JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Étteremlánc modellezése

Készítette: Szabó Kristóf

Neptunkód: IDA58U

Dátum: 2023.12.01

Tartalomjegyzék

1. Feladat

1.1. A feladat leírása:

A feladatomban egy 4 egyedből álló adatbázis modellt szeretnék bemutatni, amely a tulajdonosok, éttermek, ételek és vevők kapcsolatáról szól.

Az **Étterem** egyedhez tartozik a **EtID** tulajdonság, amely az elsődleges kulcsa az egyednek. Ezen kívül még a **Telefonszám**, amely egy többértékű tulajdonsága, hiszem egy Étteremnek lehet több telefonszáma is, például egy vezetékes és egy mobil. Továbbá a **Név** tulajdonság és a **Cím**, amely egy összetett tulajdonság, hiszen az Irszám, a Város, utca és házszám elemei vannak.

Az **Étel** egyed **EtelID** tulajdonsága kulcsként funkcionál, másik két tulajdonsága a **Név** és az **Ár**, valamint a **Köret**, amely többértékű tulajdonság, hiszen egy ételhez kérhetünk több köretet is.

A **Tulajdonos** egyed **Tkód** tulajdonsága egyedi, számjegyekből álló azonosítója, azaz kulcsként funkcionál. Van egy származtatott tulajdonsága, az **Életkor**, amelyet a **Szül.idő** tulajdonságból származtathatunk hiszen, ha a jelenlegi dátumból kivonjuk a születési dátumot, akkor megkapjuk a Tulajdonos életkorát.

A **Vevő** egyed elsődleges kulcsa a **Vkód**, amely egyértelműen azonosítja a vevőt. Másik két tulajdonsága a **Név** és a **Telefonszám**, ezek egyszerű tulajdonságok. Végül pedig a **Cím** tulajdonság, amely egy összetett tulajdonság, az Irszámból, Városból, Utcából és a Házszámból tevődik össze.

Az **Étterem** és a **Étel** egyedek között a **Készíti** kapcsolat van, amely egy egy **több a többhöz kapcsolat (N:M),** hiszen egy étterem több ételt is készíthet, és egy ételt több étterem is el tud készíteni.

Az **Étterem** és a **Tulajdonos** között **egy a többhöz (1:N)** kapcsolat van, amit a **Birtokolja** jelöl, mert egy étteremnek több tulajdonosa is lehet, viszont az én modellemben egy étteremnek csak egy tulajdonosa lehet.

Az **Étel** és a **Vevő** egyedek között **Megveszi** kapcsolat van, ami szintén egy **több a többhöz kapcsolat (N:M),** mivel egy féle ételt több vevő is megvehet, valamint egy vevő több étteremben is étkezhet.

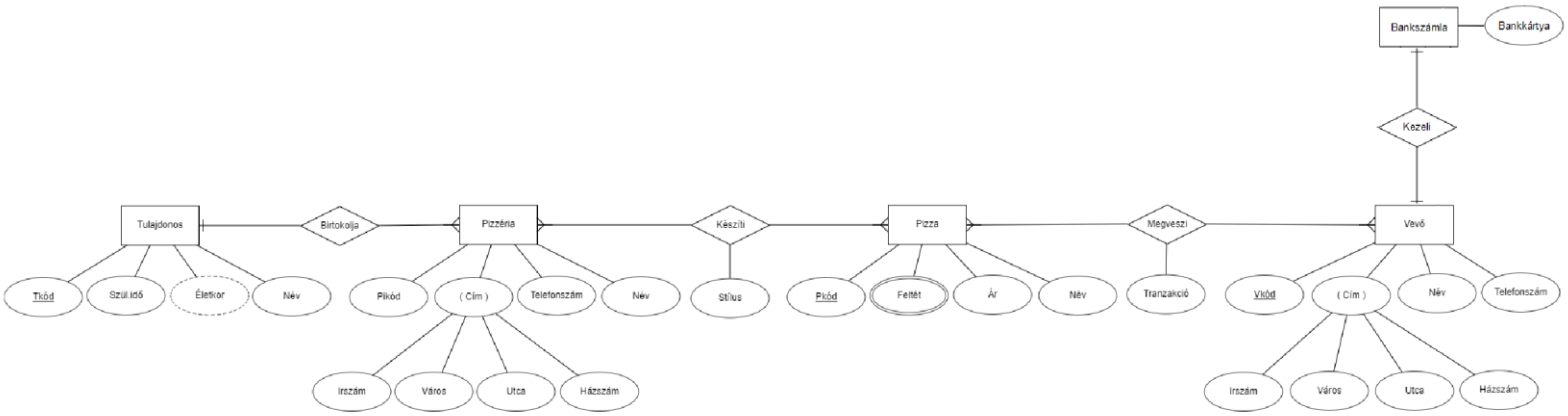
1.2. ER Modell:

Egy alapvető ER-modell egyedtípusokból áll, és meghatározza az egyedek között létező kapcsolatokat. Az ER Modell felépítéséhez szükséges elemek bemutatása és leírása.

