

Jegyzőkönyv
Adatkezelés XML környezetben
Féléves feladat

Szabó Bálint

EJX162

Pizzéria hálózat modellezése

1. Feladat

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, hiszen 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.

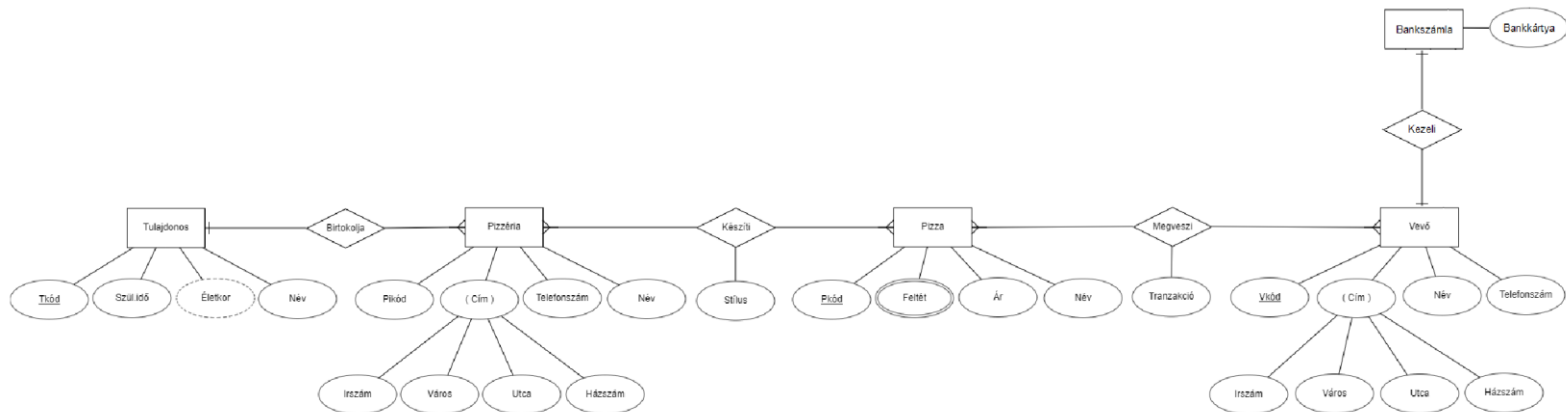
A **Pizzéria** és a **Pizza** egyedek között a **Készíti** kapcsolat van, amely egy egy **több a többhöz kapcsolat (N:M)**, hiszen egy pizzéria több pizzát is készíthet, és egy pizzát több pizzéria is el tud készíteni.

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.

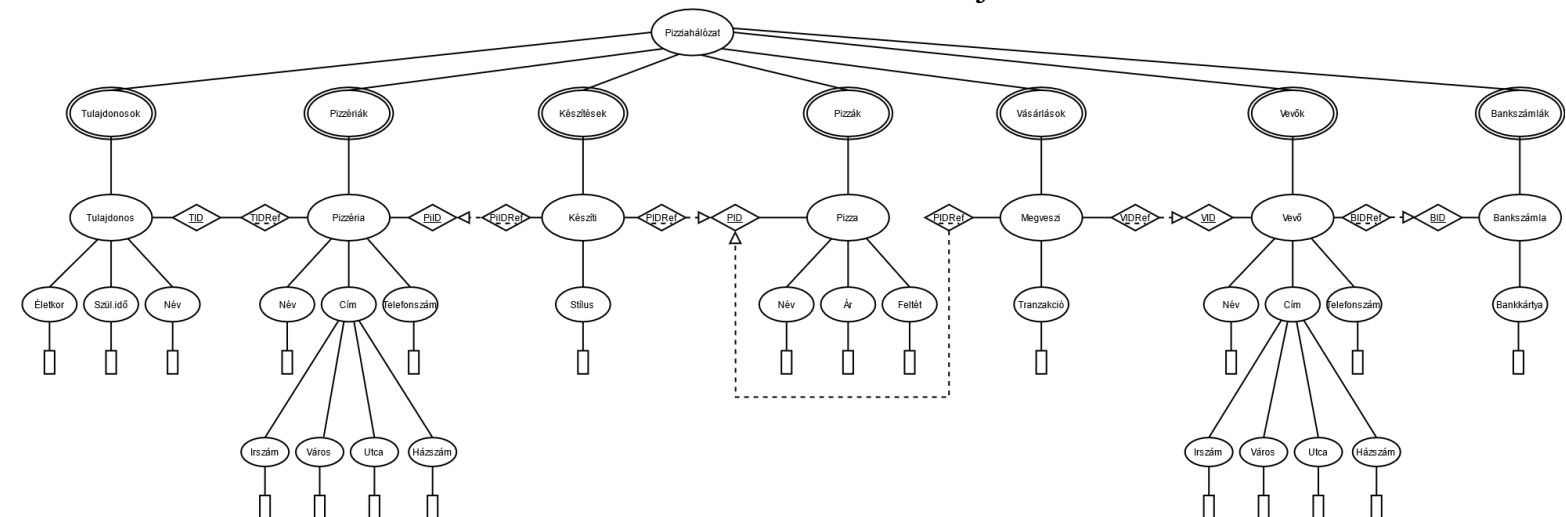
a) ER Modell:

