

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

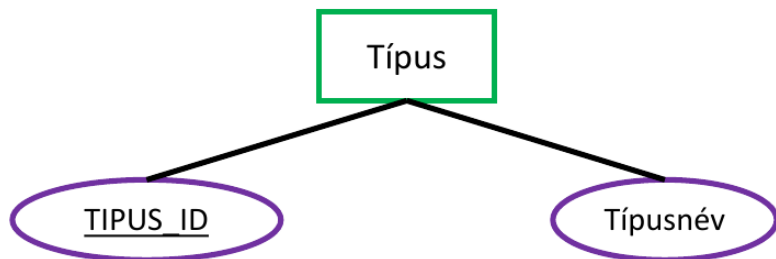
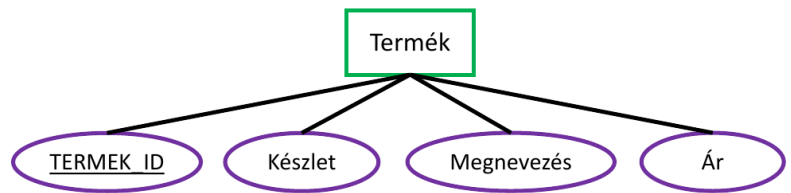
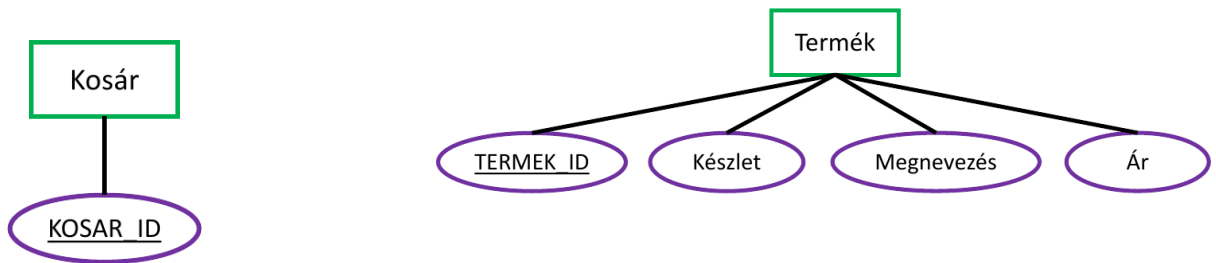
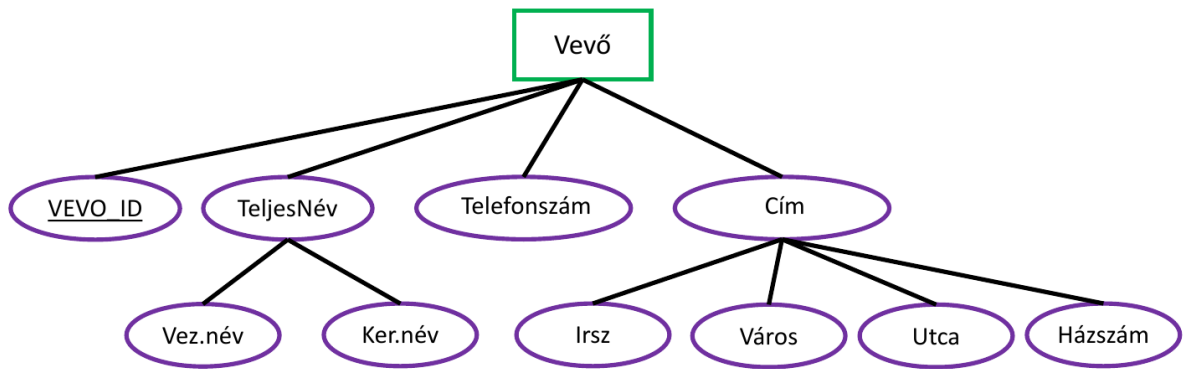
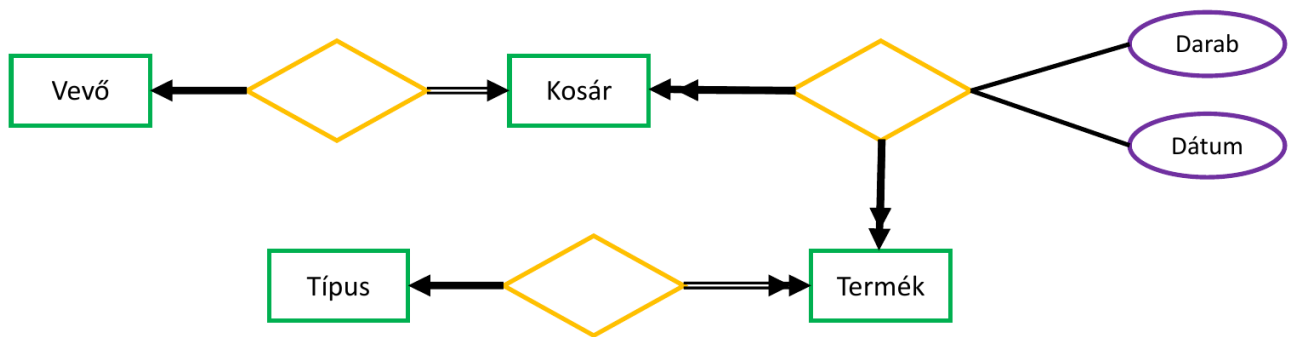
Készítette: **Szabó Péter**
Neptunkód: **HX35EU**

