JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Hell cég

Készítette: **Csicsely Gábor**

Neptunkód: **C7H5VB**

Dátum: 2023.12.03

Tartalomjegyzék

Bevezetés

A feladat leírása:

A feladatban a Hell cég működésével kapcsolatos ER modellből készítettem egy XDM modellt, amiből készült egy XML és XSD valamint hozzá egy azt kezelő DOM program az Eclipse IDE használatával.

Az XSD és XML validálásához egy online XML validátor volt használva.

A rendszer nyilvántartja a Dolgozókat(ID, Név, Beosztás, Fizetés, Telefonszám, Lakcím), a Hell termékeket gyártó Gyárakat(Gyár Azonosító, Gyártási ráta, Kiadások, Címe) a Termékeket(Vállalati Azonosító, Fajtája, Bevétel, Kiadások) amiket gyártanak.
Ezen felül nyilván vannak tartva a Megrendelők(Megrendelő Azonosító, Elérhetőség, Címe) és a Hell cégnek egyik legnagyobb látványossága, üdülő központja, az Avalon Park(Azonosító, Bevétel, Kiadások, Címe) és annak Vezetője(ID, Név, Lakcím)

A rendszer megbízhatóan nyilván tudja tartani így a bevételeket, fizetéseket, kiadásokat, valamint a gyárak gyártási rátáját, azt, hogy melyik megrendelő mennyit, és melyik gyárból rendel.

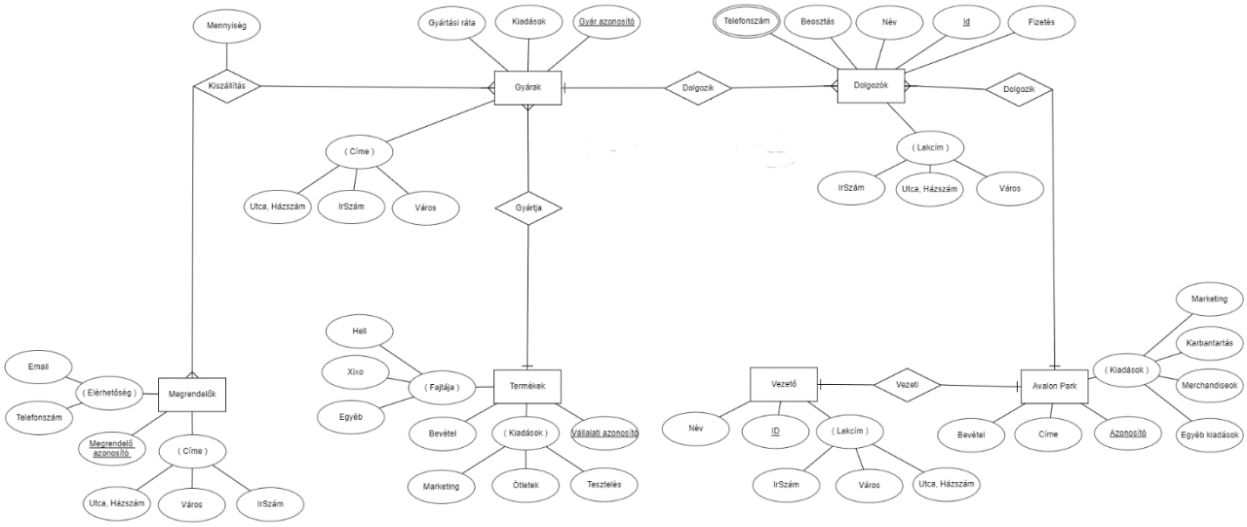
Az XML dokumentum szabályosságára egy hozzá írt XSD dokumentum figyel, ezzel biztosítva a hibátlan működést.

A hozzá írt Java DOM program képes egyszerűbb lekérdezésekre, az XML kiírására, valamint módosítására és annak mentésére.

1. feladat

1a) Az adatbázis ER modell tervezése

Az ER Modell:



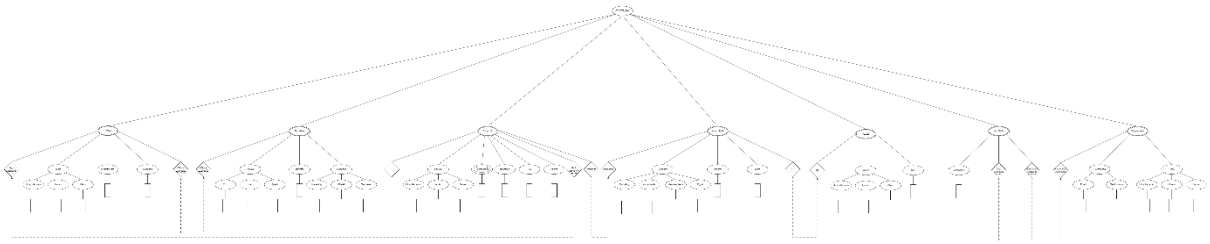
A modell 6 egyedet tartalmaz: Megrendelők, Gyárak, Dolgozók, Avalon Park, Vezető és a Termékek. Ezek között megtalálhatóak 1:1 , 1:N és N:M kapcsolatok is.

A Megrendelők és a Gyárak közötti kapcsolatnak van egy tulajdonsága(Mennyiség), amivel a rendelés méretét tudjuk kezelni.

Az ER Modell az ERDPlus használatával készült el.

1b) Az adatbázis konvertálása XDM modellre

Az XDM Modell:



A modell készítése alatt ügyeltem arra, hogy rendezett legyen és a vonalak ne keresztezzék egymást.

A PK gyémánt alakkal és aláhúzással van ábrázolva, a FK pedig szaggatott vonalas gyémánt alakkal, aláhúzással.

Amelyik egyedből több lehet, azok dupla ellipszissel lettek ábrázolva.

1c) Az XDM modell alapján XML dokumentum készítése:

Itt egyszerűen át kellett alakítani az XDM modellt XML-re.

A Gyökérelem neve C7H5VB_Hell lett.

Ahol több elem lehetett, ott mindegyikből készült 3 példány.

A példányok előtt megjegyzésként ott van a nevük átláthatóság érdekében.

Az XML kódja:

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

<C7H5VB_Hell>

<!--Gyárak-->
<Gyarak GyarAzonosito="1" VallalatiAzonosito="1">
<Cime>
        <UtcaHazszam>Vas utca 7</UtcaHazszam>
        <IrSzam>3534</IrSzam>
        <Varos>Miskolc</Varos>
</Cime>
<GyartasiRata>10000</GyartasiRata>
<Kiadasok>10000000</Kiadasok>
</Gyarak>

<Gyarak GyarAzonosito="2" VallalatiAzonosito="2">
<Cime>
        <UtcaHazszam>Gyár utca 1</UtcaHazszam>
        <IrSzam>3525</IrSzam>
        <Varos>Miskolc</Varos>
</Cime>
<GyartasiRata>15000</GyartasiRata>
<Kiadasok>13000000</Kiadasok>
```

```
</Gyarak>

<Gyarak GyarAzonosito="3" VallalatiAzonosito="3">
<Cime>
        <UtcaHazszam>Hell utca 1</UtcaHazszam>
        <IrSzam>2092</IrSzam>
        <Varos>Budakeszi</Varos>
</Cime>
<GyartasiRata>35000</GyartasiRata>
<Kiadasok>19500000</Kiadasok>
</Gyarak>