1.2.1.Egyedek:

- **Étterem**:
  Attribútumok: EtID (elsődleges kulcs),**Telefonszám** (többértékű tulajdonság), **Név**, **Cím** (összetett tulajdonság: **Irszám**, **Város**, **utca**, **házszám**)

- **Étel:**
  Attribútumok: EtelID (elsődleges kulcs), **Név**, **Ár**, **Köret** (többértékű tulajdonság)



- **Tulajdonos**
  Attribútumok: Tkód (kulcsként), **Életkor** (származtatott tulajdonság: **Szül.idő**)

- **Vevő:**
  Attribútumok: Vkód (elsődleges kulcsa), **Név**, **Telefonszám**, **Cím** (összetett tulajdonság: **Irszám**, **Város**, **utca**, **házszám**)

1.2.2.Kapcsolatok:

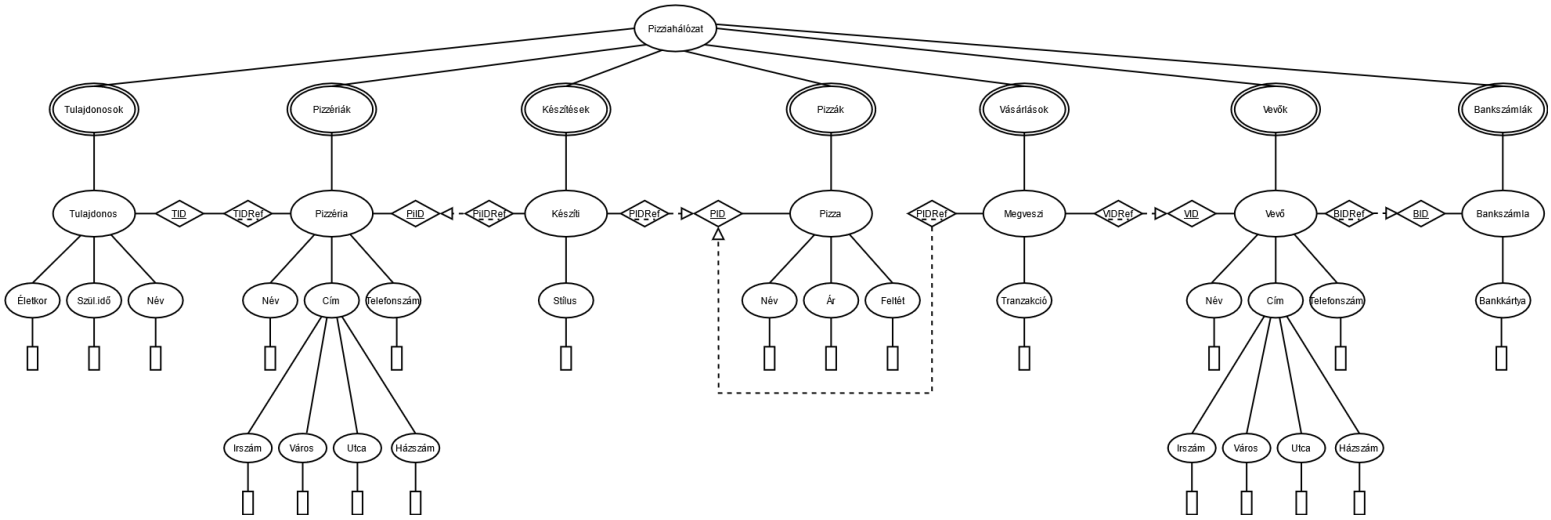
- **Készíti:** Több a többhöz kapcsolat**(N:M)** az **Étterem** és az **Étel** egyedek között.
- **Birtokolja:** Egy a többhöz kapcsolat**(1:N)** az **Étterem** és a **Tulajdonos** egyedek között.
- **Megveszi:** Több a többhöz kapcsolat**(N:M)** az **Étel** és a **Vevő** egyedek között.

1.3. XDM Modell:

Az XDM Modell-nél fontos a dokumentum sorrend kezelése. A gyökér az első elem (Étteremlánc). Minden csomópont megelőzi a leszármazottjait (Étteremlánc //Tulajdonosok//Tulajdonos//Név). A testvér balról jobbra következnek.

Az ER modell XDM modellre való konvertálása után létrejöttek idegen kulcsok, ezek a TIDRef, EtelIDRef, EtIDRef és a VIDRef, ezek a hozzájuk tartozó elsődleges kulcsra mutatnak.

Ezen kívül a két több-több (N:M) kapcsolatnál létrejött két objektum. Az Éttermek és az Ételek objektum közötti kapcsolatnál létrejött a Készítések objektum, valamint az Ételek és a Vevők között a Vásárlások objektum.



