JEGYZŐKÖNYV

Webes adatkezelő környezetek

Féléves feladat

Gyárak

Készítette: Szabó Marcell

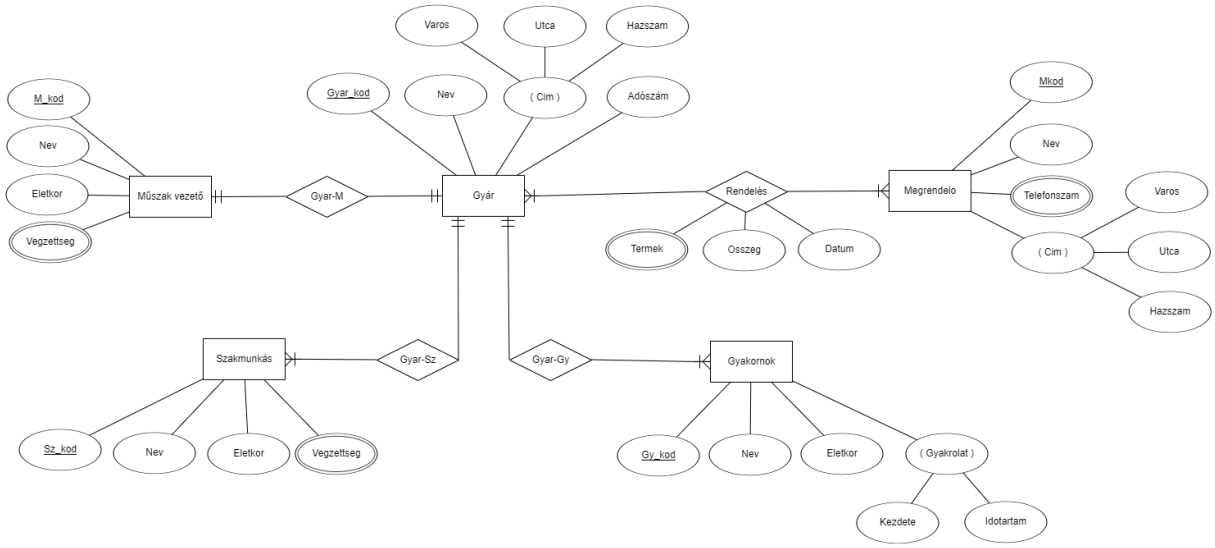
Neptunkód: FU7OMC

Dátum: 2024.11.08.

Tartalomjegyzék

1. Feladat leírása

1.1 Az adatbázis ER modell tervezése



Az egyedek tulajdonságai:

- **Gyár és Műszakvezető:**
  A Gyár és a Műszakvezető között 1:N típusú kapcsolat van, mivel egy gyár több műszakvezetőt is alkalmazhat, de egy műszakvezető csak egy gyárban dolgozhat. Például a SínCsináló Kft.-nél Trombitás Elemér és Nagy Ferenc is műszakvezetőként dolgozik.

- **Gyár és Szakmunkás:**
  A Gyár és a Szakmunkás között 1:N típusú kapcsolat van, mivel egy gyárban több szakmunkás is dolgozhat, de egy szakmunkás csak egy adott gyár alkalmazottja lehet. Például Szabó Gábor és Kovács László (Villanyszerelő) is a SínCsináló Kft.-nél dolgozik.

- **Gyár és Gyakornok:**
  A Gyár és a Gyakornok között 1:N típusú kapcsolat van, mivel egy gyár több gyakornokot is fogadhat, de egy gyakornok csak egy gyárban végezheti gyakorlatát. Illés Dániel és Hegedűs Emese például a SínCsináló Kft.-nél tölti gyakorlati idejét.

- **Gyár és Megrendelő:**
  A Gyár és a Megrendelő között 1:N típusú kapcsolat van, mivel egy gyár több megrendelővel is kapcsolatban állhat, de egy megrendelő egy adott gyártól rendel. Például a SínCsináló Kft. megrendelői között van a Máv Zrt. és a Volánbusz Zrt.

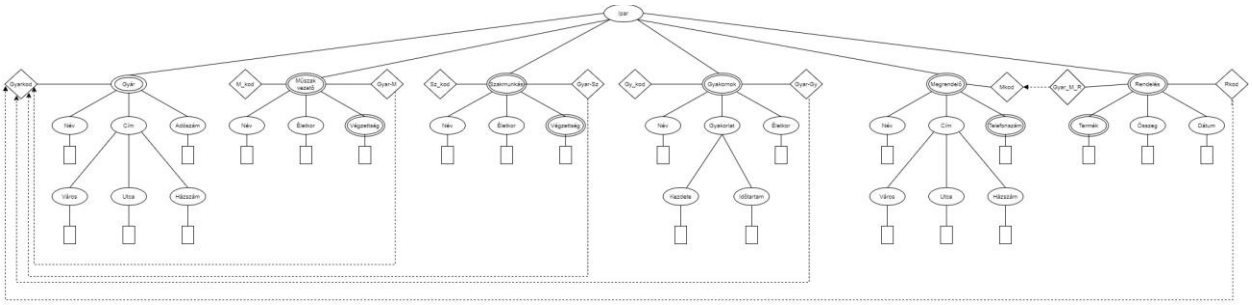
- **Rendelés és Termék:**
  A Rendelés és a Termék között 1:N típusú kapcsolat van, mivel egy rendelésben több termék is szerepelhet, de egy termék csak egy adott rendeléshez tartozik. Például a Rendelés R1-ben Vasúti Sín és Csavar is szerepel.

- **Gyár és Rendelés:**
  A Gyár és a Rendelés között 1:N típusú kapcsolat van, mivel egy gyár több rendelést is teljesíthet, de egy rendelés csak egy gyártól származik. A SínCsináló Kft. például több rendelést is kapott, mint R1 és R2.

