

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Könyvelőiroda

Készítette: Bartók-Balogh Gábor

Neptunkód: QVTQO8

Dátum:2022.11.25

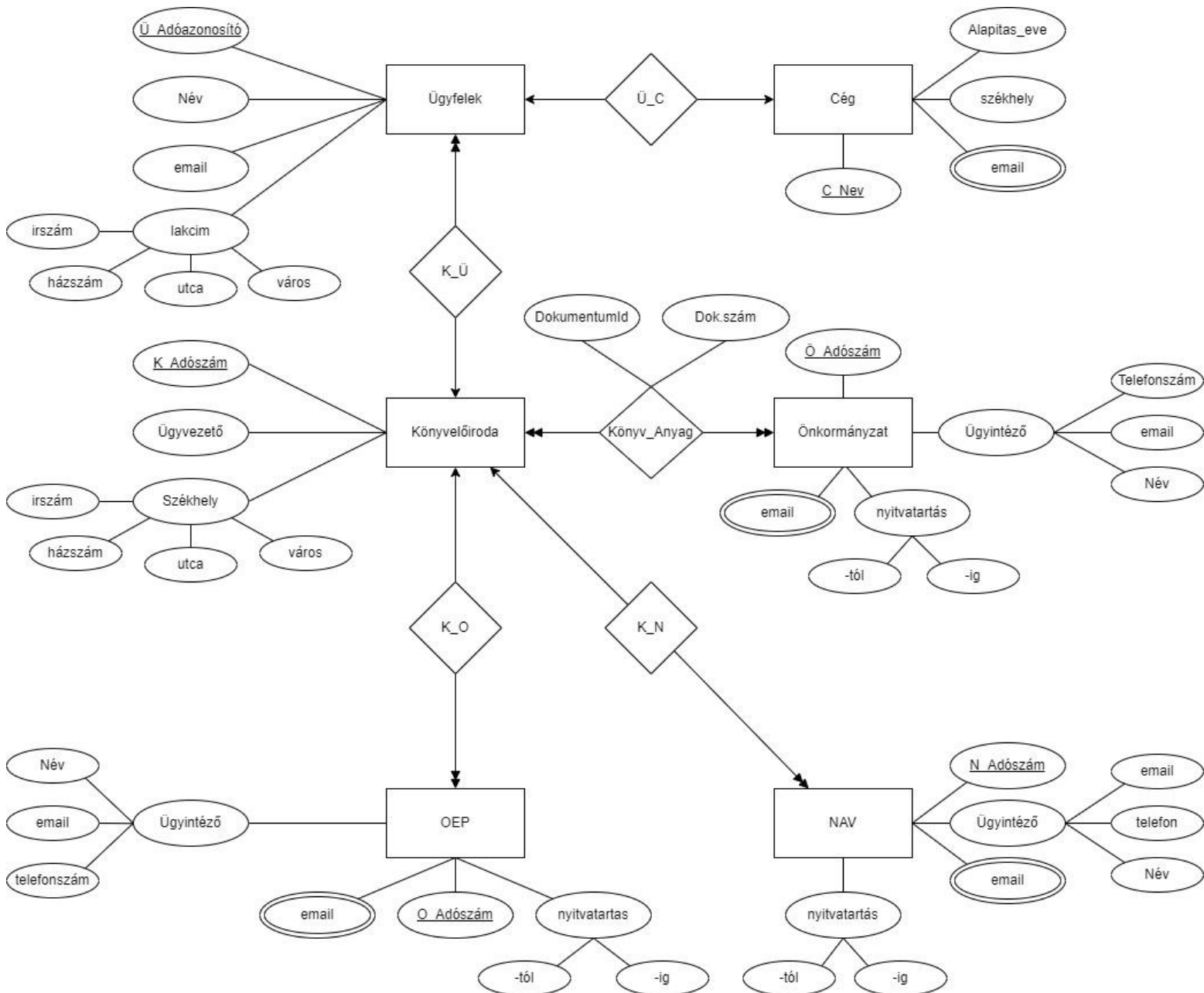
## Feladat leírás:

Féléves feladatom témájának egy Könyvelőiroda lemodellezését választottam. A féléves feladatban miután tanár úr elfogadta a modellt alapvető XML adatkezeléseket (módosítás, olvasás, lekérdezés) kell megvalósítani validált XML fájl alapján. Az ER modellt át kell konvertálni XDM modellre az ER modellben látott kapcsolatok és kulcsok megfelelő ábrázolásával. Az XDM modell alapján egy XML fájlt kellett létrehozni az XDM modellben ábrázolt egyedek legalább háromszori előfordulásával. Az XML fájl alapján egy XMLSchemát kellett megvalósítani saját típusokkal, valamint kulcsokkal, amelyek a kapcsolatokat alakítják ki az XML fájlban. Második feladatban DOM programokat kellett készíteni, amelyek 3 alapvető XML adatkezelést valósítanak meg. Az elsőben ki kell olvasni az XML fájlban szereplő elementeket is azok gyerek elementjeit ezt kiírni egy fájlba majd konzolra is. A másodikban lekérdezéseket kellett csinálni, ami azt jelenti, hogy az XML fájlból tetszőleges adatokat kérdezzük le. Harmadszorra pedig egy módosításokat végrehajtó programot kellett írni melyben tetszőleges adatokat módosíthatunk az XML fájlunkban.

## 1.feladat

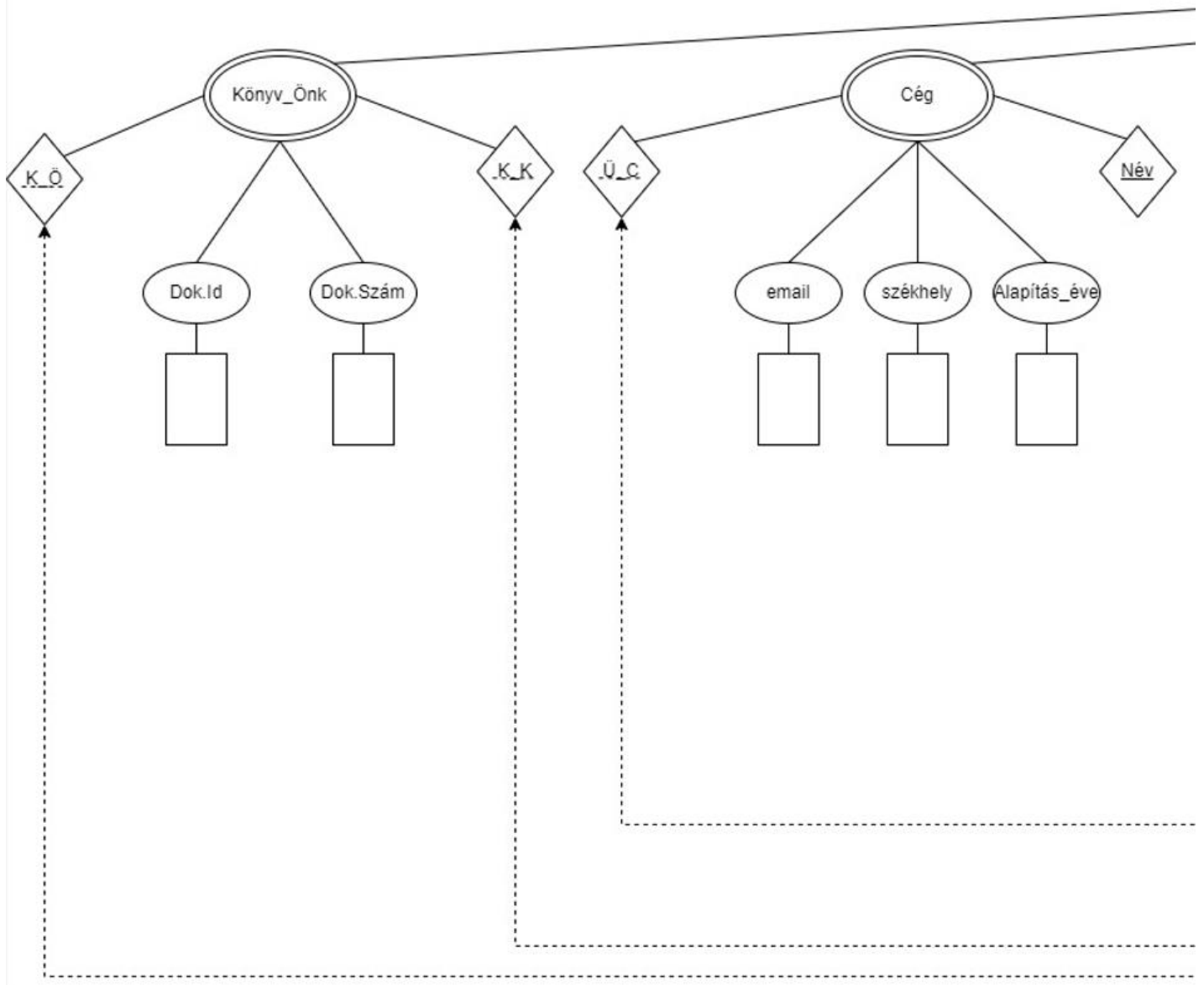
### 1a) ER modell.