b) XDM Modell:



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.



XML Dokumentum:

```
<?xml version="1.0" encoding="UTF-8"?>
<Pizzahalozat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaH7PG8U.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>
      <Hazzsam>14</Hazzsam>
    </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>
      <Hazzsam>26</Hazzsam>
    </Cim>
  </Pizzeria>
```

```
<Pizzeria PiID="3" TIDRef="3">
  <Nev>Lábas</Nev>
  <Telefonszam>035637923</Telefonszam>
  <Cim>
    <Irszam>3540</Irszam>
    <Varos>Alsózsolca</Varos>
    <Utca>Fő út</Utca>
    <Hazzsam>6</Hazzsam>
  </Cim>
</Pizzeria>
</Pizzeriak>
<Keszitesek>
  <Kesziti PIDRef="1" PiIDRef="1">
    <Stilus>Al Dente</Stilus>
  </Kesziti>

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

  <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>
      <Varos>Miskolc</Varos>
      <Utca>Fenyő utca</Utca>
      <Hazzam>23</Hazzam>
    </Cim>
  </Vevo>

  <Vevo VID="2">
    <Nev>Novák Balázs</Nev>
    <Telefonszam>078954412</Telefonszam>
    <Cim>
      <Irszam>1149</Irszam>
```

<Varos>Budapest</Varos>


```
        <Utca>Vezér utca</Utca>
        <Hazzsam>149</Hazzsam>
    </Cim>
</Vevo>

<Vevo VID="3">
    <Nev>Trab Antal</Nev>
    <Telefonszam>012258</Telefonszam>
    <Cim>
        <Irszam>3530</Irszam>
        <Varos>Miskolc</Varos>
        <Utca>Pattantyús utca</Utca>
        <Hazzsam>14</Hazzsam>
    </Cim>
</Vevo>
</Vevok>

</Pizzahalozat>
```



XML Schema:

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



```
    elementFormDefault="qualified"
attributeFormDefault="qualified">

    <xs:element name="Eletkor" type="xs:integer"/>
    <xs:element name="Szulido" type="xs:integer"/>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Irszam" type="xs:string"/>
    <xs:element name="Varos" type="xs:string"/>
    <xs:element name="Utca" type="xs:string"/>
    <xs:element name="Hazzsam" type="xs:integer"/>
    <xs:element name="Stilus" type="xs:string"/>
    <xs:element name="Ar" type="xs:integer"/>
    <xs:element name="Feltet" type="xs:string"/>
    <xs:element name="Tranzakcio" type="xs:integer"/>
    <xs:element name="Telefonszam" type="xs:integer"/>

    <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 ref="Nev"/>
            <xs:element ref="Eletkor"/>
            <xs:element ref="Szulido"/>

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

    <xs:complexType name="Pizzeria_tipus">
        <xs:sequence>
            <xs:element ref="Nev"/>
            <xs:element ref="Telefonszam"/>
            <xs:element name="Cim" type="Cim_tipus" maxOccurs="1"/>
        </xs:sequence>
```