<!--Termékek-->
<Termekek VallalatiAzonosito="1">
<Fajtaja>
        <Xixo>1</Xixo>
</Fajtaja>
<Bevetel>50000000</Bevetel>
<Kiadasok>
        <Marketing>1200000</Marketing>
        <Otletek>1000000</Otletek>
        <Teszteles>500000</Teszteles>
</Kiadasok>
</Termekek>

<Termekek VallalatiAzonosito="2">
<Fajtaja>
        <Hell>1</Hell>
</Fajtaja>
<Bevetel>80000000</Bevetel>
<Kiadasok>
```

<Marketing>3200000</Marketing>

<Otletek>1200000</Otletek>

<Teszteles>700000</Teszteles>

</Kiadasok>

</Termekek>

<Termekek VallalatiAzonosito="3">

<Fajtaja>

<Egyeb>1</Egyeb>

</Fajtaja>

<Bevetel>7000000</Bevetel>

<Kiadasok>

<Marketing>700000</Marketing>

<Otletek>250000</Otletek>

<Teszteles>150000</Teszteles>

</Kiadasok>

</Termekek>

<!--Dolgozók-->

<Dolgozok ID="1" GyarAzonosito="1">

<Lakcim>

<UtcaHazszam>Kerek utca 7</UtcaHazszam>

<IrSzam>3533</IrSzam>

<Varos>Miskolc</Varos>

</Lakcim>

<Beosztas>Gyártósoros</Beosztas>

<Nev>Nagy Ernő</Nev>

<Fizetes>250000</Fizetes>

<Telefonszam>06307687647</Telefonszam>

<Telefonszam>06508695857</Telefonszam>

</Dolgozok>

<Dolgozok ID="2" GyarAzonosito="3">
<Lakcim>
<UtcaHazszam>Kis utca 23</UtcaHazszam>
<IrSzam>3523</IrSzam>
<Varos>Miskolc</Varos>
</Lakcim>
<Beosztas>Gyártósoros</Beosztas>
<Nev>Látó Lajos</Nev>
<Fizetes>245000</Fizetes>
<Telefonszam>06307661589</Telefonszam>
</Dolgozok>

<Dolgozok ID="3" GyarAzonosito="2">
<Lakcim>
<UtcaHazszam>Mester utca 2</UtcaHazszam>
<IrSzam>2092</IrSzam>
<Varos>Budakeszi</Varos>
</Lakcim>
<Beosztas>Takarító</Beosztas>
<Nev>Hangos Ernő</Nev>
<Fizetes>220000</Fizetes>
<Telefonszam>062096758956</Telefonszam>
</Dolgozok>

<!--AvalonPark-->
<AvalonPark Azonosito="1" VezID="1">
<Kiadasok>
<Marketing>1000000</Marketing>
<Karbantartas>2500000</Karbantartas>
<Merchandiseok>3300000</Merchandiseok>

```
        <Egyeb>3000000</Egyeb>
</Kiadasok>
<Bevetel>60000000</Bevetel>
<Cime>Miskolc Iglói u. 15, 3519</Cime>
</AvalonPark>



<!--Vezető-->
<Vezeto VezID="1">
<Lakcim>
        <UtcaHazszam>Leveles utca 12</UtcaHazszam>
        <IrSzam>3520</IrSzam>
        <Varos>Miskolc</Varos>
</Lakcim>
<Nev>Lázár Ervin</Nev>
</Vezeto>



<!--Kiszállítás-->
<Kiszallitas GyarAzonosito="1" MegrendeloAzonosito="1">
<Mennyiseg>2500</Mennyiseg>
</Kiszallitas>

<Kiszallitas GyarAzonosito="2" MegrendeloAzonosito="2">
<Mennyiseg>5000</Mennyiseg>
</Kiszallitas>

<Kiszallitas GyarAzonosito="3" MegrendeloAzonosito="3">
<Mennyiseg>1200</Mennyiseg>
</Kiszallitas>
```

```
<!--Megrendelők-->
<Megrendelok MegrendeloAzonosito="1">
<Elerhetoseg>
        <Email>ugyfelszolgalat@coop.hu</Email>
        <TelefonSzam>06307658799</TelefonSzam>
</Elerhetoseg>
<Cime>
        <UtcaHazszam>Corvin utca 7</UtcaHazszam>
        <IrSzam>3525</IrSzam>
        <Varos>Miskolc</Varos>
</Cime>
</Megrendelok>

<Megrendelok MegrendeloAzonosito="2">
<Elerhetoseg>
        <Email>beszerzes@spar.hu</Email>
        <TelefonSzam>06407658788</TelefonSzam>
</Elerhetoseg>
<Cime>
        <UtcaHazszam>Magas út 26</UtcaHazszam>
        <IrSzam>3533</IrSzam>
        <Varos>Miskolc</Varos>
</Cime>
</Megrendelok>

<Megrendelok MegrendeloAzonosito="3">
<Elerhetoseg>
        <Email>megrendelo@arena.hu</Email>
        <TelefonSzam>06105758595</TelefonSzam>
</Elerhetoseg>
```

```
<Cime>
        <UtcaHazszam>Kerepesi út 9</UtcaHazszam>
        <IrSzam>1087</IrSzam>
        <Varos>Budapest</Varos>
</Cime>
</Megrendelok>