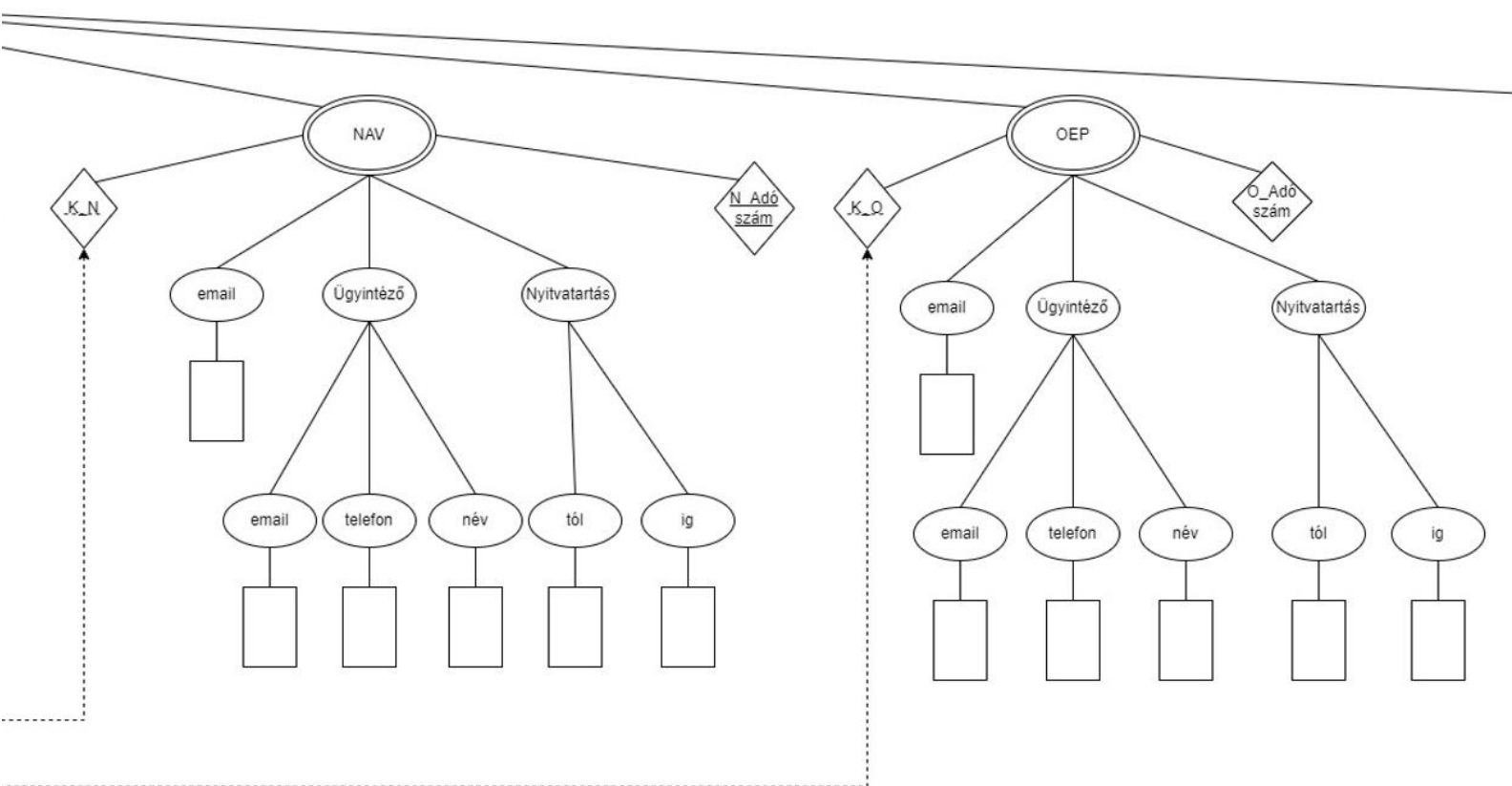
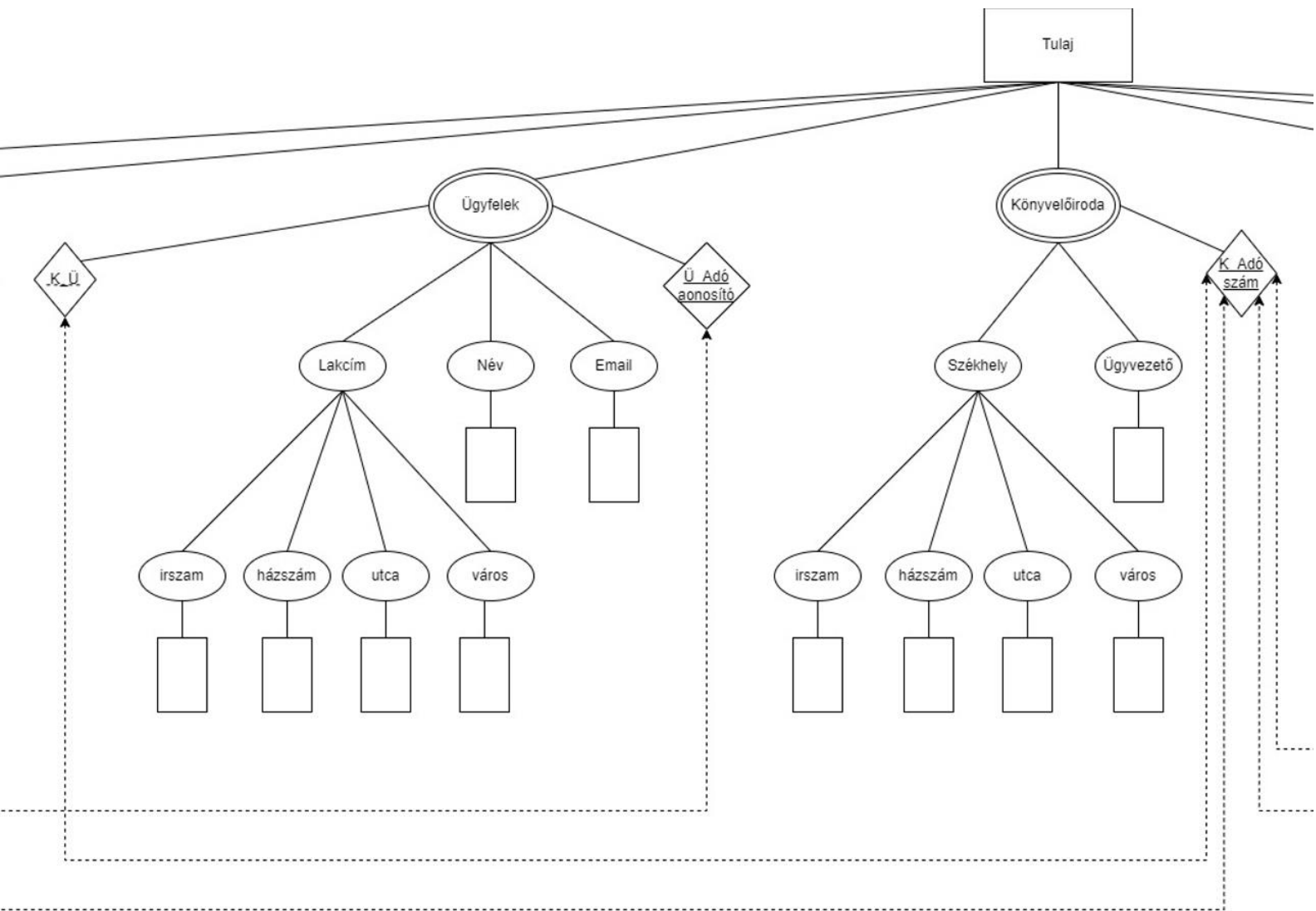
Az ER modellem egy Könyvelőiroda általános felépítését szemlélteti. Rendelkezik 1:1, 1:N, N:M kapcsolatokkal. 6 egyeddel rendelkezik amelyeknél fellelhető mindenfajta tulajdonság. A kapcsolatokat fajtáját nyilakkal szemléltetem a tulajdonságokat attól függően milyen az annak megfelelő szabványos jelöléssel ábrázolom.

