

JEGYZŐKÖNYV

Webkönyvtárak

Webes adatkezelő környezetek

Csomag tároló/szállító rendszer

Készítette: **Takács Ádám**

Neptunkód: **X16WW1**

Dátum: 2025/05/13

Tartalom

Bevezetés	3
1. feladat.....	3
1.1 Az adatbázis ER modell tervezése	3
1.2 Az adatbázis konvertálása XDM modellre	4
1.3 Az XDM modell alapján XML dokumentum készítése	5
1.4 Az XML dokumentum alapján XMLSchema készítése - saját típusok, key, keyref, speciális elemek.....	7
2. feladat	7
2.1 Adatolvasás.....	10
2.2 Adatmódosítás	14
2.3 Adatlekérdezés	16

Bevezetés

A feladat leírása: Egy olyan XML modell és a hozzá tartozó ER modell, XDM modell és bizonyos DOM utasítások elkészítése, amely segítségével szamon lehet tartani a rendszert használó cégek adatait, a cégekhez tartozó raktár épületeket, a polcok azonosítóját, a különböző termékek azonosítóját és egyéb tulajdonságait, ezen raktárkaért való felelősök adatait és elérhetőségeiket, továbbá a rendelések és azon rendelések kiszállításáért felelős futárszolgálatok adatait. Az ER és az XDM modellekre azért van szükség, hogy a kiépítendő rendszer könnyen megtervezhető és átlátható legyen. A DOM utasítások feladata az, hogy a rendszerben lévő adatok könnyen elérhetőek és ezen elemek adatai könnyen frissíthetők legyenek.

1. feladat

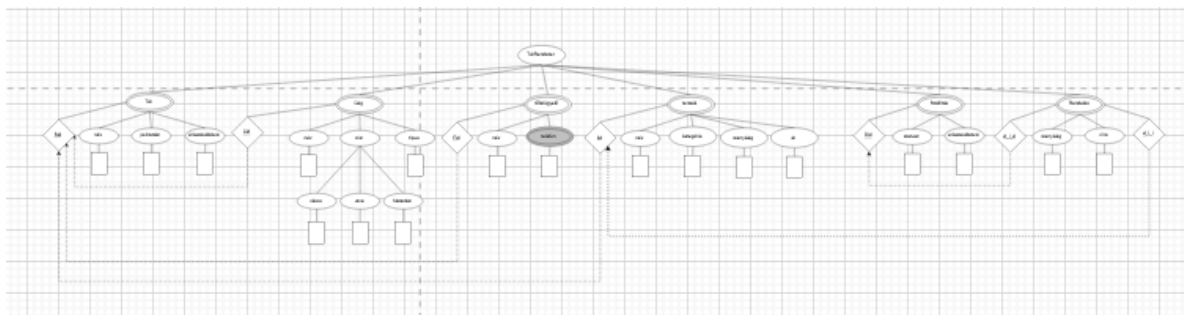
1.1 Az adatbázis ER modell tervezése (Legyen legalább 5 egyed, többféle kapcsolat (1:1; 1:N; M:N), tulajdonságok - normál, kulcs, összetett, többértékű. (Csak szerkesztő programmal rajzolt ábra megfelelő, szabványos szimbólumok használata!))

Az ER modell tervezését a „Tár” blokk elhelyezésével kezdtem az ábrán, azért a „Tár” lett a középső elem ezen az ábrán mert a kapcsolatok többsége ezen elemhez fognak visszavezetni. Következő lépésként elhelyeztem a „Tár” elem tulajdonságait az ábrán, ezek a „Név” „polcszám” „érkezésidátum” és a „Sid” mezők. Ezek azért szolgálnak mert a „Tár” elem feladata a különböző raktár épületek és ezen épületek polcainak számontartása. Ez után elkészítettem a „Főfelügyelő” elemhez szükséges mezőket. Ezen mezők a