1.4. Az XDM modell alapján XML Dokumentum készítése:

Az XML dokumentumok egy fastruktúrát alkotnak, amely a gyökérnél kezdődik és a levelekhez ágazik. Az elemek közötti kapcsolat leírására a **szülő** (A **Tulajdonos** elemnek a **Tulajdonosok** elem) , **gyermek** (a **Tulajdonosok** elemnek a **Tulajdonos** elem) és **testvér** (a **Tulajdonosoknak** elemnek az **Éttermek** elem) kifejezéseket használják. Az XML dokumentum gyökéreleme a Étteremlánc.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Etteremlanc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaEJX162.xsd">

    <Tulajdonosok>

        <Tulajdonos TID="1">
            <Nev>Nagy Béla</Nev>
            <Eletkor>26</Eletkor>
            <Szulido>1997</Szulido>
        </Tulajdonos>

        <Tulajdonos TID="2">
            <Nev>Balázs Dominika</Nev>
            <Eletkor>25</Eletkor>
            <Szulido>1998</Szulido>
        </Tulajdonos>

        <Tulajdonos TID="3">
            <Nev>Kiss Imre</Nev>
            <Eletkor>23</Eletkor>
            <Szulido>2000</Szulido>
        </Tulajdonos>

    </Tulajdonosok>

    <Ettermek>
        <Etterem EtID="1" TIDRef="1">
            <Nev>Italian Stallion</Nev>
            <Telefonszam>012432123</Telefonszam>
            <Cim>
                <Irszam>1256</Irszam>
                <Varos>Budapest</Varos>
                <Utca>Ferencziek útja</Utca>
                <Hazszam>14</Hazszam>
            </Cim>
        </Etterem>
```

```
        <Etterem EtID="2" TIDRef="2">
            <Nev>New York Style</Nev>
            <Telefonszam>05618478</Telefonszam>
            <Cim>
                <Irszam>3525</Irszam>
                <Varos>Miskolc</Varos>
                <Utca>Széchenyi István út</Utca>
                <Hazszam>26</Hazszam>
            </Cim>
        </Etterem>

        <Etterem EtID="3" TIDRef="3">
            <Nev>Véndiófa étterem</Nev>
            <Telefonszam>063025698812</Telefonszam>
            <Cim>
                <Irszam>3540</Irszam>
                <Varos>Miskolc</Varos>
                <Utca>Árpád utca</Utca>
                <Hazszam>6</Hazszam>
            </Cim>
        </Etterem>
    </Ettermek>

    <Keszitesek>
        <Kesziti EtelIDRef="1" EtelID="1">
            <Stilus>Előétel</Stilus>
        </Kesziti>

        <Kesziti EtelIDRef="2" EtelID="2">
            <Stilus>Főétel</Stilus>
        </Kesziti>

        <Kesziti EtelIDRef="3" EtelID="3">
            <Stilus>Desszert</Stilus>
        </Kesziti>
    </Keszitesek>
```

```xml
<Etelek>
    <Etel EtelID="1">
        <Nev>Pacalpörkölt</Nev>
        <Ar>4500</Ar>
        <Koret>Krumplipüré</Koret>
    </Etel>

    <Etel EtelID="2">
        <Nev>Rántotthús</Nev>
        <Ar>3200</Ar>
        <Koret>Sültkrumpli</Koret>
        <Koret>Káposztasaláta</Koret>
    </Etel>

    <Etel EtelID="3">
        <Nev>Spagetti</Nev>
        <Ar>2800</Ar>
    </Etel>

    <Etel EtelID="4">
        <Nev>Kacsacomb</Nev>
        <Ar>4600</Ar>
        <Koret>Steakburgonya</Koret>
        <Koret>Mártás</Koret>
        <Koret>Céklasaláta</Koret>
    </Etel>

    <Etel EtelID="5">
        <Nev>Burjósült</Nev>
        <Ar>6600</Ar>
        <Koret>Hagymakarika</Koret>
        <Koret>Grillezett zöldség</Koret>
    </Etel>
</Etelek>
```

```xml
    <Vasarlasok>
        <Megveszi EtelIDRef="1" VIDRef="1">
            <Tranzakcio>123</Tranzakcio>
        </Megveszi>

        <Megveszi EtelIDRef="2" VIDRef="2">
            <Tranzakcio>456</Tranzakcio>
        </Megveszi>

        <Megveszi EtelIDRef="3" VIDRef="3">
            <Tranzakcio>789</Tranzakcio>
        </Megveszi>
    </Vasarlasok>

    <Vevok>
        <Vevo VID="1">
            <Nev>Nagy Ferenc</Nev>
            <Telefonszam>06702586633</Telefonszam>
            <Cim>
                <Irszam>3519</Irszam>
                <Varos>Miskolc</Varos>
                <Utca>Fenyő utca</Utca>
                <Hazszam>23</Hazszam>
            </Cim>
        </Vevo>