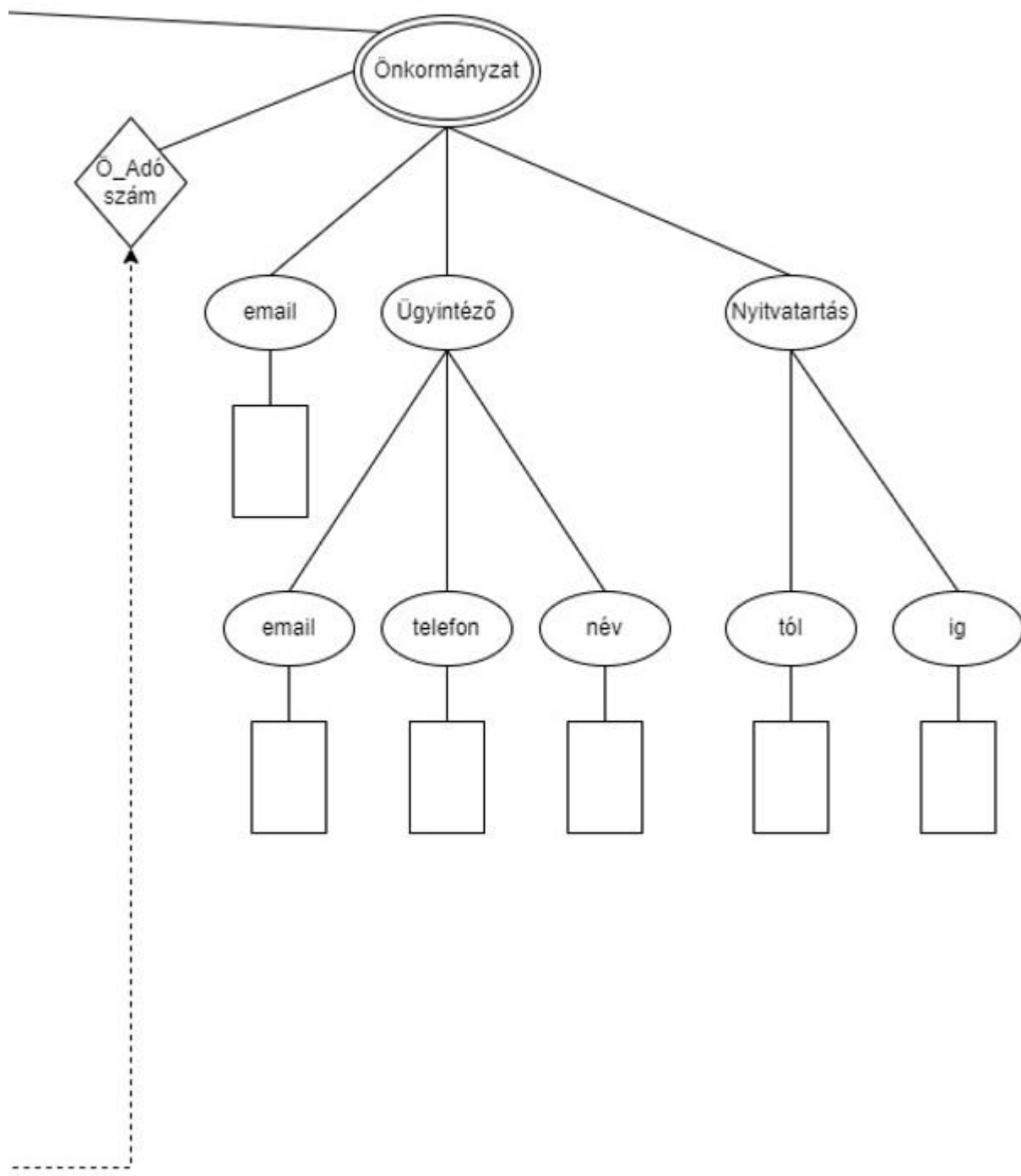


## 1b) XDM modell

Az ER modell alapján egy XDM modell került kialakításra, ahol egy fa szerű struktúrával szemléltetem a későbbi XML fájl felépítését. A több-több kapcsolat egy külön elementet kap. Az 1:1 és 1:N kapcsolatnál az idegen kulcsok külön rombuszban kerültek mint ahogyan az elsődleges kulcsok is. Majd az ER modell alapján összekötésre kerülnek az idegen kulcsok a megfelelő elsődleges kulcsokkal.







## 1c) XML

Az XDM modell alapján kialakításra került az XML. Minden egyed háromszor fordul elő az XML fájlban. Az egyedek elementekké alakulnak át amelyeknek gyermek elementjei a tulajdonságok a kapcsolatokat attribútum segítségével vannak megvalósítva.

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

<Tulaj xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemeQVTQ08.xsd">

  <!--
    Konyveloirodak
  -->
  <Konyveloiroda k_adoszam="k1">
    <szekhely>
      <irszam>2673</irszam>
      <hazszam>55</hazszam>
      <utca>Kakora</utca>
      <varos>Pingalo</varos>
    </szekhely>
    <ugyvezeto>Vigaldi Béla</ugyvezeto>
  </Konyveloiroda>

  <Konyveloiroda k_adoszam="k2">
    <szekhely>
      <irszam>2345</irszam>
      <hazszam>21</hazszam>
      <utca>Papír utca</utca>
      <varos>Eifengel</varos>
    </szekhely>
    <ugyvezeto> Pakula Anna </ugyvezeto>
  </Konyveloiroda>

  <Konyveloiroda k_adoszam="k3">
    <szekhely>
      <irszam>8654</irszam>
      <hazszam>81</hazszam>
      <utca>Tengi</utca>
      <varos>Spank</varos>
    </szekhely>
    <ugyvezeto>Dekt Elek</ugyvezeto>
  </Konyveloiroda>
```

```
<!--  
    Cegek  
-->  
<Ceg nev="egy" u_c="u1">  
    <email>cegecske@portal.hu</email>  
    <szekhely>Karácsony</szekhely>  
    <alapitas_eve>1144</alapitas_eve>  
</Ceg>  
  