</C7H5VB_Hell>.
```

1d) Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek.

Itt az XSD-ben megadjuk, hogy az XML melyik eleme milyen adat típusokat tartalmazhat, beállítjuk a kapcsolatokat FK és PK definiálásával.

Ahol az ER Modellben Composite Attribute volt, ott ComplexType segítségével hoztuk létre az element-et

Az XSD kódja:.

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

<!-- Győkérelem definiálása -->
<xs:element name="C7H5VB_Hell">
<xs:complexType>
        <xs:sequence>
                <xs:element ref="Gyarak" maxOccurs="unbounded"/>
                <xs:element ref="Termekek" maxOccurs="unbounded"/>
                <xs:element ref="Dolgozok" maxOccurs="unbounded"/>
                <xs:element ref="AvalonPark" maxOccurs="1"/>
                <xs:element ref="Vezeto" maxOccurs="1"/>
                <xs:element ref="Kiszallitas" maxOccurs="unbounded"/>
                <xs:element ref="Megrendelok" maxOccurs="unbounded"/>
        </xs:sequence>
</xs:complexType>

<!-- PK és FK létrehozása -->
```

```
<xs:key name="GyarAzonosito">
        <xs:selector xpath="Gyarak"/>
        <xs:field xpath="@GyarAzonosito"/>
</xs:key>

<xs:key name="VallalatiAzonosito">
        <xs:selector xpath="Termekek"/>
        <xs:field xpath="@VallalatiAzonosito"/>
</xs:key>
<xs:keyref name="GyarakVallalatiAzonosito" refer="VallalatiAzonosito">
        <xs:selector xpath="Gyarak"/>
        <xs:field xpath="@VallalatiAzonosito"/>
</xs:keyref>

<xs:key name="ID">
        <xs:selector xpath="Dolgozok"/>
        <xs:field xpath="@ID"/>
</xs:key>
<xs:keyref name="DolgozokGyarAzonosito" refer="GyarAzonosito">
        <xs:selector xpath="Dolgozok"/>
        <xs:field xpath="@GyarAzonosito"/>
</xs:keyref>

<xs:key name="Azonosito">
        <xs:selector xpath="AvalonPark"/>
        <xs:field xpath="@Azonosito"/>
</xs:key>
<xs:keyref name="AvalonParkVezID" refer="VezID">
        <xs:selector xpath="AvalonPark"/>
        <xs:field xpath="@VezID"/>
</xs:keyref>

<xs:key name="VezID">
        <xs:selector xpath="Vezeto"/>
        <xs:field xpath="@VezID"/>
</xs:key>
```

```
<xs:keyref name="KiszallitasMegrendeloAzonosito" refer="MegrendeloAzonosito">
        <xs:selector xpath="Kiszallitas"/>
        <xs:field xpath="@MegrendeloAzonosito"/>
</xs:keyref>

<xs:keyref name="KiszallitasGyarAzonosito" refer="GyarAzonosito">
        <xs:selector xpath="Kiszallitas"/>
        <xs:field xpath="@GyarAzonosito"/>
</xs:keyref>

<xs:key name="MegrendeloAzonosito">
        <xs:selector xpath="Megrendelok"/>
        <xs:field xpath="@MegrendeloAzonosito"/>
</xs:key>

</xs:element>


<!-- C7H5VB_Hell Gyermek elemei definiálása -->
<xs:element name="Gyarak">
<xs:complexType>
        <xs:sequence>
                <xs:element name="Cime" type="GyarakCimeType"/>
                <xs:element name="GyartasiRata" type="xs:string"/>
                <xs:element name="Kiadasok" type="penzosszeg"/>
        </xs:sequence>
        <xs:attribute name="GyarAzonosito" type="xs:integer"/>
        <xs:attribute name="VallalatiAzonosito" type="xs:integer"/>
</xs:complexType>
</xs:element>

<xs:element name="Termekek">
<xs:complexType>
        <xs:sequence>
                <xs:element name="Fajtaja" type="FajtajaType"/>
```

```
                <xs:element name="Bevetel" type="penzosszeg"/>
                <xs:element name="Kiadasok" type="TermekekKiadasokType"/>
        </xs:sequence>
        <xs:attribute name="VallalatiAzonosito" type="xs:integer"/>
</xs:complexType>
</xs:element>

<xs:element name="Dolgozok">
<xs:complexType>
        <xs:sequence>
                <xs:element name="Lakcim" type="DolgozokLakcimType"/>
                <xs:element name="Beosztas" type="xs:string"/>
                <xs:element name="Nev" type="xs:string"/>
                <xs:element name="Fizetes" type="penzosszeg"/>
                <xs:element name="Telefonszam" type="xs:string" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ID" type="xs:integer"/>
        <xs:attribute name="GyarAzonosito" type="xs:integer"/>
</xs:complexType>
</xs:element>

<xs:element name="AvalonPark">
<xs:complexType>
        <xs:sequence>
                <xs:element name="Kiadasok" type="AvalonParkKiadasokType"/>
                <xs:element name="Bevetel" type="penzosszeg"/>
                <xs:element name="Cime" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="Azonosito" type="xs:integer"/>
        <xs:attribute name="VezID" type="xs:integer"/>
</xs:complexType>
</xs:element>

<xs:element name="Vezeto">
<xs:complexType>
        <xs:sequence>
```

```
                <xs:element name="Lakcim" type="DolgozokLakcimType"/>
                <xs:element name="Nev" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="VezID" type="xs:integer"/>
</xs:complexType>
</xs:element>

<xs:element name="Kiszallitas">
<xs:complexType>
        <xs:sequence>
                <xs:element name="Mennyiseg" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="GyarAzonosito" type="xs:integer"/>
        <xs:attribute name="MegrendeloAzonosito" type="xs:integer"/>
</xs:complexType>
</xs:element>

<xs:element name="Megrendelok">
<xs:complexType>
        <xs:sequence>
                <xs:element name="Elerhetoseg" type="ElerhetosegType"/>
                <xs:element name="Cime" type="DolgozokLakcimType"/>
        </xs:sequence>
        <xs:attribute name="MegrendeloAzonosito" type="xs:integer"/>
</xs:complexType>
</xs:element>


<!-- Komplex típusok definiálása -->
<xs:complexType name="GyarakCimeType">
<xs:sequence>
        <xs:element name="UtcaHazszam" type="xs:string"/>
        <xs:element name="IrSzam" type="xs:string"/>
        <xs:element name="Varos" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="FajtajaType">
<xs:choice>
	<xs:element name="Xixo" type="xs:string"/>
	<xs:element name="Hell" type="xs:string"/>
	<xs:element name="Egyeb" type="xs:string"/>
</xs:choice>
</xs:complexType>

<xs:complexType name="TermekekKiadasokType">
<xs:sequence>
	<xs:element name="Marketing" type="penzosszeg"/>
	<xs:element name="Otletek" type="penzosszeg"/>
	<xs:element name="Teszteles" type="penzosszeg"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="AvalonParkKiadasokType">
<xs:sequence>
	<xs:element name="Marketing" type="penzosszeg"/>
	<xs:element name="Karbantartas" type="penzosszeg"/>
	<xs:element name="Merchandiseok" type="penzosszeg"/>
	<xs:element name="Egyeb" type="penzosszeg"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="DolgozokLakcimType">
<xs:sequence>
	<xs:element name="UtcaHazszam" type="xs:string"/>
	<xs:element name="IrSzam" type="xs:string"/>
	<xs:element name="Varos" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="ElerhetosegType">
<xs:sequence>
```

```
            <xs:element name="Email" type="xs:string"/>
            <xs:element name="TelefonSzam" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<!-- Egyszerű típus definiálása -->
<xs:simpleType name="penzosszeg">
<xs:restriction base="xs:integer"/>
</xs:simpleType>

</xs:schema>
```

2. feladat

A feladat egy DOM program készítése az XML dokumentum - *XMLNeptunkod.xml* – adatainak adminisztrálása alapján:

Project name: DOMParseC7h5vb

Package: hu.domparse.c7h5vb

Class names: (DomReadC7h5vb, DomModifyC7h5vb, DomQueryC7h5vb, DOMWriteC7h5vb)

2a) adatolvasás (kód – comment együtt) – fájlnév: *DOMReadNeptunkod.java* megvalósítás

Ebben a Java DOM kódban elsőnek beolvassuk az XML fájlunkat, majd a konzolba strukturáltan kiírjuk.

A fájl beolvasása után a DocumentBuildert és a DocumentBuilderFactory használatával egy DOM fát építünk fel az XML filebol.

