Jegyzőkönyv
Adatkezelés XML környezetben
Féléves feladat

Szabó Bálint
EJX162

Pizzéria hálózat modellezése

Tartalomjegyzék

1. Feladat

1.1. A feladat leírása:

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

A **Pizzéria** egyedhez tartozik a **Pikód** tulajdonság, amely az elsődleges kulcsa az egyednek. Ezen kívül még a **Telefonszám**, amely egy többértékű tulajdonsága, hiszem egy Pizzériának 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.

A **Pizza** egyed **Pkód** tulajdonsága kulcsként funkcionál, másik két tulajdonsága a **Név** és az **Ár**, valamint a **Feltét**, amely többértékű tulajdonság, hiszen egy pizzán lehet egyszerre sajt is a feltét meg szalámi 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.

A **Pizzéria** és a **Pizza** egyedek között a **Készíti** kapcsolat van, amely egy egy **több a többhöz kapcsolat (N:M),** hiszen egy pizzéria több pizzát is készíthet, és egy pizzát több pizzéria is el tud készíteni.

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

A **Pizza** és a **Vevő** egyedek között **Megveszi** kapcsolat van, ami szintén egy **több a többhöz kapcsolat (N:M),** mivel egy féle pizzát több vevő is megvehet, valamint egy vevő több pizzériától is vásárolhat.

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:

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

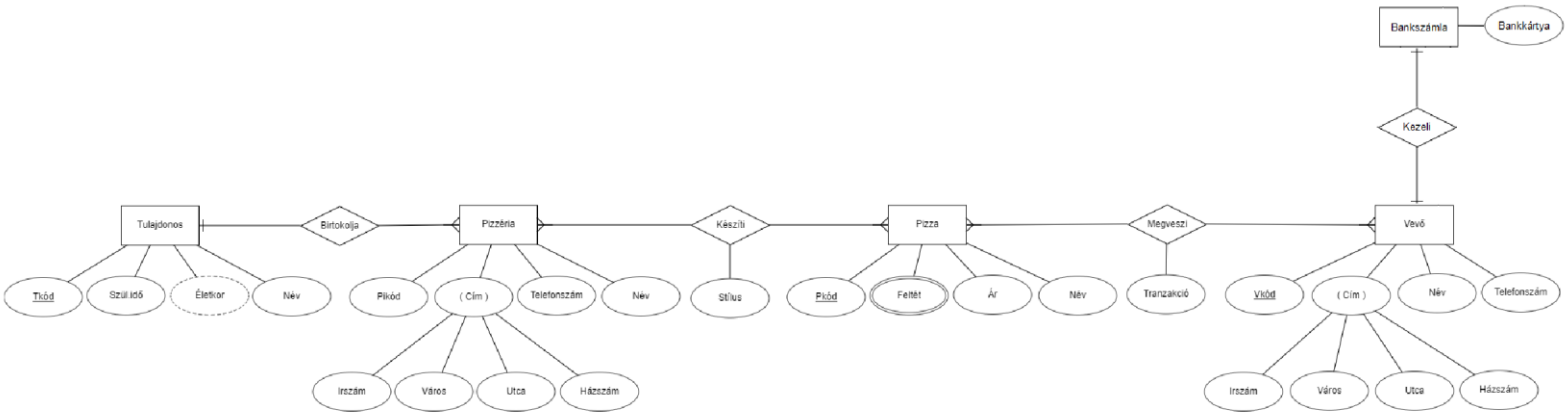
- **Pizza:**
  Attribútumok: Pkód (elsődleges kulcs), **Név**, **Ár**, **Feltét** (többértékű tulajdonság)

- **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)** a **Pizzéria** és a **Pizza** egyedek között.
- **Birtokolja:** Egy a többhöz kapcsolat**(1:N)** a **Pizzéria** és a **Tulajdonos** egyedek között.
- **Megveszi:** Több a többhöz kapcsolat**(N:M)** a **Pizza** és a **Vevő** egyedek között.