<Ceg nev="ketto" u_c="u2">  
    <email>kapucsinalo@portal.hu</email>  
    <szekhely>Husvet</szekhely>  
    <alapitas_eve>1450</alapitas_eve>  
</Ceg>  
  
<Ceg nev="harom" u_c="u3">  
    <email>lakatos@portal.hu</email>  
    <szekhely>Ujev</szekhely>  
    <alapitas_eve>1888</alapitas_eve>  
</Ceg>
```

```
<!--  
    Ugyfelek  
-->  
<Ugyfelek u_adoazonosito="u1" k_u="k1">  
    <nev>Higo Hugo</nev>  
    <email>asdasd@vaeft.hu</email>  
    <lakcim>  
        <irszam>8431</irszam>  
        <hazszam>515</hazszam>  
        <utca>Hajgu</utca>  
        <varos>Vabuga</varos>  
    </lakcim>  
</Ugyfelek>  
  
<Ugyfelek u_adoazonosito="u2" k_u="k2">  
    <nev>Verdes Elek</nev>  
    <email>hagusfa@gmail.com</email>  
    <lakcim>  
        <irszam>1020</irszam>  
        <hazszam>11</hazszam>  
        <utca>Pecsét</utca>  
        <varos>Irat</varos>  
    </lakcim>  
</Ugyfelek>  
  
<Ugyfelek u_adoazonosito="u3" k_u="k3">  
    <nev>Paktum Patrik</nev>  
    <email>ppatrik@freemail.com</email>  
    <lakcim>  
        <irszam>6723</irszam>  
        <hazszam>25</hazszam>  
        <utca>Derped</utca>  
        <varos>Rekedt</varos>  
    </lakcim>  
</Ugyfelek>
```



```

Nav
-->
<Nav n_adoszam="n1" k_n="k1">
  <email>idekuld@nav.hu</email>
  <ugyintezo>
    <email>esztikeh@nav.hu</email>
    <telefon>06304672056</telefon>
    <nev>Galamb Eszter</nev>
  </ugyintezo>
  <nyitvatartas>
    <tol>8:00</tol>
    <ig>12:00</ig>
  </nyitvatartas>
</Nav>

<Nav n_adoszam="n2" k_n="k1">
  <email>idekuld@nav.hu</email>
  <ugyintezo>
    <email>pistikehgh@nav.hu</email>
    <telefon>06304623656</telefon>
    <nev>Vad Péter</nev>
  </ugyintezo>
  <nyitvatartas>
    <tol>8:00</tol>
    <ig>12:00</ig>
  </nyitvatartas>
</Nav>

<Nav n_adoszam="n3" k_n="k1">
  <email>idekuld@nav.hu</email>
  <ugyintezo>
    <email>andikah@nav.hu</email>
    <telefon>06307353456</telefon>
    <nev>Emb Andrea</nev>
  </ugyintezo>
  <nyitvatartas>
    <tol>8:00</tol>
    <ig>12:00</ig>
  </nyitvatartas>
</Nav>

```

```

<!--
Oep
-->
<Oep o_adoszam="o1" k_o="k1">
  <email>errekuld@oep.hu</email>
  <ugyintezo>
    <email>ronaldinho@oep.hu</email>
    <telefon>06705684050</telefon>
    <nev>Ricardo Ronaldo</nev>
  </ugyintezo>
  <nyitvatartas>
    <tol>7:00</tol>
    <ig>16:00</ig>
  </nyitvatartas>
</Oep>

<Oep o_adoszam="o2" k_o="k2">
  <email>errekuld@oep.hu</email>
  <ugyintezo>
    <email>inez@oep.hu</email>
    <telefon>06707357246</telefon>
    <nev>Kardos Inez</nev>
  </ugyintezo>
  <nyitvatartas>
    <tol>8:00</tol>
    <ig>16:00</ig>
  </nyitvatartas>
</Oep>

<Oep o_adoszam="o3" k_o="k3">
  <email>errekuld@oep.hu</email>
  <ugyintezo>
    <email>hemb.sanyi@oep.hu</email>
    <telefon>06709741456</telefon>
    <nev>Hemb Sándor</nev>
  </ugyintezo>
  <nyitvatartas>
    <tol>7:00</tol>
    <ig>16:00</ig>
  </nyitvatartas>
</Oep>

```

```

<!--
    Onkormanyzatok
-->
<Onkormanyzat on_adoszam="on1">
    <email>kulddide@onk.hu</email>
    <ugyintezo>
        <email>purhab@onk.hu</email>
        <telefon>06209782132</telefon>
        <nev>Pur Hab</nev>
    </ugyintezo>
    <nyitvatartas>
        <tol>9:00</tol>
        <ig>13:30</ig>
    </nyitvatartas>
</Onkormanyzat>

<Onkormanyzat on_adoszam="on2">
    <email>kulddide@onk.hu</email>
    <ugyintezo>
        <email>tali.cska@onk.hu</email>
        <telefon>06203145469</telefon>
        <nev>Tali Cska</nev>
    </ugyintezo>
    <nyitvatartas>
        <tol>9:00</tol>
        <ig>13:30</ig>
    </nyitvatartas>
</Onkormanyzat>

<Onkormanyzat on_adoszam="on3">
    <email>kulddide@onk.hu</email>
    <ugyintezo>
        <email>szekereszter@onk.hu</email>
        <telefon>06207132875</telefon>
        <nev>Szeker Eszter</nev>
    </ugyintezo>
    <nyitvatartas>
        <tol>9:00</tol>
        <ig>13:30</ig>
    </nyitvatartas>
</Onkormanyzat>

```

```

<!--
    Konyvelo iroda és az onkormányzat közötti több-több kapcsolat segéd példányai
-->
<Konyv_Onk k_on="on1" k_k="k1">
    <dok.id>254123</dok.id>
    <dok.szam>3</dok.szam>
</Konyv_Onk>

<Konyv_Onk k_on="on2" k_k="k2">
    <dok.id>673773</dok.id>
    <dok.szam>5</dok.szam>
</Konyv_Onk>

<Konyv_Onk k_on="on3" k_k="k3">
    <dok.id>982515</dok.id>
    <dok.szam>1</dok.szam>
</Konyv_Onk>

```

## 1d) XMLSchema

Az XML dokumentum alapján egy XMLSchemat készítettem amelyben definiálódnak az XML-ben található elementek. Saját típusokat definiáltam ezenfelül pedig kigyűjtve vannak megírva a definiálások és az elementek kialakításai. Az attribútumokra ref-ekkel hivatkozok.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Tulaj" type="TulajType"/>
  <!--
    Konyveloiroda element létrehozása
  -->
  <xs:complexType name="KonyveloirodaType">
    <xs:sequence>
      <xs:element type="szekhelyType" name="szekhely"/>
      <xs:element type="xs:string" name="ugyvezeto"/>
    </xs:sequence>
    <xs:attribute ref="k_adoszam" use="required" />
  </xs:complexType>
  <!--
    Ceg element létrehozása
  -->
  <xs:complexType name="CegType">
    <xs:sequence>
      <xs:element type="xs:string" name="email"/>
      <xs:element type="xs:string" name="szekhely"/>
      <xs:element type="xs:short" name="alapitas_eve"/>
    </xs:sequence>
    <xs:attribute ref="nev" use="required" />
    <xs:attribute ref="u_c" use="required" />
  </xs:complexType>
  <!--
    Ugyfelek element létrehozása
  -->
  <xs:complexType name="UgyfelekType">
    <xs:sequence>
      <xs:element type="xs:string" name="nev"/>
      <xs:element type="xs:string" name="email"/>
      <xs:element type="lakcimType" name="lakcim"/>
    </xs:sequence>
    <xs:attribute ref="u_adoazonosito" use="required" />
    <xs:attribute ref="k_u" use="required" />
  </xs:complexType>
```

```

<!--
  Nav element létrehozása
-->
<xs:complexType name="NavType">
  <xs:sequence>
    <xs:element type="xs:string" name="email"/>
    <xs:element type="ugyintezoType" name="ugyintezo"/>
    <xs:element type="nyitvatartasType" name="nyitvatartas"/>
  </xs:sequence>
  <xs:attribute ref="n_adoszam" use="required" />
  <xs:attribute ref="k_n" use="required" />
</xs:complexType>

<!--
  Oep element létrehozása
-->
<xs:complexType name="OepType">
  <xs:sequence>
    <xs:element type="xs:string" name="email"/>
    <xs:element type="ugyintezoType" name="ugyintezo"/>
    <xs:element type="nyitvatartasType" name="nyitvatartas"/>
  </xs:sequence>
  <xs:attribute ref="o_adoszam" use="required" />
  <xs:attribute ref="k_o" use="required" />
</xs:complexType>