[REDACTED]

[REDACTED]

```
<xs:attribute ref="PiID" use="required"/>  
<xs:attribute ref="TIDRef" use="required"/>
```

[REDACTED]

[REDACTED]

[REDACTED]

```
</xs:complexType>
```

```
<xs:complexType name="Cim_tipus">
```

```
  <xs:sequence>
```

```
    <xs:element ref="Irszam"/>
```

[REDACTED]

```
<xs:element ref="Varos"/>
<xs:element ref="Utca"/>
<xs:element ref="Hazzsam"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="Kesziti_tipus">
  <xs:sequence>
    <xs:element ref="Stilus"/>
  </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 ref="Nev"/>
    <xs:element ref="Ar"/>
    <xs:element ref="Feltet" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="PID" use="required"/>
</xs:complexType>

<xs:complexType name="Megveszi_tipus">
<xs:sequence>
  <xs:element ref="Tranzakcio"/>
</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 ref="Nev"/>
    <xs:element ref="Telefonszam"/>
    <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"/>
```

[REDACTED]

[REDACTED]

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

[REDACTED]

```
<xs:complexType name="Pizzeriak_tipus">  
  <xs:sequence>
```

[REDACTED]


```
        <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"/>
</xs:sequence>
</xs:complexType>

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

<xs:element name="Pizzahalozat">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Tulajdonosok" type="Tulajdonosok_tipus"/>
            <xs:element name="Pizzeriak" type="Pizzeriak_tipus"/>
            <xs:element name="Keszitesek" type="Keszitesek_tipus"/>
            <xs:element name="Pizzak" type="Pizzak_tipus"/>
            <xs:element name="Vasarlasok" type="Vasarlasok_tipus"/>
            <xs:element name="Vevok" type="Vevok_tipus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

[Redacted]

[Redacted]

```
</xs:complexType>
```

[Redacted]

[REDACTED]

[REDACTED]

<xs:key name="Talajdonos_Ekulcs">

[REDACTED]

```

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

<xs:key name="Pizzeria_EKulcs">
  <xs:selector xpath="Pizzeriak/Pizzeria"/>
  <xs:field xpath="@PiID"/>
</xs:key>

<xs:key name="Pizza_EKulcs">
  <xs:selector xpath="Pizzak/Pizza"/>
  <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="Pizzeriak/Pizzeria"/>
  <xs:field xpath="@TIDRef"/>
</xs:keyref>

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

<xs:keyref name="Pizza_IKulcs" refer="Pizza_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>

```

DOM Read:

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

```
package hu.domparse.H7PG8U;
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 DOMReadH7PG8U {

    public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException {        File xml = new
File("C:\\Users\\Asus\\eclipseworkspace\\DOMParseH7PG8U\\src\\hu\\domparse\\H7PG8U\\XMLH7PG

        // 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);
        }
    }
} }

```

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 }
Hatszám
 { 14 }
Pizzeria [Pikod=2]
Név
 { New York Style }
Telefonszám
 { 05618478 }
Cím
 Irszám
 { 3525 }
 Varos
 { Miskolc }
 Utca
 { Széchenyi István út }
 Hatszám
 { 26 }

Pizzeria [Pikod=3]
Név
 { Láska }
Telefonszám
 { 035637923 }
Cím
 Irszám
 { 3540 }
 Varos
 { Alsószolca }
 Utca
 { Fő út }
 Hatszám
 { 6 }

Pizzák

Pizza [Pkod=1]
Név
 { Margareta }
Ár
 { 1200 }
Feltet
 { Sajt }

Pizza [Pkod=2]
Név
 { Sonkás }
Ár
 { 1400 }
Feltet
 { Sajt }
Feltet
 { Sonka }

Pizza [Pkod=3]
Név
 { Szalámis }
Ár
 { 1400 }
Feltet