        <Vevo VID="2">
            <Nev>Novák Balázs</Nev>
            <Telefonszam>06304455982</Telefonszam>
            <Cim>
                <Irszam>1149</Irszam>
                <Varos>Budapest</Varos>
                <Utca>Vezér utca</Utca>
                <Hazszam>149</Hazszam>
            </Cim>
        </Vevo>

        <Vevo VID="3">
            <Nev>Szabó Dominik</Nev>
            <Telefonszam>06205981274</Telefonszam>
            <Cim>
                <Irszam>3530</Irszam>
                <Varos>Miskolc</Varos>
                <Utca>Pattantyús utca</Utca>
                <Hazszam>14</Hazszam>
            </Cim>
        </Vevo>
    </Vevok>

</Etteremlanc>
```

1.5. Az XML dokumentum alapján XML Schema készítése:

Következik az XML Schema, amely meghatározza a fenti XMLdokumentum elemeit. Az elemek lehetnek egyszerű (elemi) jelölő elem vagy komplex (összetett) jelölő elem. Több element is **ComplexType**, mert más elemeket (gyerek) tartalmaz. Egyszerű element pl. **Nev**, komplex element pl. **Tulajdonos_tipus**. A XMLSchema készítése során saját típusokat is létrehoztam, amik segítik az ismétlődés elkerülését, és az átláthatóságot. Valamint könnyebben lehet rájuk hivatkozni.

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

    <xs:attribute name="TID" type="xs:integer"/>
    <xs:attribute name="TIDRef" type="xs:integer"/>
    <xs:attribute name="EtID" type="xs:integer"/>
    <xs:attribute name="EtIDRef" type="xs:integer"/>
    <xs:attribute name="EtelID" type="xs:integer"/>
    <xs:attribute name="EtelIDRef" type="xs:integer"/>
    <xs:attribute name="VID" type="xs:integer"/>
    <xs:attribute name="VIDRef" type="xs:integer"/>

    1 reference
    <xs:complexType name="Tulajdonos_tipus">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string"/>
            <xs:element name="Eletkor" type="xs:integer"/>
            <xs:element name="Szulido" type="xs:integer"/>

        </xs:sequence>
        <xs:attribute ref="TID" use="required"/>
    </xs:complexType>

    1 reference
    <xs:complexType name="Etterem_tipus">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string"/>
            <xs:element name="Telefonszam" type="xs:integer"/>
            <xs:element name="Cim" type="Cim_tipus" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute ref="EtID" use="required"/>
        <xs:attribute ref="TIDRef" use="required"/>
    </xs:complexType>
```

```
  2 references
  <xs:complexType name="Cim_tipus">
      <xs:sequence>
          <xs:element name="Irszam" type="xs:string"/>
          <xs:element name="Varos" type="xs:string"/>
          <xs:element name="Utca" type="xs:string"/>
          <xs:element name="Hazszam" type="xs:integer"/>
      </xs:sequence>
  </xs:complexType>

  1 reference
  <xs:complexType name="Keszitи_tipus">
      <xs:sequence>
          <xs:element name="Stilus" type="xs:string"/>
      </xs:sequence>
      <xs:attribute ref="EtelID" use="required"/>
      <xs:attribute ref="EtelIDRef" use="required"/>
  </xs:complexType>

  1 reference
  <xs:complexType name="Etel_tipus">
      <xs:sequence>
          <xs:element name="Nev" type="xs:string"/>
          <xs:element name="Ar" type="xs:integer"/>
          <xs:element name="Koret" type="xs:string" maxOccurs="unbounded" minOccurs="0"/
      </xs:sequence>
      <xs:attribute ref="EtelID" use="required"/>
  </xs:complexType>
```

```
1 reference
<xs:complexType name="Megveszi_tipus">
    <xs:sequence>
        <xs:element name="Tranzakcio" type="xs:integer"/>
    </xs:sequence>
    <xs:attribute ref="EtelIDRef" use="required"/>
    <xs:attribute ref="VIDRef" use="required"/>
</xs:complexType>

1 reference
<xs:complexType name="Vevo_tipus">
    <xs:sequence>
        <xs:element name="Nev" type="xs:string"/>
        <xs:element name="Telefonszam" type="xs:integer"/>
        <xs:element name="Cim" type="Cim_tipus" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute ref="VID" use="required"/>
</xs:complexType>

1 reference
<xs:complexType name="Tulajdonosok_tipus">
    <xs:sequence>
        <xs:element name="Tulajdonos" type="Tulajdonos_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

```
1 reference
<xs:complexType name="Ettermek_tipus">
    <xs:sequence>
        <xs:element name="Etterem" type="Etterem_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

1 reference
<xs:complexType name="Keszitesek_tipus">
    <xs:sequence>
        <xs:element name="Keszit" type="Keszit_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