„Főfelügyelő”, „Név”, „Fid” és a „Telefon”. Mivel a „Telefon” az egy olyan mező, amelynek egy elemhez több érték is rendelhető ezért ezt külön fel kellett tüntetnem az ábrán. Ez után összekötöttem a kettő mezőt a „s_f” kapcsolat segítségével. A „Cég” elem feldolgozása következik, amely ezért különleges a többi elemtől mivel rendelkezik a „Cím” elemmel, amely 3 további tulajdonságból áll ezért ezt a 3 tulajdonságot a „Cím” elemhez kellett kötnöm, amit azután a „Cég” elemhez kötöttem. Miután elkészült a „Cég” utána a „Tár” elemhez kötöttem a „c_s” kapcsolat segítségével. Magában a „Termék” elem nem különleges, mint a többi ezért egyszerűen a tulajdonságok elhelyezése után a „s_i” kapcsolat segítségével ismételt a „Tár” hoz kötöttem. A „Termék” elem akkor lesz különleges amikor a „Szállítás” elemmel együtt tekintjük hiszen ezen két elem közötti kapcsolat egy olyan kapcsolat amely 2 saját tulajdonsággal rendelkezik, ezen kapcsolat neve a modellben a „Rendelés”.

1.2 Az adatbázis konvertálása XDM modellre (Csak szerkesztő programmal rajzolt ábra megfelelő, szabványos szimbólumok használata)

Már egy készen lévő ER modell rendelkezésre állásával könnyű dolgunk van elkészíteni a XDM modellt. Azokat a részeket, amelyek azonosak az ER modellel itt nem fogom részletezni. Ami az ER modellben nem okozott akkora problémát, mint itt az XDM modellben az a kapcsolatok jelzése az ábrán. Mivel a kapcsolatok többsége a Tár Sid elemébe tart ezért úgy döntöttem, hogy a Tár elem lesz az ábra bal szélén a kapcsolatok egyszerű átláthatósága érdekében. Ez után jobbra elhelyeztem a Cég és a Főfelügyelő elemeket hiszen ezek azok az elemek, amelyek a Sid-be kapcsolódnak és más nem kapcsolódik hozzájuk. A Sid-be kapcsolódó elemek jobb szélére helyeztem a Termék és a Szállítás elemeket hiszen majd ide fog kapcsolódni a Rendelés elem. Az ábra jobb szélére helyeztem a Rendelés elemet, hogy az innen induló kapcsolatok ne zavarják a Sid-be menő kapcsolatokat.



A modell a hossza miatt nem jól látható a képen. Egy jobb minőségű ábra megtalálható a jegyzetmellet megtalálható fájlok között.

1.3 Az XDM modell alapján XML dokumentum készítése: (Ide kerül az XML kódja!)

A XDM modell sikeres elkészítése után egyszerű dolgom volt a XML dokumentum előállításával hiszen már minden fontos struktúra megvolt tervezve és csak bekellet gépelnem őket és feltöltenem adattal. Az adattal feltöltés során arra odakellet figyelnem hogy a többszörös előfordulású elemekből minimum 3-at rakjak az adatbázisba.

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

  <!-- Ceg-->
  <Ceg Cid="C1">
    <nev>Iroda es Furok kft.</nev>
    <cim>
      <varos>Tesztujvaros</varos>
      <utca>Teszt utca</utca>
      <hazszam>3</hazszam>
    </cim>
    <tipus>Irodai kellekek es banyasz eszkozok</tipus>
  </Ceg>

  <Ceg Cid="C2">
    <nev>Levego tarolasi kft.</nev>
    <cim>
      <varos>Tesztujvaros</varos>
      <utca>Teszt utca</utca>
      <hazszam>1</hazszam>
    </cim>
    <tipus>levego tarolas</tipus>
  </Ceg>

  <!-- Tarolo epuletek-->
  <Tar Sid="S1" c_s="C1">
    <nev>Fö Raktar A</nev>
    <polcszam>23B</polcszam>
    <erkezesidatum>2025-05-10</erkezesidatum>
  </Tar>

  <Tar Sid="S2" c_s="C1">
    <nev>Masodlagos Raktar B</nev>
    <polcszam>15A</polcszam>
    <erkezesidatum>2025-04-22</erkezesidatum>
  </Tar>

  <Tar Sid="S3" c_s="C1">
    <nev>Raktar C</nev>
```

```
    <polcszam>3C</polcszam>
    <erkezesidatum>2025-03-15</erkezesidatum>
</Tar>
```

```
<!--Föfelügyelők-->
<Föfelügyelő Fid="F1" s_f="S1">
    <nev>Kis Ferenc</nev>
    <telefon>??</telefon>
</Föfelügyelő>
```