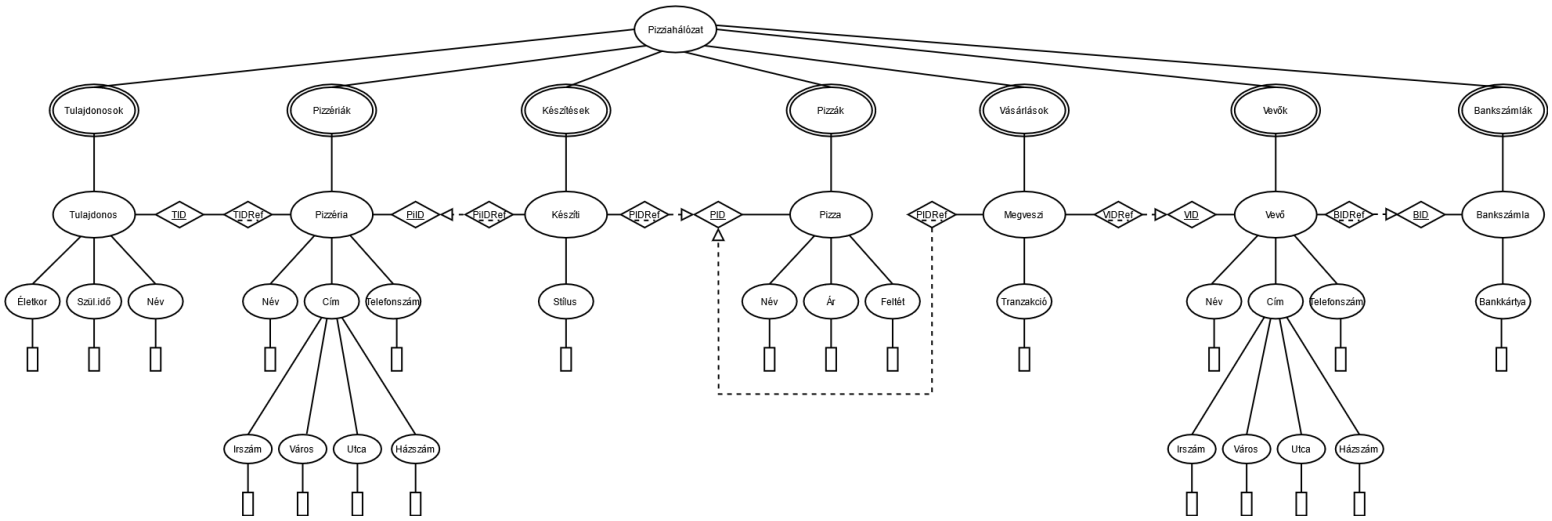


1.3. XDM Modell:

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

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

Ezen kívül a két több-több (N:M) kapcsolatnál létrejött két objektum. A Pizzériák és a Pizzák objektum közötti kapcsolatnál létrejött a Készítések objektum, valamint a Pizzák 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 a **Pizzériák** elem) kifejezéseket használják. Az XML dokumentum gyökéreleme a Pizzahálózat.

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

    <Tulajdonosok>

        <Tulajdonos TID="1">
            <Nev>Nagy Béla</Nev>
            <Eletkor>21</Eletkor>
            <Szulido>1997</Szulido>
        </Tulajdonos>

        <Tulajdonos TID="2">
            <Nev>Lőre István</Nev>
            <Eletkor>27</Eletkor>
            <Szulido>1998</Szulido>
        </Tulajdonos>

        <Tulajdonos TID="3">
            <Nev>Kiss Imre</Nev>
            <Eletkor>45</Eletkor>
            <Szulido>2000</Szulido>
        </Tulajdonos>

    </Tulajdonosok>

    <Pizzeriak>
        <Pizzeria PiID="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>
```

```
        </Pizzeria>

        <Pizzeria PiID="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>
        </Pizzeria>

