

JEGYZŐKÖNYV

Webes adatkezelő környezetek

Féléves feladat
Könyvtári nyilvántartó

Készítette: Kurucz Milán
Neptunkód: OVWBWL
Dátum: 2024.December 9.

Tartalomjegyzék

Bevezetés

A feladat leírása

1.1 Az adatbázis ER modell tervezése

1.2 Az adatbázis konvertálása XDM modellre

1.3 Az XDM modell alapján XML dokumentum készítése

1.4 Az XML dokumentum alapján XMLSchema készítése

2.1 Adatolvasás

2.2 Adatírás

2.3 Adatlekérdezés

2.4 Adatmódosítás

Bevezetés

A féléves feladatomban egy Könyvtári nyilvántartó rendszert választottam, melyet a megadottak szerint végigvezettem a feladatkiírásom.

A feladat leírása

A könyvtári rendszerhez tartozó egyedek:

- Könyv
- Kiadó
- Példány (egy adott könyvpéldány)
- Olvasó
- Tagdíj

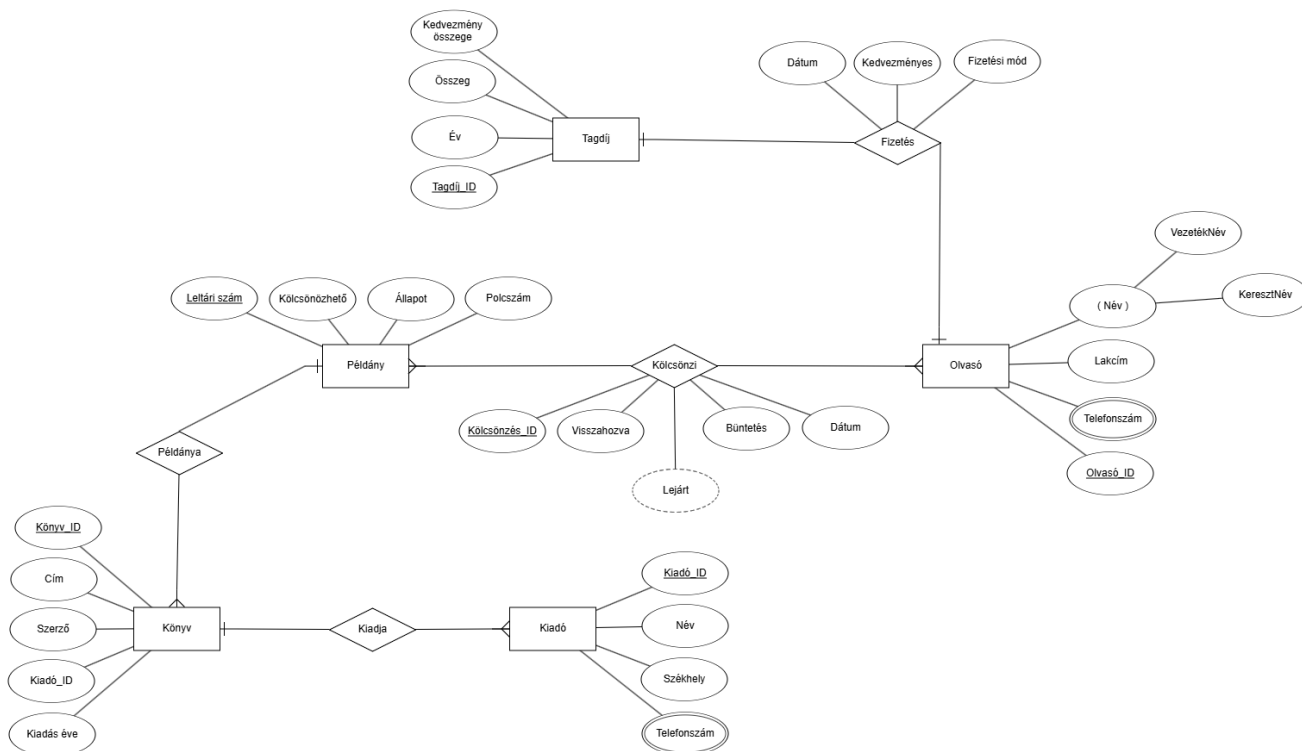
Az egyes egyedek közötti kapcsolatokat felépítettem ER modellben, illetve mindegyik legalább 4 attribútummal rendelkezik.

