

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Nemcsik Dániel**

Neptunkód: **I3I4BP**

## Feladat leírás:

A feladatomban egy könyvtár működését mutatnám be. Egy könyvhöz tartozik szerző, könyvesbolt, kiadó és vásárló, mint ahogy az ER modellben és XDM modellbe is látszik majd. Egy könyvet egy szerzőhöz köthetünk (1:1 kapcsolat), egy könyvet több kiadó is kiadhat (1:n kapcsolat), több könyvet több könyvesbolt is megvehet illetve eladhat (n:n kapcsolat), illetve több könyvesboltból több vásárló is vásárolhat (n:n kapcsolat). Tudom, hogy ebbe a logikába bele lehetne kötni, például egy könyvnek több szerzője is lehet stb..., de annak érdekében lett így elkészítve, hogy mindenféle kapcsolat szerepeljen a feladatban.

Az XDM modellben nem szerepelnek olyan adatok, amiket az ER modellben kapcsolatokkal valósítottam meg (pl.: beszerez, eladás kapcsolat, bÁr, eDarab, stb.), de szerintem ettől függetlenül is sikerült egy értelmes XML file-t létrehoznom.

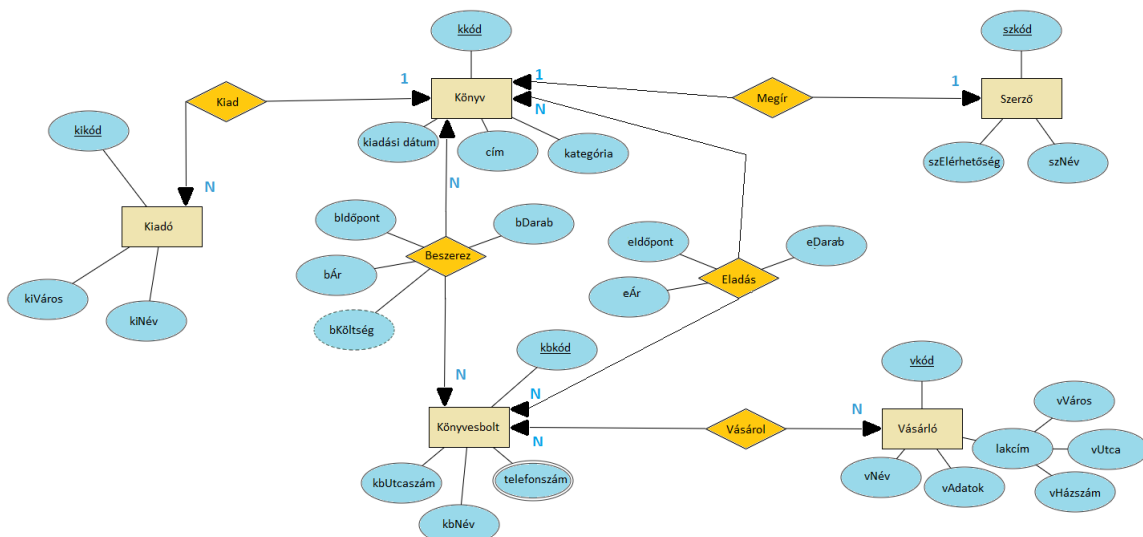
[https://github.com/daninemcsik/I3I4BP\\_XMLGyak](https://github.com/daninemcsik/I3I4BP_XMLGyak)

A feladatomat két külön mappába töltöttem fel, egyik az XMLTaskI3I4BP mappa, amibe az xml, xsd, képek és ez a jegyzőkönyv található a másik pedig a DOMParserI3I4BP, amibe a java dolgok találhatók, illetve az xml-ről egy másolat a könnyebb elérés érdekében.

## 1.feladat

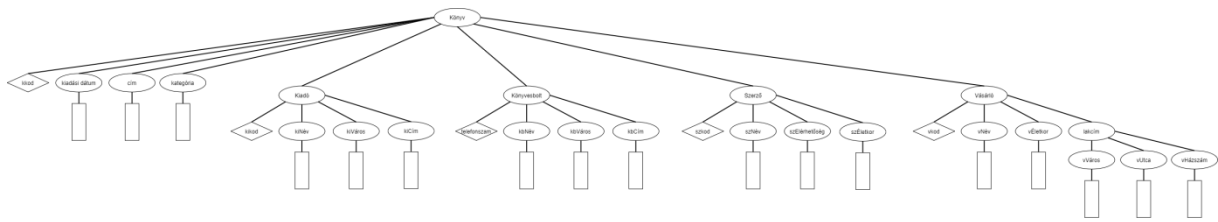
### 1a)