- **Rendelés és Megrendelő:**
  A Rendelés és a Megrendelő között N:1 típusú kapcsolat van, mivel egy rendelést egy megrendelő adhat le, de egy megrendelő több rendelést is leadhat. Például a Máv Zrt. is és a Volánbusz Zrt. is adott már le különböző rendeléseket.

1.2 Az adatbázis konvertálása XDM modellre

XDM modellnél háromféle jelölés használunk: ellipszist, rombuszt és téglalapot. Az ellipszis jelöli az elemeket, minden egyedből elem lesz, illetve a tulajdonságokból is. A rombusz jelöli az attribútumokat, amelyek a kulcs tulajdonságokból keletkeznek. A téglalap jelöli a szöveget, amely majd az XML dokumentumban fog megjelenni. Azoknak az elemeknek, amelyek többször is előfordulhatnak, a jelölése dupla ellipszissel történik. Az idegenkulcsok és a kulcsok közötti kapcsolatot szaggatott vonalas nyíllal jelöljük.



1.3 Az XDM modell alapján XML dokumentum készítése

Az XDM modell alapján készítettem el az XML dokumentumot. Legelőször a gyökérelemmel kezdtem, amelynek a **„XMLFU7OMC.xml”** nevet adtam. Ezek után a gyermekelemeiből eltérő módon hoztam létre példányokat.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Ipar xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="XMLSchemaFU7OMC.xsd">
    <Gyar Gyarkod="A1">
```

```
        <Nev>SínCsináló Kft.</Nev>
        <Cim>
            <Varos>Miskolc</Varos>
            <Utca>Szép lány</Utca>
            <Hazszam>54</Hazszam>
        </Cim>
        <Adoszam>111111111-1-11</Adoszam>
        <!-- Több műszakvezető -->
        <MuszakVezeto M_kod="M1">
            <Nev>Trombitás Elemér</Nev>
            <Vegzettseg>Lakatos</Vegzettseg>
            <Eletkor>62</Eletkor>
        </MuszakVezeto>
        <MuszakVezeto M_kod="M2">
            <Nev>Nagy Ferenc</Nev>
            <Vegzettseg>Gépész</Vegzettseg>
            <Eletkor>55</Eletkor>
        </MuszakVezeto>

        <!-- Több szakmunkás -->
        <Szakmunkas Sz_kod="SZ1">
            <Nev>Szabó Gábor</Nev>
            <Vegzettseg>Kisegítő</Vegzettseg>
            <Eletkor>30</Eletkor>
        </Szakmunkas>
        <Szakmunkas Sz_kod="SZ2">
            <Nev>Kovács László</Nev>
            <Vegzettseg>Villanyszerelő</Vegzettseg>
            <Eletkor>40</Eletkor>
        </Szakmunkas>
        <Szakmunkas Sz_kod="SZ3">
            <Nev>Kovács László</Nev>
            <Vegzettseg>Hegesztő</Vegzettseg>
            <Eletkor>50</Eletkor>
        </Szakmunkas>

        <!-- Több gyakornok -->
        <Gyakornok Gy_kod="Gyak1">
            <Nev>Illés Dániel</Nev>
            <Eletkor>20</Eletkor>
            <Gyakorlat>
                <Kezdete>2024.02.14</Kezdete>
                <Idotartam>8 hét</Idotartam>
```

```
            </Gyakorlat>
        </Gyakornok>
        <Gyakornok Gy_kod="Gyak2">
            <Nev>Hegedűs Emese</Nev>
            <Eletkor>22</Eletkor>
            <Gyakorlat>
                <Kezdete>2024.02.14</Kezdete>
                <Idotartam>8 hét</Idotartam>
            </Gyakorlat>
        </Gyakornok>
        <!-- Több megrendelő -->
        <Megrendelo Mkod="M1">
            <Nev>Máv Zrt.</Nev>
            <Cim>
                <Varos>Budapest</Varos>
                <Utca>Máv</Utca>
                <Hazszam>32</Hazszam>
            </Cim>
            <Telefonszam>+36 20 456 4353</Telefonszam>
        </Megrendelo>
        <Megrendelo Mkod="M2">
            <Nev>Volánbusz Zrt.</Nev>
            <Cim>
                <Varos>Debrecen</Varos>
                <Utca>Nagyerdei körút</Utca>
                <Hazszam>5</Hazszam>
            </Cim>
            <Telefonszam>+36 30 123 4567</Telefonszam>
        </Megrendelo>
        <!-- Több termék a rendelésben -->
        <Rendeles GyarMR="GYR1" Rkod="R1">
            <Termek>Vasúti Sín</Termek>
            <Termek>Acél Lemez</Termek>
            <Termek>Csavar</Termek>
            <Osszeg>12345654.00</Osszeg>
            <Datum>2024-03-21</Datum>
        </Rendeles>
        <Rendeles GyarMR="GYR2" Rkod="R2">
            <Termek>Betongerenda</Termek>
            <Termek>Acélcső</Termek>
            <Osszeg>5789000.00</Osszeg>
            <Datum>2024-04-10</Datum>
        </Rendeles>
    </Gyar>
```

