

Jegyzőkönyv
Adatkezelés XML környezetben
Féléves feladat

Laboda Dániel Balázs
H7PG8U

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.

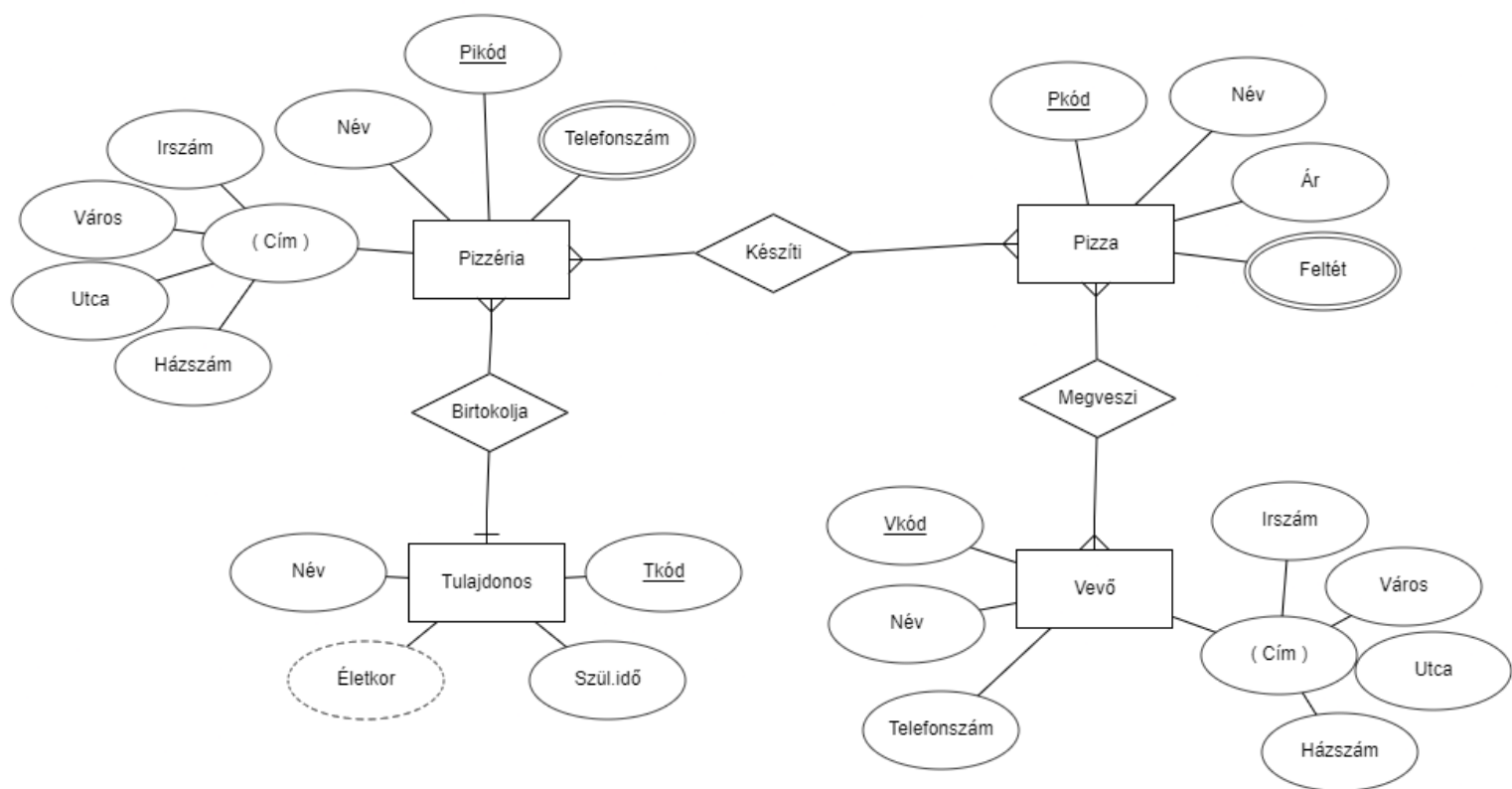
A **Tulajdonos** egyed **Tkód** tulajdonsága egyedi, számjegyekből álló azonosítója, azaz kulcsként funkcionál. Van egy származtatott tulajdonsága, az **Életkor**, amelyet a **Szül.idő** tulajdonságból származtathatunk hiszen, ha a jelenlegi dátumból kivonjuk a születési dátumot, akkor megkapjuk a Tulajdonos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