ER modell:



### 1b)

## XDM modell:



1c)

```
<?xml version="1.0" encoding="UTF-8"?>
<Könyvek xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="http://www.example.com/XMLSchemaI3I48P.xsd">

  <Könyv kkod="01">
    <kiadási_dátum>03/15/2019</kiadási_dátum>
    <cím>Senki sem az, akinek látszik</cím>
    <kategória>Irodalom</kategória>

    <Szerző szkod="01">
      <szElérhetőség>305256314</szElérhetőség>
      <szNév>Kerry Fisher</szNév>
      <szÉletkor>60</szÉletkor>
    </Szerző>

    <Kiadó kikod="01">
      <kiVáros>Szeged</kiVáros>
      <kiCím>Szegedi utca 3.</kiCím>
      <kiNév>Szegedi kiadó</kiNév>
    </Kiadó>

    <Könyvesbolt telefonszam ="704813192">
      <kbVáros>Szeged</kbVáros>
      <kbCím>Vitéz utca 30.</kbCím>
      <kbNév>Szögödi könyvek</kbNév>
    </Könyvesbolt>

    <Vásárló vkod="04">
      <vnév>Kovács Tímea</vnév>
      <vÉletkor>25</vÉletkor>
      <lakcím>
        <vVáros>Szeged</vVáros>
        <vUtca>Zoltán utca</vUtca>
        <vHázszám>44</vHázszám>
      </lakcím>
    </Vásárló>
  </Könyv>

  <Könyv kkod="02">
    <kiadási_dátum>02/10/2018</kiadási_dátum>
    <cím>Nem titok, amit ketten tudnak</cím>
    <kategória>Irodalom</kategória>

    <Szerző szkod="02">
      <szElérhetőség>708312943</szElérhetőség>
      <szNév>Robin Cuddy</szNév>
      <szÉletkor>42</szÉletkor>
    </Szerző>

    <Kiadó kikod="02">
      <kiVáros>Budapest</kiVáros>
      <kiCím>Pesti út 65.</kiCím>
      <kiNév>Pesti kiadó</kiNév>
    </Kiadó>

    <Könyvesbolt telefonszam ="307318491">
      <kbVáros>Budapest</kbVáros>
      <kbCím>Szugló utca 40.</kbCím>
      <kbNév>Book-A-Pest</kbNév>
    </Könyvesbolt>

    <Vásárló vkod="01">
      <vnév>Nagy Ferenc</vnév>
      <vÉletkor>42</vÉletkor>
      <lakcím>
        <vVáros>Budapest</vVáros>
        <vUtca>Lovarda utca</vUtca>
        <vHázszám>22</vHázszám>
      </lakcím>
    </Vásárló>
  </Könyv>

  <Könyv kkod="03">
    <kiadási_dátum>01/11/2020</kiadási_dátum>
    <cím>Tanuljunk rajzolni! - Szafari</cím>
    <kategória>Hobbi</kategória>

    <Szerző szkod="03">
      <szElérhetőség>207318219</szElérhetőség>
      <szNév>Ken Follett</szNév>
      <szÉletkor>72</szÉletkor>
    </Szerző>

    <Kiadó kikod="03">
      <kiVáros>Miskolc</kiVáros>
      <kiCím>Miskolci utca 11.</kiCím>
      <kiNév>Miskolci kiadó</kiNév>
    </Kiadó>

    <Könyvesbolt telefonszam ="207431975">
      <kbVáros>Miskolci</kbVáros>
      <kbCím>Széchenyi utca 20.</kbCím>
      <kbNév>Libri Miskolc</kbNév>
    </Könyvesbolt>

    <Vásárló vkod="05">
      <vnév>Huszár Zsófia</vnév>
      <vÉletkor>33</vÉletkor>
      <lakcím>
        <vVáros>Miskolc</vVáros>
        <vUtca>Huba utca</vUtca>
        <vHázszám>55</vHázszám>
      </lakcím>
    </Vásárló>
  </Könyv>
```

```
<Könyv kkod="04">
  <kiadási_dátum>05/03/2015</kiadási_dátum>
  <cím>Tanuljunk rajzolni! - Házi kedvencek</cím>
  <kategória>Hobbi</kategória>

  <Szerző szkod="04">
    <szElérhetőség>304891398</szElérhetőség>
    <szNév>Fehér Krisztián</szNév>
    <szÉletkor>45</szÉletkor>
  </Szerző>

  <Kiadó kikod="04">
    <kiVáros>Eger</kiVáros>
    <kiCím>Egri út 7.</kiCím>
    <kiNév>Egri kiadó</kiNév>
  </Kiadó>

  <Könyvesbolt telefonszam ="309184751">
    <kbVáros>Eger</kbVáros>
    <kbCím>Sóház utca 11.</kbCím>
    <kbNév>Lira Eger</kbNév>
  </Könyvesbolt>

  <Vásárló vkod="03">
    <vNév>Fekete Roland</vNév>