```

```

<!--
  Onkormanyzat element létrehozása
-->
<xs:complexType name="OnkormanyzatType">
  <xs:sequence>
    <xs:element type="xs:string" name="email"/>
    <xs:element type="ugyintezoType" name="ugyintezo"/>
    <xs:element type="nyitvatartasType" name="nyitvatartas"/>
  </xs:sequence>
  <xs:attribute ref="on_adoszam" use="required" />
</xs:complexType>

<!-- Konyviroda-Onkormanyzat kapcs. tábla -->
<xs:complexType name="Konyv_OnkType">
  <xs:sequence>
    <xs:element type="xs:int" name="dok.id"/>
    <xs:element type="xs:byte" name="dok.szam"/>
  </xs:sequence>
  <xs:attribute ref="k_on" use="required" />
  <xs:attribute ref="k_k" use="required" />
</xs:complexType>

```

```
<!--  
  Attribútumok kialakítása  
-->  
<!-- Konyviroda attr. -->  
<xs:attribute type="xs:string" name="k_adoszam"/>  
  
<!-- Ceg attr. -->  
<xs:attribute type="xs:string" name="nev"/>  
<xs:attribute type="xs:string" name="u_c"/>  
  
<!-- Ugyfelek attr. -->  
<xs:attribute type="xs:string" name="u_adoazonosito"/>  
<xs:attribute type="xs:string" name="k_u"/>  
  
<!-- Nav attr. -->  
<xs:attribute type="xs:string" name="n_adoszam"/>  
<xs:attribute type="xs:string" name="k_n"/>  
  
<!-- Oep attr. -->  
<xs:attribute type="xs:string" name="o_adoszam"/>  
<xs:attribute type="xs:string" name="k_o"/>  
  
<!-- Onkormanyzat attr. -->  
<xs:attribute type="xs:string" name="on_adoszam"/>  
  
<!-- Konyv_Onk attr. -->  
<xs:attribute type="xs:string" name="k_on"/>  
<xs:attribute type="xs:string" name="k_k"/>
```

```

<!--
  Complex típusu elemek definiálása
-->
<xs:complexType name="szekhelyType">
  <xs:sequence>
    <xs:element type="xs:short" name="irszam"/>
    <xs:element type="xs:byte" name="hazszam"/>
    <xs:element type="xs:string" name="utca"/>
    <xs:element type="xs:string" name="varos"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="lakcimType">
  <xs:sequence>
    <xs:element type="xs:short" name="irszam"/>
    <xs:element type="xs:short" name="hazszam"/>
    <xs:element type="xs:string" name="utca"/>
    <xs:element type="xs:string" name="varos"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ugyintezoType">
  <xs:sequence>
    <xs:element type="xs:string" name="email"/>
    <xs:element type="xs:long" name="telefon"/>
    <xs:element type="xs:string" name="nev"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="nyitvatartasType">
  <xs:sequence>
    <xs:element type="xs:string" name="tol"/>
    <xs:element type="xs:string" name="ig"/>
  </xs:sequence>
</xs:complexType>

```

```

<!--
  Egyedi típusok meghatározása
-->
<xs:complexType name="TulajType">
  <xs:sequence>

    <xs:element type="KonyveloirodaType" name="Konyveloiroda" maxOccurs="unbounded" minOccurs="3">
    </xs:element>

    <xs:element type="CegType" name="Ceg" maxOccurs="unbounded" minOccurs="3">
    </xs:element>

    <xs:element type="UgyfelekType" name="Ugyfelek" maxOccurs="unbounded" minOccurs="3">
    </xs:element>

    <xs:element type="NavType" name="Nav" maxOccurs="unbounded" minOccurs="3">
    </xs:element>

    <xs:element type="OepType" name="Oep" maxOccurs="unbounded" minOccurs="3">
    </xs:element>

    <xs:element type="OnkormanyzatType" name="Onkormanyzat" maxOccurs="unbounded" minOccurs="3">
    </xs:element>

    <xs:element type="Konyv_OnkType" name="Konyv_Onk" maxOccurs="unbounded" minOccurs="3">
    </xs:element>

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

```

```

<!--
  Elsodleges kulcsok
-->
<xs:key name="Konyveliroda_kulcs">
  <xs:selector xpath="Konyveloiroda"></xs:selector>
  <xs:field xpath="@k_adoszam"></xs:field>
</xs:key>

<xs:key name="Onkormanyzat_kulcs">
  <xs:selector xpath="Onkormanyzat"></xs:selector>
  <xs:field xpath="@on_adoszam"></xs:field>
</xs:key>

<xs:key name="Ugyfelek_kulcs">
  <xs:selector xpath="Ugyfelek"></xs:selector>
  <xs:field xpath="@u_adoazonosito"></xs:field>
</xs:key>

<!--
  Idegen kulcsok
-->
<xs:keyref refer="Ugyfelek_kulcs" name="Ugyfelek_ceg_f_key">
  <xs:selector xpath="Ugyfelek"></xs:selector>
  <xs:field xpath="@u_c"></xs:field>
</xs:keyref>

<xs:keyref refer="Konyveliroda_kulcs" name="konyveloiroda_ugyfel_f_key">
  <xs:selector xpath="Konyveloiroda"></xs:selector>
  <xs:field xpath="@k_u"></xs:field>
</xs:keyref>

<xs:keyref refer="Konyveliroda_kulcs" name="konyveloiroda_nav_f_key">
  <xs:selector xpath="Konyveloiroda"></xs:selector>
  <xs:field xpath="@k_n"></xs:field>
</xs:keyref>

<xs:keyref refer="Konyveliroda_kulcs" name="konyveloiroda_oep_f_key">
  <xs:selector xpath="Konyveloiroda"></xs:selector>
  <xs:field xpath="@k_o"></xs:field>
</xs:keyref>

```

```

<!--
  Konyveloiroda és az onkormányzat közötti több-több kapcs. kialakítása
-->
<xs:keyref refer="Onkormanyzat_kulcs" name="Onkormanyzat_konyveloiroda_f_key">
  <xs:selector xpath="Konyv_Onk"></xs:selector>
  <xs:field xpath="@k_on"></xs:field>
</xs:keyref>

<xs:keyref refer="Konyveliroda_kulcs" name="konyveloiroda_onkormanyzat_f_key">
  <xs:selector xpath="Konyv_Onk"></xs:selector>
  <xs:field xpath="@k_k"></xs:field>
</xs:keyref>

```

## 2.feladat

### 2a) DOMRead

Az XML fájlból minden elementet kiolvas. Mindegyik element kiolvasására külön metódus van amelyek a mainbe vannak meghívva. A kiolvasott adatokat egy .txt fájlba írja ki.

