

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Oktatás témájú XML adatkezelő rendszer és a
hozzátartozó DOM API programok
megvalósítása

Készítette: **Varga Bence**

Neptunkód: **HK74CE**

Dátum: 2024.12.09.

Tartalomjegyzék

1. Bevezetés.....	2
2. A feladat leírása	3
3. Első feladat	3
1. Az adatbázis ER modell tervezése	3
2. Az adatbázis konvertálása XDM modellre	4
3. Az XDM modell alapján XML dokumentum készítése	5
4. Az XML dokumentum alapján XMLSchema készítése	8
4. Második feladat.....	14
1. Adatolvasás (DOMReadHK74CE.java).....	14
2. Adatírás (DOMWriteHK74CE.java)	15
3. Adatlekérdezés (DOMQueryHK74CE.java)	16
4. Adatmódosítás (DOMModifyHK74CE.java)	17

Bevezetés

A feladat egy XML adatkezelő rendszer megvalósítása és a dokumentum DOM API-val történő feldolgozása. Az első részben az adatbázis tervezésére és az XML dokumentum generálására történik. A második részben DOM API segítségével végzünk adatbeolvasást, írást, lekérdezést és módosítást.

A feladat leírása:

A projekt során egy XML adatkezelő rendszert készítettünk, amely egy fiktív rendszer adatainak kezelésére szolgál. A témakör az oktatás. Az adatbázis tervezése során 5 entitást és több kapcsolatot definiáltunk: 1:1, 1:N és N:M kapcsolatok. A rendszer az alábbi lépéseket valósította meg:

1. Az ER modell megtervezése és vizualizációja szabványos szimbólumokkal.
 2. Az ER modell átalakítása XDM modellre.
 3. Az XDM modell alapján validált XML dokumentum generálása, többszörös előfordulási elemekkel.
 4. Az XML dokumentumhoz tartozó XMLSchema elkészítése, saját típusok definiálásával.
 5. DOM API segítségével négy különböző Java osztály készült, amelyek az XML adatok kezelését végzik.
-