    <vÉletkor>20</vÉletkor>
    <lakCím>
      <vVáros>Eger</vVáros>
      <vUtca>Gáz utca</vUtca>
      <vHázszám>33</vHázszám>
    </lakCím>
  </Vásárló>
</Könyv>
```

```
<Könyv kkod="06">
  <kiadási_dátum>10/09/2010</kiadási_dátum>
  <cím>The Pillars of The Earth</cím>
  <kategória>Történelem</kategória>

  <Szerző szkod="01">
    <szElérhetőség>305256314</szElérhetőség>
    <szNév>Kerry Fisher</szNév>
    <szÉletkor>60</szÉletkor>
  </Szerző>

  <Kiadó kikod="01">
    <kiVáros>Szeged</kiVáros>
    <kiCím>Szegedi utca 3.</kiCím>
    <kiNév>Szegedi kiadó</kiNév>
  </Kiadó>

  <Könyvesbolt telefonszam ="704813192">
    <kbVáros>Szeged</kbVáros>
    <kbCím>Vitéz utca 30.</kbCím>
    <kbNév>Szögödi könyvek</kbNév>
  </Könyvesbolt>

  <Vásárló vkod="09">
    <vNév>Lakatos Noel</vNév>
    <vÉletkor>50</vÉletkor>
    <lakCím>
      <vVáros>Szeged</vVáros>
      <vUtca>Forró utca</vUtca>
      <vHázszám>99</vHázszám>
    </lakCím>
  </Vásárló>
</Könyv>
```

```
<Könyv kkod="05">
  <kiadási_dátum>08/09/2009</kiadási_dátum>
  <cím>World Without End</cím>
  <kategória>Történelem</kategória>

  <Szerző szkod="05">
    <szElérhetőség>703189418</szElérhetőség>
    <szNév>Stephen King</szNév>
    <szÉletkor>74</szÉletkor>
  </Szerző>

  <Kiadó kikod="05">
    <kiVáros>Szombathely</kiVáros>
    <kiCím>Szombathely utca 13.</kiCím>
    <kiNév>Szombathelyi kiadó</kiNév>
  </Kiadó>

  <Könyvesbolt telefonszam ="701831951">
    <kbVáros>Szombathely</kbVáros>
    <kbCím>Forró utca 3.</kbCím>
    <kbNév>Office Könyvesbolt</kbNév>
  </Könyvesbolt>

  <Vásárló vkod="02">
    <vNév>Kis Tibor</vNév>
    <vÉletkor>30</vÉletkor>
    <lakCím>
      <vVáros>Szombathely</vVáros>
      <vUtca>Ferencsik utca</vUtca>
      <vHázszám>11</vHázszám>
    </lakCím>
  </Vásárló>
</Könyv>
```

```
<Könyv kkod="07">
  <kiadási_dátum>03/11/2020</kiadási_dátum>
  <cím>Programozz C nyelven</cím>
  <kategória>Számítástechnika</kategória>

  <Szerző szkod="02">
    <szElérhetőség>708312943</szElérhetőség>
    <szNév>Robin Cuddy</szNév>
    <szÉletkor>42</szÉletkor>
  </Szerző>

  <Kiadó kikod="02">
    <kiVáros>Budapest</kiVáros>
    <kiCím>Pesti út 65.</kiCím>
    <kiNév>Pesti kiadó</kiNév>
  </Kiadó>

  <Könyvesbolt telefonszam ="307318491">
    <kbVáros>Budapest</kbVáros>
    <kbCím>Szugló utca 40.</kbCím>
    <kbNév>Book-A-Pest</kbNév>
  </Könyvesbolt>

  <Vásárló vkod="10">
    <vNév>Bognár Éva</vNév>
    <vÉletkor>66</vÉletkor>
    <lakCím>
      <vVáros>Budapest</vVáros>
      <vUtca>Zöld utca</vUtca>
      <vHázszám>111</vHázszám>
    </lakCím>
  </Vásárló>
</Könyv>
```

```

<Könyv kkod="08">
  <kiadási_dátum>09/11/2020</kiadási_dátum>
  <cím>Számítógépvásárlási túlélőkönyv</cím>
  <kategória>Számítástechnika</kategória>

  <Szerző szkod="03">
    <szElérhetőség>207318219</szElérhetőség>
    <szNév>Ken Follett</szNév>
    <szÉletkor>72</szÉletkor>
  </Szerző>

  <Kiadó kikod="03">
    <kiVáros>Miskolc</kiVáros>
    <kiCím>Miskolci utca 11.</kiCím>
    <kiNév>Miskolci kiadó</kiNév>
  </Kiadó>

  <Könyvesbolt telefonszam ="301829231">
    <kbVáros>Miskolc</kbVáros>
    <kbCím>Iroda utca 4.</kbCím>
    <kbNév>Office Könyvesbolt</kbNév>
  </Könyvesbolt>

  <Vásárló vkod="08">
    <vnév>Mezei Gábor</vnév>
    <véletkor>39</véletkor>
    <lakcím>
      <vVáros>Miskolc</vVáros>
      <vUtca>Gyula utca</vUtca>
      <vházzsám>88</vházzsám>
    </lakcím>
  </Vásárló>
</Könyv>