1 reference
<xs:complexType name="Etelek_tipus">
    <xs:sequence>
        <xs:element name="Etel" type="Etel_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

1 reference
<xs:complexType name="Vasarlasok_tipus">
    <xs:sequence>
        <xs:element name="Megveszi" type="Megveszi_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

```
1 reference
<xs:complexType name="Vevok_tipus">
    <xs:sequence>
        <xs:element name="Vevo" type="Vevo_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="Etteremlanc">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Tulajdonosok" type="Tulajdonosok_tipus"/>
            <xs:element name="Ettermek" type="Ettermek_tipus"/>
            <xs:element name="Keszitesek" type="Keszitesek_tipus"/>
            <xs:element name="Etelek" type="Etelek_tipus"/>
            <xs:element name="Vasarlasok" type="Vasarlasok_tipus"/>
            <xs:element name="Vevok" type="Vevok_tipus"/>
        </xs:sequence>
    </xs:complexType>

    <xs:key name="Tulajdonos_EKulcs">
        <xs:selector xpath="Tulajdonosok/Tulajdonos"/>
        <xs:field xpath="@TID"/>
    </xs:key>
```

```
        <xs:key name="Etterem_EKulcs">
            <xs:selector xpath="Ettermek/Etterem"/>
            <xs:field xpath="@PiID"/>
        </xs:key>

        <xs:key name="Etel_EKulcs">
            <xs:selector xpath="Etelek/Etle"/>
            <xs:field xpath="@PID"/>
        </xs:key>

        <xs:key name="Vevo_EKulcs">
            <xs:selector xpath="Vevok/Vevo"/>
            <xs:field xpath="@VID"/>
        </xs:key>

        <xs:keyref name="Tulajdonos_IKulcs" refer="Tulajdonos_EKulcs">
            <xs:selector xpath="Ettermek/Etterem"/>
            <xs:field xpath="@TIDRef"/>
        </xs:keyref>

        <xs:keyref name="Etterem_IKulcs" refer="Etterem_EKulcs">
            <xs:selector xpath="Reszvetelek/Reszvetel"/>
            <xs:field xpath="@PiIDRef"/>
        </xs:keyref>

        <xs:keyref name="Etel_IKulcs" refer="Etel_EKulcs">
            <xs:selector xpath="Vasarlasok/Megveszi"/>
            <xs:field xpath="@PIDRef"/>
        </xs:keyref>

        <xs:keyref name="Vevo_IKulcs" refer="Vevo_EKulcs">
            <xs:selector xpath="Vasarlasok/Megveszi"/>
            <xs:field xpath="@VIDRef"/>
        </xs:keyref>
    </xs:element>
</xs:schema>
```

2. Feladat

2.1. DOM Read:

A feladat egy DOM program készítése XML dokumentum adatainak adminisztrálása alapján (comment-tel együtt). A DOM Read kiírja az XML dokumentumom adatait.

```java
package hu.domparse.ida58u;

import java.io.File;

public class DomReadEJX162 {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        File xml = new File("C:\\Egyetem\\XML\\IDA58U_XML_gyak\\XMLTaskIDA58U\\DOMParseIDA58U\\bin\\hu\\domparse\\ida58u\\XMLIDA58U.xml");


        // XML fájl DOM document alakítása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse(xml);

        // DOM document átalakítása DOM DocumentTraversal formába
        DocumentTraversal traversal = (DocumentTraversal) document;

        TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),
                NodeFilter.SHOW_ELEMENT | NodeFilter.SHOW_TEXT, null, true);

        //DOM bejárása és kiíratása
        DomTraverser.traverseLevel(walker, "");
    }
```

```java
        private static void printElementAttributes(NamedNodeMap attributes) {
            int length = attributes.getLength();

            if (length > 0) {
                System.out.print(" [ ");

                for (int i = 0; i < length; i++) {
                    Node attribute = attributes.item(i);

                    System.out.printf("%s=%s%s", attribute.getNodeName(), attribute.getNodeValue(),
                            i != length - 1 ? ", " : "");
                }

                System.out.println(" ]");
            } else {
                System.out.println();
            }
        }

        private static void printTextNode(Node node, String indent) {
            String content_trimmed = node.getTextContent().trim();

            if (content_trimmed.length() > 0) {
                System.out.print(indent);
                System.out.printf("{ %s }%n", content_trimmed);
            }
        }
    }
}
```

```java
private static class DomTraverser {
    public static void traverseLevel(TreeWalker walker, String indent) {
        // Aktuális csomópont
        Node node = walker.getCurrentNode();

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            printElementNode(node, indent);
        } else {
            printTextNode(node, indent);
        }

        // Rekurzívan meghívjuk a bejárást a DOM fában
        for (Node n = walker.firstChild(); n != null; n = walker.nextSibling()) {
            traverseLevel(walker, indent + "    ");
        }

        walker.setCurrentNode(node);
    }

    private static void printElementNode(Node node, String indent) {
        System.out.print(indent + node.getNodeName());

        printElementAttributes(node.getAttributes());
    }
```

