element zeneE = createElement(document, "zene", zene);
    prod.appendChild(zeneE);

    Element letszamE = createElement(document, "letszam", String.valueOf(letszam));
    prod.appendChild(letszamE);

    Element p_nevE = createElement(document, "p_nev", p_nev);
    prod.appendChild(p_nevE);

    Element p_stilusE = document.createElement("p_stilus");
    for (String p : p_stiluslista){
        Element temp = createElement(document, "p_stilus", p);
        p_stilusE.appendChild(temp);
    }
    prod.appendChild(p_stilusE);
}

```

```

        rootElement.appendChild(prod);
    }

    /*Táncsoport (Táncsport szövetség rövidítve)*/

    private static void processTancsoport(Document document, Element rootElement,
String egy_idgkulcs, String prod_idgkulcs, String vers_idgkulcs, String tanc_idgkulcs)
    {
        Element tancsoport = document.createElement("tancsoport");
        tancsoport.setAttribute("egy_idgkulcs", egy_idgkulcs);
        tancsoport.setAttribute("prod_idgkulcs", prod_idgkulcs);
        tancsoport.setAttribute("vers_idgkulcs", vers_idgkulcs);
        tancsoport.setAttribute("tanc_idgkulcs", tanc_idgkulcs);

        rootElement.appendChild(tancsoport);
    }

    /*Nevezés*/

    private static void processNevez(Document document, Element rootElement, String
nkod, String n_v_idgkulcs, int nevdij) {
        Element nevez = document.createElement("nevez");
        nevez.setAttribute("nkod", nkod);

        Element nevdijE = createElement(document, "nevdij", String.valueOf(nevdij));
        nevez.appendChild(nevdijE);

        nevez.setAttribute("n_v_idgkulcs", n_v_idgkulcs);

        rootElement.appendChild(nevez);
    }

    /*Elem létrehozása*/

    private static Element createElement(Document document, String tagName, String
textContent) {
        Element element = document.createElement(tagName);
        element.setTextContent(textContent);
        return element;
    }

    /* Kimenet létrehozása */

    private static void printDocument(Document document) {
        try {
            TransformerFactory transformerFactory = TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            transformer.setOutputProperty(OutputKeys.INDENT, "yes");

```

```

        /*Létrehozza az XMLvbnyzw.xml nevű fájlt.*/
        transformer.transform(new          DOMSource(document),          new
StreamResult("XMLvbnyzw.xml"));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```


2.feladat

c. DOMQuery elkészítése

```
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;

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 DomQueryVBNYZW{
    public static void main(String[] args) {
        File xmlFile = new File("XMLvbnyzw.xml"); //XML fájl elérési útja
        Document doc = parseXmlFile(xmlFile);

        if (doc != null) {
            doc.getDocumentElement().normalize();

            int tancEgyesuletekSzama = tancEgyesuletekSzamaVarosban(doc, "Miskolc");
            System.out.println("Táncegyesületek száma Miskolcon: " + tancEgyesuletekSzama);

            listaNemEgyedulTancolok(doc);

            listaEgyesuletSzurkoloi(doc, "Ködmön");

            listaSavariaTancosai(doc, "1993");

            listCompetitionsInYear(doc, "2024");

        } else {
            System.out.println("The document is null");
            System.exit(-1);
        }
    }

    private static Document parseXmlFile(File xmlFile) {
        Document doc = null;

        try {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            doc = dbBuilder.parse(xmlFile);
        } catch (ParserConfigurationException | IOException | org.xml.sax.SAXException e) {
            e.printStackTrace();
        }
    }
}
```

```

        return doc;
    }
    //Miskolci tánc egyesületek száma
    private static int tancEgyesuletekSzamaVarosban(Document doc, String varos) {
        NodeList tancEgyesulet = doc.getDocumentElement().getElementsByTagName("egyesulet");
        int tancEgyesuletekSzama = 0;

        for (int i = 0; i < tancEgyesuletekSzama.getLength(); i++) {
            Element tancEgyesulet = (Element) tancEgyesulet.item(i);
            String egyesuletVaros = tancEgyesulet.getElementsByTagName("varos").item(0).getTextContent();

            if (egyesuletVaros.equals(varos)) {
                tancEgyesuletekSzama++;
            }
        }

        return tancEgyesuletekSzama;
    }

    //Azok a produkciók, amik nem szólók (2 vagy annál többen táncolják)
    private static void listaNemEgyedulTancolok(Document doc) {
        NodeList produk = doc.getDocumentElement().getElementsByTagName("produkcio");

        System.out.println("A nem szolo produkciók:");

        for (int i = 0; i < produk.getLength(); i++) {
            Element produk = (Element) produk.item(i);
            int tancosSzamolo = Integer.parseInt(produk.getElementsByTagName("letszam").item(0).getTextContent());

            if (tancosSzamolo > 2) {
                String ProdukcioNev = produk.getElementsByTagName("p_nev").item(0).getTextContent();
                System.out.println("Produkció: " + ProdukcioNev + ", Táncolók létszáma: " + tancosSzamolo);
            }
        }
    }

    //A Ködmönnek szurkolók
    private static void listaEgyesuletSzurkoloi(Document doc, String tancEgyesuletNev) {
        NodeList listaz = doc.getDocumentElement().getElementsByTagName("egyesulet");

        System.out.println("Az adott egyesület szurkolói: " + tancEgyesuletNev + ":");

        for (int i = 0; i < listaz.getLength(); i++) {
            Element tancStudio = (Element) listaz.item(i);