```
<Föfelügyelő Fid="F2" s_f="S2">
    <nev>Nagy Ferenc</nev>
    <telefon>??</telefon>
    <telefon>????</telefon>
    <telefon>????</telefon>
</Föfelügyelő>
```

```
<!--Termekek-->
<Termek Iid="I1" s_i="S1">
    <nev>Elektromos furo</nev>
    <kategoria>Eszközök</kategoria>
    <menyiseg>10</menyiseg>
    <ar>299</ar>
</Termek>
```

```
<Termek Iid="I2" s_i="S2">
    <nev>Nyomtato papir</nev>
    <kategoria>Irodai kellek</kategoria>
    <menyiseg>500</menyiseg>
    <ar>1.50</ar>
</Termek>
```

```
<Termek Iid="I3" s_i="S3">
    <nev>LED Feny</nev>
    <kategoria>vilagitas</kategoria>
    <menyiseg>200</menyiseg>
    <ar>3</ar>
</Termek>
```

```
<!-- Rendelesek-->
<Rendeles d_i_i="I2" d_i_d="D3">
    <menyiseg>250</menyiseg>
    <cim>??</cim>
</Rendeles>
```

```
<Rendeles d_i_i="I3" d_i_d="D2">
    <menyiseg>100</menyiseg>
```

```

        <cim>??</cim>
    </Rendeles>

    <Rendeles d_i_i="I1" d_i_d="D1">
        <menyiseg>2</menyiseg>
        <cim>??</cim>
    </Rendeles>

    <!-- Szallitas-->
    <Szallitas Did="D1">
        <statusz>kiszallitva</statusz>
        <erkezesidatum>2025-05-13</erkezesidatum>
    </Szallitas>

    <Szallitas Did="D2">
        <statusz>szallitas allat</statusz>
        <erkezesidatum>2025-05-15</erkezesidatum>
    </Szallitas>

    <Szallitas Did="D3">
        <statusz>felvetelre var</statusz>
        <erkezesidatum>2025-05-20</erkezesidatum>
    </Szallitas>

</TarRendszer>

```

1.4 Az XML dokumentum alapján XMLSchema készítése - saját típusok, key, keyref, speciális elemek. (Ide kerül az XML Schema kódja!)

A készlévő XML dokumentummal egyszerűen előlehetett állítani az XMLSchema-t is. Az elemek nagy részét az órán tanult módszerrel valósítottam meg. Saját típusnak a dátum mezőket választottam és a kulcsok pedig a különböző id mezők lettek.

```

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

    <xs:element name="TárRendszer">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ceg" type="cegTipus" maxOccurs="unbounded"
            />
                <xs:element name="tar" type="tarTipus" maxOccurs="unbounded"
            />
            />
    </xs:element>
</xs:schema>

```

```

        <xs:element name="fofelugyelo" type="felugyeloTipus"
maxOccurs="unbounded" />
        <xs:element name="termek" type="termekTipus"
maxOccurs="unbounded" />
        <xs:element name="rendeles" type="rendelesTipus"
maxOccurs="unbounded" />
        <xs:element name="szallitas" type="szallitasTipus"
maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<!-- Kulcsok -->
<xs:key name="ceg_kulcs">
    <xs:selector xpath="Cég" />
    <xs:field xpath="@Cid" />
</xs:key>

<xs:key name="tar_kulcs">
    <xs:selector xpath="Tár" />
    <xs:field xpath="@Sid" />
</xs:key>

<xs:key name="termek_kulcs">
    <xs:selector xpath="Termék" />
    <xs:field xpath="@Iid" />
</xs:key>

<xs:key name="szallitas_kulcs">
    <xs:selector xpath="Szállítás" />
    <xs:field xpath="@Did" />
</xs:key>

<!-- Idegen kulcs -->
<xs:keyref refer="ceg_kulcs" name="ceg_tar_idegen_kulcs">
    <xs:selector xpath="Tár" />
    <xs:field xpath="@c_s" />
</xs:keyref>

<xs:keyref refer="fofelugyelo_kulcs"
name="tar_fofelugyelo_idegen_kulcs">
    <xs:selector xpath="Főfelügyelő" />
    <xs:field xpath="@s_f" />
</xs:keyref>

<xs:keyref refer="termek_kulcs" name="tar_termek_idegen_kulcs">
    <xs:selector xpath="Termék" />
    <xs:field xpath="@s_i" />
</xs:keyref>

```