        <Pizzeria PiID="3" TIDRef="3">
            <Nev>Lábas</Nev>
            <Telefonszam>035637923</Telefonszam>
            <Cim>
                <Irszam>3540</Irszam>
                <Varos>Alsózsolca</Varos>
                <Utca>Fő út</Utca>
                <Hazszam>6</Hazszam>
            </Cim>
        </Pizzeria>
    </Pizzeriak>

    <Keszitesek>
        <Kesziti PIDRef="1" PiIDRef="1">
            <Stilus>Al Dente</Stilus>
        </Kesziti>

        <Kesziti PIDRef="2" PiIDRef="2">
            <Stilus>Közepes</Stilus>
```

```
        </Keszitesek>

        <Kesziti PIDRef="3" PiIDRef="3">
            <Stilus>Pirított</Stilus>
        </Kesziti>
    </Keszitesek>

    <Pizzak>
        <Pizza PID="1">
            <Nev>Margareta</Nev>
            <Ar>1200</Ar>
            <Feltet>Sajt</Feltet>
        </Pizza>

        <Pizza PID="2">
            <Nev>Sonkás</Nev>
            <Ar>1400</Ar>
            <Feltet>Sajt</Feltet>
            <Feltet>Sonka</Feltet>
        </Pizza>

        <Pizza PID="3">
            <Nev>Szalámis</Nev>
            <Ar>1400</Ar>
            <Feltet>Sajt</Feltet>
            <Feltet>Szalámi</Feltet>
        </Pizza>

        <Pizza PID="4">
            <Nev>Sonkás-Kukoricás</Nev>
            <Ar>1600</Ar>
            <Feltet>Sajt</Feltet>
            <Feltet>Sonka</Feltet>
```

```
            <Feltet>Kukorica</Feltet>
        </Pizza>

        <Pizza PID="5">
            <Nev>Lábas Speciál</Nev>
            <Ar>2100</Ar>
            <Feltet>Sajt</Feltet>
            <Feltet>BBQ szósz</Feltet>
            <Feltet>Pulled Pork</Feltet>
            <Feltet>Bacon</Feltet>
        </Pizza>
    </Pizzak>

    <Vasarlasok>
        <Megveszi PIDRef="1" VIDRef="1">
            <Tranzakcio>123</Tranzakcio>
        </Megveszi>

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

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

    <Vevok>
        <Vevo VID="1">
            <Nev>Nagy Ferenc</Nev>
            <Telefonszam>77895412</Telefonszam>
            <Cim>
                <Irszam>3519</Irszam>
```

```
135                 <Varos>Miskolc</Varos>
136                 <Utca>Fenyő utca</Utca>
137                 <Hazszam>23</Hazszam>
138             </Cim>
139         </Vevo>
140
141         <Vevo VID="2">
142             <Nev>Novák Balázs</Nev>
143             <Telefonszam>078954412</Telefonszam>
144             <Cim>
145                 <Irszam>1149</Irszam>
146                 <Varos>Budapest</Varos>
147                 <Utca>Vezér utca</Utca>
148                 <Hazszam>149</Hazszam>
149             </Cim>
150         </Vevo>
151
152         <Vevo VID="3">
153             <Nev>Trab Antal</Nev>
154             <Telefonszam>012258</Telefonszam>
155             <Cim>
156                 <Irszam>3530</Irszam>
157                 <Varos>Miskolc</Varos>
158                 <Utca>Pattantyús utca</Utca>
159                 <Hazszam>14</Hazszam>
160             </Cim>
161         </Vevo>
162     </Vevok>
163 </Pizzahalozat>
```

1.5. Az XML dokumentum alapján XML Schema készítése:

Következik az XML Schema, amely meghatározza a fenti XML-dokumentum 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="PiID" type="xs:integer"/>
    <xs:attribute name="PiIDRef" type="xs:integer"/>
    <xs:attribute name="PID" type="xs:integer"/>
    <xs:attribute name="PIDRef" type="xs:integer"/>
    <xs:attribute name="VID" type="xs:integer"/>
    <xs:attribute name="VIDRef" type="xs:integer"/>

    <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>

    <xs:complexType name="Pizzeria_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="PiID" use="required"/>
        <xs:attribute ref="TIDRef" use="required"/>
    </xs:complexType>

    <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>

    <xs:complexType name="Kesziti_tipus">
        <xs:sequence>
            <xs:element name="Stilus" type="xs:string"/>
        </xs:sequence>
        <xs:attribute ref="PIDRef" use="required"/>
        <xs:attribute ref="PiIDRef" use="required"/>
    </xs:complexType>

    <xs:complexType name="Pizza_tipus">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string"/>
            <xs:element name="Ar" type="xs:integer"/>
            <xs:element name="Feltet" type="xs:string" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute ref="PID" use="required"/>
    </xs:complexType>

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

    <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>

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

    <xs:complexType name="Pizzeriak_tipus">
        <xs:sequence>
            <xs:element name="Pizzeria" type="Pizzeria_tipus" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="Keszitesek_tipus">
        <xs:sequence>
            <xs:element name="Kesziti" type="Kesziti_tipus" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="Pizzak_tipus">
        <xs:sequence>
            <xs:element name="Pizza" type="Pizza_tipus" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

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