```
</Ipar>
```

1.4 Az XML dokumentum alapján XMLSchema készítése

Az XML dokumentum validálásának megkönnyítésére egy sémát készítettem. Először definiáltam az egyszerű típusokat, majd létrehoztam saját típusokat, összesen ötöt. Például a telefonszámhoz egy olyan típust készítettem, amely reguláris kifejezéssel ellenőrzi a formátumot, és egy másik típust az érvényes kódok listájával enumeráció segítségével. Ezután megterveztem a komplex típusokat az összes elemre, beleértve az összetett cím és személyzet adatstruktúrákat. Végül definiáltam elsődleges kulcsokat az egyedi azonosításhoz, valamint idegen kulcsokat az elemek közötti kapcsolatok biztosítására. A rendszerben megvalósítottam egy 1:N kapcsolatot a Gyár és a Rendelés között, ahol egy gyárhoz több rendelés is tartozhat.

XMLSchemaFU7OMC.xsd:

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

    <!-- Kor típus (1 és 100 közötti egész szám) -->
    <xs:simpleType name="korTipus">
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="1" />
            <xs:maxExclusive value="100" />
        </xs:restriction>
    </xs:simpleType>

    <!-- Személy típus -->
    <xs:complexType name="szemelyzet">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string" />
            <xs:element name="Vegzettseg" type="xs:string" />
            <xs:element name="Eletkor" type="korTipus" />
        </xs:sequence>
    </xs:complexType>

    <!-- Cím típus -->
    <xs:complexType name="CimTipus">
        <xs:sequence>
            <xs:element name="Varos" type="xs:string" />
            <xs:element name="Utca" type="xs:string" />
            <xs:element name="Hazszam" type="xs:string" />
        </xs:sequence>
```

```
        <xs:attribute name="Rkod" type="xs:string" />
    </xs:complexType>

    <!-- Műszakvezető típus -->
    <xs:complexType name="MuszakVezetoTipus">
        <xs:complexContent>
            <xs:extension base="szemelyzet">
                <xs:attribute name="M_kod" type="xs:string" use="required" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <!-- Szakmunkás típus -->
    <xs:complexType name="SzakmunkasTipus">
        <xs:complexContent>
            <xs:extension base="szemelyzet">
                <xs:attribute name="Sz_kod" type="xs:string" use="required" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <!-- Gyakornok típus -->
    <xs:complexType name="GyakornokTipus">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string" />
            <xs:element name="Eletkor" type="korTipus" />
            <xs:element name="Gyakorlat">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Kezdete" type="xs:string" />
                        <xs:element name="Idotartam" type="xs:string" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="Gy_kod" type="xs:string" use="required" />
    </xs:complexType>

    <!-- Megrendelő típus -->
    <xs:complexType name="MegrendeloTipus">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string" />
            <xs:element name="Cim" type="CimTipus" />
```

```
            <xs:element name="Telefonszam" type="xs:string" />
        </xs:sequence>
        <xs:attribute name="Mkod" type="xs:string" use="required" />
    </xs:complexType>

    <!-- Rendelés típus -->
    <xs:complexType name="RendelesTipus">
        <xs:sequence>
            <xs:element name="Termek" type="xs:string" minOccurs="1"
maxOccurs="unbounded" />
            <xs:element name="Osszeg" type="xs:string" />
            <xs:element name="Datum" type="xs:string" />
        </xs:sequence>
        <xs:attribute name="Rkod" type="xs:string" />
        <xs:attribute name="GyarMR" type="xs:string" />
    </xs:complexType>

    <!-- Ipar fő struktúra -->
    <xs:element name="Ipar">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Gyar" minOccurs="1" maxOccurs="100">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Nev" type="xs:string" />
                            <xs:element name="Cim" type="CimTipus" />
                            <xs:element name="Adoszam" type="xs:string" />
                            <xs:element name="MuszakVezeto"
type="MuszakVezetoTipus" minOccurs="1" maxOccurs="100" />
                            <xs:element name="Szakmunkas"
type="SzakmunkasTipus" minOccurs="1" maxOccurs="100" />
                            <xs:element name="Gyakornok" type="GyakornokTipus"
minOccurs="1" maxOccurs="100" />
                            <xs:element name="Megrendelo"
type="MegrendeloTipus" minOccurs="1" maxOccurs="100" />
                            <xs:element name="Rendeles" type="RendelesTipus"
minOccurs="1" maxOccurs="100" />
                        </xs:sequence>
                        <xs:attribute name="Gyarkod" type="xs:string" />
                    </xs:complexType>
                    <!-- Primary key -->
                    <xs:key name="GyarPrimaryKey">
                        <xs:selector xpath="Gyar" />
```

```
                        <xs:field xpath="@Gyarkod" />
                    </xs:key>
                    <!-- Foreign key kapcsolatok -->
                    <xs:keyref name="RendelesGyarForeignKey"
refer="GyarPrimaryKey">
                        <xs:selector xpath="Gyar/Rendeles" />
                        <xs:field xpath="@GyarMR" />
                    </xs:keyref>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2. DOM program készítése

2.1 Adatolvasás – fájlnév: DOMReadNeptunkod.java

main metódus:

A program belépési pontja, amely inicializálja az XML-feldolgozást:

Fájlbeolvasás: XMLFU7OMC.xml fájl betöltése.

Az XML feldolgozását külön metódusok végzik az adatok (gyárak, műszakvezetők, szakmunkások stb.) lekérdezésére és kiírására.

parseXML metódus:

Funkció: XML fájl beolvasása és DOM-parszolása.

Paraméter: File file – a beolvasandó XML fájl.

Visszatérési érték: Document – a DOM dokumentum, amely az XML adatszerkezetet reprezentálja.

queryXMLGyarkodok metódus:

Funkció: Az XML-ben található gyár adatok kiírása.

Minden Gyar elemre iterálva megjeleníti a gyár azonosítóját (Gyarkod), nevét, címét és adószámát.

queryXMLMuszakVezetok metódus:

Funkció: Műszakvezetők adatainak megjelenítése.

A MuszakVezeto elemek kiolvasása, amelyek tartalmazzák a műszakvezető kódját (M_kod), nevét, végzettségét és életkorát.

queryXMLSzakmunkasok metódus:

Funkció: Szakmunkások adatainak megjelenítése.

A Szakmunkas elemek kiolvasása, amelyek tartalmazzák a szakmunkás kódját (Sz_kod), nevét, végzettségét és életkorát.

queryXMLGyakornokok metódus:

Funkció: Gyakornokok és gyakorlatok adatainak megjelenítése.

A Gyakornok elemek kiírása, beleértve a gyakornok kódját (Gy_kod), nevét és életkorát.

A gyakornokok gyakorlati időszakainak (Gyakorlat) részletei, mint a kezdete és időtartama.

queryXMLMegrendelok metódus:

Funkció: Megrendelők adatainak kiírása.

Minden Megrendelo elem attribútumai: azonosító (Mkod), név, cím (város, utca, házszám) és telefonszám kiíratása.

queryXMLRendelesek metódus:

Funkció: Rendelések adatainak kiírása.

Rendeles elemek kiolvasása, beleértve a gyárkódot (GyarMR), rendeléskódot (Rkod), termékeket és a rendelés összegét, dátumát.

Forráskód:

```
package hu.domparse.fu7omc;

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

public class DOMReadFU7OMC {

    public static void main(String[] args) {
        try {
            File inputFile = new File("XMLFU7OMC.xml");
            Document document = parseXML(inputFile);

            queryXMLGyarkodok(document);
            queryXMLMuszakVezetok(document);
```

```
            queryXMLSzakmunkasok(document);
            queryXMLGyakornokok(document);
            queryXMLMegrendelok(document);
            queryXMLRendelesek(document);

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

    // XML beolvasása
    public static Document parseXML(File file) throws Exception {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        return builder.parse(file);
    }

    // Gyárak kiírása
    private static void queryXMLGyarkodok(Document document) {
        NodeList gyarNodeList = document.getElementsByTagName("Gyar");
        for (int i = 0; i < gyarNodeList.getLength(); i++) {
            Node gyarNode = gyarNodeList.item(i);
            if (gyarNode.getNodeType() == Node.ELEMENT_NODE) {
                Element gyarElement = (Element) gyarNode;
                String gyarkod = gyarElement.getAttribute("Gyarkod");
                String nev =
gyarElement.getElementsByTagName("Nev").item(0).getTextContent();
                String varos =
gyarElement.getElementsByTagName("Varos").item(0).getTextContent();
                String utca =
gyarElement.getElementsByTagName("Utca").item(0).getTextContent();
                String hazszam =
gyarElement.getElementsByTagName("Hazszam").item(0).getTextContent();
                String adoszam =
gyarElement.getElementsByTagName("Adoszam").item(0).getTextContent();

                System.out.println("------------------------------");
                System.out.println("Gyar:");
                System.out.println("    Gyarkod=\"" + gyarkod + "\"");
                System.out.println("    Nev: " + nev);
                System.out.println("    Cim:");
                System.out.println("        Varos: " + varos);
                System.out.println("        Utca: " + utca);
```

```
                System.out.println("        Hazszam: " + hazszam);
                System.out.println("    Adoszam: " + adoszam);
            }
        }
    }

    // Műszakvezetők kiírása
    private static void queryXMLMuszakVezetok(Document document) {
        NodeList muszakVezetoNodeList =
document.getElementsByTagName("MuszakVezeto");
        for (int i = 0; i < muszakVezetoNodeList.getLength(); i++) {
            Node muszakVezetoNode = muszakVezetoNodeList.item(i);
            if (muszakVezetoNode.getNodeType() == Node.ELEMENT_NODE) {
                Element muszakVezetoElement = (Element) muszakVezetoNode;
                String M_kod = muszakVezetoElement.getAttribute("M_kod");
                String nev =
muszakVezetoElement.getElementsByTagName("Nev").item(0).getTextContent();
                String vegzettseg =
muszakVezetoElement.getElementsByTagName("Vegzettseg").item(0).getTextContent()
;
                String eletkor =
muszakVezetoElement.getElementsByTagName("Eletkor").item(0).getTextContent();

                System.out.println("-------------------------------");
                System.out.println("MuszakVezeto:");
                System.out.println("    MuszakVezetoKod=\"" + M_kod + "\"");
                System.out.println("    Nev: " + nev);
                System.out.println("    Vegzettseg: " + vegzettseg);
                System.out.println("    Eletkor: " + eletkor);
            }
        }
    }

    // Szakmunkások kiírása
    private static void queryXMLSzakmunkasok(Document document) {
        NodeList szakmunkasNodeList =
document.getElementsByTagName("Szakmunkas");
        for (int i = 0; i < szakmunkasNodeList.getLength(); i++) {
            Node szakmunkasNode = szakmunkasNodeList.item(i);
            if (szakmunkasNode.getNodeType() == Node.ELEMENT_NODE) {
                Element szakmunkasElement = (Element) szakmunkasNode;
                String Sz_kod = szakmunkasElement.getAttribute("Sz_kod");
```

```
                String nev =
szakmunkasElement.getElementsByTagName("Nev").item(0).getTextContent();
                String vegzettseg =
szakmunkasElement.getElementsByTagName("Vegzettseg").item(0).getTextContent();
                String eletkor =
szakmunkasElement.getElementsByTagName("Eletkor").item(0).getTextContent();

                System.out.println("------------------------------");
                System.out.println("Szakmunkas:");
                System.out.println("    SzakmunkasKod=\"" + Sz_kod + "\"");
                System.out.println("    Nev: " + nev);
                System.out.println("    Vegzettseg: " + vegzettseg);
                System.out.println("    Eletkor: " + eletkor);
            }
        }
    }

    // Gyakornokok kiírása
    private static void queryXMLGyakornokok(Document document) {
        NodeList gyakornokNodeList =
document.getElementsByTagName("Gyakornok");
        for (int i = 0; i < gyakornokNodeList.getLength(); i++) {
            Node gyakornokNode = gyakornokNodeList.item(i);
            if (gyakornokNode.getNodeType() == Node.ELEMENT_NODE) {
                Element gyakornokElement = (Element) gyakornokNode;
                String Gy_kod = gyakornokElement.getAttribute("Gy_kod");
                String nev =
gyakornokElement.getElementsByTagName("Nev").item(0).getTextContent();
                String eletkor =
gyakornokElement.getElementsByTagName("Eletkor").item(0).getTextContent();

                System.out.println("------------------------------");
                System.out.println("Gyakornok:");
                System.out.println("    GyakornokKod=\"" + Gy_kod + "\"");
                System.out.println("    Nev: " + nev);
                System.out.println("    Eletkor: " + eletkor);