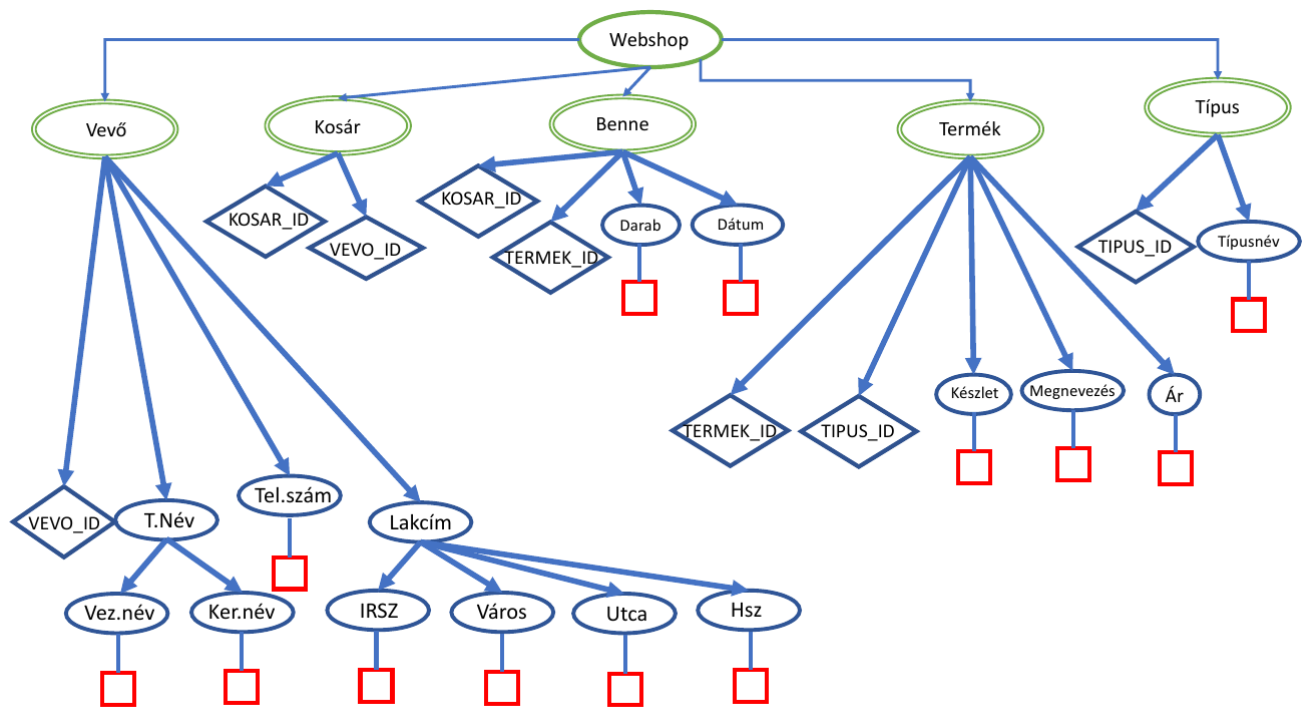
Rövid leírás:

Az xml beadandóhoz egy webshop adatbázist álmodtam meg. Az adatbázis alapelemei a vevő, a termék és a típus. A vevők regisztrálhatnak a webshopban, termékeket adhatnak hozzá a kosárhoz ezeknek az adatait tarolom. Két kiegészítő egyed is szerepel a webshopban, egyik a kosár amit a vevőkhöz rendelve valósítottam meg, tehát minden vevőnek 1 kosara lehet. Hogy mi szerepel a kosárban azt egy másik elemmel realizáltam, mely a kosárazonosítóhoz a termékazonítókat rendeli hozzá, és nyilvántartja hogy hány darabot szeretne a vevő, illetve mikor adta a kosárhoz.

A vevőnél nyilvántartjuk a személyes adatait, teljes nevét, lakcímét és a telefonszámát. A lakcím és a név összetett típus, a telefonszám pedig egyedi típus, csak magyar telenszámot fogad el.

A termék elem tartalmazza az elérhető készletet, a termék nevét és árát.





Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><Webshop
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaHx35eu.xsd">
  <Vevő VEVO_ID="0">
    <TeljesNév>
      <Vezetéknév>Szabó</Vezetéknév>
      <Keresztnév>Péter</Keresztnév>
    </TeljesNév>
    <Telefonszám>301234657</Telefonszám>
    <Lakcím>
      <Irányítószám>3524</Irányítószám>
      <Város>Miskolc</Város>
      <Utca>Stadion</Utca>
      <Házzsám>12</Házzsám>
    </Lakcím>
  </Vevő>
  <Vevő VEVO_ID="1">
    <TeljesNév>
      <Vezetéknév>Kiss</Vezetéknév>
      <Keresztnév>Béla Aurélió</Keresztnév>
    </TeljesNév>
    <Telefonszám>701234567</Telefonszám>
    <Lakcím>
      <Irányítószám>1015</Irányítószám>
      <Város>Budapest</Város>
      <Utca>Fahéj</Utca>
      <Házzsám>22</Házzsám>
    </Lakcím>
  </Vevő>
  <Vevő VEVO_ID="2">
    <TeljesNév>
      <Vezetéknév>Nincs</Vezetéknév>
      <Keresztnév>Ko Sara</Keresztnév>
    </TeljesNév>
    <Telefonszám>201234567</Telefonszám>
    <Lakcím>
      <Irányítószám>1015</Irányítószám>
      <Város>Budapest</Város>
      <Utca>Perc</Utca>
      <Házzsám>10</Házzsám>
    </Lakcím>
  </Vevő>
  <Kosár KOSAR_ID="0" VEVO_ID="0"/>
  <Kosár KOSAR_ID="1" VEVO_ID="1"/>
  <Termék TERMEK_ID="0" TIPUS_ID="0">
    <Készlet>12</Készlet>
    <Megnevezés>LG OLED88ZX9LA</Megnevezés>
```

<Ár>110220</Ár>
</Termék>
<Termék TERMEK_ID="1" TIPUS_ID="0">
 <Készlet>5</Készlet>
 <Megnevezés>Samsung QE82Q950RB</Megnevezés>
 <Ár>142780</Ár>
</Termék>
<Termék TERMEK_ID="2" TIPUS_ID="2">
 <Készlet>120</Készlet>
 <Megnevezés>Apple iPhone 12 Pro Max</Megnevezés>
 <Ár>259160</Ár>
</Termék>
<Termék TERMEK_ID="3" TIPUS_ID="1">
 <Készlet>12000</Készlet>
 <Megnevezés>Apple MacBook Pro 16</Megnevezés>
 <Ár>409420</Ár>
</Termék>
<Termék TERMEK_ID="4" TIPUS_ID="1">
 <Készlet>10</Készlet>
 <Megnevezés>Apple MacBook Pro 13</Megnevezés>
 <Ár>551100</Ár>
</Termék>
<Típus TIPUS_ID="0">
 <Típusnév>Televízió</Típusnév>
</Típus>
<Típus TIPUS_ID="1">
 <Típusnév>Laptop</Típusnév>