A kód:

```
public class DomReadC7h5vb {

   public static void main(String[] args) {

      try {

         //Beolvassa az XML fájlt

         File xmlFile = new File("XMLC7H5VB.xml");

         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

         DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

         Document doc = dBuilder.parse(xmlFile);
```

```
        doc.getDocumentElement().normalize();

        // Gyökér elem lekérése
        Element rootElement = doc.getDocumentElement();
        System.out.println("Gyökér elem: " + rootElement.getNodeName());

        // Fa teljes tartalmának kiírása
        printNode(rootElement, "");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private static void printNode(Node node, String indent) {
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        System.out.println(indent + "Elem: " + node.getNodeName());

        if (node.hasAttributes()) {
            NamedNodeMap attributes = node.getAttributes();
            for (int i = 0; i < attributes.getLength(); i++) {
                Node attribute = attributes.item(i);
                System.out.println(indent + "  Attribútum: " + attribute.getNodeName() + " = " + attribute.getNodeValue());
            }
        }

        if (node.hasChildNodes()) {
            NodeList children = node.getChildNodes();
            for (int i = 0; i < children.getLength(); i++) {
                Node child = children.item(i);
                printNode(child, indent + "  ");
            }
        }
```

```
        } else if (node.getNodeType() == Node.TEXT_NODE) {
            String textContent = node.getNodeValue().trim();
            if (!textContent.isEmpty()) {
                System.out.println(indent + "Tartalom: " + textContent);
            }
        }
    }
}
```

2b) adatmódosítás (kód – comment együtt) – fájlnév: *DOMModifyNeptunkod.java*

Itt miután beolvastuk az XML filet, végrehajtottunk 5 fajta módosítást:
- Dolgozóhoz telefonszám hozzáadása
- Vezető nevének módosítása
- Avalon Park bevételének növelése
- Eltávolítja az első megrendelőt
- Gyártósoros dolgozók fizetésének emelése

A Kód:

```
public class DomModifyC7h5vb {

    public static void main(String[] args) {
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();

            // XML dokumentum beolvasása
            Document document = builder.parse("XMLC7H5VB.xml");

            // Dolgozóhoz telefonszám hozzáadása
            addPhoneNumber(document, 2, "06101234567");

            // Vezető nevének módosítása
            modifyLeaderName(document, "Nagy Ervin");
```

```
        // Avalon Park bevétel növelése
        increaseAvalonParkBevetel(document, 10000);
        // Eltávolítja az első megrendelőt
        removeFirstCustomer(document);

        // Gyártósoros dolgozók fizetésének emelése
        increaseSalary(document, 1500);



        // Fájlba írás és konzolra írás
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        transformer.setOutputProperty(OutputKeys.INDENT, "yes");
        DOMSource source = new DOMSource(document);

        // Konzolra írás
        StreamResult console = new StreamResult(System.out);
        transformer.transform(source, console);

        // Fájlba mentés
        StreamResult file = new StreamResult("XMLC7H5VBModify.xml");
        transformer.transform(source, file);

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

private static void addPhoneNumber(Document doc, int dolgozoID, String newPhoneNumber) {
    NodeList dolgozok = doc.getElementsByTagName("Dolgozok");
    for (int i = 0; i < dolgozok.getLength(); i++) {
        Element dolgozo = (Element) dolgozok.item(i);
```

```
        if (Integer.parseInt(dolgozo.getAttribute("ID")) == dolgozoID) {
            Element phoneElem = doc.createElement("Telefonszam");
            phoneElem.setTextContent(newPhoneNumber);
            dolgozo.appendChild(phoneElem);
            break;
        }
    }
}

private static void modifyLeaderName(Document doc, String newName) {
    NodeList vezetok = doc.getElementsByTagName("Vezeto");
    if (vezetok.getLength() > 0) {
        Element vezeto = (Element) vezetok.item(0);
        vezeto.getElementsByTagName("Nev").item(0).setTextContent(newName);
    }
}

private static void increaseAvalonParkBevetel(Document doc, int increaseAmount) {
    NodeList avalonParks = doc.getElementsByTagName("AvalonPark");
    for (int i = 0; i < avalonParks.getLength(); i++) {
        Element avalonPark = (Element) avalonParks.item(i);
        Element bevetelElem = (Element) avalonPark.getElementsByTagName("Bevetel").item(0);
        try {
            int currentRevenue = Integer.parseInt(bevetelElem.getTextContent());
            int newRevenue = currentRevenue + increaseAmount;
            bevetelElem.setTextContent(Integer.toString(newRevenue));
        } catch (NumberFormatException e) {
            System.err.println("Nem sikerült az érték növelése a 'Bevetel' elemnél: " + bevetelElem.getTextContent());
        }
    }
}
```

```
private static void removeFirstCustomer(Document doc) {
    NodeList megrendelok = doc.getElementsByTagName("Megrendelok");
    if (megrendelok.getLength() > 0) {
        Node firstCustomer = megrendelok.item(0);
        firstCustomer.getParentNode().removeChild(firstCustomer);
    }
}


    private static void increaseSalary(Document doc, int amount) {
        NodeList dolgozok = doc.getElementsByTagName("Dolgozok");
        for (int i = 0; i < dolgozok.getLength(); i++) {
            Element dolgozo = (Element) dolgozok.item(i);
            if (dolgozo.getElementsByTagName("Beosztas").item(0).getTextContent().equals("Gyártósoros")) {
                int currentSalary = Integer.parseInt(dolgozo.getElementsByTagName("Fizetes").item(0).getTextContent());
                dolgozo.getElementsByTagName("Fizetes").item(0).setTextContent(String.valueOf(currentSalary + amount));
            }
        }
    }
}
```

2c) *adatlekérdezés* (kód – comment együtt) – fájlnév: *DOMQueryNeptunkod.java*

Itt 5 lekérdezést valósítottam meg:

- Az összes gyár azonosítójának kiírása

-Gyárak címeinek kiírása

-Összes dolgozónak a fizetésének összege

- Gyártósoros dolgozóknak a fizetésének összege

-Összetett lekérdezés: a 3000-nél nagyobb rendeléseknek a gyártójának címe és megrendelőjének a címe, valamint rendelés mérete.