2.1.1.Dom Read Output:

Etteremlanc [xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance, xsi:noNamespaceSchemaLocation=XMLSchemaEJX162.xsd]

Tulajdonosok
Tulajdonos [TID=1]
Nev
{ Nagy Béla }
Eletkor
{ 26 }
Szulido
{ 1997 }
Tulajdonos [TID=2]
Nev
{ Balázs Dominika }
Eletkor
{ 25 }
Szulido
{ 1998 }
Tulajdonos [TID=3]
Nev
{ Kiss Imre }
Eletkor
{ 23 }
Szulido
{ 2000 }
Ettermek
Etterem [EtID=1, TIDRef=1]

Nev
{ Italian Stallion }
Telefonszam
{ 012432123 }
Cim
Irszam
{ 1256 }
Varos
{ Budapest }
Utca
{ Ferenciek útja }
Hazszam
{ 14 }
Etterem [EtID=2, TIDRef=2]
Nev
{ New York Style }
Telefonszam
{ 05618478 }
Cim
Irszam
{ 3525 }
Varos
{ Miskolc }
Utca
{ Széchenyi István út }
Hazszam
{ 26 }
Etterem [EtID=3, TIDRef=3]
Nev
{ Véndiófa étterem }
Telefonszam
{ 063025698812 }
Cim
Irszam
{ 3540 }
Varos
{ Miskolc }
Utca
{ Árpád utca }
Hazszam
{ 6 }
Keszitesek
Kesziti [EtelID=1, EtelIDRef=1]
Stilus
{ Előétel }
Kesziti [EtelID=2, EtelIDRef=2]
Stilus
{ Főétel }
Kesziti [EtelID=3, EtelIDRef=3]
Stilus
{ Desszert }
Etelek
Etel [EtelID=1]

Nev
{ Pacalpörkölt }
Ar
{ 4500 }
Koret
{ Krumplipüré }
Etel [EtelID=2]
Nev
{ Rántotthús }
Ar
{ 3200 }
Koret
{ Sültkrumpli }
Koret
{ Káposztasaláta }
Etel [EtelID=3]
Nev
{ Spagetti }
Ar
{ 2800 }
Etel [EtelID=4]
Nev
{ Kacsacomb }
Ar
{ 4600 }
Koret
{ Steakburgonya }
Koret
{ Mártás }
Koret
{ Céklasaláta }
Etel [EtelID=5]
Nev
{ Burjósült }
Ar
{ 6600 }
Koret
{ Hagymakarika }
Koret
{ Grillezett zöldség }
Vasarlasok
Megveszi [EtelIDRef=1, VIDRef=1]
Tranzakcio
{ 123 }
Megveszi [EtelIDRef=2, VIDRef=2]
Tranzakcio
{ 456 }
Megveszi [EtelIDRef=3, VIDRef=3]
Tranzakcio
{ 789 }
Vevok
Vevo [VID=1]
Nev

{ Nagy Ferenc }
Telefonszam
{ 06702586633 }
Cim
Irszam
{ 3519 }
Varos
{ Miskolc }
Utca
{ Fenyő utca }
Hazszam
{ 23 }
Vevo [VID=2]
Nev
{ Novák Balázs }
Telefonszam
{ 06304455982 }
Cim
Irszam
{ 1149 }
Varos
{ Budapest }
Utca
{ Vezér utca }
Hazszam
{ 149 }
Vevo [VID=3]
Nev
{ Szabó Dominik }
Telefonszam
{ 06205981274 }
Cim
Irszam
{ 3530 }
Varos
{ Miskolc }
Utca
{ Pattantyús utca }
Hazszam
{ 14 }