A feladatkiírás szerint a dokumentumban (vagy csatolt formában) megtalálható egy ER modell, az ER modellről konvertált XDM modell, az XDM modell szerint felépített XML dokumentum, melyet feltöltöttem kitalált példányokkal a további adatmanipulációs és lekérdezési feladatok megoldásához. Az XML dokumentum validálása megtörtént egy XMLSchema szerint, amely tartalmaz saját típusokat és referenciákat is. A projekt mellé csatolva van egy Java program is, amely megvalósítja az adatok olvasását, írását, lekérdezését és módosítását.

1.1 ER modell tervezése

Az ER modell tervezését az ERDPlus weboldalon végeztem el a megadottak szerint:

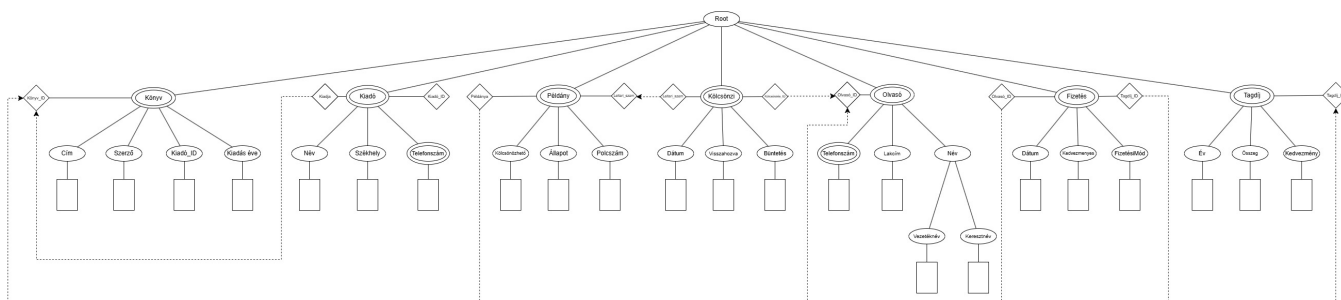
- Legalább 5 egyed
- Többféle kapcsolat (1:1, 1:N, N:M)
- Normál kulcs, összetött, többértékű
- Szabványos ábra



1. ábra: ER modell

1.2 XDM Modell

Az XDM modell (XML Data Model) az XML dokumentumok adatszerkezetének formalizált leírására szolgáló modell. Az XDM a W3C által definiált specifikáció, amely az XML, az XPath, az XQuery és az XSLT technológiák alapját képezi. Ez az adatmodell az XML-t hierarchikus struktúrájú fa formájában ábrázolja, amely csomópontokból (node) áll, és lehetővé teszi az XML dokumentumok egyértelmű és formális kezelését.



1.3 XDM Modell alapján dokumentum

Az XML dokumentumomat Visual Studio Code-ban írtam meg a megadottak szerint, minden többszörösen előforduló elemből legalább 3 szerepel a dokumentum-

ban, illetve az egyes példányok definiálásánál kommentezve.

Maga a kód:

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

<KönyvtariRendszer xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:
  noNamespaceSchemaLocation="XMLSchemaOVWBWL.xsd">
  <!-- Könyvek példányai -->
  <Könyvek>
    <Könyv>
      <KönyvID>1</KönyvID>
      <Cim>A világ története</Cim>
      <Szerzo>Erkel Maria</Szerzo>
      <KiadasEve>1989</KiadasEve>
    </Könyv>
    <Könyv>
      <KönyvID>2</KönyvID>
      <Cim>Utazás alapjai</Cim>
      <Szerzo>Alma Aladar</Szerzo>
      <KiadasEve>2001</KiadasEve>
    </Könyv>
    <Könyv>
      <KönyvID>3</KönyvID>
      <Cim>Matematika alapjai</Cim>
      <Szerzo>Gyari Ferenc</Szerzo>
      <KiadasEve>1977</KiadasEve>
    </Könyv>
  </Könyvek>

  <!-- Az egyes könyvek egyes példányai -->
  <Peldanyok>
    <Peldany>
      <LeltariSzam>001</LeltariSzam>
      <KönyvID>1</KönyvID>
      <Allapot>Jo</Allapot>
      <Polcszam>101A</Polcszam>
      <Kolcsonozheto>igen</Kolcsonozheto>
    </Peldany>
    <Peldany>
      <LeltariSzam>002</LeltariSzam>
      <KönyvID>2</KönyvID>
      <Allapot>Kozepes</Allapot>
      <Polcszam>202B</Polcszam>
      <Kolcsonozheto>nem</Kolcsonozheto>
    </Peldany>
    <Peldany>
      <LeltariSzam>003</LeltariSzam>
      <KönyvID>2</KönyvID>
      <Allapot>Jo</Allapot>
      <Polcszam>202C</Polcszam>
      <Kolcsonozheto>igen</Kolcsonozheto>
    </Peldany>
    <Peldany>
      <LeltariSzam>004</LeltariSzam>
      <KönyvID>3</KönyvID>
```