```

```

<Könyv kkod="09">
  <kiadási_dátum>11/11/2011</kiadási_dátum>
  <cím>The Shining</cím>
  <kategória>Horror</kategória>

  <Szerző szkod="04">
    <szElérhetőség>304891398</szElérhetőség>
    <szNév>Fehér Krisztián</szNév>
    <szÉletkor>45</szÉletkor>
  </Szerző>

  <Kiadó kikod="04">
    <kiVáros>Eger</kiVáros>
    <kiCím>Egri út 7.</kiCím>
    <kiNév>Egri kiadó</kiNév>
  </Kiadó>

  <Könyvesbolt telefonszam ="309184751">
    <kbVáros>Eger</kbVáros>
    <kbCím>Sóház utca 11.</kbCím>
    <kbNév>Lira Eger</kbNév>
  </Könyvesbolt>

  <Vásárló vkod="06">
    <vnév>Kiss Levente</vnév>
    <véletkor>18</véletkor>
    <lakcím>
      <vVáros>Eger</vVáros>
      <vUtca>Fényes utca</vUtca>
      <vházzsám>66</vházzsám>
    </lakcím>
  </Vásárló>
</Könyv>

```

```

<Könyv kkod="10">
  <kiadási_dátum>10/01/2018</kiadási_dátum>
  <cím>It</cím>
  <kategória>Horror</kategória>

  <Szerző szkod="05">
    <szElérhetőség>703189418</szElérhetőség>
    <szNév>Stephen King</szNév>
    <szÉletkor>74</szÉletkor>
  </Szerző>

  <Kiadó kikod="05">
    <kiVáros>Szombathely</kiVáros>
    <kiCím>Szombathely utca 13.</kiCím>
    <kiNév>Szombathelyi kiadó</kiNév>
  </Kiadó>

  <Könyvesbolt telefonszam ="208194851">
    <kbVáros>Szombathely</kbVáros>
    <kbCím>Vezér utca 6.</kbCím>
    <kbNév>Office Könyvesbolt</kbNév>
  </Könyvesbolt>

  <Vásárló vkod="07">
    <vnév>Barna Virág</vnév>
    <véletkor>55</véletkor>
    <lakcím>
      <vVáros>Szombathely</vVáros>
      <vUtca>Szepes utca</vUtca>
      <vházzsám>77</vházzsám>
    </lakcím>
  </Vásárló>
</Könyv>
</Könyvek>

```

1d)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.example.org/XMLSchemaI3I4BP" xmlns:tns="http://www.example.org/XMLSchemaI3I4BP" elementFormDefault="qualified">

  <xs:element name="Könyvek">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Könyv" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="kiadási_dátum" type="xs:string" />
              <xs:element name="cím" type="xs:string" />
              <xs:element name="kategória" type="xs:string" />

              <xs:element name="Szerző">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="szÉlérhetőség" type="xs:unsignedInt" />
                    <xs:element name="szNév" type="xs:string" />
                    <xs:element name="szÉletkor" type="xs:unsignedInt" />
                  </xs:sequence>
                  <xs:attribute name="szkod" type="xs:unsignedInt" use="required" />
                </xs:complexType>
              </xs:element>