2.2. DOM Modify:

A feladat egy DOM program készítése, amely módosít néhány elemet az XML dokumentumomban. Én Kiss Imre nevét írtam át Nagy Imrére, valamint a 3000 Ft-nál drágább ételek árát csökkentem 10%-kal.

```java
package hu.domparse.ida58u;

import java.io.File;

public class DomModifyEJX162 {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException,
    XPathExpressionException, DOMException, ParseException {

        File xml = new File("C:\\Egyetem\\XML\\IDA58U_XML_gyak\\XMLTaskIDA58U\\DOMParseIDA58U\\bin\\hu\\domparse\\ida58u\\XMLIDA58U.xml");

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse(xml);

        // a DOM document módosítása
        DomModifier.modifyDom(document);

        // DOM document átalakatisa DOM DocumentTraversal formába
        DocumentTraversal traversal = (DocumentTraversal) document;

        //TreeWalker inicializálása
        TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),
                NodeFilter.SHOW_ELEMENT | NodeFilter.SHOW_TEXT, null, true);

        //DOM bejárása
        DomTraverser.traverseLevel(walker, "");

    }
```

```java
    private static class DomModifier {
        public static void modifyDom(Document document) throws XPathExpressionException, DOMException, ParseException {
            XPathFactory factory = XPathFactory.newInstance();
            XPath xpath = factory.newXPath();

            // 1.) Kiss Imre nevenek megváltoztatása Nagy Imrére
            Node tulajdonos = (Node) xpath.evaluate("//Tulajdonos[./Nev='Kiss Imre']",
                    document, XPathConstants.NODE);

            tulajdonos.setTextContent("Nagy Imre");

            // 2.) Minden 3000 forintnál drágább ételnek az ára csökken 10%-al
            NodeList etelek = (NodeList) xpath.evaluate("//Etel[./Ar>3000]/Ar", document, XPathConstants.NODESET);
            System.out.println(etelek);
            for (int i = 0; i < etelek.getLength(); i++) {
                Node etel = etelek.item(i);

                double price = Double.parseDouble(etel.getTextContent());
                etel.setTextContent(Double.toString(price * 0.9));

            }
        }
    }
```

```java
    private static class DomTraverser {
        public static void traverseLevel(TreeWalker walker, String indent) {
            //AktuÁlis csomópont
            Node node = walker.getCurrentNode();

            if (node.getNodeType() == Node.ELEMENT_NODE) {
                printElementNode(node, indent);
            } else {
                printTextNode(node, indent);
            }

            // Rekurzívan meghívjuk a bejárást a DOM fában
            for (Node n = walker.firstChild(); n != null; n = walker.nextSibling()) {
                traverseLevel(walker, indent + "    ");
            }

            walker.setCurrentNode(node);
        }

        private static void printElementNode(Node node, String indent) {
            System.out.print(indent + node.getNodeName());

            printElementAttributes(node.getAttributes());
        }
```

```
    private static void printElementAttributes(NamedNodeMap attributes) {
        int length = attributes.getLength();

        if (length > 0) {
            System.out.print(" [ ");

            for (int i = 0; i < length; i++) {
                Node attribute = attributes.item(i);

                System.out.printf("%s=%s%s", attribute.getNodeName(), attribute.getNodeValue(),
                        i != length - 1 ? ", " : "");
            }

            System.out.println(" ]");
        } else {
            System.out.println();
        }
    }

    private static void printTextNode(Node node, String indent) {
        String content_trimmed = node.getTextContent().trim();

        if (content_trimmed.length() > 0) {
            System.out.print(indent);
            System.out.printf("{ %s }%n", content_trimmed);
        }
    }
}
```

2.2.1.DOM Modify Output:

com.sun.org.apache.xml.internal.dtm.ref.DTMNodeList@7823a2f9

Etteremlanc [xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance, xsi:noNamespaceSchemaLocation=XMLSchemaEJX162.xsd]

Tulajdonosok

Tulajdonos [TID=1]

Nev

{ Nagy Béla }

Eletkor

{ 26 }

Szulido

{ 1997 }

Tulajdonos [TID=2]

Nev

{ Balázs Dominika }

Eletkor

{ 25 }
Szulido
{ 1998 }
Tulajdonos [TID=3]
{ Nagy Imre }
Ettermek
Etterem [EtID=1, TIDRef=1]
Nev
{ Italian Stallion }
Telefonszam
{ 012432123 }
Cim
Irszam
{ 1256 }
Varos
{ Budapest }
Utca
{ Ferencziek útja }
Hazszam
{ 14 }
Etterem [EtID=2, TIDRef=2]
Nev
{ New York Style }
Telefonszam
{ 05618478 }
Cim
Irszam
{ 3525 }
Varos
{ Miskolc }
Utca
{ Széchenyi István út }
Hazszam
{ 26 }
Etterem [EtID=3, TIDRef=3]
Nev
{ Véndiófa étterem }
Telefonszam

{ 063025698812 }
Cim
Irszam
{ 3540 }
Varos
{ Miskolc }
Utca
{ Árpád utca }
Hazszam
{ 6 }
Keszitesek
Kesziti [EtelID=1, EtelIDRef=1]
Stilus
{ Előétel }
Kesziti [EtelID=2, EtelIDRef=2]
Stilus
{ Főétel }
Kesziti [EtelID=3, EtelIDRef=3]
Stilus
{ Desszert }
Etelek
Etel [EtelID=1]
Nev
{ Pacalpörkölt }
Ar
{ 4050.0 }
Koret
{ Krumplipüré }
Etel [EtelID=2]
Nev
{ Rántotthús }
Ar
{ 2880.0 }
Koret
{ Sültkrumpli }
Koret
{ Káposztasaláta }
Etel [EtelID=3]