```

<xs:keyref refer="termek_kulcs" name="termek_idegen_kulcs">
  <xs:selector xpath="Rendelés" />
  <xs:field xpath="@d_i_i" />
</xs:keyref>

<xs:keyref refer="szallitas_kulcs" name="szallitas_idegen_kulcs">
  <xs:selector xpath="Rendelés" />
  <xs:field xpath="@d_i_i" />
</xs:keyref>

<!-- 1:1 Kapcsolat -->
<xs:unique name="fofelugyelo_kulcs">
  <xs:selector xpath="Főfelügyelő" />
  <xs:field xpath="@Fid" />
</xs:unique>
</xs:element>

<!-- Komplex típusok -->
<xs:complexType name="cegTípus">
  <xs:sequence>
    <xs:element name="név" type="xs:string" />
    <xs:element name="cím" minOccurs="0" maxOccurs="5">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="város" type="xs:string" />
          <xs:element name="utca" type="xs:string" />
          <xs:element name="házszám" type="xs:integer" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="típus" type="xs:string" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="tarTípus">
  <xs:sequence>
    <xs:element name="név" type="xs:string" />
    <xs:element name="polcszám" type="xs:string" />
    <xs:element name="érkezésidátum" type="xs:date" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="felugyeloTípus">
  <xs:sequence>
    <xs:element name="név" type="xs:string" />
    <xs:element name="telefon" type="xs:string" />
  </xs:sequence>

```

```

</xs:complexType>

<xs:complexType name="termekTipus">
  <xs:sequence>
    <xs:element name="név" type="xs:string" />
    <xs:element name="kategória" type="xs:string" />
    <xs:element name="menyiség" type="xs:integer" />
    <xs:element name="ár" type="xs:integer" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="rendelesTipus">
  <xs:sequence>
    <xs:element name="menyiség" type="xs:integer" />
    <xs:element name="cím" type="xs:string" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="szallitasTipus">
  <xs:sequence>
    <xs:element name="statusz" type="xs:string" />
    <xs:element name="erkezesidatum" type="xs:date" />
  </xs:sequence>
</xs:complexType>

</xs:schema>

```

2. feladat

A feladat egy DOM program készítése Java nyelven az XML dokumentum - *XMLNeptunkod.xml* – adatainak adminisztrálása alapján: (ide kerül a kód - comment együtt)

Project name: DOMParseNeptunkod

Package: hu.domparse.neptunkod

Class names: (DomReadNeptunkod, DomModifyNeptunkod, DomQueryNeptunkod, DOMWriteNeptunkod)

2.1 Adatolvasás (kód – comment együtt) – fájlnev: *DOMReadNeptunkod.java*

Ezt a feladatot röviden úgy csináltam meg, hogy ciklusok segítségével bejártam az XML dokumentum elemeit és ezeket a konzolra kiírtam. A különböző főbb elemeket pedig egy speciális kiírással választottam el.

```

package hu.domparse.neptunkod;

// Szükséges Java osztályok importálása fájlkezeléshez és DOM feldolgozáshoz

```

```

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

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

public class DOMReadX16WW1 {

    public static void main(String[] args) {
        try {
            // XML fájl betöltése
            File xmlFile = new File("XMLX16WW1.xml"); // Az XML fájl
helyének megadása
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance(); // DOM parser gyár példányosítása
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder(); //
Dokumentum építő létrehozása

            Document doc = dBuilder.parse(xmlFile); // XML dokumentum
betöltése
            doc.getDocumentElement().normalize(); // Dokumentum
normalizálása (pl. whitespace-ek eltávolítása)

            // A gyökérelem (root element) nevének kiírása
            System.out.println("Gyökérelem: " +
doc.getDocumentElement().getNodeName());

            // ----- Cégek bejárása -----
            NodeList cegList = doc.getElementsByTagName("Ceg"); // 'Ceg'
elemek lekérése
            System.out.println("\n-- Cégek --");
            for (int i = 0; i < cegList.getLength(); i++) {
                Element ceg = (Element) cegList.item(i); // Egy adott 'Ceg'
elem kiválasztása
                // A cég attribútumainak és al-elemeinek kiírása
                System.out.println("Cid: " + ceg.getAttribute("Cid"));
                System.out.println("Név: " +
ceg.getElementsByTagName("nev").item(0).getTextContent());
                Element cim = (Element)
ceg.getElementsByTagName("cim").item(0); // 'cim' elem lekérése
                System.out.println("Cím: " +
cim.getElementsByTagName("varos").item(0).getTextContent() + ", " +
                    cim.getElementsByTagName("utca").item(0).getTextCont
ent() + " " +

```