              <xs:element name="Kiadó">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="kiVáros" type="xs:string" />
                    <xs:element name="kiCím" type="xs:string" />
                    <xs:element name="kiNév" type="xs:string" />
                  </xs:sequence>
                  <xs:attribute name="kkod" type="xs:unsignedInt" use="required" />
                </xs:complexType>
              </xs:element>

              <xs:element name="Könyvesbolt">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="kbVáros" type="xs:string" />
                    <xs:element name="kbCím" type="xs:string" />
                    <xs:element name="kbNév" type="xs:string" />
                  </xs:sequence>
                  <xs:attribute name="telefonszam" type="xs:unsignedInt" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

```

    <xs:element name="VásárLó">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="vNév" type="xs:string" />
          <xs:element name="vÉletkor" type="xs:unsignedInt" />
          <xs:element name="LakCím">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="vVáros" type="xs:string" />
                <xs:element name="vUtca" type="xs:string" />
                <xs:element name="vHázszám" type="xs:unsignedInt" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="vkod" type="xs:unsignedInt" use="required" />
      </xs:complexType>
    </xs:element>

  </xs:sequence>
  <xs:attribute name="kkod" type="xs:unsignedInt" use="required" />
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>

</xs:schema>
```

## 2.feladat

A kommentek a kódon belül találhatóak

2a)

Adatolvasás: - DOMReadI3I4BP.java

```
package hu.domparse.i3i4bp;

import org.w3c.dom.Document;

public class DOMReadI3I4BP {

    public static void main(String[] args) throws IOException, ParserConfigurationException, SAXException{
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        try {
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document doc = (Document) db.parse(new File("src\\XMLI3I4BP.xml"));
            doc.getDocumentElement().normalize();

            System.out.println("Gyökérelem meghatározása");
            System.out.println();

            NodeList könyv = doc.getElementsByTagName("Könyv"); //Könyv elemek elérése
            for(int i = 0; i < könyv.getLength(); i++) {
                Node node = könyv.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element element = (Element) node;

                    String id = element.getAttribute("kkod");

                    String datum = element.getElementsByTagName("kiadási dátum").item(0).getTextContent(); //Könyv elem belső tulajdonságainak elérése
                    String cim = element.getElementsByTagName("cim").item(0).getTextContent();
                    String kategoria = element.getElementsByTagName("kategória").item(0).getTextContent();

                    System.out.println("Jelenlegi elem : " + node.getNodeName()); //Ezek kiírása
                    System.out.println("Könyv kód : " + id);
                    System.out.println("Dátum : " + datum);
                    System.out.println("Cím : " + cim);
                    System.out.println("Kategória : " + kategoria);
                    System.out.println();

                    NodeList szerzo = doc.getElementsByTagName("Szerző"); //Szerző elem elérése
                    Node sznode = szerzo.item(i);
                    if (sznode.getNodeType() == Node.ELEMENT_NODE) {
                        Element szelement = (Element) sznode;

                        String sid = szelement.getAttribute("szkod");
                        String szelerhetoseg = szelement.getElementsByTagName("szElérhetőség").item(0).getTextContent(); //Szerző elem belső
                        String sznev = szelement.getElementsByTagName("szNév").item(0).getTextContent(); //tulajdonságainak elérése
                        String szetkor = szelement.getElementsByTagName("szÉletkor").item(0).getTextContent();

                        System.out.println("Szerző kódja : " + sid);
                        System.out.println("Szerző elérhetősége : " + szelerhetoseg); //Ezek kiírása
                        System.out.println("Szerző neve : " + sznev);
                        System.out.println("Szerző életkora : " + szetkor);
                        System.out.println();
                    }
                }
            }
        }
    }
}
```

```
NodeList kiado = doc.getElementsByTagName("Kiadó"); //Kiadó elem elérése
Node knode = kiado.item(i);
if (knode.getNodeType() == Node.ELEMENT_NODE) {
    Element kelement = (Element) knode;

    String kikod = kelement.getAttribute("kikod");
    String kiVaros = kelement.getElementsByTagName("kiVáros").item(0).getTextContent(); //Kiadó elem belső tulajdonságainak elérése
    String kiCim = kelement.getElementsByTagName("kiCím").item(0).getTextContent();
    String kiNev = kelement.getElementsByTagName("kiNév").item(0).getTextContent();

    System.out.println("Kiadó kódja : " + kikod);
    System.out.println("Kiadó városa : " + kiVaros); //Ezek kiírása
    System.out.println("Kiadó címe : " + kiCim);
    System.out.println("Kiadó neve : " + kiNev);
    System.out.println();
}

NodeList kbolt = doc.getElementsByTagName("Könyvesbolt"); //Könyvesbolt elem elérése
Node kbnode = kbolt.item(i);
if (kbnode.getNodeType() == Node.ELEMENT_NODE) {
    Element kbelement = (Element) kbnode;

    String kbszam = kbelement.getAttribute("telefonszam");
    String kbVaros = kbelement.getElementsByTagName("kbVáros").item(0).getTextContent(); //Könyvesbolt elem belső
    String kbCim = kbelement.getElementsByTagName("kbCím").item(0).getTextContent(); //tulajdonságainak elérése
    String kbNev = kbelement.getElementsByTagName("kbNév").item(0).getTextContent();

    System.out.println("Könyvesbolt telefonszáma : " + kbszam);
    System.out.println("Könyvesbolt városa : " + kbVaros);
    System.out.println("Könyvesbolt címe : " + kbCim); //Ezek kiírása
    System.out.println("Könyvesbolt neve : " + kbNev);
    System.out.println();
}
}
```

```

        NodeList vararlo = doc.getElementsByTagName("Vásárló"); //Vásárló elem elérése
        Node Vnode = vararlo.item(i);
        if(Vnode.getNodeType() == Node.ELEMENT_NODE) {
            Element Velement = (Element) Vnode;

            String vNev = Velement.getElementsByTagName("vNév").item(0).getTextContent(); //Vásárló elem belső tulajdonságainak elérése
            String vEletkor = Velement.getElementsByTagName("vEletkor").item(0).getTextContent();

            System.out.println("Vásárló neve : " + vNev);
            System.out.println("Vásárló életkora : " + vEletkor); //Vásárló kiírása

            NodeList lakcim = doc.getElementsByTagName("lakcím"); //Vásárló elem lakcím összetett elemének elérése
            Node lnode = lakcim.item(i);
            if(lnode.getNodeType() == Node.ELEMENT_NODE) {
                Element lelement = (Element) lnode;

                String vVaros = Velement.getElementsByTagName("vVáros").item(0).getTextContent(); //lakcím elem belső tulajdonságainak elérése
                String vUtc = Velement.getElementsByTagName("vUtc").item(0).getTextContent();
                String vHazszam = Velement.getElementsByTagName("vHázzszám").item(0).getTextContent();

                System.out.println("Vásárló városa : " + vVaros);
                System.out.println("Vásárló utcája : " + vUtc);
                System.out.println("Vásárló házzsám : " + vHazszam); //Ezek kiírása
                System.out.println();
            }
        }
        System.out.println("-----");
    }
}

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