</Típus>
<Típus TIPUS_ID="2">
 <Típusnév>Mobiltelefon</Típusnév>
</Típus>
<Típus TIPUS_ID="3">
 <Típusnév>Tablet</Típusnév>
</Típus>
<Benne KOSAR_ID="0" TERMEK_ID="1">
 <Darab>2</Darab>
 <Dátum>2020-11-20</Dátum>
</Benne>
<Benne KOSAR_ID="0" TERMEK_ID="4">
 <Darab>1</Darab>
 <Dátum>2020-11-20</Dátum>
</Benne>
<Benne KOSAR_ID="0" TERMEK_ID="3">
 <Darab>10</Darab>
 <Dátum>2020-11-20</Dátum>
</Benne>
</Webshop>

Az XML dokumentum alapján XMLSchema készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="Webshop">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Vevő" maxOccurs="unbounded">
          <xs:complexType mixed="true">
            <xs:sequence>
              <xs:element name="TeljesNév" type="NevType"/>
              <xs:element name="Telefonszám" type="TelefonszamType"/>
              <xs:element name="Lakcím" type="LakcimType"/>
            </xs:sequence>
            <xs:attribute name="VEVO_ID" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>

        <xs:element name="Kosár" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="KOSAR_ID" type="xs:integer" use="required"/>
            <xs:attribute name="VEVO_ID" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>

        <xs:element name="Termék" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Készlet" type="xs:integer"/>
              <xs:element name="Megnevezés" type="xs:string"/>
              <xs:element name="Ár" type="xs:integer"/>
            </xs:sequence>
            <xs:attribute name="TERMEK_ID" type="xs:integer" use="required"/>
            <xs:attribute name="TIPUS_ID" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>

        <xs:element name="Típus" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Típusnév" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="TIPUS_ID" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>

        <xs:element name="Benne" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Darab" type="xs:integer"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
        <xs:element name="Dátum" type="DatumType"/>
    </xs:sequence>
    <xs:attribute name="KOSAR_ID" type="xs:integer" use="required"></xs:attribute>
    <xs:attribute name="TERMEK_ID" type="xs:integer" use="required"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

```
<xs:key name="VEVO_KULCS">
    <xs:selector xpath="./Vevő" />
    <xs:field xpath="@VEVO_ID" />
</xs:key>
```

```
<xs:key name="KOSAR_KULCS">
    <xs:selector xpath="./Kosár" />
    <xs:field xpath="@KOSAR_ID" />
</xs:key>
```

```
<xs:key name="TERMEK_KULCS">
    <xs:selector xpath="./Termék" />
    <xs:field xpath="@TERMEK_ID" />
</xs:key>
```

```
<xs:key name="TIPUS_KULCS">
    <xs:selector xpath="./Típus"></xs:selector>
    <xs:field xpath="@TIPUS_ID"></xs:field>
</xs:key>
```