Első feladat

1.1 Az adatbázis ER modell tervezése

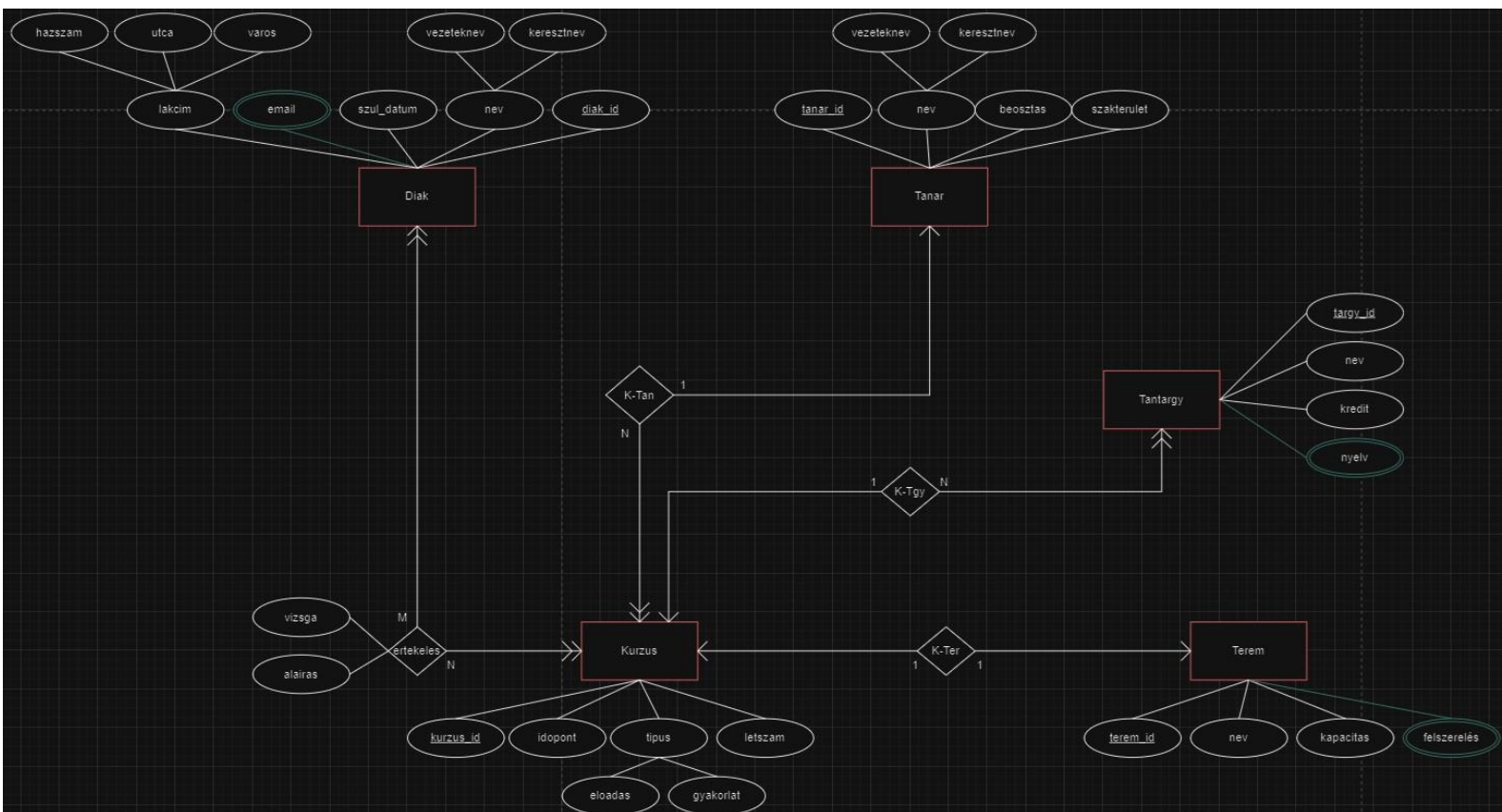
Megvalósítás:

Az ER modell 5 entitást tartalmaz: **Diák**, **Tanár**, **Tantárgy**, **Terem** és **Kurzus**. Az entitások kapcsolatai:

- **Diák-Kurzus:** N:M kapcsolat.
- **Tanár-Kurzus:** 1:N kapcsolat.
- **Kurzus-Tantárgy:** 1:N kapcsolat.
- **Kurzus-Terem:** 1:1 kapcsolat.

A modell tartalmaz összetett (pl. lakcím) és többértékű attribútumokat (pl. felszerelés).