```

2b)

## Adatmódosítás – DOMModifyI3I4BP.java

```

package hu.domparse.i3i4bp;

import java.io.File;

public class DOMModifyI3I4BP {

    public static void main(String[] args) throws IOException, ParserConfigurationException, SAXException {
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        try {
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document doc = (Document) db.parse(new File("src\\XML\\I3I4BP.xml"));
            Node könyv = doc.getElementsByTagName("Könyv").item(0); //első könyv elem elérése

            NamedNodeMap attr = könyv.getAttributes(); //első könyv elem tulajdonságának lekérdezése
            Node nodeAttr = attr.getNamedItem("kkod"); //kkod tulajdonság megkeresése
            NodeAttr.setTextContent("444"); //tulajdonság értékének megváltoztatása

            NodeList list = könyv.getChildNodes(); //könyv elem gyerekelemeinek elérése

            for(int i = 0; i < list.getLength(); i++) { //iteráció elemek között
                Node node = list.item(i);
                if(node.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) node;
                    if("cím".equals(eElement.getNodeName())) { //cím megegyezik-e az éppen vizsgált elem nevével? Ha igen akkor...
                        if("Senki sem az, akinek látszik".equals(eElement.getTextContent())) { //...megnézi, hogy a cím elem tartalma is
                            //megegyezik-e az éppen vizsgált elem tartalmával? Ha igen...
                            eElement.setTextContent("Mindenki az, akinek látszik"); //...akkor megváltoztatja erre.
                        }
                    }
                    if("kategória".equals(eElement.getNodeName())) { //kategória megegyezik-e az éppen vizsgált elem nevével? Ha igen akkor...
                        if("Irodalom".equals(eElement.getTextContent())) { //...megnézi, hogy a kategória elem tartalma is
                            // megegyezik-e az éppen vizsgált elem tartalmával? Ha igen...
                            eElement.setTextContent("Történelem"); //...akkor megváltoztatja erre.
                        }
                    }
                }
            }

            TransformerFactory transformerFactory = TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer(); //transformer létrehozása, ami lehetővé teszi majd az eredmény különféle kiírását
            DOMSource source = new DOMSource(doc); //transzformációs 'source tree' számára egy tároló DOM fáként
            transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8"); //elvilleg ennek a két sornak kéne a kimenetet utf-8 karakterkészlettel
            transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "yes"); //létrehozni, de nem működik. Konzolba nem írja ki rendesen a dolgokat,
            //de egyébként ki lehet olvasni, hogy a megfelelő dolgok kerültek változtatásra
            // csak az ékezetes betűk helyett random karakterek szerepelnek

            StreamResult consoleResult = new StreamResult(System.out); //transzformáció eredményének tárolója, xml, html, sima szöveg, vagy egyéb módon
            transformer.transform(source, consoleResult); //a forrást (source) belerakja az eredmény tárolóba (consoleResult)

        } catch (Exception e) { //hibakezelés, ha csak Exception-t adunk meg, akkor minden hibát kezel, ami nem saját hiba, mivel ez az összes hiba ebből ered.
            e.printStackTrace();
        }
    }
}