```
<xs:unique name="TERMEK_EGYEDI_KULCS">
    <xs:selector xpath="./Termék"></xs:selector>
    <xs:field xpath="@TERMEK_ID"></xs:field>
</xs:unique>
```

```
<xs:unique name="EGY_VEVO_EGY_KOSAR">
    <xs:selector xpath="./Kosár"></xs:selector>
    <xs:field xpath="@VEVO_ID"></xs:field>
</xs:unique>
```

```
<!-- Kosárhoz kötelező a vevő -->
<xs:keyref name="VEVO_KOSARA" refer="VEVO_KULCS">
    <xs:selector xpath="./Kosár" />
    <xs:field xpath="@VEVO_ID" />
</xs:keyref>
```

```
<!-- Termékhez kötelező típus -->
<xs:keyref name="TERMEK_TIPUSA" refer="TIPUS_KULCS">
    <xs:selector xpath="./Termék" />
    <xs:field xpath="@TIPUS_ID" />
```



```

</xs:keyref>

<!-- A kosár - termék kapcsolat megvalósítása -->
<xs:keyref name="BENNE_KOSAR" refer="KOSAR_KULCS">
  <xs:selector xpath="/Benne" />
  <xs:field xpath="@KOSAR_ID" />
</xs:keyref>

<xs:keyref name="BENNE_TERMEK" refer="TERMEK_KULCS">
  <xs:selector xpath="/Benne" />
  <xs:field xpath="@TERMEK_ID" />
</xs:keyref>
</xs:element>

<xs:simpleType name="IRSZType">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="1000"/>
    <xs:maxInclusive value="9999"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TelefonszamType">
  <xs:restriction base="xs:string">
    <xs:pattern value="(20|30|70)[0-9]{7}" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DatumType">
  <xs:restriction base="xs:string">
    <xs:pattern value="([12]\d{3}-(0[1-9]|1[0-2])-(0[1-9]|([12]\d|3[01])))" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="LakcimType">
  <xs:sequence>
    <xs:element name="Írányítószám" type="IRSZType"/>
    <xs:element name="Város" type="xs:string"/>
    <xs:element name="Utca" type="xs:string"/>
    <xs:element name="Házszám" type="xs:positiveInteger"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="NevType">
  <xs:sequence>
    <xs:element name="Vezetéknév" type="xs:string"/>
    <xs:element name="Keresztnév" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

2. feladat

A feladat egy DOM program készítése az XML dokumentum adatainak adminisztrálása alapján:

Adatolvasás:

```
package hu.domparse.hx35eu;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMReadHx35eu {

    public static void main(String[] args) throws SAXException, IOException,
ParserConfigurationException {
        File file = new File("XMLHx35eu.xml");

        Document doc =
DocumentBuilderFactory.newInstance().newDocumentBuilder().parse(file);
        Element root = doc.getDocumentElement();
        root.normalize();

        printXML(root, "");
    }

    public static void printXML(Node root, String indentation) {
        String nodename = root.getNodeName();

        if (!nodename.contains("text")) {
            System.out.println(indentation + nodename + printAttrs(root));
        }

        indentation += "\t";

        NodeList childNodes = root.getChildNodes();

        for (int i = 0; i < childNodes.getLength(); i++) {
            Node child = childNodes.item(i);
            boolean complexType = child.getTextContent().contains("\n");

            if (complexType) {
                printXML(child, indentation);
            } else {
                System.out
                    .println(indentation + child.getNodeName() +
printAttrs(child) + ": " + child.getTextContent());
            }
        }
    }
}
```



```
eElement.getElementsByTagName("Készlet").item(0).setTextContent("12000");
    }
}

Transformer transformer =
TransformerFactory.newInstance().newTransformer();
DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(file);
transformer.transform(source, result);
}
}
```