```

        <Allapot>Kivalo</Allapot>
        <Polcszam>303C</Polcszam>
        <Kolcsonozheto>igen</Kolcsonozheto>
    </Peldany>
</Peldanyok>

<!-- A konyvkiadok paldanyai -->
<Kiadok>
    <Kiado>
        <KiadoID>1001</KiadoID>
        <Nev>Mentor Kiado</Nev>
        <Szekhely>Budapest</Szekhely>
        <Telefonszam>+36 1 234 5678</Telefonszam>
        <Telefonszam>+36 1 356 3765</Telefonszam>
    </Kiado>
    <Kiado>
        <KiadoID>1002</KiadoID>
        <Nev>Urtudomanyi Kiado</Nev>
        <Szekhely>Debrecen</Szekhely>
        <Telefonszam>+36 52 345 6789</Telefonszam>
    </Kiado>
    <Kiado>
        <KiadoID>1003</KiadoID>
        <Nev>Amicus Kiado</Nev>
        <Szekhely>Pecs</Szekhely>
        <Telefonszam>+36 72 456 7890</Telefonszam>
        <Telefonszam>+36 98 346 3169</Telefonszam>
    </Kiado>
</Kiadok>

<!-- Olvasok peldanyai -->
<Olvasok>
    <Olvaso>
        <OlvasoID>101</OlvasoID>
        <Nev>
            <Vezeteknev>Kovacs</Vezeteknev>
            <Keresztnev>Peter</Keresztnev>
        </Nev>
        <Lakcim>Budapest, Fo utca 1.</Lakcim>
        <Telefonszam>+36 30 123 4567</Telefonszam>
        <Telefonszam>+36 20 342 1927</Telefonszam>
    </Olvaso>
    <Olvaso>
        <OlvasoID>102</OlvasoID>
        <Nev>
            <Vezeteknev>Nagy</Vezeteknev>
            <Keresztnev>Anna</Keresztnev>
        </Nev>
        <Lakcim>Debrecen, Kossuth utca 2.</Lakcim>
        <Telefonszam>+36 30 234 5678</Telefonszam>
    </Olvaso>
    <Olvaso>
        <OlvasoID>103</OlvasoID>
        <Nev>
            <Vezeteknev>Toth</Vezeteknev>
            <Keresztnev>Janos</Keresztnev>
        </Nev>
        <Lakcim>Pecs, Rakoczi ut 3.</Lakcim>
        <Telefonszam>+36 30 345 6789</Telefonszam>
    </Olvaso>

```

```

</Olvasok>

<!-- A tagdijak -->
<Tagdijak>
  <Tagdij ev="2023">
    <TagdijID>100</TagdijID>
    <Osszeg>1000</Osszeg>
    <Kedvezmeny>200</Kedvezmeny>
  </Tagdij>
</Tagdijak>

<!-- Kiadja kapcsolat -->
<Kiadasok>
  <Kiadja>
    <KonyvID>1</KonyvID>
    <KiadoID>1001</KiadoID>
  </Kiadja>
  <Kiadja>
    <KonyvID>2</KonyvID>
    <KiadoID>1002</KiadoID>
  </Kiadja>
  <Kiadja>
    <KonyvID>3</KonyvID>
    <KiadoID>1003</KiadoID>
  </Kiadja>
</Kiadasok>

<!-- Kolcsonzi kapcsolat -->
<Kolcsonzesek>
  <Kolcsonzi>
    <KolcsonzesID>001</KolcsonzesID>
    <OlvasoID>101</OlvasoID>
    <LeltariSzam>001</LeltariSzam>
    <Datum>2024-01-01</Datum>
    <Visszahozva>2024-01-10</Visszahozva>
    <Buntetes>0</Buntetes>
  </Kolcsonzi>
  <Kolcsonzi>
    <KolcsonzesID>002</KolcsonzesID>
    <OlvasoID>102</OlvasoID>
    <LeltariSzam>002</LeltariSzam>
    <Datum>2024-01-05</Datum>
    <Visszahozva>2024-01-15</Visszahozva>
    <Buntetes>500</Buntetes>
  </Kolcsonzi>
  <Kolcsonzi>
    <KolcsonzesID>003</KolcsonzesID>
    <OlvasoID>103</OlvasoID>
    <LeltariSzam>003</LeltariSzam>
    <Datum>2024-01-10</Datum>
    <Visszahozva>2024-01-20</Visszahozva>
    <Buntetes>0</Buntetes>
  </Kolcsonzi>
</Kolcsonzesek>