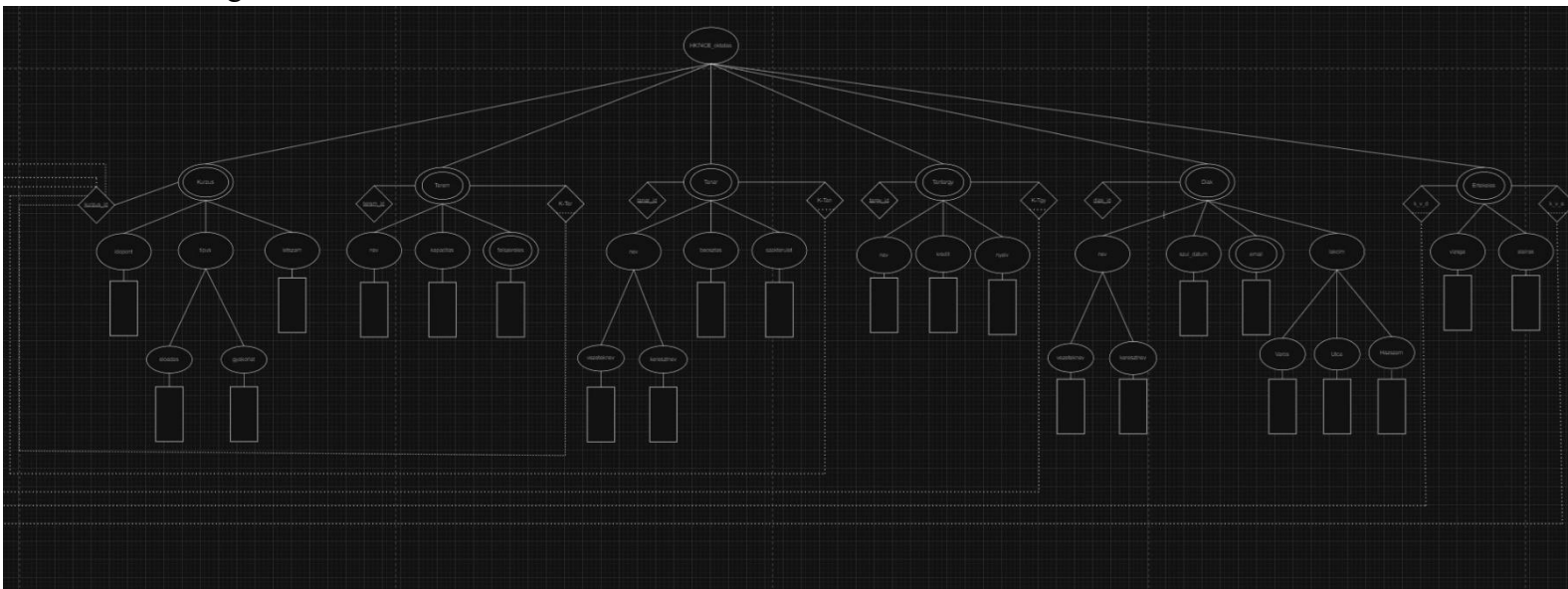
ER modell:



1.2 Az adatbázis konvertálása XDM modellre

Megvalósítás:

Az XDM modell az ER modell hierarchikus reprezentációja. Az entitások csomópontként, az attribútumok pedig alsomópontként jelennek meg. Az idegen kulcsok a kapcsolatok megvalósításához kerülnek felhasználásra. Az **XDM modell**:



1.3 Az XDM modell alapján XML dokumentum készítése

Megvalósítás:

Az XML dokumentumot a XDM modell alapján hoztuk létre. Minden entitásból legalább három példányt generáltunk, a kapcsolatok attribútumai pedig alárendelt csomópontként szerepelnek.

A kód: [XML_HK74CE.xml](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<HK74CE_oktatas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="XMLSchemaHK74CE.xsd">

  <!-- Diákok -->
  <Diakok>
    <Diak DiakID="1">
      <Nev>
        <Keresztnev>Bence</Keresztnev>
        <Vezeteknev>Varga</Vezeteknev>
      </Nev>
      <SzulDatum>2000-05-12</SzulDatum>
      <Email>varga.bence2000@gmail.com</Email>
      <Email>varga.bence@student.uni-miskolc.hu</Email>
      <Lakcim>
        <Varos>Tiszafüred</Varos>
        <Utca>Bán Zsigmond</Utca>
        <Hazzsam>02</Hazzsam>
      </Lakcim>
    </Diak>
    <Diak DiakID="2">
      <Nev>
        <Keresztnev>Anna</Keresztnev>
        <Vezeteknev>Nagy</Vezeteknev>
      </Nev>
      <SzulDatum>1999-08-22</SzulDatum>
      <Email>anna.nagy@example.com</Email>
      <Email>anna.student@example.com</Email>
      <Email>anna.office@example.hu</Email>
      <Lakcim>
        <Varos>Szeged</Varos>
        <Utca>Kossuth tér</Utca>
        <Hazzsam>5</Hazzsam>
      </Lakcim>
    </Diak>
```

```

<Diak DiakID="3">
  <Nev>
    <Keresztnev>Péter</Keresztnev>
    <Vezeteknev>Kis</Vezeteknev>
  </Nev>
  <SzulDatum>2001-11-03</SzulDatum>
  <Email>peter.kis@example.com</Email>
  <Email>peter.kis@student.uni-miskolc.hu</Email>
  <Lakcim>
    <Varos>Debrecen</Varos>
    <Utca>Nagy utca</Utca>
    <Hazzsam>8</Hazzsam>
  </Lakcim>
</Diak>
</Diakok>