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.xml.parsers.ParserConfigurationException;
import org.xml.sax.SAXException;

public class DOMReadQVTQ08 {

    private static Document doc;

    Run | Debug
    public static void main(String argv[]) throws SAXException,
        IOException, ParserConfigurationException {

        File xmlFile = new File(
            pathname: "./XMLQVTQ08.xml");

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
    }
}
```

```
// Konyveloirodak kiirasa
readKonyveloIroda();

// Cegek kiiratasa
readCegek();

// Ugyfelek kiiratasa
readUgyfelek();

// Nav kiiratasa
readNav();

// Oep kiiratasa
readOep();

// Onkormanyzat kiiratasa
readOnkormanyzat();

// Konyvelesi anyagok kiiratasa
readKonyv_Onk();
```



```

private static void readKonyv_Onk() {
    NodeList konyvanyagokList = doc.getElementsByTagName(tagname: "Konyv_Onk");

    for (int i = 0; i < konyvanyagokList.getLength(); i++) {

        Node nNode = konyvanyagokList.item(i);

        System.out.println("\nElement: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            Element elem = (Element) nNode;

            Node node1 = elem.getElementsByTagName(name: "dok.id").item(index: 0);
            String dokId = node1.getTextContent();

            Node node2 = elem.getElementsByTagName(name: "dok.szam").item(index: 0);
            String dokSzam = node2.getTextContent();

            System.out.printf(format: "Dokumentum Id: %s\n", dokId);
            System.out.printf(format: "Dokumentumok Szama: %s\n", dokSzam);

            // Elementek es ertekeik kiirasa a fajlba
            try (FileWriter fw = new FileWriter(fileName: "DOMRead.txt", append: true)); {

                // fajlba iras megkezdése
                fw.write(str: "Konyvelesi Anyag:-- \n");
                fw.write("Dokumentum Id:" + dokId + "\n");
                fw.write("Dokumentumok szama:" + dokSzam + "\n\n");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

private static void readOnkormanyzat() {
NodeList onkormanyzatList = doc.getElementsByTagName(tagname: "Onkormanyzat");

for (int i = 0; i < onkormanyzatList.getLength(); i++) {

    Node nNode = onkormanyzatList.item(i);

    System.out.println("\nElement: " + nNode.getNodeName());

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element elem = (Element) nNode;

        String uid = elem.getAttribute(name: "on_adoszam");

        Node node1 = elem.getElementsByTagName(name: "email").item(index: 0);
        String email = node1.getTextContent();

        System.out.printf(format: "Adoszam: %s\n", uid);
        System.out.printf(format: "Email: %s\n", email);
        System.out.printf("Ugyintezo: " +
            "%n\tEmail: " + elem.getElementsByTagName(name: "email").item(index: 0).getTextContent() +
            "%n\tTelefon: " + elem.getElementsByTagName(name: "telefon").item(index: 0).getTextContent() +
            "%n\tNev: " + elem.getElementsByTagName(name: "nev").item(index: 0).getTextContent() + "\n");
        System.out.printf("Nyitvatartas: " +
            "%n\t" + elem.getElementsByTagName(name: "tol").item(index: 0).getTextContent() + "-tol." +
            "%n\t" + elem.getElementsByTagName(name: "ig").item(index: 0).getTextContent() + "-ig." + "\n");

        // Elementek es ertekeik kiirasa a fajlba
        try (FileWriter fw = new FileWriter(fileName: "DOMRead.txt", append: true);) {

            // fajlba iras megkezdese
            fw.write(str: "Onkormanyzat:----- \n");
            fw.write("Adoszam: " + uid + "\n");
            fw.write("Email: \n" + email + "\n");
            fw.write("Ugyintezo: \n" +
                "\tIrszam: " + elem.getElementsByTagName(name: "email").item(index: 0).getTextContent() + "\n" +
                "\tHazzsam: " + elem.getElementsByTagName(name: "telefon").item(index: 0).getTextContent() + "\n" +
                "\tUtca: " + elem.getElementsByTagName(name: "nev").item(index: 0).getTextContent() + "\n");
            fw.write("Nyitvatartas: \n" +
                "\t" + elem.getElementsByTagName(name: "tol").item(index: 0).getTextContent() + "-tol." + "\n" +
                "\t" + elem.getElementsByTagName(name: "ig").item(index: 0).getTextContent() + "-ig." + "\n\n");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
}

```

```

private static void readCegek() {
    NodeList cegList = doc.getElementsByTagName(tagname: "Ceg");

    for (int i = 0; i < cegList.getLength(); i++) {

        Node nNode = cegList.item(i);

        System.out.println("\nElement: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            Element elem = (Element) nNode;

            String nev = elem.getAttribute(name: "nev");

            Node node1 = elem.getElementsByTagName(name: "email").item(index: 0);
            String email = node1.getTextContent();

            Node node2 = elem.getElementsByTagName(name: "szekhely").item(index: 0);
            String szekhely = node2.getTextContent();

            Node node3 = elem.getElementsByTagName(name: "alapitas_eve").item(index: 0);
            String alapitas_eve = node3.getTextContent();

            System.out.printf(format: "Nev: %s\n", nev);
            System.out.printf(format: "Email: %s\n", email);
            System.out.printf(format: "Szekhely: %s\n", szekhely);
            System.out.printf(format: "Alapitas: %s\n", alapitas_eve);

            // Elementek es ertekeik kiirasa a fajlba
            try (FileWriter fw = new FileWriter(fileName: "DOMRead.txt", append: true);) {

                // fajlba iras megkezdése
                fw.write(str: "Ceg:----- \n");
                fw.write("Nev: " + nev + "\n");
                fw.write("Email: \n" + email + "\n");
                fw.write("Szekhely: \n" + szekhely + "\n");
                fw.write("Alapitas: " + alapitas_eve + "\n\n");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

private static void readOep() {
    NodeList oepList = doc.getElementsByTagName(tagname: "Oep");

    for (int i = 0; i < oepList.getLength(); i++) {

        Node nNode = oepList.item(i);

        System.out.println("\nElement: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            Element elem = (Element) nNode;

            String uid = elem.getAttribute(name: "o_adoszam");

            Node node1 = elem.getElementsByTagName(name: "email").item(index: 0);
            String email = node1.getTextContent();

            System.out.printf(format: "Adoszam: %s\n", uid);
            System.out.printf(format: "Email: %s\n", email);
            System.out.printf("Ugyintezo: " +
                "%n\tEmail::" + elem.getElementsByTagName(name: "email").item(index: 0).getTextContent() +
                "%n\tTelefon:" + elem.getElementsByTagName(name: "telefon").item(index: 0).getTextContent() +
                "%n\tNev:" + elem.getElementsByTagName(name: "nev").item(index: 0).getTextContent() + "\n");
            System.out.printf("Nyitvatartas: " +
                "%n\t" + elem.getElementsByTagName(name: "tol").item(index: 0).getTextContent() + "-tol." +
                "%n\t" + elem.getElementsByTagName(name: "ig").item(index: 0).getTextContent() + "-ig." + "\n");

            // Elementek es ertekeik kiirasa a fajlba
            try (FileWriter fw = new FileWriter(fileName: "DOMRead.txt", append: true);) {