<!-- Fizetesek kapcsolata -->
<Fizetesek>
  <Fizetes>
    <TagdijID>100</TagdijID>
    <OlvasoID>101</OlvasoID>

```

```

        <Datum>2024-01-01</Datum>
        <Kedvezmenyes>Igen</Kedvezmenyes>
        <FizetesiMod>Keszpenz</FizetesiMod>
    </Fizetes>
    <Fizetes>
        <TagdijID>100</TagdijID>
        <OlvasoID>102</OlvasoID>
        <Datum>2024-08-05</Datum>
        <Kedvezmenyes>Nem</Kedvezmenyes>
        <FizetesiMod>Bankkartya</FizetesiMod>
    </Fizetes>
    <Fizetes>
        <TagdijID>100</TagdijID>
        <OlvasoID>103</OlvasoID>
        <Datum>2024-01-10</Datum>
        <Kedvezmenyes>Igen</Kedvezmenyes>
        <FizetesiMod>Atutalas</FizetesiMod>
    </Fizetes>
</Fizetesek>
</KonyvtariRendszer>

```

1.4 XMLSchema dokumentum készítése

A létrehozott séma biztosítja az adatok érvényesítését és a kapcsolatok fenntartását, például a példányok és a könyvek közötti összekapcsolást. A létrehozott egyedi típusok (pld. telefonszámra) segítik az újrafelhasználhatóságot és a jobb olvashatóságot.

Az XMLSchema kódja:

```

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

    <!-- Tipusok -->
    <xs:simpleType name="TelefonSzamTipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="\+?\d{1,3}( \d{1,4}){2,3}" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="KolcsonozhetoTipus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="igen" />
            <xs:enumeration value="nem" />
        </xs:restriction>
    </xs:simpleType>

    <xs:element name="KonyvtariRendszer">
        <xs:complexType>
            <xs:sequence>

                <!-- Konyvek -->
                <xs:element name="Konyvek">

```



```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Konyv" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="KonyvID" type="xs:integer" />
          <xs:element name="Cim" type="xs:string" />
          <xs:element name="Szerzo" type="xs:string" />
          <xs:element name="KiadasEve" type="xs:integer" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

<!-- Peldanyok -->
<xs:element name="Peldanyok">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Peldany" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="LeltariSzam" type="xs:string" />
            <xs:element name="KonyvID" type="xs:integer" />
            <xs:element name="Allapot" type="xs:string" />
            <xs:element name="Polcszam" type="xs:string" />
            <xs:element name="Kolcsonozheto" type="KolcsonozhetoTipus" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Kiadok -->
<xs:element name="Kiadok">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Kiado" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="KiadoID" type="xs:integer" />
            <xs:element name="Nev" type="xs:string" />
            <xs:element name="Szekhely" type="xs:string" />
            <xs:element name="Telefonszam" type="TelefonSzamTipus" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<!-- Olvasok -->
<xs:element name="Olvasok">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Olvaso" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="OlvasoID" type="xs:integer" />
                        <xs:element name="Nev">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Vezeteknev" type="xs:string" />
                                    <xs:element name="Keresztnev" type="xs:string" />
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Lakcim" type="xs:string" />
                        <xs:element name="Telefonszam" type="TelefonSzamTipus" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!-- Tagdijak -->
<xs:element name="Tagdijak">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Tagdij" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="TagdijID" type="xs:integer" />
                        <xs:element name="Osszeg" type="xs:integer" />
                        <xs:element name="Kedvezmeny" type="xs:integer" />
                    </xs:sequence>
                    <xs:attribute name="ev" type="xs:gYear" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!-- Kiadasok -->
<xs:element name="Kiadasok">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Kiadja" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="KonyvID" type="xs:integer" />
          <xs:element name="KiadoID" type="xs:integer" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