A Kód:

```
package hu.domparse.c7h5vb;
```

```
import org.w3c.dom.*;
import javax.xml.parsers.*;
import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class DomQueryC7h5vb {
    public static void main(String[] args) {
        try {
            File xmlFile = new File("XMLC7H5VB.xml");
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(xmlFile);
            doc.getDocumentElement().normalize();

            System.out.println("\nGyárak:");
            List<String> gyarak = getGyarak(doc);
            for (String gyar : gyarak) {
                System.out.println(gyar);
            }

            System.out.println("\nGyárak címei:");
            List<String> cimek = getGyarCimek(doc);
            for (String cim : cimek) {
                System.out.println(cim);
            }
            System.out.println("\nDolgozók fizetéseinek összege:");
            int osszFizetes = getOsszFizetes(doc);
            System.out.println("Összesített fizetés: " + osszFizetes);
```

```
        System.out.println("\n'Gyártósoros' beosztásban dolgozók összfizetése:");
        int osszFizetesGyartosoros = getOsszFizetesGyartosoros(doc);
        System.out.println("Gyártósoros dolgozók összfizetése: " + osszFizetesGyartosoros);

        System.out.println("\nÖsszetett lekérdezés eredménye:");
        List<String> osszetettLekerdezesEredmeny = osszetettLekerdezes(doc);
        for (String eredmeny : osszetettLekerdezesEredmeny) {
            System.out.println(eredmeny);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private static List<String> getGyarak(Document doc) {
    List<String> gyarak = new ArrayList<>();
    NodeList gyarakElements = doc.getElementsByTagName("Gyarak");
    for (int i = 0; i < gyarakElements.getLength(); i++) {
        Element gyarElement = (Element) gyarakElements.item(i);
        String gyarAzonosito = gyarElement.getAttribute("GyarAzonosito");
        gyarak.add("Gyár Azonosító: " + gyarAzonosito);
    }
    return gyarak;
}

private static List<String> getGyarCimek(Document doc) {
    List<String> cimek = new ArrayList<>();
    NodeList gyarakElements = doc.getElementsByTagName("Gyarak");
    for (int i = 0; i < gyarakElements.getLength(); i++) {
```

```
            Element gyarElement = (Element) gyarakElements.item(i);
            Element cimElement = (Element) gyarElement.getElementsByTagName("Cime").item(0);
            String utca = cimElement.getElementsByTagName("UtcaHazszam").item(0).getTextContent();
            String varos = cimElement.getElementsByTagName("Varos").item(0).getTextContent();
            cimek.add("Cím: " + utca + ", " + varos);
        }
        return cimek;
    }

    private static List<String> osszetettLekerdezes(Document doc) {
        List<String> eredmeny = new ArrayList<>();
        NodeList kiszallitasElements = doc.getElementsByTagName("Kiszallitas");
        NodeList gyarakElements = doc.getElementsByTagName("Gyarak");
        NodeList megrendelokElements = doc.getElementsByTagName("Megrendelok");

        for (int i = 0; i < kiszallitasElements.getLength(); i++) {
            Element kiszallitasElement = (Element) kiszallitasElements.item(i);
            int mennyiseg =
Integer.parseInt(kiszallitasElement.getElementsByTagName("Mennyiseg").item(0).getTextContent());

            if (mennyiseg > 3000) {
                String gyarAzonosito = kiszallitasElement.getAttribute("GyarAzonosito");
                String megrendeloAzonosito = kiszallitasElement.getAttribute("MegrendeloAzonosito");

                String gyarCim = getCim(gyarakElements, gyarAzonosito);
                String megrendeloCim = getCim(megrendelokElements, megrendeloAzonosito);

                eredmeny.add("Mennyiség: " + mennyiseg + ", Gyár Címe: " + gyarCim + ", Megrendelő Címe: " +
megrendeloCim);
            }
```

```
    }
    return eredmeny;
}

private static int getOsszFizetes(Document doc) {
    int osszeg = 0;
    NodeList dolgozok = doc.getElementsByTagName("Dolgozok");
    for (int i = 0; i < dolgozok.getLength(); i++) {
        Element dolgozo = (Element) dolgozok.item(i);
        int fizetes = Integer.parseInt(dolgozo.getElementsByTagName("Fizetes").item(0).getTextContent());
        osszeg += fizetes;
    }
    return osszeg;
}

private static int getOsszFizetesGyartosoros(Document doc) {
    int osszeg = 0;
    NodeList dolgozok = doc.getElementsByTagName("Dolgozok");
    for (int i = 0; i < dolgozok.getLength(); i++) {
        Element dolgozo = (Element) dolgozok.item(i);
        String beosztas = dolgozo.getElementsByTagName("Beosztas").item(0).getTextContent();
        if ("Gyártósoros".equals(beosztas)) {
            int fizetes = Integer.parseInt(dolgozo.getElementsByTagName("Fizetes").item(0).getTextContent());
            osszeg += fizetes;
        }
    }
    return osszeg;
}

private static String getCim(NodeList elements, String azonosito) {
```

```
        for (int i = 0; i < elements.getLength(); i++) {
            Element element = (Element) elements.item(i);
            if (element.getAttribute("GyarAzonosito").equals(azonosito) ||
element.getAttribute("MegrendeloAzonosito").equals(azonosito)) {
                NodeList cimek = element.getElementsByTagName("Cime");
                if (cimek.getLength() > 0) {
                    Element cimElement = (Element) cimek.item(0);
                    String utca = cimElement.getElementsByTagName("UtcaHazszam").item(0).getTextContent();
                    String varos = cimElement.getElementsByTagName("Varos").item(0).getTextContent();
                    return utca + ", " + varos;
                }
            }
        }
        return "Cím nem található";
    }

}
```

.

2d) *adatírás* - készítsen egy DOM API programot, amely egy *XMLNeptunkod.xml* dokumentum tartalmát fa struktúra formában kiírja a *konzolra és egy XMLNeptunkod1.xml* fájlba. (kód – comment együtt) – fájlnév: DOMWriteNeptunkod.java

Az XML Filet beolvassuk és létrehozunk rá egy Document objektumot.
A „Gyarak”, „Dolgozok” stb. elemeket az Element objektumok segítségével lettek hozzá adva a gyökér elemhez.
A Kommenteket a „Comment” org.w3c.dom.Comment-nek a „createComment” segítségével tudtuk hozzá adni.