                // fajlba iras megkezdése
                fw.write(str: "Oep:----- \n");
                fw.write("Adoszam:" + uid + "\n");
                fw.write("Email: \n" + email + "\n");
                fw.write("Ugyintezo: \n" +
                    "\tIrszam:" + elem.getElementsByTagName(name: "email").item(index: 0).getTextContent() + "\n" +
                    "\tHazzsam:" + elem.getElementsByTagName(name: "telefon").item(index: 0).getTextContent() + "\n" +
                    "\tUtca:" + elem.getElementsByTagName(name: "nev").item(index: 0).getTextContent() + "\n");
                fw.write("Nyitvatartas: \n" +
                    "\t" + elem.getElementsByTagName(name: "tol").item(index: 0).getTextContent() + "-tol" + "\n" +
                    "\t" + elem.getElementsByTagName(name: "ig").item(index: 0).getTextContent() + "-ig" + "\n\n");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

private static void readKonyveloIroda() {
    NodeList kirolaList = doc.getElementsByTagName(tagname: "Konyveloiroda");

    for (int i = 0; i < kirolaList.getLength(); i++) {

        Node nNode = kirolaList.item(i);

        System.out.println("\nElement: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            Element elem = (Element) nNode;

            String uid = elem.getAttribute(name: "k_adoszam");

            Node node2 = elem.getElementsByTagName(name: "ugyvezeto").item(index: 0);
            String uvezeto = node2.getTextContent();

            System.out.printf(format: "Adoszam: %s\n", uid);
            System.out.printf("Szekhely: " +
                "%n\tIrszam:" + elem.getElementsByTagName(name: "irszam").item(index: 0).getTextContent() +
                "%n\tHazzsam:" + elem.getElementsByTagName(name: "hazzsam").item(index: 0).getTextContent() +
                "%n\tUtca:" + elem.getElementsByTagName(name: "utca").item(index: 0).getTextContent() +
                "%n\tVaros:" + elem.getElementsByTagName(name: "varos").item(index: 0).getTextContent());
            System.out.printf(format: "%nUgyvezeto: %s\n", uvezeto);

            // Elementek es ertekeik kiirasa a fajlba
            try (FileWriter fw = new FileWriter(fileName: "DOMRead.txt", append: true);) {

                // fajlba iras megkezdése
                fw.write(str: "Konyveloiroda:----- \n");
                fw.write("Adoszam: " + uid + "\n");
                fw.write("Szekhely: \n" +
                    "\tIrszam:" + elem.getElementsByTagName(name: "irszam").item(index: 0).getTextContent() + "\n" +
                    "\tHazzsam:" + elem.getElementsByTagName(name: "hazzsam").item(index: 0).getTextContent() + "\n" +
                    "\tUtca:" + elem.getElementsByTagName(name: "utca").item(index: 0).getTextContent() + "\n" +
                    "\tVaros:" + elem.getElementsByTagName(name: "varos").item(index: 0).getTextContent() + "\n");
                fw.write("Ugyvezeto: " + uvezeto + "\n\n");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

private static void readUgyfelek() {
    NodeList ugyfelekList = doc.getElementsByTagName(tagname: "Ugyfelek");

    for (int i = 0; i < ugyfelekList.getLength(); i++) {

        Node nNode = ugyfelekList.item(i);

        System.out.println("\nElement: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            Element elem = (Element) nNode;

            String uid = elem.getAttribute(name: "u_adoazonosito");

            Node node1 = elem.getElementsByTagName(name: "nev").item(index: 0);
            String nev = node1.getTextContent();

            Node node2 = elem.getElementsByTagName(name: "email").item(index: 0);
            String email = node2.getTextContent();

            System.out.printf(format: "Adoszam: %s\n", uid);
            System.out.printf(format: "Nev: %s\n", nev);
            System.out.printf(format: "Email: %s\n", email);
            System.out.printf("Lakcim:" +
                "\n\tIrszam:" + elem.getElementsByTagName(name: "irszam").item(index: 0).getTextContent() +
                "\n\tHazszam:" + elem.getElementsByTagName(name: "hazszam").item(index: 0).getTextContent() +
                "\n\tUtca:" + elem.getElementsByTagName(name: "utca").item(index: 0).getTextContent() +
                "\n\tVaros:" + elem.getElementsByTagName(name: "varos").item(index: 0).getTextContent() + "\n");

            // Elementek es ertekeik kiirasa a fajlba
            try (FileWriter fw = new FileWriter(fileName: "DOMRead.txt", append: true);) {

                // fajlba iras megkezdese
                fw.write(str: "Ugyfelek:----- \n");
                fw.write("Adoszam:" + uid + "\n");
                fw.write("Nev: " + nev + "\n");
                fw.write("Email: \n" + email + "\n");
                fw.write("Lakcim:" +
                    "\n\tIrszam:" + elem.getElementsByTagName(name: "irszam").item(index: 0).getTextContent() + "\n" +
                    "\n\tHazszam:" + elem.getElementsByTagName(name: "hazszam").item(index: 0).getTextContent() + "\n" +
                    "\n\tUtca:" + elem.getElementsByTagName(name: "utca").item(index: 0).getTextContent() + "\n" +
                    "\n\tVaros:" + elem.getElementsByTagName(name: "varos").item(index: 0).getTextContent() + "\n\n");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

private static void readNav() {
    NodeList navList = doc.getElementsByTagName(tagname: "Nav");

    for (int i = 0; i < navList.getLength(); i++) {

        Node nNode = navList.item(i);

        System.out.println("\nElement: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            Element elem = (Element) nNode;

            String uid = elem.getAttribute(name: "n_adoszam");

            Node node1 = elem.getElementsByTagName(name: "email").item(index: 0);
            String email = node1.getTextContent();

            System.out.printf(format: "Adoszam: %s\n", uid);
            System.out.printf(format: "Email: %s\n", email);
            System.out.printf("Ugyintezo: " +
                "%n\tEmail: " + elem.getElementsByTagName(name: "email").item(index: 0).getTextContent() +
                "%n\tTelefon: " + elem.getElementsByTagName(name: "telefon").item(index: 0).getTextContent() +
                "%n\tNev: " + elem.getElementsByTagName(name: "nev").item(index: 0).getTextContent() + "\n");
            System.out.printf("Nyitvatartas: " +
                "%n\t" + elem.getElementsByTagName(name: "tol").item(index: 0).getTextContent() + "-tol." +
                "%n\t" + elem.getElementsByTagName(name: "ig").item(index: 0).getTextContent() + "-ig." + "\n");

            // Elementek es ertekeik kiirasa a fajlba
            try (FileWriter fw = new FileWriter(fileName: "DOMRead.txt", append: true);) {

                // fajlba iras megkezdese
                fw.write(str: "Nav:----- \n");
                fw.write("Adoszam: " + uid + "\n");
                fw.write("Email: \n" + email + "\n");
                fw.write("Ugyintezo: \n" +
                    "\tIrszam: " + elem.getElementsByTagName(name: "email").item(index: 0).getTextContent() + "\n" +
                    "\tHazszam: " + elem.getElementsByTagName(name: "telefon").item(index: 0).getTextContent() + "\n" +
                    "\tUtca: " + elem.getElementsByTagName(name: "nev").item(index: 0).getTextContent() + "\n");
                fw.write("Nyitvatartas: \n" +
                    "\t" + elem.getElementsByTagName(name: "tol").item(index: 0).getTextContent() + "-tol" + "\n" +
                    "\t" + elem.getElementsByTagName(name: "ig").item(index: 0).getTextContent() + "-ig" + "\n\n");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