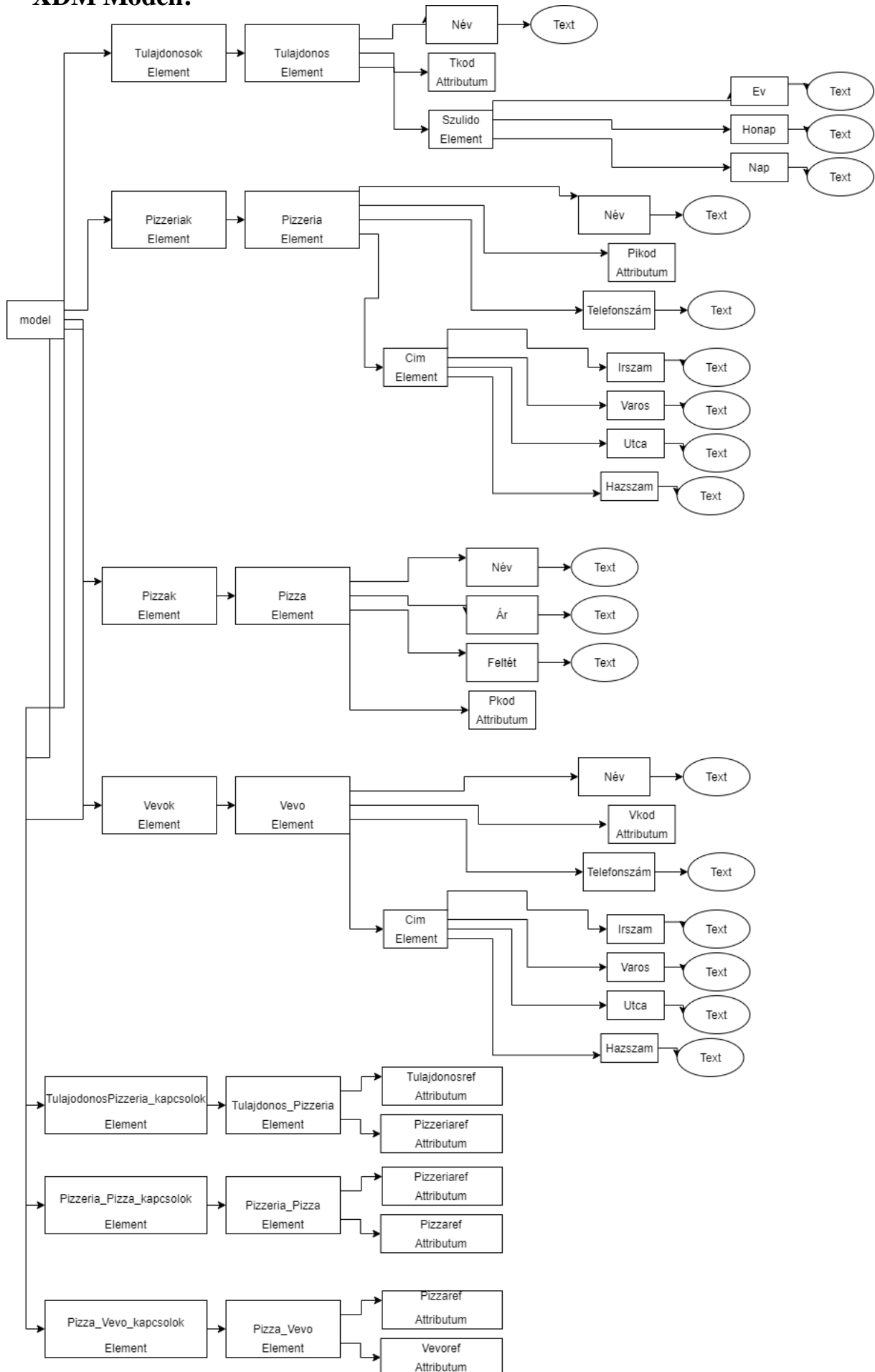
A **Pizzéria** és a **Pizza** egyedek között a **Készíti** kapcsolat van, amely egy egy **több a többhöz kapcsolat (N:M)**, hiszen egy pizzéria több pizzát is készíthet, és egy pizzát több pizzéria is el tud készíteni.

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.