<!-- Tanárok -->
<Tanarok>
  <Tanar TanarID="101">
    <Nev>
      <Keresztnev>Anna</Keresztnev>
      <Vezeteknev>Szabó</Vezeteknev>
    </Nev>
    <Beosztas>Docens</Beosztas>
    <Szakterulet>Informatika</Szakterulet>
  </Tanar>
  <Tanar TanarID="102">
    <Nev>
      <Keresztnev>Balázs</Keresztnev>
      <Vezeteknev>Fekete</Vezeteknev>
    </Nev>
    <Beosztas>Adjunktus</Beosztas>
    <Szakterulet>Matematika</Szakterulet>
  </Tanar>
  <Tanar TanarID="103">
    <Nev>
      <Keresztnev>Mária</Keresztnev>
      <Vezeteknev>Fehér</Vezeteknev>
    </Nev>
    <Beosztas>Oktató</Beosztas>
    <Szakterulet>Fizika</Szakterulet>
  </Tanar>
</Tanarok>

<!-- Tantárgyak -->
<Tantargyak>
  <Tantargy TargyID="MAT101">
    <Nev>Matematika</Nev>

```

```

        <Kredit>5</Kredit>
        <Nyelv>Magyar</Nyelv>
        <Nyelv>Angol</Nyelv>
    </Tantargy>
    <Tantargy TargyID="INF202">
        <Nev>Informatika</Nev>
        <Kredit>5</Kredit>
        <Nyelv>Magyar</Nyelv>
    </Tantargy>
    <Tantargy TargyID="FIZ303">
        <Nev>Fizika</Nev>
        <Kredit>4</Kredit>
        <Nyelv>Angol</Nyelv>
    </Tantargy>
</Tantargyak>

<!-- Termek -->
<Termek>
    <Terem TeremID="101">
        <Nev>Eloadoterem A</Nev>
        <Kapacitas>100</Kapacitas>
        <Felszerelés>Projektor</Felszerelés>
        <Felszerelés>Tábla</Felszerelés>
        <Felszerelés>Függöny</Felszerelés>
    </Terem>
    <Terem TeremID="102">
        <Nev>Labor B</Nev>
        <Kapacitas>30</Kapacitas>
        <Felszerelés>Számítógépek</Felszerelés>
    </Terem>
    <Terem TeremID="103">
        <Nev>Eloadoterem C</Nev>
        <Kapacitas>50</Kapacitas>
        <Felszerelés>Tábla</Felszerelés>
    </Terem>
</Termek>

<!-- Kurzusok -->
<Kurzusok>
    <Kurzus KurzusID="KURZ001">
        <Idopont>2024-12-05</Idopont>
        <Tipus>
            <Eloadas>true</Eloadas>
            <Gyakorlat>false</Gyakorlat>
        </Tipus>
        <Letszam>30</Letszam>
    </Kurzus>
    <Kurzus KurzusID="KURZ002">

```

```

        <Idopont>2024-12-06</Idopont>
        <Tipus>
            <Eloadas>false</Eloadas>
            <Gyakorlat>true</Gyakorlat>
        </Tipus>
        <Letszam>25</Letszam>
    </Kurzus>
    <Kurzus KurzusID="KURZ003">
        <Idopont>2024-12-07</Idopont>
        <Tipus>
            <Eloadas>true</Eloadas>
            <Gyakorlat>true</Gyakorlat>
        </Tipus>
        <Letszam>40</Letszam>
    </Kurzus>
</Kurzusok>