```
104             <xs:element name="Megveszi" type="Megveszi_tipus" maxOccurs="unbounded"/>
105         </xs:sequence>
106     </xs:complexType>
107
108     <xs:complexType name="Vevok_tipus">
109         <xs:sequence>
110             <xs:element name="Vevo" type="Vevo_tipus" maxOccurs="unbounded"/>
111         </xs:sequence>
112     </xs:complexType>
113
114     <xs:element name="Pizzahalozat">
115         <xs:complexType>
116             <xs:sequence>
117                 <xs:element name="Tulajdonosok" type="Tulajdonosok_tipus"/>
118                 <xs:element name="Pizzeriak" type="Pizzeriak_tipus"/>
119                 <xs:element name="Keszitesek" type="Keszitesek_tipus"/>
120                 <xs:element name="Pizzak" type="Pizzak_tipus"/>
121                 <xs:element name="Vasarlasok" type="Vasarlasok_tipus"/>
122                 <xs:element name="Vevok" type="Vevok_tipus"/>
123             </xs:sequence>
124         </xs:complexType>
125
126         <xs:key name="Talajdonos_EKulcs">
127             <xs:selector xpath="Tulajdonosok/Tulajdonos"/>
128             <xs:field xpath="@TID"/>
129         </xs:key>
130
131         <xs:key name="Pizzeria_EKulcs">
132             <xs:selector xpath="Pizzeriak/Pizzeria"/>
133             <xs:field xpath="@PiID"/>
134         </xs:key>
135
136         <xs:key name="Pizza_EKulcs">
137             <xs:selector xpath="Pizzak/Pizza"/>
```

```
138             <xs:field xpath="@PID"/>
139         </xs:key>
140
141         <xs:key name="Vevo_EKulcs">
142             <xs:selector xpath="Vevok/Vevo"/>
143             <xs:field xpath="@VID"/>
144         </xs:key>
145
146         <xs:keyref name="Tualjdonos_IKulcs" refer="Talajdonos_EKulcs">
147             <xs:selector xpath="Pizzeriak/Pizzeria"/>
148             <xs:field xpath="@TIDRef"/>
149         </xs:keyref>
150
151         <xs:keyref name="Pizzeria_IKulcs" refer="Pizzeria_EKulcs">
152             <xs:selector xpath="Reszvetelek/Reszvetel"/>
153             <xs:field xpath="@PiIDRef"/>
154         </xs:keyref>
155
156         <xs:keyref name="Pizza_IKulcs" refer="Pizza_EKulcs">
157             <xs:selector xpath="Vasarlasok/Megveszi"/>
158             <xs:field xpath="@PIDRef"/>
159         </xs:keyref>
160
161         <xs:keyref name="Vevo_IKulcs" refer="Vevo_EKulcs">
162             <xs:selector xpath="Vasarlasok/Megveszi"/>
163             <xs:field xpath="@VIDRef"/>
164         </xs:keyref>
165     </xs:element>
166 </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.ejx162;