                NodeList gyakorlatNodes =
gyakornokElement.getElementsByTagName("Gyakorlat");
                for (int j = 0; j < gyakorlatNodes.getLength(); j++) {
                    Element gyakorlatElement = (Element)
gyakorlatNodes.item(j);
```

```
                    String kezdet =
gyakorlatElement.getElementsByTagName("Kezdete").item(0).getTextContent();
                    String idotartam =
gyakorlatElement.getElementsByTagName("Idotartam").item(0).getTextContent();

                    System.out.println("    Gyakorlat:");
                    System.out.println("        Kezdete: " + kezdet);
                    System.out.println("        Idotartam: " + idotartam);
                }
            }
        }
    }

    // Megrendelők kiírása
    private static void queryXMLMegrendelok(Document document) {
        NodeList megrendeloNodeList =
document.getElementsByTagName("Megrendelo");
        for (int i = 0; i < megrendeloNodeList.getLength(); i++) {
            Node megrendeloNode = megrendeloNodeList.item(i);
            if (megrendeloNode.getNodeType() == Node.ELEMENT_NODE) {
                Element megrendeloElement = (Element) megrendeloNode;
                String Mkod = megrendeloElement.getAttribute("Mkod");
                String nev =
megrendeloElement.getElementsByTagName("Nev").item(0).getTextContent();
                String varos =
megrendeloElement.getElementsByTagName("Varos").item(0).getTextContent();
                String utca =
megrendeloElement.getElementsByTagName("Utca").item(0).getTextContent();
                String hazszam =
megrendeloElement.getElementsByTagName("Hazszam").item(0).getTextContent();
                String telefonszam =
megrendeloElement.getElementsByTagName("Telefonszam").item(0).getTextContent();

                System.out.println("------------------------------");
                System.out.println("Megrendelo:");
                System.out.println("    MegrendeloKod=\"" + Mkod + "\"");
                System.out.println("    Nev: " + nev);
                System.out.println("    Cim:");
                System.out.println("        Varos: " + varos);
                System.out.println("        Utca: " + utca);
                System.out.println("        Hazszam: " + hazszam);
                System.out.println("    Telefonszam: " + telefonszam);
            }
```

```
        }
    }

    // Rendelések kiírása
    private static void queryXMLRendelesek(Document document) {
        NodeList rendelesNodeList = document.getElementsByTagName("Rendeles");
        for (int i = 0; i < rendelesNodeList.getLength(); i++) {
            Node rendelesNode = rendelesNodeList.item(i);
            if (rendelesNode.getNodeType() == Node.ELEMENT_NODE) {
                Element rendelesElement = (Element) rendelesNode;
                String GyarMR = rendelesElement.getAttribute("GyarMR");
                String Rkod = rendelesElement.getAttribute("Rkod");
                NodeList termekNodes =
rendelesElement.getElementsByTagName("Termek");

                System.out.println("------------------------------");
                System.out.println("Rendeles:");
                System.out.println("    RendelesKodok=");
                System.out.println("        GyarMR= " + GyarMR);
                System.out.println("        Rkod= " + Rkod);
                for (int j = 0; j < termekNodes.getLength(); j++) {
                    String termek = termekNodes.item(j).getTextContent();
                    System.out.println("    Termek: " + termek);
                }
                String osszeg =
rendelesElement.getElementsByTagName("Osszeg").item(0).getTextContent();
                String datum =
rendelesElement.getElementsByTagName("Datum").item(0).getTextContent();
                System.out.println("    Osszeg: " + osszeg);
                System.out.println("    Datum: " + datum);
            }
        }
    }
}
```

2.2 Adatírás – fájlnév: DOMWriteNeptunkod.java

writeFactoryDataToXML(): Ez a metódus létrehoz egy új XML dokumentumot, hozzáad egy gyökérelemet ("Ipar"), majd különböző elemeket (gyár, műszakvezetők, szakmunkások, gyakornokok, megrendelők, rendelések) szúr be.

Adatok hozzáadása az XML-hez:

addCimElement: Hozzáadja a gyár címét "Cim" néven, mely tartalmazza a várost, utcát és házszámot.

addMuszakVezeto: Új műszakvezető hozzáadása az adott gyárhoz.

addSzakmunkas: Szakmunkás adatainak beszúrása a gyár elem alá.

addGyakornok: Gyakornok hozzáadása, beleértve a név, életkor, gyakorlat kezdete és időtartama adatokat.

addMegrendelo: Megrendelő létrehozása és címeinek hozzáadása (város, utca, házszám, telefonszám).

addRendeles: Rendelés hozzáadása, amely termékeket, összeget és rendelési dátumot tartalmaz.

Forráskód:

```
package hu.domparse.fu7omc;

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 org.w3c.dom.*;

public class DOMWriteFU7OMC {

    public static void main(String[] args) {
        // Metódus meghívása az XML fájl generálásához
        writeFactoryDataToXML();
    }

    private static void writeFactoryDataToXML() {
        try {
            // Dokumentum létrehozása
            DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document document = builder.newDocument();

            // Gyökérelem létrehozása
            Element rootElement = document.createElement("Ipar");
```

```
            rootElement.setAttribute("xmlns:xsi",
"http://www.w3.org/2001/XMLSchema-instance");
            rootElement.setAttribute("xsi:noNamespaceSchemaLocation",
"XMLSchemaFU7OMC.xsd");
            document.appendChild(rootElement);

            // Gyar elem hozzáadása
            Element gyar = document.createElement("Gyar");
            gyar.setAttribute("Gyarkod", "A1");

            // Gyar adatai
            addTextElement(document, gyar, "Nev", "AcélMűvek");
            addCimElement(document, gyar);
            addTextElement(document, gyar, "Adoszam", "111111111-1-11");