XDM Modell:



XML Dokumentum:

```
<?xml version="1.0" encoding="UTF-8"?>
<model xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaH7PG8U.xsd">
  <Tulajdonosok>

    <Tulajdonos Tkod="1">
      <Nev>Nagy Béla</Nev>
      <Szulido>
        <Ev>1987</Ev>
        <Honap>03</Honap>
        <Nap>12</Nap>
      </Szulido>
    </Tulajdonos>

    <Tulajdonos Tkod="2">
      <Nev>Lőre István</Nev>
      <Szulido>
        <Ev>1967</Ev>
        <Honap>10</Honap>
        <Nap>25</Nap>
      </Szulido>
    </Tulajdonos>

    <Tulajdonos Tkod="3">
      <Nev>Kiss Imre</Nev>
      <Szulido>
        <Ev>1980</Ev>
        <Honap>08</Honap>
        <Nap>20</Nap>
      </Szulido>
    </Tulajdonos>

  </Tulajdonosok>

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

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

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

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

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

```
</Pizza>

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

<Vevok>
  <Vevo Vkod="1">
    <Nev>Nagy Ferenc</Nev>
    <Telefonszam>77895412</Telefonszam>
    <Cim>
      <Irszam>3519</Irszam>
      <Varos>Miskolc</Varos>
      <Utca>Fenyő utca</Utca>
      <Hazzsam>23</Hazzsam>
    </Cim>
  </Vevo>

  <Vevo Vkod="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 Vkod="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>

<Tulajdonos_Pizzeria_kapcsolok>
```

```
        <Tulajdonos_Pizzeria Tulajdonosref="1"
Pizzeriaref="1"></Tulajdonos_Pizzeria>
        <Tulajdonos_Pizzeria Tulajdonosref="2"
Pizzeriaref="2"></Tulajdonos_Pizzeria>
        <Tulajdonos_Pizzeria Tulajdonosref="2"
Pizzeriaref="3"></Tulajdonos_Pizzeria>
    </Tulajdonos_Pizzeria_kapcsolok>

    <Pizzeria_Pizza_kapcsolok>
        <Pizzeria_Pizza Pizzeriaref="1" Pizzaref="1"></Pizzeria_Pizza>
        <Pizzeria_Pizza Pizzeriaref="1" Pizzaref="3"></Pizzeria_Pizza>
        <Pizzeria_Pizza Pizzeriaref="2" Pizzaref="2"></Pizzeria_Pizza>
        <Pizzeria_Pizza Pizzeriaref="2" Pizzaref="4"></Pizzeria_Pizza>
        <Pizzeria_Pizza Pizzeriaref="3" Pizzaref="5"></Pizzeria_Pizza>
    </Pizzeria_Pizza_kapcsolok>

    <Pizza_Vevo_kapcsolok>
        <Pizza_Vevo Vevoref="1" Pizzaref="2"></Pizza_Vevo>
        <Pizza_Vevo Vevoref="2" Pizzaref="4"></Pizza_Vevo>
        <Pizza_Vevo Vevoref="3" Pizzaref="5"></Pizza_Vevo>
    </Pizza_Vevo_kapcsolok>

</model>
```


XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="model">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Tulajdonosok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Tulajdonos"
type="Tulajdonos_type" minOccurs="0" maxOccurs="unbounded"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Pizzeriak">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Pizzeria" type="Pizzeria_type"
minOccurs="0" maxOccurs="unbounded"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Pizzak">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Pizza" type="Pizza_type"
minOccurs="0" maxOccurs="unbounded"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Vevok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Vevo" type="Vevo_type"
minOccurs="0" maxOccurs="unbounded"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Tulajdonos_Pizzeria_kapcsolok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Tulajdonos_Pizzeria"
type="Tulajdonos_Pizzeria_kapcsolok" minOccurs="0"
maxOccurs="unbounded"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Pizzeria_Pizza_kapcsolok">
```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element name="Pizzeria_Pizza"
type="Pizzeria_Pizza_kapcsoló" minOccurs="0"
maxOccurs="unbounded"></xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Pizza_Vevo_kapcsoló">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Pizza_Vevo"
type="Pizza_Vevo_kapcsoló" minOccurs="0" maxOccurs="unbounded"></xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:complexType name="Tulajdonos_type">
    <xs:sequence>
        <xs:element name="Név"></xs:element>
        <xs:element name="Szulido" type="Szulido_type"></xs:element>
    </xs:sequence>
    <xs:attribute name="Tkod" type="xs:integer"
use="required"></xs:attribute>
</xs:complexType>