A Kód:

```
package hu.domparse.c7h5vb;

import org.w3c.dom.Comment;
```

```
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;

public class DomWriteC7h5vb {

public static void main(String[] args) {
try {
		// XML fájl betöltése és Document objektum létrehozása
		DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
		DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
		Document doc = dBuilder.parse(new File("XMLC7H5VB.xml"));
		doc.getDocumentElement().normalize();

		// Új XML dokumentum létrehozása
		Document newDoc = dBuilder.newDocument();

		// Gyökérelem (C7H5VB_Hell) hozzáadása az új dokumentumhoz
		Element rootElement = newDoc.createElement("C7H5VB_Hell");
		newDoc.appendChild(rootElement);

		Comment gyarakComment = newDoc.createComment("Gyárak");
		rootElement.appendChild(gyarakComment);

		// Gyarak elemek létrehozása és hozzáadása
		Element gyarakElement1 = newDoc.createElement("Gyarak");
		gyarakElement1.setAttribute("GyarAzonosito", "1");
		gyarakElement1.setAttribute("VallalatiAzonosito", "1");
		// Cime elem és alárendelt elemei
		Element cimeElement1 = newDoc.createElement("Cime");
		Element utcaHazszamElement1 = newDoc.createElement("UtcaHazszam");
		utcaHazszamElement1.setTextContent("Vas utca 7");
		Element irSzamElement1 = newDoc.createElement("IrSzam");
		irSzamElement1.setTextContent("3534");
```

```
Element varosElement1 = newDoc.createElement("Varos");
varosElement1.setTextContent("Miskolc");
cimeElement1.appendChild(utcaHazszamElement1);
cimeElement1.appendChild(irSzamElement1);
cimeElement1.appendChild(varosElement1);
// GyartasiRata és Kiadasok elemek
Element gyartasiRataElement1 = newDoc.createElement("GyartasiRata");
gyartasiRataElement1.setTextContent("10000");
Element kiadasokElement1 = newDoc.createElement("Kiadasok");
kiadasokElement1.setTextContent("10000000");
gyarakElement1.appendChild(cimeElement1);
gyarakElement1.appendChild(gyartasiRataElement1);
gyarakElement1.appendChild(kiadasokElement1);
rootElement.appendChild(gyarakElement1);

// Második Gyarak elem létrehozása és hozzáadása
Element gyarakElement2 = newDoc.createElement("Gyarak");
gyarakElement2.setAttribute("GyarAzonosito", "2");
gyarakElement2.setAttribute("VallalatiAzonosito", "2");

Element cimeElement2 = newDoc.createElement("Cime");
Element utcaHazszamElement2 = newDoc.createElement("UtcaHazszam");
utcaHazszamElement2.setTextContent("Gyár utca 1");
Element irSzamElement2 = newDoc.createElement("IrSzam");
irSzamElement2.setTextContent("3525");
Element varosElement2 = newDoc.createElement("Varos");
varosElement2.setTextContent("Miskolc");

cimeElement2.appendChild(utcaHazszamElement2);
cimeElement2.appendChild(irSzamElement2);
cimeElement2.appendChild(varosElement2);

Element gyartasiRataElement2 = newDoc.createElement("GyartasiRata");
gyartasiRataElement2.setTextContent("15000");
Element kiadasokElement2 = newDoc.createElement("Kiadasok");
kiadasokElement2.setTextContent("13000000");

gyarakElement2.appendChild(cimeElement2);
gyarakElement2.appendChild(gyartasiRataElement2);
gyarakElement2.appendChild(kiadasokElement2);
rootElement.appendChild(gyarakElement2);
```

```
// Harmadik Gyarak elem létrehozása és hozzáadása
Element gyarakElement3 = newDoc.createElement("Gyarak");
gyarakElement3.setAttribute("GyarAzonosito", "3");
gyarakElement3.setAttribute("VallalatiAzonosito", "3");

Element cimeElement3 = newDoc.createElement("Cime");
Element utcaHazszamElement3 = newDoc.createElement("UtcaHazszam");
utcaHazszamElement3.setTextContent("Hell utca 1");
Element irSzamElement3 = newDoc.createElement("IrSzam");
irSzamElement3.setTextContent("2092");
Element varosElement3 = newDoc.createElement("Varos");
varosElement3.setTextContent("Budakeszi");

cimeElement3.appendChild(utcaHazszamElement3);
cimeElement3.appendChild(irSzamElement3);
cimeElement3.appendChild(varosElement3);

Element gyartasiRataElement3 = newDoc.createElement("GyartasiRata");
gyartasiRataElement3.setTextContent("35000");
Element kiadasokElement3 = newDoc.createElement("Kiadasok");
kiadasokElement3.setTextContent("19500000");

gyarakElement3.appendChild(cimeElement3);
gyarakElement3.appendChild(gyartasiRataElement3);
gyarakElement3.appendChild(kiadasokElement3);
rootElement.appendChild(gyarakElement3);

Comment termekekComment = newDoc.createComment("Termékek");
rootElement.appendChild(termekekComment);
// Termekek elemek létrehozása és hozzáadása
Element termekekElement1 = newDoc.createElement("Termekek");
termekekElement1.setAttribute("VallalatiAzonosito", "1");
Element fajtajaElement1 = newDoc.createElement("Fajtaja");
Element xixoElement1 = newDoc.createElement("Xixo");
xixoElement1.setTextContent("1");
fajtajaElement1.appendChild(xixoElement1);
Element bevetelElement1 = newDoc.createElement("Bevetel");
bevetelElement1.setTextContent("50000000");
Element kiadasokTermekekElement1 = newDoc.createElement("Kiadasok");
Element marketingElement1 = newDoc.createElement("Marketing");
```

```
marketingElement1.setTextContent("1200000");
Element otletekElement1 = newDoc.createElement("Otletek");
otletekElement1.setTextContent("1000000");
Element tesztelesElement1 = newDoc.createElement("Teszteles");
tesztelesElement1.setTextContent("500000");
kiadasokTermekekElement1.appendChild(marketingElement1);
kiadasokTermekekElement1.appendChild(otletekElement1);
kiadasokTermekekElement1.appendChild(tesztelesElement1);
kiadasokTermekekElement1.appendChild(fajtajaElement1);
kiadasokTermekekElement1.appendChild(bevetelElement1);
termekekElement1.appendChild(kiadasokTermekekElement1);
rootElement.appendChild(termekekElement1);

Element termekekElement2 = newDoc.createElement("Termekek");
termekekElement2.setAttribute("VallalatiAzonosito", "2");
Element fajtajaElement2 = newDoc.createElement("Fajtaja");
Element hellElement2 = newDoc.createElement("Hell");
hellElement2.setTextContent("1");
fajtajaElement2.appendChild(hellElement2);
Element bevetelElement2 = newDoc.createElement("Bevetel");
bevetelElement2.setTextContent("80000000");
Element kiadasokTermekekElement2 = newDoc.createElement("Kiadasok");
Element marketingElement2 = newDoc.createElement("Marketing");
marketingElement2.setTextContent("3200000");
Element otletekElement2 = newDoc.createElement("Otletek");
otletekElement2.setTextContent("1200000");
Element tesztelesElement2 = newDoc.createElement("Teszteles");
tesztelesElement2.setTextContent("700000");
kiadasokTermekekElement2.appendChild(marketingElement2);
kiadasokTermekekElement2.appendChild(otletekElement2);
kiadasokTermekekElement2.appendChild(tesztelesElement2);
termekekElement2.appendChild(fajtajaElement2);
termekekElement2.appendChild(bevetelElement2);
termekekElement2.appendChild(kiadasokTermekekElement2);
rootElement.appendChild(termekekElement2);

// Termekek elemek létrehozása és hozzáadása a harmadik termékhez
Element termekekElement3 = newDoc.createElement("Termekek");
termekekElement3.setAttribute("VallalatiAzonosito", "3");