<!-- Kapcsolatok -->
<Kapcsolatok>
    <DiakKurzusKapcsolatok>
        <Kapcsolat DiakID="1" KurzusID="KURZ001">
            <Vizsga>90</Vizsga>
            <Alairas>aláírva</Alairas>
        </Kapcsolat>
        <Kapcsolat DiakID="2" KurzusID="KURZ002">
            <Vizsga>10</Vizsga>
            <Alairas>megtagadva</Alairas>
        </Kapcsolat>
        <Kapcsolat DiakID="3" KurzusID="KURZ003">
            <Vizsga>80</Vizsga>
            <Alairas>aláírva</Alairas>
        </Kapcsolat>
    </DiakKurzusKapcsolatok>
</Kapcsolatok>
</HK74CE_oktatas>

```

1.4 Az XML dokumentum alapján XMLSchema készítése

Megvalósítás:

Az XMLSchema fájl saját típusokat is definiál, amelyek tartalmazzák az entitások összetett attribútumait. Továbbá az elsődleges kulcsok és idegen kulcsok biztosítják a kapcsolatok integritását.

A kód: [XMLSchemaHK74CE.xsd](#)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <!-- Összetett típus a névhez -->
  <xsd:complexType name="NevTipus">
    <xsd:sequence>
      <xsd:element name="Keresztnev" type="xsd:string"/>
      <xsd:element name="Vezeteknev" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <!-- Összetett típus a lakcímhez -->
  <xsd:complexType name="LakcimTipus">
    <xsd:sequence>
      <xsd:element name="Varos" type="xsd:string"/>
      <xsd:element name="Utca" type="xsd:string"/>
      <xsd:element name="Hazzsam" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <!-- Összetett típus a kurzus típusához -->
  <xsd:complexType name="TipusTipus">
    <xsd:sequence>
      <xsd:element name="Eloadas" type="xsd:boolean"/>
      <xsd:element name="Gyakorlat" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>

  <!-- Egyszerű típus a kredit értékéhez-->
  <xsd:simpleType name="KreditTipus">
    <xsd:restriction base="xsd:integer">
      <xsd:minInclusive value="1"/>
      <xsd:maxInclusive value="5"/>
    </xsd:restriction>
  </xsd:simpleType>

  <!-- Gyökérelem -->
  <xsd:element name="HK74CE_oktatas">
    <xsd:complexType>
      <xsd:sequence>
        <!-- Diákok -->
        <xsd:element name="Diakok">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Diak" maxOccurs="unbounded">
```

```

        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Nev"
type="NevTipus"/>
                <xsd:element name="SzulDatum"
type="xsd:date"/>
                <xsd:element name="Email"
type="xsd:string" maxOccurs="unbounded"/>
                <xsd:element name="Lakcim"
type="LakcimTipus"/>
            </xsd:sequence>
            <xsd:attribute name="DiakID"
type="xsd:string" use="required"/>
        </xsd:complexType>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

    <!-- Tanárok -->
    <xsd:element name="Tanarok">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Tanar" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="Nev"
type="NevTipus"/>
                            <xsd:element name="Beosztas"
type="xsd:string"/>
                            <xsd:element name="Szakterulet"
type="xsd:string"/>
                        </xsd:sequence>
                        <xsd:attribute name="TanarID"
type="xsd:string" use="required"/>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <!-- Tantárgyak -->
    <xsd:element name="Tantargyak">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Tantargy"
maxOccurs="unbounded">
                    <xsd:complexType>

```

```

                                <xsd:sequence>
                                    <xsd:element name="Nev"
type="xsd:string"/>
                                <xsd:element name="Kredit"
type="KreditTipus"/> <!-- Egyszerű típus -->
                                <xsd:element name="Nyelv"
type="xsd:string" maxOccurs="unbounded"/>
                                </xsd:sequence>
                                <xsd:attribute name="TargyID"
type="xsd:string" use="required"/>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>

                <!-- Termek -->
                <xsd:element name="Termek">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="Terem" maxOccurs="unbounded">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="Nev"
type="xsd:string"/>
                                        <xsd:element name="Kapacitas"
type="xsd:integer"/>
                                        <xsd:element name="Felszerelés"
type="xsd:string" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                    <xsd:attribute name="TeremID"
type="xsd:string" use="required"/>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>

                <!-- Kurzusok -->
                <xsd:element name="Kurzusok">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="Kurzus" maxOccurs="unbounded">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="Idopont"
type="xsd:date"/>