```

```

        String                                studioNev                                =
tancStudio.getElementsByTagName("nev").item(0).getTextContent();

        if (studioNev.equals(tancEgyesuletNev)) {
            NodeList fanok = tancStudio.getElementsByTagName("szurkolo");

            for (int j = 0; j < fanok.getLength(); j++) {
                Element fan = (Element) fanok.item(j);
                String                                fanNev                                =
fan.getElementsByTagName("sz_nev").item(0).getTextContent();
                String                                fanSzuldat                            =
fan.getElementsByTagName("sz_szuldat").item(0).getTextContent();

                System.out.println("Szurkolo neve: " + fanNev + ", születési dátuma: " +
fanSzuldat);
            }
        }
    }
}

//Savaria 1993.után született táncosai
private static void listaSavariaTancosai(Document doc, String szulnap) {
    NodeList lista = doc.getElementsByTagName("tancos");

    System.out.println("Savaria táncosa, aki 1993. után ekkor született" + szulnap + ":");

    for (int i = 0; i < lista.getLength(); i++) {
        Element tancolo = (Element) lista.item(i);
        String tancoloStudio = getElementValueByTagName(tancolo, "egyesulet/nev");
        String tancoloNev = getElementValueByTagName(tancolo, "t_nev");
        String tancoloSzuldat = getElementValueByTagName(tancolo, "t_szuldat");

        String[] szuletEs = szulDate.split("\\.");
        String szulEv = szuletEs.length > 0 ? szuletEs[0] : "";

        if (tancoloStudio.equals("Savaria") && Integer.parseInt(szulEv) >
Integer.parseInt(szulnap)) {
            System.out.println("Név: " + tancoloNev + ", születési dátum: " + szulDate);
        }
    }
}

private static String getElementValueByTagName(Element element, String tagName) {
    NodeList nodeList = element.getElementsByTagName(tagName);
    if (nodeList.getLength() > 0) {
        return nodeList.item(0).getTextContent();
    }
    return "";
}

```

```

//2024-es versenyek listája
private static void listaVersenyekEv(Document doc, String year) {
    NodeList versenyek = doc.getElementsByTagName("verseny");

    System.out.println("Versenyek az adott évben: " + year + ":");

    for (int i = 0; i < versenyek.getLength(); i++) {
        Element versen = (Element) versenyek.item(i);
        String versenyDatum = getElementValueByTagName(versen, "idopont");

        String[] datumResz = versenyDatum.split("\\.");
        String versenyEv = datumResz.length > 0 ? datumResz[0] : "";

        if (versenyEv.equals(year)) {
            String versenyHelyszin = getElementValueByTagName(versen, "helyszin");
            System.out.println("Dátum: " + versenyDatum + ", Helyszin: " + versenyHelyszin);
        }
    }
}
}

```

2.feladat

d. DOMModify készítése

```
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
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;

public class DomModifyVBNYZW {
    public static void main(String[] args) {
        File xmlFile = new File("XMLvbnyzw.xml");
        Document doc = parseXmlFile(xmlFile);

        if (doc == null) {
            System.out.println("A dokumentum nem található");
            System.exit(-1);
        } else {
            doc.getDocumentElement().normalize();
        }

        modositLazurDij(doc.getDocumentElement(), "Lazúr", 15000);

        modositTancosKorcsoport(doc.getDocumentElement(), "Kiss Szimonetta", "A", "B");

        // Kiírja a képernyőre a módosított XML-t
        printXml(doc);
    }

    private static Document parseXmlFile(File xmlFile) {
        Document doc = null;

        try {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            doc = dbBuilder.parse(xmlFile);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

    }

    return doc;
}

//A Lazur egyesület havidíjának módosítása 15000-re
private static void modositLazurDij(Element element, String egyesuletNev, int ujDij) {
    NodeList productLepes = element.getElementsByTagName("egyesulet");

    for (int i = 0; i < productLepes.getLength(); i++) {
        Element productElement = (Element) productLepes.item(i);
        String egyesuletNevXml =
productElement.getElementsByTagName("nev").item(0).getTextContent();

        if (egyesuletNev.equals(egyesuletNevXml)) {

            Element dijElement =
productElement.getElementsByTagName("havidij").item(0);
            dijElement.setTextContent(Integer.toString(ujDij));
            break;
        }
    }
}

//Kiss Szimonetta táncos korcsoportjának módosítása
private static void modisitTancosKorcsoport(Element element, String tancosNev, String
korCsoport, String ujKorCsoport) {
    NodeList tancosNodes = element.getElementsByTagName("tancos");

    for (int i = 0; i < tancosNodes.getLength(); i++) {
        Element tancosElement = (Element) tancosNodes.item(i);
        String tancosNevXml =
tancosElement.getElementsByTagName("t_nev").item(0).getTextContent();
        String korCsoportXML =
tancosElement.getElementsByTagName("korcsoport").item(0).getTextContent();

        if (tancosNev.equals(tancosNevXml) && korCsoport.equals(korCsoportXML)) {
            Element korCsoportElement =
tancosElement.getElementsByTagName("korcsoport").item(0);
            korCsoportElement.setTextContent(ujKorCsoport);
            break;
        }
    }
}

private static void printXml(Document doc) {
    try {

        TransformerFactory transformerFactory = TransformerFactory.newInstance();

```

```
Transformer transformer = transformerFactory.newTransformer();
transformer.setOutputProperty(OutputKeys.INDENT, "yes");

DOMSource source = new DOMSource(doc);
StreamResult consoleResult = new StreamResult(System.out);
transformer.transform(source, consoleResult);
} catch (Exception e) {
    e.printStackTrace();
}
}
```