```

        { Sajt }
    Feltet
        { Szalámi }
Pizza [ Pkod=4 ]
    Nev
        { Sonkás-Kukoricás }
    Ar
        { 1600 }
    Feltet
        { Sajt }
    Feltet
        { Sonka }
    Feltet
        { Kukorica }
Pizza [ Pkod=5 ]
    Nev
        { Lábás 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 ]
```

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.

```

package hu.dompase.H7PG8U;
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 DOMModifyH7PG8U {

    public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException,
XPathExpressionException, DOMException, ParseException {
        File xml = new
File("C:\\Users\\Asus\\eclipseworkspace\\DOMParseH7PG8U\\src\\hu\\domparse\\H7PG8U\\XMLH7PG
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);
                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);
        }
    }
}
}

```

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 }
Hatszám
 { 26 }
Pizzeria [Pikod=3]
Nev
 { Lábás }
Telefonszam
 { 035637923 }
Cim
 Irszam
 { 3540 }
 Varos
 { Alsózsolca }
 Utca
{ Fő út }
Hatszám
 { 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
    { Lábás 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]

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.

```

package hu.domparse.H7PG8U;
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.Element;
import org.w3c.dom.Node; import org.w3c.dom.NodeList; import
org.xml.sax.SAXException;
public class DOMQueryH7PG8U {

    public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException {
        File file = new
File("C:\\Users\\Asus\\eclipseworkspace\\DOMParseH7PG8U\\src\\hu\\domparse\\H7PG8U\\XMLH7PG
// 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 kiiratas

        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"));
            }
        }
    }
}

```

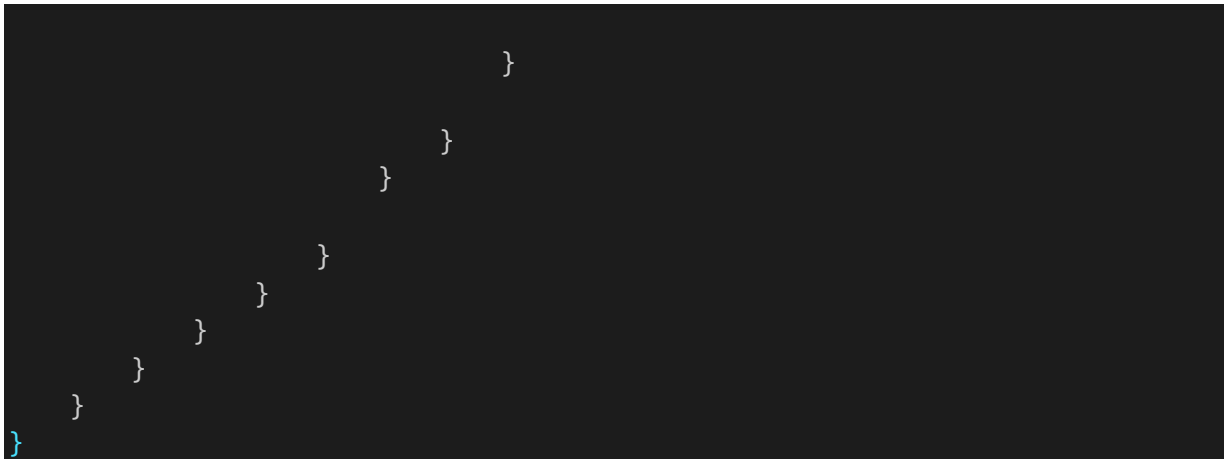
```

        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());
                }
            }
        }
    }
}
}
}
}

```



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 : Lábás Telefonszam
: 035637923 Cim:
Irszam : 3540
Varos : Alsózsolca
Utca : Fő út
Hazszam : 6

PESTI PIZZERIA

Nev: Italian Stallion