```

```
<xsd:element name="Tipus"
type="TipusTipus"/>
<xsd:element name="Letszam"
type="xsd:integer"/>
</xsd:sequence>
<xsd:attribute name="KurzusID"
type="xsd:string" use="required"/>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<!-- Egyedi kurzus ID (Nem lehet két azonos a kurzus kód) -->
<xsd:unique name="UniqueKurzusID">
<xsd:selector xpath="Kurzus"/>
<xsd:field xpath="@KurzusID"/>
</xsd:unique>
</xsd:element>
<!-- Kapcsolatok -->
<xsd:element name="Kapcsolatok">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="DiakKurzusKapcsolatok">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Kapcsolat"
maxOccurs="unbounded">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Vizsga"
type="xsd:integer"/>
<xsd:element
name="Alairas" type="xsd:string"/>
</xsd:sequence>
<xsd:attribute name="DiakID"
type="xsd:string" use="required"/>
<xsd:attribute name="KurzusID"
type="xsd:string" use="required"/>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
```

```

</xsd:complexType>

<!-- Kulcsok és referenciák -->
<xsd:key name="DiakPrimaryKey">
  <xsd:selector xpath="Diakok/Diak"/>
  <xsd:field xpath="@DiakID"/>
</xsd:key>
<xsd:key name="KurzusPrimaryKey">
  <xsd:selector xpath="Kurzusok/Kurzus"/>
  <xsd:field xpath="@KurzusID"/>
</xsd:key>
<xsd:key name="TanarPrimaryKey">
  <xsd:selector xpath="Tanarok/Tanar"/>
  <xsd:field xpath="@TanarID"/>
</xsd:key>
<xsd:key name="TantargyPrimaryKey">
  <xsd:selector xpath="Tantargyak/Tantargy"/>
  <xsd:field xpath="@TargyID"/>
</xsd:key>
<xsd:key name="TeremPrimaryKey">
  <xsd:selector xpath="Termek/Terem"/>
  <xsd:field xpath="@TeremID"/>
</xsd:key>
<xsd:keyref name="DiakKapcsolatRef" refer="DiakPrimaryKey">
  <xsd:selector
xpath="Kapcsolatok/DiakKurzusKapcsolatok/Kapcsolat"/>
  <xsd:field xpath="@DiakID"/>
</xsd:keyref>
<xsd:keyref name="KurzusKapcsolatRef" refer="KurzusPrimaryKey">
  <xsd:selector
xpath="Kapcsolatok/DiakKurzusKapcsolatok/Kapcsolat"/>
  <xsd:field xpath="@KurzusID"/>
</xsd:keyref>
</xsd:element>
</xsd:schema>

```

Második feladat

2.1 Adatolvasás (DOMReadHK74CE.java)

Cél: Az XML dokumentum feldolgozása, az adatok blokk formájú kiírása konzolra és mentése fájlba.

kódrészlet:

```
// Az összes "Diak" elem lekérése
NodeList diakok = doc.getElementsByTagName("Diak");
// Minden diák feldolgozása
for (int i = 0; i < diakok.getLength(); i++) {
    Node node = diakok.item(i);
    // Csak az elemtípusú csomópontokat dolgozza fel
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element diak = (Element) node;

        // Diák adatainak kiolvasása (pl. DiakID, név, születési dátum)
        String diakID =
diak.getElementsByTagName("DiakID").item(0).getTextContent();

        String keresztnév =
diak.getElementsByTagName("Keresztnév").item(0).getTextContent();

        String vezeteknev =
diak.getElementsByTagName("Vezeteknev").item(0).getTextContent();

        String szulDatum =
diak.getElementsByTagName("SzulDatum").item(0).getTextContent();

        // Eredmények konzolra és fájlba írása
        String output = String.format("Diák ID: %s\nNév: %s %s\nSzületési dátum: %s\n",
diakID, keresztnév, vezeteknev, szulDatum);

        System.out.println(output);
        writer.println(output);
    }
}
```

Eredmény: Az adatok blokk formában jelennek meg a konzolon, és [text fájl](#)-ba mentésre kerül.