<xs:complexType name="Pizzeria_type">
    <xs:sequence>
        <xs:element name="Név"></xs:element>
        <xs:element name="Telefonszam" maxOccurs="unbounded"></xs:element>
        <xs:element name="Cím" type="Cím_type"></xs:element>
    </xs:sequence>
    <xs:attribute name="Pikod" type="xs:integer"
use="required"></xs:attribute>
</xs:complexType>

<xs:complexType name="Pizza_type">
    <xs:sequence>
        <xs:element name="Név"></xs:element>
        <xs:element name="Ar"></xs:element>
        <xs:element name="Feltet" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
    <xs:attribute name="Pkod" type="xs:integer"
use="required"></xs:attribute>
</xs:complexType>

```

```

<xs:complexType name="Vevo_type">
  <xs:sequence>
    <xs:element name="Nev"></xs:element>
    <xs:element name="Telefonszam"></xs:element>
    <xs:element name="Cim" type="Cim_type"></xs:element>
  </xs:sequence>
  <xs:attribute name="Vkod" type="xs:integer"
use="required"></xs:attribute>
</xs:complexType>

  <xs:complexType name="Tulajdonos_Pizzeria_kapcsoló">
    <xs:attribute name="Tulajdonosref" type="xs:integer"
use="required"></xs:attribute>
    <xs:attribute name="Pizzeriaref" type="xs:integer"
use="required"></xs:attribute>
  </xs:complexType>

  <xs:complexType name="Pizzeria_Pizza_kapcsoló">
    <xs:attribute name="Pizzeriaref" type="xs:integer"
use="required"></xs:attribute>
    <xs:attribute name="Pizzaref" type="xs:integer"
use="required"></xs:attribute>
  </xs:complexType>

  <xs:complexType name="Pizza_Vevo_kapcsoló">
    <xs:attribute name="Pizzaref" type="xs:integer"
use="required"></xs:attribute>
    <xs:attribute name="Vevoref" type="xs:integer"
use="required"></xs:attribute>
  </xs:complexType>

  <xs:complexType name="Cim_type">
    <xs:sequence>
      <xs:element name="Irszam"></xs:element>
      <xs:element name="Varos"></xs:element>
      <xs:element name="Utca"></xs:element>
      <xs:element name="Hazzsam"></xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Szulido_type">
    <xs:sequence>
      <xs:element name="Ev"></xs:element>
      <xs:element name="Honap"></xs:element>
      <xs:element name="Nap"></xs:element>
    </xs:sequence>
  </xs:complexType>

</xs:schema>

```

DOM Read:

```
package hu.domparse.H7PG8U;

import java.io.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import org.w3c.dom.traversal.*;
import org.xml.sax.*;

public class DOMReadH7PG8U {

    public static void main(String[] args) throws
        ParserConfigurationException, SAXException, IOException {
        File xml = new File("XMLH7PG8U.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 kiírása
        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 }

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

{ Lábás }

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

{ 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:

```
package hu.domparse.H7PG8U;

import java.io.*;
import java.text.ParseException;

import javax.xml.parsers.*;
import javax.xml.xpath.*;

import org.w3c.dom.*;
import org.w3c.dom.traversal.*;
import org.xml.sax.*;

public class DOMModifyH7PG8U {

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

        File xml = new File("XMLH7PG8U.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);

                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 }

Hazsam

{ 14 }

Pizzeria [Pikod=2]

Nev

{ New York Style }

Telefonszam

{ 05618478 }

Cim

Irszam

{ 3525 }

Varos

{ Miskolc }

Utca

{ Széchenyi István út }

Hazsam

{ 26 }

Pizzeria [Pikod=3]

Nev

{ Lábás }

Telefonszam

{ 035637923 }

Cim

Irszam

{ 3540 }

Varos

{ Alsózsolca }

Utca

{ Fő út }

Haszam

{ 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 }

Haszam

{ 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]


```

        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

Hazsam : 14

Element nev : Pizzeria

ID:2

Nev : New York Style

Telefonszam : 05618478

Cim:

Irszam : 3525

Varos : Miskolc

Utca : Széchenyi István út

Hazsam : 26

Element nev : Pizzeria

ID:3

Nev : Lábás

Telefonszam : 035637923

Cim:

Irszam : 3540

Varos : Alsózsolca

Utca : Fő út

Hazsam : 6

PESTI PIZZERIA

Nev: Italian Stallion