```



2c)

## Adatlekérdezés – DOMQueryI3I4BP.java

```
1 package hu.domparse.i3i4bp;
2
3 import java.io.File;
4
5
6
7
8
9
10
11
12
13
14
15
16
17 public class DOMQueryI3I4BP {
18     public static void main(String[] args) throws IOException, ParserConfigurationException, SAXException {
19         DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
20
21         try {
22             DocumentBuilder db = dbf.newDocumentBuilder();
23             Document doc = (Document) db.parse(new File("src\\XMLI3I4BP.xml"));
24             doc.getDocumentElement().normalize();
25
26             System.out.println("Könyvek címe, amik irodalm kategóriába sorolhatók:");
27             ListLiteratureBookNames(doc);
28             System.out.println("-----\n"); //kiíratás
29             ListKerryFisherBookBuyersNamesAndAges(doc);
30             System.out.println("-----\n");
31             ListShopPhonenumberAddressAndName(doc);
32             System.out.println("-----\n");
33             ListPublisherAndWriterNames(doc);
34             System.out.println("-----\n");
35         } catch (Exception e) {
36             e.printStackTrace();
37         }
38     }
39
40     //40-83 sor Szerző, Kiadó, Könyvesbolt, Vásárló elem elérése kódismétlés elkerülése érdekében.
41     static Element getSzerzoElement(Document doc, int i) {
42         NodeList nodeList = doc.getElementsByTagName("Szerző");
43         Node node = nodeList.item(i);
44         if (node.getNodeType() == Node.ELEMENT_NODE) {
45             Element element = (Element) node;
46             return element;
47         } else {
48             return null;
49         }
50     }
51
52     static Element getKiadoElement(Document doc, int i) {
53         NodeList nodeList = doc.getElementsByTagName("Kiadó");
54         Node node = nodeList.item(i);
55         if (node.getNodeType() == Node.ELEMENT_NODE) {
56             Element element = (Element) node;
57             return element;
58         } else {
59             return null;
60         }
61     }
62 }
```

```

62● static Element getKonyvesboltElement(Document doc, int i) {
63     NodeList nodelist = doc.getElementsByTagName("Könyvesbolt");
64     Node node = nodelist.item(i);
65     if(node.getNodeType() == Node.ELEMENT_NODE) {
66         Element element = (Element) node;
67         return element;
68     }else {
69         return null;
70     }
71 }
72
73● static Element getVasarloElement(Document doc, int i) {
74     NodeList nodelist = doc.getElementsByTagName("Vásárló");
75     Node node = nodelist.item(i);
76     if(node.getNodeType() == Node.ELEMENT_NODE) {
77         Element element = (Element) node;
78         return element;
79     }else {
80         return null;
81     }
82 }
83
84● static void listLiteratureBookNames(Document doc) { //Irodalom kategóriájú könyvek nevének lekérdezése.
85     NodeList könyv = doc.getElementsByTagName("Könyv");
86     for(int i = 0; i < könyv.getLength(); i++) {
87         Node node = könyv.item(i);
88         if(node.getNodeType() == Node.ELEMENT_NODE) {
89             Element element = (Element) node;
90             //kiírás
91             if((element.getElementsByTagName("kategória").item(0).getTextContent()).equalsIgnoreCase("Irodalom")) {
92                 System.out.println(" " + element.getElementsByTagName("cim").item(0).getTextContent());
93             }
94         }
95     }
96 }
97
98● static void listKerryFisherBookBuyersNamesAndAges(Document doc) { //Kerry Fisher könyveket vásárló emberek nevének és korának lekérdezése.
99     NodeList könyv = doc.getElementsByTagName("Könyv");
100     System.out.println("Kerry Fisher könyveit 6k vették meg:");
101     for(int i = 0; i < könyv.getLength(); i++) {
102         Node node = könyv.item(i);
103         if(node.getNodeType() == Node.ELEMENT_NODE) {
104             Element szerzo = getSzerzoElement(doc, i);
105             Element vasarlo = getVasarloElement(doc, i);
106             //kiírás
107             if((szerzo.getElementsByTagName("szNév").item(0).getTextContent()).equalsIgnoreCase("Kerry Fisher")) {
108                 System.out.println(" Vásárló neve: " + vasarlo.getElementsByTagName("vNév").item(0).getTextContent()
109                     + ", " + vasarlo.getElementsByTagName("vÉletkor").item(0).getTextContent() + " éves."
110                 );
111             }
112         }
113     }
114 }
115 }
116 }
117 }