            // Műszakvezetők
            addMuszakVezeto(document, gyar, "M1", "Trombitás Elemér",
"Lakatos", 62);
            addMuszakVezeto(document, gyar, "M2", "Nagy Ferenc", "Gépész", 55);

            // Szakmunkások
            addSzakmunkas(document, gyar, "SZ1", "Szabó Gábor", "Kisegítő",
32);
            addSzakmunkas(document, gyar, "SZ2", "Kovács László",
"Villanyszerelő", 45);

            // Gyakornokok
            addGyakornok(document, gyar, "Gyak1", "Illés Dániel", 20,
"2024.02.14","8 hét");
            addGyakornok(document, gyar, "Gyak2", "Hegedűs Emese",
22,"2024.02.14","8 hét");

            // Megrendelők
            addMegrendelo(document, gyar, "M1","Máv Zrt.", "Budapest", "Máv",
"32", "+36 20 456 4353");
            addMegrendelo(document, gyar, "M2","Volánbusz Zrt.", "Debrecen",
"Nagyerdei körút", "5", "+36 30 123 4567");

            // Rendelések
            addRendeles(document, gyar, "GYR1","R1","Vasúti Sín", "Acél Lemez",
"Csavar", "12345654.00", "2024-03-21");
```

```
            addRendeles(document, gyar, "GYR2","R2","Betongerenda", "Acélcső",
"", "5789000.00", "2024-04-10");

            // A gyár elem hozzáadása a gyökér elemhez
            rootElement.appendChild(gyar);

            // Dokumentum kiírása fájlba és konzolra
            writeXMLToFileAndConsole(document);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    // Gyar címének hozzáadása
    private static void addCimElement(Document document, Element gyar) {
        Element cim = document.createElement("Cim");

        // Város, Utca, Házszám
        addTextElement(document, cim, "Varos", "Miskolc");
        addTextElement(document, cim, "Utca", "Szép lány");
        addTextElement(document, cim, "Hazszam", "54");

        gyar.appendChild(cim);
    }

    // Műszakvezető hozzáadása
    private static void addMuszakVezeto(Document document, Element gyar, String
kod, String nev, String vegzettseg, int eletkor) {
        Element muszakVezeto = document.createElement("MuszakVezeto");
        muszakVezeto.setAttribute("M_kod", kod);

        addTextElement(document, muszakVezeto, "Nev", nev);
        addTextElement(document, muszakVezeto, "Vegzettseg", vegzettseg);
        addTextElement(document, muszakVezeto, "Eletkor",
String.valueOf(eletkor));

        gyar.appendChild(muszakVezeto);
    }

    // Szakmunkás hozzáadása
    private static void addSzakmunkas(Document document, Element gyar, String
kod, String nev, String vegzettseg, int eletkor) {
```

```
        Element szakmunkas = document.createElement("Szakmunkas");
        szakmunkas.setAttribute("Sz_kod", kod);

        addTextElement(document, szakmunkas, "Nev", nev);
        addTextElement(document, szakmunkas, "Vegzettseg", vegzettseg);
        addTextElement(document, szakmunkas, "Eletkor",
String.valueOf(eletkor));

        gyar.appendChild(szakmunkas);
    }

    // Gyakornok hozzáadása
    private static void addGyakornok(Document document, Element gyar, String
kod, String nev, int eletkor, String kezd, String ido) {
        Element gyakornok = document.createElement("Gyakornok");
        gyakornok.setAttribute("Gy_kod", kod);

        addTextElement(document, gyakornok, "Nev", nev);
        addTextElement(document, gyakornok, "Eletkor",
String.valueOf(eletkor));
        Element gyak = document.createElement("Gyakorlat");
        addTextElement(document, gyak, "Kezdete", kezd);
        addTextElement(document, gyak, "Idotartam", ido);
        gyakornok.appendChild(gyak);
        gyar.appendChild(gyakornok);
    }

    // Megrendelő hozzáadása
    private static void addMegrendelo(Document document, Element gyar, String
kod, String nev, String varos, String utca, String hazszam, String telefonszam)
{
        Element megrendelo = document.createElement("Megrendelo");
        megrendelo.setAttribute("Mkod", kod);

        addTextElement(document, megrendelo, "Nev", nev);
        Element cim = document.createElement("Cim");
        addTextElement(document, cim, "Varos", varos);
        addTextElement(document, cim, "Utca", utca);
        addTextElement(document, cim, "Hazszam", hazszam);
        megrendelo.appendChild(cim);

        addTextElement(document, megrendelo, "Telefonszam", telefonszam);
```

```
        gyar.appendChild(megrendelo);
    }

    // Rendelés hozzáadása
    private static void addRendeles(Document document, Element gyar, String
kod1, String kod2, String termek1, String termek2, String termek3, String
osszeg, String datum) {
        Element rendeles = document.createElement("Rendeles");
        rendeles.setAttribute("GyarMR", kod1);
        rendeles.setAttribute("Rkod", kod2);

        addTextElement(document, rendeles, "Termek", termek1);
        if (!termek2.isEmpty()) addTextElement(document, rendeles, "Termek",
termek2);
        if (!termek3.isEmpty()) addTextElement(document, rendeles, "Termek",
termek3);
        addTextElement(document, rendeles, "Osszeg", osszeg);
        addTextElement(document, rendeles, "Datum", datum);

        gyar.appendChild(rendeles);
    }

    // Szöveg elem hozzáadása
    private static void addTextElement(Document document, Element
parentElement, String tagName, String textContent) {
        Element element = document.createElement(tagName);
        element.appendChild(document.createTextNode(textContent));
        parentElement.appendChild(element);
    }

    // Az XML fájlba és konzolra történő írás
    private static void writeXMLToFileAndConsole(Document document) {
        try {
            // Transformer létrehozása és beállítása
            TransformerFactory transformerFactory =
TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
            transformer.setOutputProperty(OutputKeys.INDENT, "yes");
            transformer.setOutputProperty("{http://xml.apache.org/xslt}indent-
amount", "4");
```

```
            // Fájlba írás
            File xmlFile = new File("XMLFU7OMC1.xml");
            transformer.transform(new DOMSource(document), new
StreamResult(xmlFile));

            // Konzolra írás
            transformer.transform(new DOMSource(document), new
StreamResult(System.out));

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

2.3 Adatlekérdezés – fájlnév: DOMQueryNeptunkod.java

DOMQueryFU7OMC osztály: Ez az osztály olyan metódusokat tartalmaz, amelyek egy XML fájl tartalmát képes beolvasni és feldolgozni DOM segítségével, ráadásul lekérdezéseket is képesek vagyunk végrehajtani.