```

        cim.getElementsByTagName("hazszam").item(0).getTextC
ontent());
        System.out.println("Típus: " +
ceg.getElementsByTagName("tipus").item(0).getTextContent());
        System.out.println();
    }

    // ----- Tárolók bejárása -----
    NodeList tarList = doc.getElementsByTagName("Tar"); // 'Tar'
elemek lekérése
    System.out.println("-- Tárolók --");
    for (int i = 0; i < tarList.getLength(); i++) {
        Element tar = (Element) tarList.item(i);
        // Tároló attribútumainak és al-elemeinek kiírása
        System.out.println("Sid: " + tar.getAttribute("Sid") + ",
Céges azonosító: " + tar.getAttribute("c_s"));
        System.out.println("Név: " +
tar.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Polcszám: " +
tar.getElementsByTagName("polcszam").item(0).getTextContent());
        System.out.println("Érkezési dátum: " +
tar.getElementsByTagName("erkezesidatum").item(0).getTextContent());
        System.out.println();
    }

    // ----- Főfelügyelők bejárása -----
    NodeList felugyeloList =
doc.getElementsByTagName("Főfelügyelő");
    System.out.println("-- Főfelügyelők --");
    for (int i = 0; i < felugyeloList.getLength(); i++) {
        Element fel = (Element) felugyeloList.item(i);
        // Főfelügyelő attribútumainak és al-elemeinek kiírása
        System.out.println("Fid: " + fel.getAttribute("Fid") + ",
Raktár: " + fel.getAttribute("s_f"));
        System.out.println("Név: " +
fel.getElementsByTagName("nev").item(0).getTextContent());
        // Több telefonszám bejárása
        NodeList telefonok = fel.getElementsByTagName("telefon");
        for (int j = 0; j < telefonok.getLength(); j++) {
            System.out.println("Telefon: " +
telefonok.item(j).getTextContent());
        }
        System.out.println();
    }

    // ----- Termékek bejárása -----
    NodeList termékList = doc.getElementsByTagName("Termek");
    System.out.println("-- Termékek --");

```

```

        for (int i = 0; i < termékList.getLength(); i++) {
            Element termék = (Element) termékList.item(i);
            // Termék attribútumainak és al-elemeinek kiírása
            System.out.println("Id: " + termék.getAttribute("Id") + ",
Raktár: " + termék.getAttribute("s_i"));
            System.out.println("Név: " +
termék.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("Kategória: " +
termék.getElementsByTagName("kategoria").item(0).getTextContent());
            System.out.println("Mennyiség: " +
termék.getElementsByTagName("menyiseg").item(0).getTextContent());
            System.out.println("Ár: " +
termék.getElementsByTagName("ar").item(0).getTextContent());
            System.out.println();
        }

        // ----- Rendelések bejárása -----
        NodeList rendelesList = doc.getElementsByTagName("Rendeles");
        System.out.println("-- Rendelések --");
        for (int i = 0; i < rendelesList.getLength(); i++) {
            Element rendeles = (Element) rendelesList.item(i);
            // Rendeléshez kapcsolódó attribútumok és al-elemek kiírása
            System.out.println("Termék ID: " +
rendeles.getAttribute("d_i_i") + ", Szállítás ID: " +
rendeles.getAttribute("d_i_d"));
            System.out.println("Mennyiség: " +
rendeles.getElementsByTagName("menyiseg").item(0).getTextContent());
            System.out.println("Cím: " +
rendeles.getElementsByTagName("cim").item(0).getTextContent());
            System.out.println();
        }

        // ----- Szállítások bejárása -----
        NodeList szallitasList = doc.getElementsByTagName("Szallitas");
        System.out.println("-- Szállítások --");
        for (int i = 0; i < szallitasList.getLength(); i++) {
            Element szallitas = (Element) szallitasList.item(i);
            // Szállítás részleteinek kiírása
            System.out.println("Did: " + szallitas.getAttribute("Did"));
            System.out.println("Státusz: " +
szallitas.getElementsByTagName("statusz").item(0).getTextContent());
            System.out.println("Érkezési dátum: " +
szallitas.getElementsByTagName("erkezesidatum").item(0).getTextContent());
            System.out.println();
        }