Element fajtajaElement3 = newDoc.createElement("Fajtaja");
Element egyebElement3 = newDoc.createElement("Egyeb");
```

```
egyebElement3.setTextContent("1");
fajtajaElement3.appendChild(egyebElement3);
Element bevetelElement3 = newDoc.createElement("Bevetel");
bevetelElement3.setTextContent("7000000");
Element kiadasokTermekekElement3 = newDoc.createElement("Kiadasok");
Element marketingElement3 = newDoc.createElement("Marketing");
marketingElement3.setTextContent("700000");
Element otletekElement3 = newDoc.createElement("Otletek");
otletekElement3.setTextContent("250000");
Element tesztelesElement3 = newDoc.createElement("Teszteles");
tesztelesElement3.setTextContent("150000");
kiadasokTermekekElement3.appendChild(marketingElement3);
kiadasokTermekekElement3.appendChild(otletekElement3);
kiadasokTermekekElement3.appendChild(tesztelesElement3);
termekekElement3.appendChild(fajtajaElement3);
termekekElement3.appendChild(bevetelElement3);
termekekElement3.appendChild(kiadasokTermekekElement3);
rootElement.appendChild(termekekElement3);

Comment dolgozokComment = newDoc.createComment("Dolgozók");
rootElement.appendChild(dolgozokComment);
// Dolgozok elemek létrehozása és hozzáadása
Element dolgozokElement1 = newDoc.createElement("Dolgozok");
dolgozokElement1.setAttribute("ID", "1");
dolgozokElement1.setAttribute("GyarAzonosito", "1");
Element lakcimElement1 = newDoc.createElement("Lakcim");
Element utcaHazszamElementD1 = newDoc.createElement("UtcaHazszam");
utcaHazszamElementD1.setTextContent("Kerek utca 7");
Element irSzamElementD1 = newDoc.createElement("IrSzam");
irSzamElementD1.setTextContent("3533");
Element varosElementD1 = newDoc.createElement("Varos");
varosElementD1.setTextContent("Miskolc");
lakcimElement1.appendChild(utcaHazszamElementD1);
lakcimElement1.appendChild(irSzamElementD1);
lakcimElement1.appendChild(varosElementD1);
Element beosztasElement1 = newDoc.createElement("Beosztas");
beosztasElement1.setTextContent("Gyártósoros");
Element nevElement1 = newDoc.createElement("Nev");
nevElement1.setTextContent("Nagy Ernő");
Element fizetesElement1 = newDoc.createElement("Fizetes");
fizetesElement1.setTextContent("250000");
```

```
Element telefonszamElement1 = newDoc.createElement("Telefonszam");
telefonszamElement1.setTextContent("06307687647");
dolgozokElement1.appendChild(lakcimElement1);
dolgozokElement1.appendChild(beosztasElement1);
dolgozokElement1.appendChild(nevElement1);
dolgozokElement1.appendChild(fizetesElement1);
dolgozokElement1.appendChild(telefonszamElement1);
rootElement.appendChild(dolgozokElement1);
// Dolgozok elemek létrehozása és hozzáadása a második dolgozóhoz
Element dolgozokElement2 = newDoc.createElement("Dolgozok");
dolgozokElement2.setAttribute("ID", "2");
dolgozokElement2.setAttribute("GyarAzonosito", "3");
Element lakcimElement2 = newDoc.createElement("Lakcim");
Element utcaHazszamElementD2 = newDoc.createElement("UtcaHazszam");
utcaHazszamElementD2.setTextContent("Kis utca 23");
Element irSzamElementD2 = newDoc.createElement("IrSzam");
irSzamElementD2.setTextContent("3523");
Element varosElementD2 = newDoc.createElement("Varos");
varosElementD2.setTextContent("Miskolc");
lakcimElement2.appendChild(utcaHazszamElementD2);
lakcimElement2.appendChild(irSzamElementD2);
lakcimElement2.appendChild(varosElementD2);
Element beosztasElement2 = newDoc.createElement("Beosztas");
beosztasElement2.setTextContent("Gyártósoros");
Element nevElement2 = newDoc.createElement("Nev");
nevElement2.setTextContent("Látó Lajos");
Element fizetesElement2 = newDoc.createElement("Fizetes");
fizetesElement2.setTextContent("245000");
Element telefonszamElement2 = newDoc.createElement("Telefonszam");
telefonszamElement2.setTextContent("06307661589");
dolgozokElement2.appendChild(lakcimElement2);
dolgozokElement2.appendChild(beosztasElement2);
dolgozokElement2.appendChild(nevElement2);
dolgozokElement2.appendChild(fizetesElement2);
dolgozokElement2.appendChild(telefonszamElement2);
rootElement.appendChild(dolgozokElement2);

// Dolgozok elemek létrehozása és hozzáadása a harmadik dolgozóhoz
Element dolgozokElement3 = newDoc.createElement("Dolgozok");
dolgozokElement3.setAttribute("ID", "3");
dolgozokElement3.setAttribute("GyarAzonosito", "2");
```

```
Element lakcimElement3 = newDoc.createElement("Lakcim");
Element utcaHazszamElementD3 = newDoc.createElement("UtcaHazszam");
utcaHazszamElementD3.setTextContent("Mester utca 2");
Element irSzamElementD3 = newDoc.createElement("IrSzam");
irSzamElementD3.setTextContent("2092");
Element varosElementD3 = newDoc.createElement("Varos");
varosElementD3.setTextContent("Budakeszi");
lakcimElement3.appendChild(utcaHazszamElementD3);
lakcimElement3.appendChild(irSzamElementD3);
lakcimElement3.appendChild(varosElementD3);
Element beosztasElement3 = newDoc.createElement("Beosztas");
beosztasElement3.setTextContent("Takarító");
Element nevElement3 = newDoc.createElement("Nev");
nevElement3.setTextContent("Hangos Ernő");
Element fizetesElement3 = newDoc.createElement("Fizetes");
fizetesElement3.setTextContent("220000");
Element telefonszamElement3 = newDoc.createElement("Telefonszam");
telefonszamElement3.setTextContent("062096758956");
dolgozokElement3.appendChild(lakcimElement3);
dolgozokElement3.appendChild(beosztasElement3);
dolgozokElement3.appendChild(nevElement3);
dolgozokElement3.appendChild(fizetesElement3);
dolgozokElement3.appendChild(telefonszamElement3);
rootElement.appendChild(dolgozokElement3);

Comment avalonComment = newDoc.createComment("Avalon Park");
rootElement.appendChild(avalonComment);
// AvalonPark elem létrehozása és hozzáadása
Element avalonParkElement = newDoc.createElement("AvalonPark");
avalonParkElement.setAttribute("Azonosito", "1");
avalonParkElement.setAttribute("VezID", "1");

// AvalonPark Kiadasok és Bevetel elemek
Element avalonParkKiadasok = newDoc.createElement("Kiadasok");
Element avalonParkMarketing = newDoc.createElement("Marketing");
avalonParkMarketing.setTextContent("1000000");
Element avalonParkKarbantartas = newDoc.createElement("Karbantartas");
avalonParkKarbantartas.setTextContent("2500000");
Element avalonParkMerchandiseok = newDoc.createElement("Merchandiseok");
avalonParkMerchandiseok.setTextContent("3300000");
Element avalonParkEgyeb = newDoc.createElement("Egyeb");
```

```
avalonParkEgyeb.setTextContent("3000000");
avalonParkKiadasok.appendChild(avalonParkMarketing);
avalonParkKiadasok.appendChild(avalonParkKarbantartas);
avalonParkKiadasok.appendChild(avalonParkMerchandiseok);
avalonParkKiadasok.appendChild(avalonParkEgyeb);

Element avalonParkBevetel = newDoc.createElement("Bevetel");
avalonParkBevetel.setTextContent("60000000");

Element avalonParkCime = newDoc.createElement("Cime");
avalonParkCime.setTextContent("Miskolc Iglói u. 15, 3519");

avalonParkElement.appendChild(avalonParkKiadasok);
avalonParkElement.appendChild(avalonParkBevetel);
avalonParkElement.appendChild(avalonParkCime);
rootElement.appendChild(avalonParkElement);