<!-- Kolcsonzesek -->
<xs:element name="Kolcsonzesek">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Kolcsonzi" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="KolcsonzesID" type="xs:string" />
            <xs:element name="OlvasoID" type="xs:integer" />
            <xs:element name="LeltariSzam" type="xs:string" />
            <xs:element name="Datum" type="xs:date" />
            <xs:element name="Visszahozva" type="xs:date" />
            <xs:element name="Buntetes" type="xs:integer" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Fizetesek -->
<xs:element name="Fizetesek">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Fizetes" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="TagdijID" type="xs:integer" />
            <xs:element name="OlvasoID" type="xs:integer" />
            <xs:element name="Datum" type="xs:date" />
            <xs:element name="Kedvezmenyes">
              <xs:simpleType>
                <xs:restriction base="xs:string">

```

```

        <xs:enumeration value="Igen"
            />
        <xs:enumeration value="Nem"
            />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="FizetesMod">
    <xs:simpleType>
        <xs:restriction base="xs:string"
            ">
            <xs:enumeration value="
                Keszpenz" />
            <xs:enumeration value="
                Bankkartya" />
            <xs:enumeration value="
                Atutalas" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>

<!-- Egyedi kulcsok es referenciak -->
<xs:key name="UniqueKonyvID">
    <xs:selector xpath="Konyvek/Konyv" />
    <xs:field xpath="KonyvID" />
</xs:key>
<xs:key name="UniqueKiadoID">
    <xs:selector xpath="Kiadok/Kiado" />
    <xs:field xpath="KiadoID" />
</xs:key>
<xs:key name="UniqueLeltariSzam">
    <xs:selector xpath="Peldanyok/Peldany" />
    <xs:field xpath="LeltariSzam" />
</xs:key>
<xs:key name="UniqueOlvasoID">
    <xs:selector xpath="Olvasok/Olvaso" />
    <xs:field xpath="OlvasoID" />
</xs:key>
<xs:key name="UniqueKolcsonzesID">
    <xs:selector xpath="Kolcsonzesek/Kolcsonzi" />
    <xs:field xpath="KolcsonzesID" />
</xs:key>
<xs:key name="UniqueTagdijID">
    <xs:selector xpath="Tagdijak/Tagdij" />
    <xs:field xpath="TagdijID" />
</xs:key>

<!-- Hivatkozasok -->
<xs:keyref name="KonyvPeldanyReferencia" refer="UniqueKonyvID">
    <xs:selector xpath="Peldanyok/Peldany" />
    <xs:field xpath="KonyvID" />

```

```

</xs:keyref>
<xs:keyref name="KiadoKonyvReferencia" refer="UniqueKiadoID">
  <xs:selector xpath="Kiadasok/Kiadja" />
  <xs:field xpath="KiadoID" />
</xs:keyref>
<xs:keyref name="OlvasoKolcsonzesReferencia" refer="UniqueOlvasoID">
  <xs:selector xpath="Kolcsonzesek/Kolcsonzi" />
  <xs:field xpath="OlvasoID" />
</xs:keyref>
<xs:keyref name="PeldanyKolcsonzesReferencia" refer="UniqueLeltariSzam">
  <xs:selector xpath="Kolcsonzesek/Kolcsonzi" />
  <xs:field xpath="LeltariSzam" />
</xs:keyref>
<xs:keyref name="FizetesTagdijReferencia" refer="UniqueTagdijID">
  <xs:selector xpath="Fizetesek/Fizetes" />
  <xs:field xpath="TagdijID" />
</xs:keyref>
</xs:element>
</xs:schema>

```

2. Feladat bevezetés

Az alábbi feladatokat Java nyelven az IntelliJ IDE-ben készítettem el.

2.1 Adatolvasás

Az adatolvasást DOM API-val oldottam meg, pár kódrészletet mellékelve.

```

private static void processNodes(NodeList nodes, int depth) {
    for (int i = 0; i < nodes.getLength(); i++) {
        Node node = nodes.item(i);

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (!hasTextNode(node)) {
                System.out.println(node.getNodeName() + ":");
            } else {
                System.out.print(node.getNodeName() + ": ");
            }
        }

        NamedNodeMap attributes = node.getAttributes();
        if (attributes != null) {
            for (int j = 0; j < attributes.getLength(); j++) {
                Node attribute = attributes.item(j);
                System.out.print(attribute.getNodeName() + " = " +
                    attribute.getNodeValue() + " ");
            }
        }

        if (node.hasChildNodes()) {
            NodeList childNodes = node.getChildNodes();
            StringBuilder textContentBuilder = new StringBuilder();

```