    } catch (Exception e) {
        // Hibakezelés: a kivétel stack trace-ének kiírása

```

```

        e.printStackTrace();
    }
}
}

```

2.2 Adatmódosítás (kód – comment együtt) – fájlnev: *DOMModifyNeptunkod.java*

Ezen a résznél először egy ciklussal körbejártam a dokumentumot, minden lépésnél egy „if” parancs segítségével megnéztem azt, hogy a megfelelő mezőnél tartok-e. Miután megtaláltam a mezőt után egy újabb ciklussal végig mentem az elemein a módosítandó elem keresése céljából, miután megtaláltam az elemet kicseréltem az új adattal és kiírtam a konzolra.

```

package hu.domparse.neptunkod;

import java.io.File;

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;

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

/**
 *
 * @author Takács Ádám (X16WW1)
 */
public class DOMModifyX16WW1 {

    public static void main(String[] args) {
        try {
            // XML fájl betöltése
            File xmlFile = new File("XMLX16WW1.xml");

            // DOM parser létrehozása a fájl feldolgozásához
            DocumentBuilderFactory facotry =
DocumentBuilderFactory.newInstance(); // Gyár létrehozása

```

```

        DocumentBuilder dBuilder = facotry.newDocumentBuilder(); //
Dokumentum építő létrehozása

        Document doc = dBuilder.parse(xmlFile); // Az XML fájl
beolvasása és DOM dokumentummá alakítása

        doc.getDocumentElement().normalize(); // A dokumentum
normalizálása (pl. whitespace-ek kezelése)

        // 'Tar' elemek lekérése (tárolók)
        NodeList nList = doc.getElementsByTagName("Tar");

        // Tárolók bejárása
        for (int i = 0; i < nList.getLength(); i++){
            Node tar = doc.getElementsByTagName("Tar").item(i); // Egy
adott 'Tar' csomópont
            NodeList list = tar.getChildNodes(); // A tároló
alcsomópontjai (pl. név, polcszám, stb.)

            for (int temp = 0; temp < list.getLength(); temp++){
                Node node = list.item(temp);
                if (node.getNodeType() == Node.ELEMENT_NODE ) { // Csak
elem típusú csomópontokkal dolgozunk
                    Element eElement = (Element) node;

                    // Polcszám "3C" értékének cseréje "3D"-re
                    if ("polcszám".equals(eElement.getNodeName())){
                        if ("3C".equals(eElement.getTextContent())){
                            eElement.setTextContent("3D");
                        }
                    }

                    // Tároló azonosító "S1" cseréje "S4"-re
                    if ("Sid".equals(eElement.getNodeName())){
                        if ("S1".equals(eElement.getTextContent())){
                            eElement.setTextContent("S4");
                        }
                    }

                    // Tároló nevének módosítása "Masodlagos Raktar B" →
"Masodlagos Raktar C"
                    if ("nev".equals(eElement.getNodeName())){
                        if ("Masodlagos Raktar
B".equals(eElement.getTextContent())){
                            eElement.setTextContent("Masodlagos Raktar
C");
                        }
                    }
                }
            }
        }

```

```

        // Cég azonosító módosítása "C1" → "C2"
        if ("c_s".equals(eElement.getNodeName())){
            if ("C1".equals(eElement.getTextContent())){
                eElement.setTextContent("C2");
            }
        }

        // Érkezési dátum módosítása "2025-03-15" → "2025-
04-15"

        if ("erkezesidatum".equals(eElement.getNodeName())){
            if ("2025-03-
15".equals(eElement.getTextContent())){
                eElement.setTextContent("2025-04-15");
            }
        }
    }
}

// A módosított dokumentum mentése és kiírása konzolra
TransformerFactory transformerFactory =
TransformerFactory.newInstance(); // Transformer gyár
Transformer transformer = transformerFactory.newTransformer();
// Transformer példány
DOMSource source = new DOMSource(doc); // DOM forrás objektum
létrehozása

System.out.println("----Modified File----");
StreamResult consoResult = new StreamResult(System.out); //
Eredmény kiírása konzolra
transformer.transform(source, consoResult); // A DOM dokumentum
átalakítása és kiírása

    } catch (Exception e) {
        // Hiba esetén a kivétel kiírása
        e.printStackTrace();
    }
}
}

```

2.3 Adatlekérdezés (kód – comment együtt) – fájlnev: *DOMQueryNeptunkod.java*

A feladat ezen részét ez előző kettő részfeladatban használt módszerekkel valósítottam meg. Először kiszűrtem a kereset elemeket és utána ezeket kiírtam a konzolra.