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

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

import org.w3c.dom.Document;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.traversal.DocumentTraversal;
import org.w3c.dom.traversal.NodeFilter;
import org.w3c.dom.traversal.TreeWalker;
import org.xml.sax.SAXException;

public class DomReadEJX162 {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        File xml = new
File("C:\\Users\\Bálint\\Desktop\\EGYETEM\\2022-23. 5.félév\\xml_beadando\\DOMParseEJX162"
        + "\\src\\hu\\domparse\\ejx162\\XMLEJX162.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 inicializálása
        TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),
                NodeFilter.SHOW_ELEMENT | NodeFilter.SHOW_TEXT, null, true);

        //DOM bejárása és kiíratása
```

```
        DomTraverser.traverseLevel(walker, "");
    }

    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);
            }
        }
    }
}
```

2.1.1.Dom Read Output:

model [xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance,
xsi:noNamespaceSchemaLocation=XMLSchemaH7PG8U.xsd]
  Tulajdonosok
    Tulajdonos [Tkod=1]
      Nev
        { Nagy Béla }
      Szulido
        Ev
          { 1987 }
        Honap
          { 03 }
        Nap
          { 12 }
    Tulajdonos [Tkod=2]
Nev
        { Lőre István }
      Szulido
        Ev
          { 1967 }
        Honap
          { 10 }
        Nap
          { 25 }
    Tulajdonos [Tkod=3]
      Nev
        { Kiss Imre }
      Szulido
        Ev
          { 1980 }
        Honap
          { 08 }
        Nap
          { 20 }
  Pizzeriak
    Pizzeria [Pikod=1]
      Nev
        { Italian Stallion }
      Telefonszam
        { 012432123 }
      Cim
        Irszam
          { 1256 }
        Varos
          { Budapest }
        Utca
          { Ferencziek útja }
        Hazszam
          { 14 }
    Pizzeria [Pikod=2]
      Nev
        { New York Style }
      Telefonszam
        { 05618478 }
      Cim
        Irszam

{ 3525 }
Varos
{ Miskolc }
Utca
{ Széchenyi István út }
Hazszam
{ 26 }
Pizzeria [Pikod=3]
Nev
{ Lábas }
Telefonszam
{ 035637923 }
Cim
Irszam
{ 3540 }
Varos
{ Alsózsolca }
Utca
{ Fő út }
Hazszam
{ 6 }
Pizzak
Pizza [Pkod=1]
Nev
{ Margareta }
Ar
{ 1200 }
Feltet
{ Sajt }
Pizza [Pkod=2]
Nev
{ Sonkás }
Ar
{ 1400 }
Feltet
{ Sajt }
Feltet
{ Sonka }
Pizza [Pkod=3]
Nev
{ Szalámis }
Ar
{ 1400 }
Feltet
{ Sajt }
Feltet
{ Szalámi }
Pizza [Pkod=4]
Nev
{ Sonkás-Kukoricás }
Ar
{ 1600 }
Feltet
{ Sajt }
Feltet

{ Sonka }
Feltet
{ Kukorica }
Pizza [Pkod=5]
Nev
{ Lábas Speciál }
Ar
{ 2100 }
Feltet
{ Sajt }
Feltet
{ BBQ szósz }
Feltet
{ Pulled Pork }
Feltet
{ Bacon }
Vevok
Vevo [Vkod=1]
Nev
{ Nagy Ferenc }
Telefonszam
{ 77895412 }
Cim
Irszam
{ 3519 }
Varos
{ Miskolc }
Utca
{ Fenyő utca }
Hazszam
{ 23 }
Vevo [Vkod=2]
Nev
{ Novák Balázs }
Telefonszam
{ 078954412 }
Cim
Irszam
Tulajdonos_Pizzeria [Pizzeriaref=1, Tulajdonosref=1]
Tulajdonos_Pizzeria [Pizzeriaref=2, Tulajdonosref=2]
Tulajdonos_Pizzeria [Pizzeriaref=3, Tulajdonosref=2]
Pizzeria_Pizza_kapcsolok
Pizzeria_Pizza [Pizzaref=1, Pizzeriaref=1]
Pizzeria_Pizza [Pizzaref=3, Pizzeriaref=1]
Pizzeria_Pizza [Pizzaref=2, Pizzeriaref=2]
Pizzeria_Pizza [Pizzaref=4, Pizzeriaref=2]
Pizzeria_Pizza [Pizzaref=5, Pizzeriaref=3]
Pizza_Vevo_kapcsolok
Pizza_Vevo [Pizzaref=2, Vevoref=1]
Pizza_Vevo [Pizzaref=4, Vevoref=2]
Pizza_Vevo [Pizzaref=5, Vevoref=3]

2.2. DOM Modify:

A feladat egy DOM program készítése, amely módosít néhány elemet az XML dokumentumomban. Én Kiss Imre nevét írtam át Nagyobb Imrére, valamint a 1300 Ft-nál drágább pizzák árát csökkentem 10%-kal.