Lekérdezések és metódusok:

QueryXMLAllMuszakVezetok:

Leírás: Az összes műszakvezető adatainak kiírása.

Művelet: Kilistázza a MuszakVezeto elemek összes adatát, beleértve a kódot, nevet, végzettséget és életkort.

QueryXMLAllSzakmunkasok:

Leírás: Az összes szakmunkás adatainak kiírása.

Művelet: Kilistázza a Szakmunkas elemek adatait, például a kódot, nevet, végzettséget és életkort.

QueryXMLAllGyakornokok:

Leírás: Az összes gyakornok adatainak kiírása.

Művelet: Kilistázza a Gyakornok elemek kódját, nevét és életkorát.

QueryXMLAllMegrendelok:

Leírás: Az összes megrendelő adatainak kiírása.

Művelet: Kiírja a Megrendelo elemek összes adatát, beleértve a kódot, nevet, várost, utcát, házszámot és telefonszámot.

Forráskód:

```
package hu.domparse.fu7omc;

import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.*;

public class DOMQueryFU7OMC {

    // A fő metódus, ami meghívja a queryXMLDocument metódust a megadott XML
fájllal
    public static void main(String[] args) {
        queryXMLDocument("./XMLFU7OMC.xml");
    }

    // Metódus, amely az XML fájl beolvasására és feldolgozására szolgál
    private static void queryXMLDocument(String filePath) {
        try {
            File xmlFile = new File(filePath);
            DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            Document document = dBuilder.parse(xmlFile);
            document.getDocumentElement().normalize();

            //1. Az összes műszakvezető adatainak kiírása
            queryXMLAllMuszakVezetok(document);
            System.out.println("-------------------------------");
            //2. Az összes szakmunkás adatainak kiírása
            queryXMLAllSzakmunkasok(document);
            System.out.println("-------------------------------");
            //3. Az összes gyakornok adatainak kiírása
            queryXMLAllGyakornokok(document);
            System.out.println("-------------------------------");
            //4. Az összes megrendelő adatainak kiírása
            queryXMLAllMegrendelok(document);
            System.out.println("-------------------------------");
```

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

    // Műszakvezetők kiírása
    private static void queryXMLAllMuszakVezetok(Document document) {
        System.out.println("\n1. \033 Az összes műszakvezető adatainak
kiírása" );

        NodeList muszakVezetoNodeList =
document.getElementsByTagName("MuszakVezeto");
        for (int i = 0; i < muszakVezetoNodeList.getLength(); i++) {
            Node muszakVezetoNode = muszakVezetoNodeList.item(i);
            if (muszakVezetoNode.getNodeType() == Node.ELEMENT_NODE) {
                Element muszakVezetoElement = (Element) muszakVezetoNode;
                String kod = muszakVezetoElement.getAttribute("M_kod");
                String nev =
muszakVezetoElement.getElementsByTagName("Nev").item(0).getTextContent();
                String vegzettseg =
muszakVezetoElement.getElementsByTagName("Vegzettseg").item(0).getTextContent()
;
                String eletkor =
muszakVezetoElement.getElementsByTagName("Eletkor").item(0).getTextContent();

                System.out.println("<MuszakVezeto Kod=\"" + kod + "\">");
                System.out.println("    <Nev>" + nev + "</Nev>");
                System.out.println("    <Vegzettseg>" + vegzettseg +
"</Vegzettseg>");
                System.out.println("    <Eletkor>" + eletkor + "</Eletkor>");
                System.out.println("</MuszakVezeto>");
            }
        }
    }

    // Szakmunkások kiírása
    private static void queryXMLAllSzakmunkasok(Document document) {
        System.out.println("\n2. Az összes szakmunkás adatainak kiírása");

        NodeList szakmunkasNodeList =
document.getElementsByTagName("Szakmunkas");
        for (int i = 0; i < szakmunkasNodeList.getLength(); i++) {
```

```
            Node szakmunkasNode = szakmunkasNodeList.item(i);
            if (szakmunkasNode.getNodeType() == Node.ELEMENT_NODE) {
                Element szakmunkasElement = (Element) szakmunkasNode;
                String kod = szakmunkasElement.getAttribute("Sz_kod");
                String nev =
szakmunkasElement.getElementsByTagName("Nev").item(0).getTextContent();
                String vegzettseg =
szakmunkasElement.getElementsByTagName("Vegzettseg").item(0).getTextContent();
                String eletkor =
szakmunkasElement.getElementsByTagName("Eletkor").item(0).getTextContent();

                System.out.println("<Szakmunkas Kod=\"" + kod + "\">");
                System.out.println("    <Nev>" + nev + "</Nev>");
                System.out.println("    <Vegzettseg>" + vegzettseg +
"</Vegzettseg>");
                System.out.println("    <Eletkor>" + eletkor + "</Eletkor>");
                System.out.println("</Szakmunkas>");
            }
        }
    }

    // Gyakornokok kiírása
    private static void queryXMLAllGyakornokok(Document document) {
        System.out.println("\n3. Az összes gyakornok adatainak kiírása");