```

package hu.domparse.neptunkod;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

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

public class DOMQueryX16WW1 {

    public static void main(String[] args) {
        try {
            // XML fájl betöltése
            File inputFile = new File("XMLX16WW1.xml");

            // DOM parser inicializálása
            DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();

            // Dokumentum objektum létrehozása a fájlból
            Document doc = builder.parse(inputFile);
            doc.getDocumentElement().normalize();

            // 1. LEKÉRDEZÉS: Az összes cég neve
            System.out.println("1. Összes cég neve:");
            NodeList cegList = doc.getElementsByTagName("Ceg");
            for (int i = 0; i < cegList.getLength(); i++) {
                Element ceg = (Element) cegList.item(i);
                // A <nev> elem tartalmának kiírása
                System.out.println(" - " +
ceg.getElementsByTagName("nev").item(0).getTextContent());
            }

            // 2. LEKÉRDEZÉS: Raktárak nevei és polcszámai
            System.out.println("\n2. Raktárak nevei és polcszámai:");
            NodeList tarList = doc.getElementsByTagName("Tar");
            for (int i = 0; i < tarList.getLength(); i++) {
                Element tar = (Element) tarList.item(i);
                String nev =
tar.getElementsByTagName("nev").item(0).getTextContent();
                String polc =
tar.getElementsByTagName("polcszam").item(0).getTextContent();
                System.out.println(" - " + nev + " (Polcszám: " + polc +
")");
            }
        }
    }
}

```

```

    }

    // 3. LEKÉRDEZÉS: Termékek nevei, árai és raktár-azonosítójuk
    System.out.println("\n3. Termékek nevei, árai és raktár
azonosítójuk:");
    NodeList termékList = doc.getElementsByTagName("Termek");
    for (int i = 0; i < termékList.getLength(); i++) {
        Element termék = (Element) termékList.item(i);
        String nev =
termék.getElementsByTagName("nev").item(0).getTextContent();
        String ar =
termék.getElementsByTagName("ar").item(0).getTextContent();
        String sid = termék.getAttribute("s_i"); // raktár ID
        System.out.println(" - " + nev + ", Ár: " + ar + ", Raktár
ID: " + sid);
    }

    // 4. LEKÉRDEZÉS: Rendelések termékei és mennyisége
    System.out.println("\n4. Rendelések termékei és mennyisége:");
    NodeList rendelésList = doc.getElementsByTagName("Rendeles");
    for (int i = 0; i < rendelésList.getLength(); i++) {
        Element rendelés = (Element) rendelésList.item(i);
        String iid = rendelés.getAttribute("d_i_i"); // Termék ID
        String meny =
rendelés.getElementsByTagName("mennyiség").item(0).getTextContent();

        // A hozzá tartozó termék nevének megkeresése
        for (int j = 0; j < termékList.getLength(); j++) {
            Element termék = (Element) termékList.item(j);
            if (termék.getAttribute("Iid").equals(iid)) {
                String nev =
termék.getElementsByTagName("nev").item(0).getTextContent();
                System.out.println(" - " + nev + ", Mennyiség: " +
meny);

                break; // ha megtaláltuk, nem kell tovább keresni
            }
        }
    }

    // 5. LEKÉRDEZÉS: Szállítások státuszai és érkezési dátumai
    System.out.println("\n5. Szállítások státuszai és érkezési
dátumai:");
    NodeList szállitList = doc.getElementsByTagName("Szallitas");
    for (int i = 0; i < szállitList.getLength(); i++) {
        Element szállit = (Element) szállitList.item(i);
        String statusz =
szállit.getElementsByTagName("statusz").item(0).getTextContent();

```

```
        String datum =  
szallit.getElementsByTagName("erkezesidatum").item(0).getTextContent();  
        System.out.println(" - Státusz: " + statusz + ", Érkezés: "  
+ datum);  
    }  
  
    } catch (Exception e) {  
        // Hiba esetén a kivétel kiírása  
        e.printStackTrace();  
    }  
}  
}
```