A kód: [DomReadHK74CE.java](#)

2.2 Adatírás (DOMWriteHK74CE.java)

Cél: Az XML dokumentum tartalmának fa struktúra formában történő konzolra írása és mentése új fájlba.

kódrészlet:

//TransformerFactory inicializálása

```
TransformerFactory transformerFactory = TransformerFactory.newInstance();  
Transformer transformer = transformerFactory.newTransformer();
```

// A kimenet behúzásainak engedélyezése

```
transformer.setOutputProperty(OutputKeys.INDENT, "yes");
```

// Az XML dokumentum konzolra írása

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

// Az XML dokumentum fájlba mentése

```
StreamResult fileResult = new StreamResult(new File("XML_HK74CE_1.xml"));  
transformer.transform(new DOMSource(doc), fileResult); // Átalakítás és fájlba írás
```

Eredmény: Az XML tartalom mentve lett az [XML_HK74CE_1.xml](#) fájlba.

A kód: [DOMWriteHK74CE.java](#)

2.3 Adatlekérdezés (DOMQueryHK74CE.java)

Cél: Az XML dokumentum adatai alapján legalább 4 lekérdezés végrehajtása.

kódrészlet:

// Összes "Terem" elem lekérése

```
NodeList termek = doc.getElementsByTagName("Terem");
```

```
for (int i = 0; i < termek.getLength(); i++) { // Minden terem feldolgozása
```

```
    Node node = termek.item(i);
```

```
    if (node.getNodeType() == Node.ELEMENT_NODE) { // Csak az elemtípusú  
        csomópontokat dolgozza fel
```

```
        Element terem = (Element) node;
```

// szűrés ülőhely alapján

```
        int kapacitas =  
Integer.parseInt(terem.getElementsByTagName("Kapacitas").item(0).getTextContent(  
));
```

// Ha a kapacitás nagyobb, mint 100

```
        if (kapacitas > 100) {
```

```
            String teremID =  
terem.getElementsByTagName("TeremID").item(0).getTextContent();
```

```
            String nev = terem.getElementsByTagName("Nev").item(0).getTextContent();
```

// Szűrt adatok kiírása

```
            System.out.printf("Terem ID: %s, Név: %s, Kapacitás: %d\n", teremID, nev,  
kapacitas);
```

```
        }
```

```
    }
```

```
}
```

Eredmény: A konzolon megjelenik a szűrt adatok listája (pl. 100 ülőhelynél nagyobb kapacitású termék listája).

A kód: [DOMQueryHK74CE.java](#)

2.4 Adatmódosítás (DOMModifyHK74CE.java)

Cél: Az XML dokumentum módosítása legalább 4 adatnál, majd mentés új fájlba.

kód:

```
// Összes "Kurzus" elem lekérése
NodeList kurzusok = doc.getElementsByTagName("Kurzus");
for (int i = 0; i < kurzusok.getLength(); i++) { // Minden kurzus feldolgozása
    Node node = kurzusok.item(i);

    // Csak az elemtípusú csomópontokat dolgozza fel
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element kurzus = (Element) node;

        // Ha az ID megegyezik, növeljük a létszámot
        if
(kurzus.getElementsByTagName("KurzusID").item(0).getTextContent().equals("KURZ
001")) {
            int jelenlegiLetszam =
Integer.parseInt(kurzus.getElementsByTagName("Letszam").item(0).getTextContent()
);

            // Létszám növelése
kurzus.getElementsByTagName("Letszam").item(0).setTextContent(String.valueOf(jel
enlegiLetszam + 5));

            // Módosítás kiírása
            System.out.println("Kurzus KURZ001 létszáma növelve: " + (jelenlegiLetszam
+ 5));

            break;
        }
    }
}
```

Eredmény: A módosított XML tartalom mentve lett az [XML_HK74CE_Modified.xml](#) fájlba.

A kód: [DOMModifyHK74CE.java](#)