```java
package hu.domparse.ejx162;

import java.io.File;
import java.io.IOException;
import java.text.ParseException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

import org.w3c.dom.DOMException;
import org.w3c.dom.Document;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.traversal.DocumentTraversal;
import org.w3c.dom.traversal.NodeFilter;
import org.w3c.dom.traversal.TreeWalker;
import org.xml.sax.SAXException;

public class DomModifyEJX162 {

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

        File xml = new
File("C:\\Users\\Bálint\\Desktop\\EGYETEM\\2022-23. 5.félév\\xml_beadando"
        + "\\DOMParseEJX162\\src\\hu\\domparse\\ejx162\\XMLEJX162.xml");

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

        // a DOM document módosítása
```

```
        DomModifier.modifyDom(document);

        // DOM document átalakítása 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, "");

    }

    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 Nagyobb Imrére
            Node owner = (Node) xpath.evaluate("//Tulajdonos[./Nev='Kiss Imre']",
                    document, XPathConstants.NODE);

            owner.setTextContent("Nagyobb Imre");

            // 2.) Minden 1300 forintnál drágább pizzának az ára csökken 10%-al
            NodeList pizzas = (NodeList) xpath.evaluate("//Pizza[./Ar>1300]/Ar", document, XPathConstants.NODESET);
            System.out.println(pizzas);
            for (int i = 0; i < pizzas.getLength(); i++) {
                Node pizza = pizzas.item(i);