```

        for (int j = 0; j < childNodes.getLength(); j++) {
            Node childNode = childNodes.item(j);

            if (childNode.getNodeType() == Node.TEXT_NODE) {
                String textContent = childNode.getTextContent().trim();
                if (!textContent.isEmpty()) {
                    textContentBuilder.append(textContent).append("
");
                }
            }
        }

        if (textContentBuilder.length() > 0) {
            System.out.println(textContentBuilder.toString().trim());
        }
        else {
            processNodes(childNodes, depth + 1);
        }
    }

    if (depth == 2) {
        System.out.println();
    }
}
}

```

Az alábbi kódrészlet a megadott útvonalon lévő beolvasott fájlból a DOM API segítségével feldolgozza és kiírja az XML dokumentum tartalmát.

2.2 Adatírás

A fa struktúra felépítését és fájlba kiírását DOM API-val készítettem el, mely először Fa struktúrában a konzolra írja a felépített XML dokumentum elemeit, majd kiírja őket egy fájlba.

```

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document doc = builder.newDocument();

// Gyoker elem
Element root = doc.createElement("KonyvtariRendszer");
root.setAttribute("xmlns:xs", "http://www.w3.org/2001/XMLSchema-instance");
root.setAttribute("xs:noNamespaceSchemaLocation", "XMLSchemaOVWBWL.xsd");
doc.appendChild(root);

Element konyvek = doc.createElement("Konyvek");
root.appendChild(konyvek);
addKonyv(doc, konyvek, "1", "A világ története", "Erkel Maria", "1989");
addKonyv(doc, konyvek, "2", "Urutazás alapjai", "Alma Aladar", "2001");

```

```
addKonyv(doc, konyvek, "3", "Matematika alapjai", "Gyari Ferenc", "1977");

Element peldanyok = doc.createElement("Peldanyok");
root.appendChild(peldanyok);
addPeldany(doc, peldanyok, "001", "1", "Jo", "101A", "igen");
addPeldany(doc, peldanyok, "002", "2", "Kozepes", "202B", "nem");
addPeldany(doc, peldanyok, "003", "2", "Jo", "202C", "igen");
addPeldany(doc, peldanyok, "004", "3", "Kivalo", "303C", "igen");
```

Az alábbi kódrészlet végzi az XML dokumentumunk replikálásának az első részét. Felépítjük vele magát az XML dokumentumot, amit aztán a már korábban megírt DOMRead tömbösített formában kiír a konzolra, illetve egy másik kódrészlet element egy fájlba.

2.3 Adatlekérdezés

Az adatlekérdezésnél 4 lekérdezést valósítottam meg:

- Könyvek címe és szerzője
- Kölcsönözhető, még nem kint lévő példányok
- Olvasók neve és lakcíme
- Előző kölcsönzések büntetés nélkül

Kódrészlet a harmadik lekérdezésre:

```
System.out.println("\n3. lekerdezes: Olvasok neve es lakcime:");
NodeList olvasok = doc.getElementsByTagName("Olvaso");
for (int i = 0; i < olvasok.getLength(); i++) {
    Element olvaso = (Element) olvasok.item(i);
    String vezeteknev = olvaso.getElementsByTagName("Vezeteknev").item(0).
        gettextContent();
    String keresztnév = olvaso.getElementsByTagName("Keresztnév").item(0).
        gettextContent();
    String lakcim = olvaso.getElementsByTagName("Lakcim").item(0).
        gettextContent();
    System.out.println("Nev: " + vezeteknev + " " + keresztnév + ", Lakcim:
        " + lakcim);
}
```

2.4 Adatmódosítás

Az adatmódosítás feladatnál összesen 4 módosítást hajtottam végre, egy új könyv hozzáadását a meglévők mellé, egy példány fizikai állapotának a módosítását, egy meglévő könyv címét átírtam illetve frissítettem egy olvasó lakcímét.

Kódrészlet az első módosításról:

```
Element newBook = doc.createElement("Konyv");
newBook.appendChild(createElement(doc, "KonyvID", "4"));
newBook.appendChild(createElement(doc, "Cim", "Java programozas"));
newBook.appendChild(createElement(doc, "Szerzo", "Kurucz Milan"));
newBook.appendChild(createElement(doc, "KiadasEve", "2024"));
doc.getElementsByTagName("Konyvek").item(0).appendChild(newBook);
```