Comment vezetoComment = newDoc.createComment("Vezető");
rootElement.appendChild(vezetoComment);
// Vezeto elem létrehozása és hozzáadása
Element vezetoElement = newDoc.createElement("Vezeto");
vezetoElement.setAttribute("VezID", "1");

Element vezetoLakcim = newDoc.createElement("Lakcim");
Element vezetoUtcaHazszam = newDoc.createElement("UtcaHazszam");
vezetoUtcaHazszam.setTextContent("Leveles utca 12");
Element vezetoIrSzam = newDoc.createElement("IrSzam");
vezetoIrSzam.setTextContent("3520");
Element vezetoVaros = newDoc.createElement("Varos");
vezetoVaros.setTextContent("Miskolc");

vezetoLakcim.appendChild(vezetoUtcaHazszam);
vezetoLakcim.appendChild(vezetoIrSzam);
vezetoLakcim.appendChild(vezetoVaros);

Element vezetoNev = newDoc.createElement("Nev");
vezetoNev.setTextContent("Lázár Ervin");

vezetoElement.appendChild(vezetoLakcim);
vezetoElement.appendChild(vezetoNev);
rootElement.appendChild(vezetoElement);
```

```
Comment kiszallitasComment = newDoc.createComment("Kiszállítás");
rootElement.appendChild(kiszallitasComment);
// Kiszallitas elemek létrehozása és hozzáadása
Element kiszallitasElement1 = newDoc.createElement("Kiszallitas");
kiszallitasElement1.setAttribute("GyarAzonosito", "1");
kiszallitasElement1.setAttribute("MegrendeloAzonosito", "1");
Element kiszallitasMennyiseg1 = newDoc.createElement("Mennyiseg");
kiszallitasMennyiseg1.setTextContent("2500");
kiszallitasElement1.appendChild(kiszallitasMennyiseg1);
rootElement.appendChild(kiszallitasElement1);

Element kiszallitasElement2 = newDoc.createElement("Kiszallitas");
kiszallitasElement2.setAttribute("GyarAzonosito", "2");
kiszallitasElement2.setAttribute("MegrendeloAzonosito", "2");
Element kiszallitasMennyiseg2 = newDoc.createElement("Mennyiseg");
kiszallitasMennyiseg2.setTextContent("5000");
kiszallitasElement2.appendChild(kiszallitasMennyiseg2);
rootElement.appendChild(kiszallitasElement2);

Element kiszallitasElement3 = newDoc.createElement("Kiszallitas");
kiszallitasElement3.setAttribute("GyarAzonosito", "3");
kiszallitasElement3.setAttribute("MegrendeloAzonosito", "3");
Element kiszallitasMennyiseg3 = newDoc.createElement("Mennyiseg");
kiszallitasMennyiseg3.setTextContent("1200");
kiszallitasElement3.appendChild(kiszallitasMennyiseg3);
rootElement.appendChild(kiszallitasElement3);

Comment megrendelokComment = newDoc.createComment("Megrendelők");
rootElement.appendChild(megrendelokComment);
// Megrendelok elemek létrehozása és hozzáadása
Element megrendelokElement1 = newDoc.createElement("Megrendelok");
megrendelokElement1.setAttribute("MegrendeloAzonosito", "1");
Element megrendelokElerhetoseg1 = newDoc.createElement("Elerhetoseg");
Element megrendelokEmail1 = newDoc.createElement("Email");
megrendelokEmail1.setTextContent("ugyfelszolgalat@coop.hu");
Element megrendelokTelefonSzam1 = newDoc.createElement("TelefonSzam");
megrendelokTelefonSzam1.setTextContent("06307658799");
megrendelokElerhetoseg1.appendChild(megrendelokEmail1);
megrendelokElerhetoseg1.appendChild(megrendelokTelefonSzam1);
```

```
Element megrendelokCime1 = newDoc.createElement("Cime");
Element megrendelokUtcaHazszam1 = newDoc.createElement("UtcaHazszam");
megrendelokUtcaHazszam1.setTextContent("Corvin utca 7");
Element megrendelokIrSzam1 = newDoc.createElement("IrSzam");
megrendelokIrSzam1.setTextContent("3525");
Element megrendelokVaros1 = newDoc.createElement("Varos");
megrendelokVaros1.setTextContent("Miskolc");

megrendelokCime1.appendChild(megrendelokUtcaHazszam1);
megrendelokCime1.appendChild(megrendelokIrSzam1);
megrendelokCime1.appendChild(megrendelokVaros1);

megrendelokElement1.appendChild(megrendelokElerhetoseg1);
megrendelokElement1.appendChild(megrendelokCime1);
rootElement.appendChild(megrendelokElement1);

// További Megrendelok elemek létrehozása és hozzáadása
Element megrendelokElement2 = newDoc.createElement("Megrendelok");
megrendelokElement2.setAttribute("MegrendeloAzonosito", "2");

Element megrendelokElerhetoseg2 = newDoc.createElement("Elerhetoseg");
Element megrendelokEmail2 = newDoc.createElement("Email");
megrendelokEmail2.setTextContent("beszerzes@spar.hu");
Element megrendelokTelefonSzam2 = newDoc.createElement("TelefonSzam");
megrendelokTelefonSzam2.setTextContent("06407658788");
megrendelokElerhetoseg2.appendChild(megrendelokEmail2);
megrendelokElerhetoseg2.appendChild(megrendelokTelefonSzam2);

Element megrendelokCime2 = newDoc.createElement("Cime");
Element megrendelokUtcaHazszam2 = newDoc.createElement("UtcaHazszam");
megrendelokUtcaHazszam2.setTextContent("Magas út 26");
Element megrendelokIrSzam2 = newDoc.createElement("IrSzam");
megrendelokIrSzam2.setTextContent("3533");
Element megrendelokVaros2 = newDoc.createElement("Varos");
megrendelokVaros2.setTextContent("Miskolc");

megrendelokCime2.appendChild(megrendelokUtcaHazszam2);
megrendelokCime2.appendChild(megrendelokIrSzam2);
megrendelokCime2.appendChild(megrendelokVaros2);

megrendelokElement2.appendChild(megrendelokElerhetoseg2);
```

```
megrendelokElement2.appendChild(megrendelokCime2);
rootElement.appendChild(megrendelokElement2);

Element megrendelokElement3 = newDoc.createElement("Megrendelok");
megrendelokElement3.setAttribute("MegrendeloAzonosito", "3");

Element megrendelokElerhetoseg3 = newDoc.createElement("Elerhetoseg");
Element megrendelokEmail3 = newDoc.createElement("Email");
megrendelokEmail3.setTextContent("megrendelo@arena.hu");
Element megrendelokTelefonSzam3 = newDoc.createElement("TelefonSzam");
megrendelokTelefonSzam3.setTextContent("06105758595");
megrendelokElerhetoseg3.appendChild(megrendelokEmail3);
megrendelokElerhetoseg3.appendChild(megrendelokTelefonSzam3);

Element megrendelokCime3 = newDoc.createElement("Cime");
Element megrendelokUtcaHazszam3 = newDoc.createElement("UtcaHazszam");
megrendelokUtcaHazszam3.setTextContent("Kerepesi út 9");
Element megrendelokIrSzam3 = newDoc.createElement("IrSzam");
megrendelokIrSzam3.setTextContent("1087");
Element megrendelokVaros3 = newDoc.createElement("Varos");
megrendelokVaros3.setTextContent("Budapest");

megrendelokCime3.appendChild(megrendelokUtcaHazszam3);
megrendelokCime3.appendChild(megrendelokIrSzam3);
megrendelokCime3.appendChild(megrendelokVaros3);

megrendelokElement3.appendChild(megrendelokElerhetoseg3);
megrendelokElement3.appendChild(megrendelokCime3);
rootElement.appendChild(megrendelokElement3);

// Az új XML fájl mentése
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
transformer.setOutputProperty(OutputKeys.INDENT, "yes");
DOMSource source = new DOMSource(newDoc);
StreamResult result = new StreamResult(new File("XMLC7H5VB1.xml"));
transformer.transform(source, result);

System.out.println("Az új XML fájl létrehozása sikeres!");
System.out.println("");
// Konzolra írás
```

```
            StreamResult consoleResult = new StreamResult(System.out);
            transformer.transform(source, consoleResult);



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

}
}
```

Dr. Bednarik László

tárgyjegyző