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

            }
        }
    }

    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);

```

```
111                     System.out.printf("%s=%s%s", attribute.getNodeName(), attribute.getNodeValue(),
112                             i != length - 1 ? ", " : "");
113                 }
114
115                 System.out.println(" ]");
116             } else {
117                 System.out.println();
118             }
119         }
120
121         private static void printTextNode(Node node, String indent) {
122             String content_trimmed = node.getTextContent().trim();
123
124             if (content_trimmed.length() > 0) {
125                 System.out.print(indent);
126                 System.out.printf("{ %s }%n", content_trimmed);
127             }
128         }
129     }
130
131 }
```

2.2.1.DOM Modify Output:

model [xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance,
xsi:noNamespaceSchemaLocation=XMLSchemaH7PG8U.xsd]
  Tulajdonosok
    Tulajdonos [Tkod=1]
      Nev
        { Nagy Béla }
      Szulido
        Ev
          { 1987 }
        Honap
          { 03 }
        Nap
          { 12 }
    Tulajdonos [Tkod=2]
      Nev
        { Lőre István }
      Szulido
        Ev
          { 1967 }
        Honap
          { 10 }
        Nap
          { 25 }
    Tulajdonos [Tkod=3]
      { Nagyobb Imre }
  Pizzeriak
    Pizzeria [Pikod=1]
      Nev
        { Italian Stallion }
      Telefonszam

{ 012432123 }
Cim
Irszam
{ 1256 }
Varos
{ Budapest }
Utca
{ Ferencziek útja }
Hazszam
{ 14 }
Pizzeria [Pikod=2]
Nev
{ New York Style }
Telefonszam
{ 05618478 }
Cim
Irszam
{ 3525 }
Varos
{ Miskolc }
Utca
{ Széchenyi István út }
Hazszam
{ 26 }
Pizzeria [Pikod=3]
Nev
{ Lábas }
Telefonszam
{ 035637923 }
Cim
Irszam
{ 3540 }
Varos
{ Alsózsolca }
Utca
{ Fő út }
Hazszam
{ 6 }
Pizzak
Pizza [Pkod=1]
Nev
{ Margareta }
Ar
{ 1200 }
Feltet
{ Sajt }
Pizza [Pkod=2]
Nev
{ Sonkás }
Ar

{ 1260.0 }
Feltet
{ Sajt }
Feltet
{ Sonka }
Pizza [Pkod=3]
Nev
{ Szalámis }
Ar
{ 1260.0 }
Feltet
{ Sajt }
Feltet
{ Szalámi }
Pizza [Pkod=4]
Nev
{ Sonkás-Kukoricás }
Ar
{ 1440.0 }
Feltet
{ Sajt }
Feltet
{ Sonka }
Feltet
{ Kukorica }
Pizza [Pkod=5]
Nev
{ Lábas Speciál }
Ar
{ 1890.0 }
Feltet
{ Sajt }
Feltet
{ BBQ szósz }
Feltet
{ Pulled Pork }
Feltet
{ Bacon }
Vevok
Vevo [Vkod=1]
Nev
{ Nagy Ferenc }
Telefonszam
{ 77895412 }
Cim
Irszam
{ 3519 }
Varos
{ Miskolc }

Utca
{ Fenyő utca }
Hazszam
{ 23 }
Vevo [Vkod=2]
Nev
{ Novák Balázs }
Telefonszam
{ 078954412 }
Cim
Irszam
{ 1149 }
Varos
Pizzeria_Pizza [Pizzaref=1, Pizzeriaref=1]
Pizzeria_Pizza [Pizzaref=3, Pizzeriaref=1]
Pizzeria_Pizza [Pizzaref=2, Pizzeriaref=2]
Pizzeria_Pizza [Pizzaref=4, Pizzeriaref=2]
Pizzeria_Pizza [Pizzaref=5, Pizzeriaref=3]
Pizza_Vevo_kapcsolok
Pizza_Vevo [Pizzaref=2, Vevoref=1]
Pizza_Vevo [Pizzaref=4, Vevoref=2]
Pizza_Vevo [Pizzaref=5, Vevoref=3]

2.3. DOM Query:

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

```java
package hu.domparse.ejx162;

import java.io.File;

public class DomQueryEJX162 {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        File file = new File("C:\\Users\\Bálint\\Desktop\\EGYETEM\\2022-23. 5.félév\\xml_beadando"
                + "\\DOMParseEJX162\\src\\hu\\domparse\\ejx162\\XMLEJX162.xml");
        // Parse-olás
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

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

```
        // Minden pizzeria attribútum kiiratasa

        System.out.println("PIZZERIAK");
        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("Pikod"));
                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 a pizzerianak a nevet, ami Pesten van
        System.out.println("\nPESTI PIZZERIA\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:

Gyökér element: model
PIZZERIAK

Element nev : Pizzeria

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

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

Element nev : Pizzeria
ID:3
Nev : Lábas Telefonszam
: 035637923 Cim:
Irszam : 3540
Varos : Alsózsolca
Utca : Fő út
Hazszam : 6

PESTI PIZZERIA

Nev: Italian Stallion