```

```

118● static void listShopPhonenumberAddressAndName(Document doc) { //Lekérdezés, hogy Ken Follett melyik könyveit hol adták el, ezek a helyek adatai.
119     NodeList könyv = doc.getElementsByTagName("Könyv");
120     System.out.println("Ken Follett alábbi könyveit adták el: ");
121     for(int i = 0; i < könyv.getLength(); i++) {
122         Node node = könyv.item(i);
123         if(node.getNodeType() == Node.ELEMENT_NODE) {
124             Element kbolt = getKonyvesboltElement(doc, i);
125             Element szerzo = getSzerzoElement(doc, i);
126             Element element = (Element) node;
127             //kiírás
128             if((szerzo.getElementsByTagName("szNév").item(0).getTextContent()).equalsIgnoreCase("Ken Follett")){
129                 System.out.println(" Könyv címe: " + element.getElementsByTagName("cim").item(0).getTextContent() + "\nAz alábbi könyvesboltba:");
130                 System.out.println(" Neve: " + kbolt.getElementsByTagName("kbNév").item(0).getTextContent() + "\n Címe: " +
131                     kbolt.getElementsByTagName("kbVáros").item(0).getTextContent() + ", " + kbolt.getElementsByTagName("kbCím").item(0).getTextContent()
132                     + "\n Telefonszáma: " + kbolt.getAttribute("telefonszam"));
133                 System.out.println();
134             }
135         }
136     }
137 }
138 }
139 }
140 }
141 }

```

```

142● static boolean isInList(ArrayList<String> a, String str) { //Megnézi, hogy egy név benne van-e már az arraylist-ben, ismétlődés elkerülése érdekében.
143     boolean state = false; //Ez csak azért szükséges bele, mert úgy írtam meg az xml dokumentumot, hogy egy szerzőhöz egy
144                             //kiadó tartozik, tehát nem lehet 2 különböző kiadó egy szerzőhöz.
145     for(int i = 0; i < a.size(); i++) { //Így 5 sort ír ki, szerző neve és a hozzá vett kiadó. Ha nem lenne ez benne, akkor mindent 2x írna ki.
146         if(a.get(i).equalsIgnoreCase(str)) { //Ha egy szerző több kiadéhoz is tartozna, akkor másképp kéne csinálni.
147             state = true;
148             break;
149         }else {
150             state = false;
151         }
152     }
153     return state;
154 }
155
156● static void listPublisherAndWriterNames(Document doc) { //Lekérdezés, ami megadja, hogy melyik író melyik kiadónál publikálta könyveit.
157     NodeList könyv = doc.getElementsByTagName("Könyv");
158     System.out.println("Melyik író melyik kiadónál publikálta könyveit:");
159     ArrayList<String> publisherList = new ArrayList<String>();
160     for(int i = 0; i < könyv.getLength(); i++) {
161         Node node = könyv.item(i);
162         if(node.getNodeType() == Node.ELEMENT_NODE) {
163             Element kiado = getKiadoElement(doc, i);
164             Element szerzo = getSzerzoElement(doc, i);
165             String sznev = szerzo.getElementsByTagName("szNév").item(0).getTextContent();
166             if(!isInList(publisherList, sznev) == false) {
167                 publisherList.add(sznev);
168                 System.out.println(" " + sznev + " a " +
169                     kiado.getElementsByTagName("kiNév").item(0).getTextContent() + " kiadónál publikálta könyveit."); //kiírás
170             }
171         }
172     }
173 }
174 }
175 }
176 }
177 }
178 }

```