## 2b) DOMModify

Az XML fájl tetszőleges adatait vagy akár egy element összes azonos tagját módosítottam. Ezeket a módosításokat kiírom konzolra majd az egész módosított XML dokumentumot lementem egy külön modified.xml fájlba.

```
public class DOMModifyQVTQ08 {  
    Run | Debug  
    public static void main(String argv[]) throws SAXException,  
        IOException, ParserConfigurationException {  
        File xmlFile = new File(  
            pathname: "./XMLQVTQ08.xml");  
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
        DocumentBuilder dBuilder = factory.newDocumentBuilder();  
        Document doc = dBuilder.parse(xmlFile);  
  
        //Második ügyfel azonosító attributóinak módosítása  
        NodeList ügyfelekList = doc.getElementsByTagName(tagname: "Ugyfelek");  
        Node ügyfel = ügyfelekList.item(index: 1);  
        ügyfel.getAttributes().getNamedItem(name: "u_adoazonosito").setTextContent(textContent: "ua2");  
  
        //-----  
  
        //Könyvelő iroda minden utca element értékeinek módosítása  
        NodeList kirodalist = doc.getElementsByTagName(tagname: "Konyveloiroda");  
        modifyElement(kirodalist, tagName: "utca", newValue: "Abla");  
  
        //Módosított elemek kiírása konzolra  
        for (int i = 0; i < kirodalist.getLength(); i++) {  
            Node nNode = kirodalist.item(i);  
            System.out.println("\nElement: " + nNode.getNodeName());  
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {  
                Element elem = (Element) nNode;  
                String uid = elem.getAttribute(name: "k_adoszam");  
                Node node2 = elem.getElementsByTagName(name: "ugyvezeto").item(index: 0);  
                String uvezeto = node2.getTextContent();  
                System.out.printf(format: "Adoszam: %s\n", uid);  
                System.out.printf("Szekhely: " +  
                    "%n\tIrszam:" + elem.getElementsByTagName(name: "irszam").item(index: 0).getTextContent() +  
                    "%n\tHatszam:" + elem.getElementsByTagName(name: "hatszam").item(index: 0).getTextContent() +  
                    "%n\tUtca:" + elem.getElementsByTagName(name: "utca").item(index: 0).getTextContent() +  
                    "%n\tVaros:" + elem.getElementsByTagName(name: "varos").item(index: 0).getTextContent());  
                System.out.printf(format: "%nUgyvezeto: %s\n \n", uvezeto);  
            }  
        }  
    }  
}
```

```

//Uj gyermek element hozzáadása a 3.Ugyfel elementhez
Node ugyfelTel = ugyfelekList.item(index: 2);

Element telefonszam = doc.createElement(tagName: "telefonszam");
telefonszam.appendChild(doc.createTextNode(data: "06205672543"));
ugyfelTel.appendChild(telefonszam);

//Modositott Ugyfel elem kiirasa
System.out.println("Ugyfel:" + ugyfelekList.item(index: 2).getTextContent() + "\n");

//-----

```

```

//Nav element ugyintezo elementjének gyerek 'email' elementjének átnevezése
NodeList navList = doc.getElementsByTagName(tagname: "Nav");

for (int i = 0; i < navList.getLength(); i++) {
    Node nav = navList.item(i);

    Element elem = (Element) nav;

    //Nav ugyintezo elementjének lekerdezese
    Node childs = elem.getElementsByTagName(name: "ugyintezo").item(index: 0);

    //az ugyintezo element gyerek elementjeinek lekerdezese
    NodeList childNodes = childs.getChildNodes();

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

        if (item.getNodeType() == Node.ELEMENT_NODE) {
            if ("email".equalsIgnoreCase(item.getNodeName())) {
                String email = item.getTextContent();

                //email element torlése
                childs.removeChild(item);

                //Uj xml element hozzáadása
                Element n = doc.createElement(tagName: "uemail");
                n.appendChild(doc.createTextNode(email));

                childs.appendChild(n);
            }
        }
    }
}

```



```

//Oep email elementjének törlése
NodeList oepList = doc.getElementsByTagName(tagname: "Oep");

for (int i = 0; i < oepList.getLength(); i++) {
    Node oep = oepList.item(i);

    NodeList childNodes = oep.getChildNodes();

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

        Node item = childNodes.item(j);

        if (item.getNodeType() == Node.ELEMENT_NODE) {

            if ("email".equalsIgnoreCase(item.getNodeName())) {

                //email element torlése
                oep.removeChild(item);

            }

        }

    }
}

```

```

        try (FileOutputStream output = new FileOutputStream(
            name: "modified.xml")) {
            writeXml(doc, output);
        } catch (TransformerException e) {
            e.printStackTrace();
        }

    }
}

```

```

//Modositott xml fajl megirása
private static void writeXml(Document doc, OutputStream output)
    throws TransformerException, UnsupportedEncodingException {

    TransformerFactory transformerFactory = TransformerFactory.newInstance();

    Transformer transformer = transformerFactory.newTransformer(
        new StreamSource(new File(pathname: "DOMParseQVTQ08/styling.xslt")));

    transformer.setOutputProperty(OutputKeys.INDENT, value: "yes");
    transformer.setOutputProperty(OutputKeys.STANDALONE, value: "no");

    DOMSource source = new DOMSource(doc);
    StreamResult result = new StreamResult(output);

    transformer.transform(source, result);

}

```

## 2c) DOMQuery

XML fájlból kérek le adatokat xPath alkalmazásával.

```
public class DOMQueryQVTQ08 {

    private static XPath xPath = XPathFactory.newInstance().newXPath();
    private static Document document;

    Run | Debug
    public static void main(String[] args) {

        try {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder;

            dBuilder = dbFactory.newDocumentBuilder();

            document = dBuilder.parse(uri: "./XMLQVTQ08.xml");

            document.getDocumentElement().normalize();

            //Minden konyveloiroda lekérdezése
            queryAllKonyveloIroda();

            //Lekérés id alapján
            queryKonyveloIrodaById(value: "k2");

            //név alapján lekérdezés
            ugyfelekQueryByName(name: "Higo Hugo");

            //Irszam es varos alapjan lekerdezes
            ugyfelekQueryByIrszamAndVaros(irszam: "6723",varos: "Rekedt");

            //7-től nyitva levő oep lekerdezes
            oepQueryByOpening(opening: "7:00");
```

```
        } catch (ParserConfigurationException e) {
            e.printStackTrace();
        } catch (SAXException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
// Az osszes Konyveloiroda lekerdezes
public static NodeList queryAllKonyveloIroda() {
    String expression = String.format(format: "/Tulaj/Konyveloiroda");
    try {

        System.out.printf(format: "Konyveloirodak lekerdezes: \n");

        NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(document, XPathConstants.NODESET);
```

```
// Konyveloiroda lekerdezese id alapjan
public static NodeList queryKonyveloirodaById(String value) {
    String id = "k_adoszam";

    String expression = String.format(format: "/Tulaj/Konyveloiroda[@%s = '%s']", id, value);
    try {
        System.out.printf(format: "-----\n\n");
        System.out.printf(format: "Id alapjan lekerdezés: \n");

        NodeList nodeList = (NodeList) XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
    }
}
```

```
//Ugyfelek lekerdezese név alapjan
public static NodeList ugyfelekQueryByName(String name) {

    String expression = String.format(format: "/Tulaj/Ugyfelek[nev='%s']", name);
    try {
        System.out.printf(format: "-----\n\n");
        System.out.printf(format: "Nev alapjan lekerdezés: \n");

        NodeList nodeList = (NodeList) XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
    }
}
```

```
//Ugyfelek lekerdezese irszam es varos alapjan
public static NodeList ugyfelekQueryByIrszamAndVaros(String irszam,String varos) {

    String expression = String.format(format: "/Tulaj/Ugyfelek[lacim/irszam='%s' and lacim/varos='%s']", irszam,varos);
    try {
        System.out.printf(format: "-----\n\n");
        System.out.printf(format: "Irszam es varos alapjan lekerdezés: \n");

        NodeList nodeList = (NodeList) XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
    }
}
```

```
//7től nyitva levő oep lekerdezése
public static NodeList oepQueryByOpening(String opening) {

    String expression = String.format(format: "/Tulaj/Oep[nyitvatartas/tol='%s']", opening);
    try {
        System.out.printf(format: "-----\n\n");
        System.out.printf(format: "7-tol nyitva levo Oep lekérdezése: \n");

        NodeList nodeList = (NodeList) XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
    }
}
```