        NodeList gyakornokNodeList =
document.getElementsByTagName("Gyakornok");
        for (int i = 0; i < gyakornokNodeList.getLength(); i++) {
            Node gyakornokNode = gyakornokNodeList.item(i);
            if (gyakornokNode.getNodeType() == Node.ELEMENT_NODE) {
                Element gyakornokElement = (Element) gyakornokNode;
                String kod = gyakornokElement.getAttribute("Gy_kod");
                String nev =
gyakornokElement.getElementsByTagName("Nev").item(0).getTextContent();
                String eletkor =
gyakornokElement.getElementsByTagName("Eletkor").item(0).getTextContent();

                System.out.println("<Gyakornok Kod=\"" + kod + "\">");
                System.out.println("    <Nev>" + nev + "</Nev>");
                System.out.println("    <Eletkor>" + eletkor + "</Eletkor>");
                System.out.println("</Gyakornok>");
            }
        }
```

```
    }

    // Megrendelők kiírása
    private static void queryXMLAllMegrendelok(Document document) {
        System.out.println("\n4. Az összes megrendelő adatainak kiírása");

        NodeList megrendeloNodeList =
document.getElementsByTagName("Megrendelo");
        for (int i = 0; i < megrendeloNodeList.getLength(); i++) {
            Node megrendeloNode = megrendeloNodeList.item(i);
            if (megrendeloNode.getNodeType() == Node.ELEMENT_NODE) {
                Element megrendeloElement = (Element) megrendeloNode;
                String kod = megrendeloElement.getAttribute("Mkod");
                String nev =
megrendeloElement.getElementsByTagName("Nev").item(0).getTextContent();
                String varos =
megrendeloElement.getElementsByTagName("Varos").item(0).getTextContent();
                String utca =
megrendeloElement.getElementsByTagName("Utca").item(0).getTextContent();
                String hazszam =
megrendeloElement.getElementsByTagName("Hazszam").item(0).getTextContent();
                String telefonszam =
megrendeloElement.getElementsByTagName("Telefonszam").item(0).getTextContent();

                System.out.println("<Megrendelo Kod=\"" + kod + "\">");
                System.out.println("    <Nev>" + nev + "</Nev>");
                System.out.println("    <Cim>");
                System.out.println("        <Varos>" + varos + "</Varos>");
                System.out.println("        <Utca>" + utca + "</Utca>");
                System.out.println("        <Hazszam>" + hazszam +
"</Hazszam>");
                System.out.println("    </Cim>");
                System.out.println("    <Telefonszam>" + telefonszam +
"</Telefonszam>");
                System.out.println("</Megrendelo>");
            }
        }
    }
}
```

2.4 Adatmódosítás – fájlnév: DOMModifyNeptunkod.java

DOMModifyFU7OMC osztály: Ez az osztály végzi el az XML fájlban a módosításokat a DOM használatával.

Adatmódosításhoz a következő segédmetódusokat használtam:

getElementsByTagName: Node-listát ad vissza, amely az adott elem nevű összes leszármazottját tartalmazza a hívott elemen belül.

setTextContext: beállítja az elem szövegtartalmát a megadott értékre.

modifyXMLElements metódus: ez a metódus fogja módosítani az 'XMLFU7OMC.xml' fájlt és megjeleníteni a konzolra.

Forráskód:

```
package hu.domparse.fu7omc;

import java.io.File;
import org.w3c.dom.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class DOMModifyFU7OMC {
    public static void main(String[] args) {
        // Módosító metódus meghívása a megadott XML fájlra
        modifyXMLElements("./XMLFU7OMC.xml");
    }

    private static void modifyXMLElements(String filePath) {
        try {
            // XML fájl beolvasása
            File xmlFile = new File(filePath);
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document document = dBuilder.parse(xmlFile);

            // 1. Módosítás: Műszakvezető név módosítása
            NodeList muszakVezetoList =
document.getElementsByTagName("MuszakVezeto");
            Element muszakVezetoElement = (Element) muszakVezetoList.item(0);
// Első műszakvezető
```

```
muszakVezetoElement.getElementsByTagName("Nev").item(0).setTextContent("Trombitás Elemér");

            // 2. Módosítás: Szakmunkás életkorának módosítása
            NodeList szakmunkasList = document.getElementsByTagName("Szakmunkas");
            Element szakmunkasElement = (Element) szakmunkasList.item(2); // Harmadik szakmunkás

szakmunkasElement.getElementsByTagName("Eletkor").item(0).setTextContent("56");

            // 3. Módosítás: Gyakornok név módosítása
            NodeList gyakornokList = document.getElementsByTagName("Gyakornok");
            Element gyakornokElement = (Element) gyakornokList.item(1); // Második gyakornok

gyakornokElement.getElementsByTagName("Nev").item(0).setTextContent("Hegedűs Emese");

            // 4. Módosítás: Megrendelő címének frissítése
            NodeList megrendeloList = document.getElementsByTagName("Megrendelo");
            Element megrendeloElement = (Element) megrendeloList.item(0); // Első megrendelő
            Element cimElement = (Element) megrendeloElement.getElementsByTagName("Cim").item(0);

cimElement.getElementsByTagName("Varos").item(0).setTextContent("Debrecen");

cimElement.getElementsByTagName("Utca").item(0).setTextContent("Kossuth Lajos utca");

cimElement.getElementsByTagName("Hazszam").item(0).setTextContent("101");

            // 5. Módosítás: Rendelés termék árának módosítása
            NodeList rendelesList = document.getElementsByTagName("Rendeles");
            Element rendelesElement = (Element) rendelesList.item(0); // Első rendelés
            NodeList osszegNodes = rendelesElement.getElementsByTagName("Osszeg");
```

```
            Element osszegElement = (Element) osszegNodes.item(0); // The
<Osszeg> element
            osszegElement.setTextContent("19999"); // Update the value

            // A módosított XML kiírása a konzolra
            TransformerFactory transformerFactory =
TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            DOMSource source = new DOMSource(document);

            // A módosított XML fájl konzolra történő kiíratása
            StreamResult consoleResult = new StreamResult(System.out);
            transformer.transform(source, consoleResult);

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