Nev
{ Spagetti }
Ar
{ 2520.0 }
Etel [EtelID=4]
Nev
{ Kacsacomb }
Ar
{ 4140.0 }
Koret
{ Steakburgonya }
Koret
{ Mártás }
Koret
{ Céklasaláta }
Etel [EtelID=5]
Nev
{ Burjósült }
Ar
{ 5940.0 }
Koret
{ Hagymakarika }
Koret
{ Grillezett zöldség }
Vasarlasok
Megveszi [EtelIDRef=1, VIDRef=1]
Tranzakcio
{ 123 }
Megveszi [EtelIDRef=2, VIDRef=2]
Tranzakcio
{ 456 }
Megveszi [EtelIDRef=3, VIDRef=3]
Tranzakcio
{ 789 }
Vevok
Vevo [VID=1]
Nev
{ Nagy Ferenc }

Telefonszam
{ 06702586633 }
Cim
Irszam
{ 3519 }
Varos
{ Miskolc }
Utca
{ Fenyő utca }
Hazszam
{ 23 }
Vevo [VID=2]
Nev
{ Novák Balázs }
Telefonszam
{ 06304455982 }
Cim
Irszam
{ 1149 }
Varos
{ Budapest }
Utca
{ Vezér utca }
Hazszam
{ 149 }
Vevo [VID=3]
Nev
{ Szabó Dominik }
Telefonszam
{ 06205981274 }
Cim
Irszam
{ 3530 }
Varos
{ Miskolc }
Utca
{ Pattantyús utca }
Hazszam

2.3. DOM Query:

A DOM Query file-ban lekérdezem az XML file-ból, az összes Étterem attributumát, valamint annak az Étteremnek a nevét, amely Pesten van.

```java
package hu.domparse.ida58u;

import java.io.File;

public class DomQueryEJX162 {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        File file = new File("C:\\Egyetem\\XML\\IDA58U_XML_gyak\\XMLTaskIDA58U\\DOMParseIDA58U\\bin\\hu\\domparse\\ida58u\\XMLIDA58U.xml");
        // Parse-ol□s
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

        Document doc = dBuilder.parse(file);
        doc.getDocumentElement().normalize();
        // Root element kiiratsa
        System.out.print("Gyoker element: ");
        System.out.println(doc.getDocumentElement().getNodeName());
        NodeList nList = doc.getElementsByTagName("Etterem");

        // Minden etterem attribútum kiiratasa

        System.out.println("ETTERMEK");
        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i);
            System.out.println("\nElement nev : " + node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                System.out.println("ID:" + elem.getAttribute("EtID"));
                NodeList nList2 = elem.getChildNodes();
                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE) {
                        Element elem2 = (Element) node2;
                        if (!node2.getNodeName().equals("Cim")) {
                            System.out.println(node2.getNodeName() + " : " + node2.getTextContent());
                        } else {
                            System.out.println("Cim:");

                            NodeList nList3 = elem2.getChildNodes();
                            for (int k = 0; k < nList3.getLength(); k++) {
                                Node node3 = nList3.item(k);

                                if (node3.getNodeType() == Node.ELEMENT_NODE) {

                                    System.out.println("    " + node3.getNodeName() + " : " + node3.getTextContent());
                                }
                            }
                        }
                    }
                }
            }
        }
```

```
        //Kiirja annak az etteremnek a nevet, ami Pesten van
        System.out.println("\nPESTI ETTEREM\n");
        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                NodeList nList2 = elem.getChildNodes();
                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE) {
                        Element elem2 = (Element) node2;
                        NodeList nList3 = elem2.getChildNodes();
                        for (int k = 0; k < nList3.getLength(); k++) {
                            Node node3 = nList3.item(k);
                            if (node3.getNodeType() == Node.ELEMENT_NODE) {
                                if (node3.getNodeName().equals("Varos")) {
                                    if (node3.getTextContent().equals("Budapest")) {
                                        node2 = nList2.item(1);
                                        System.out.println(node2.getNodeName() + ": " + node2.getTextContent());
                                    }

                                }

                            }
                        }

                    }
                }
            }
        }
    }
}
```

2.3.1.DOM Query Output:

Gyoker element: Etteremlanc

ETTERMEK

Element nev : Etterem
ID:1
Nev : Italian Stallion
Telefonszam : 012432123
Cim:
Irszam : 1256
Varos : Budapest
Utca : Ferencziek útja
Hazszam : 14

Element nev : Etterem
ID:2
Nev : New York Style
Telefonszam : 05618478
Cim:
Irszam : 3525
Varos : Miskolc
Utca : Széchenyi István út
Hazszam : 26

Element nev : Etterem
ID:3

Nev : Véndiófa étterem
Telefonszam : 063025698812
Cim:
Irszam : 3540
Varos : Miskolc
Utca : Árpád utca
Hazszam : 6

PESTI ETTEREM

Nev: Italian Stallion