

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Sportüzlet felépítése

Féléves feladat

Készítette: Kis Kornél

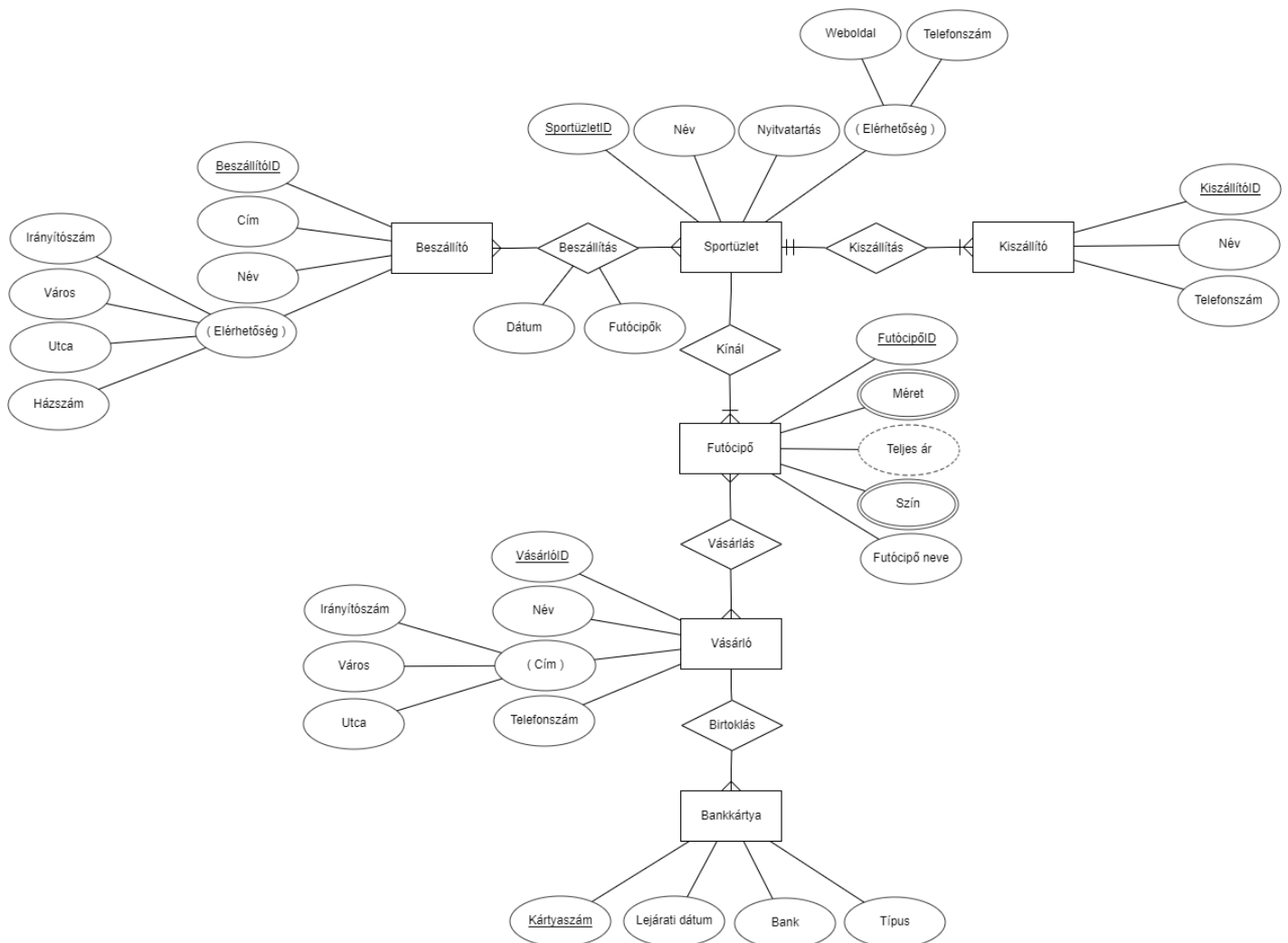
Neptunkód: B7IK21

## A feladat leírása:

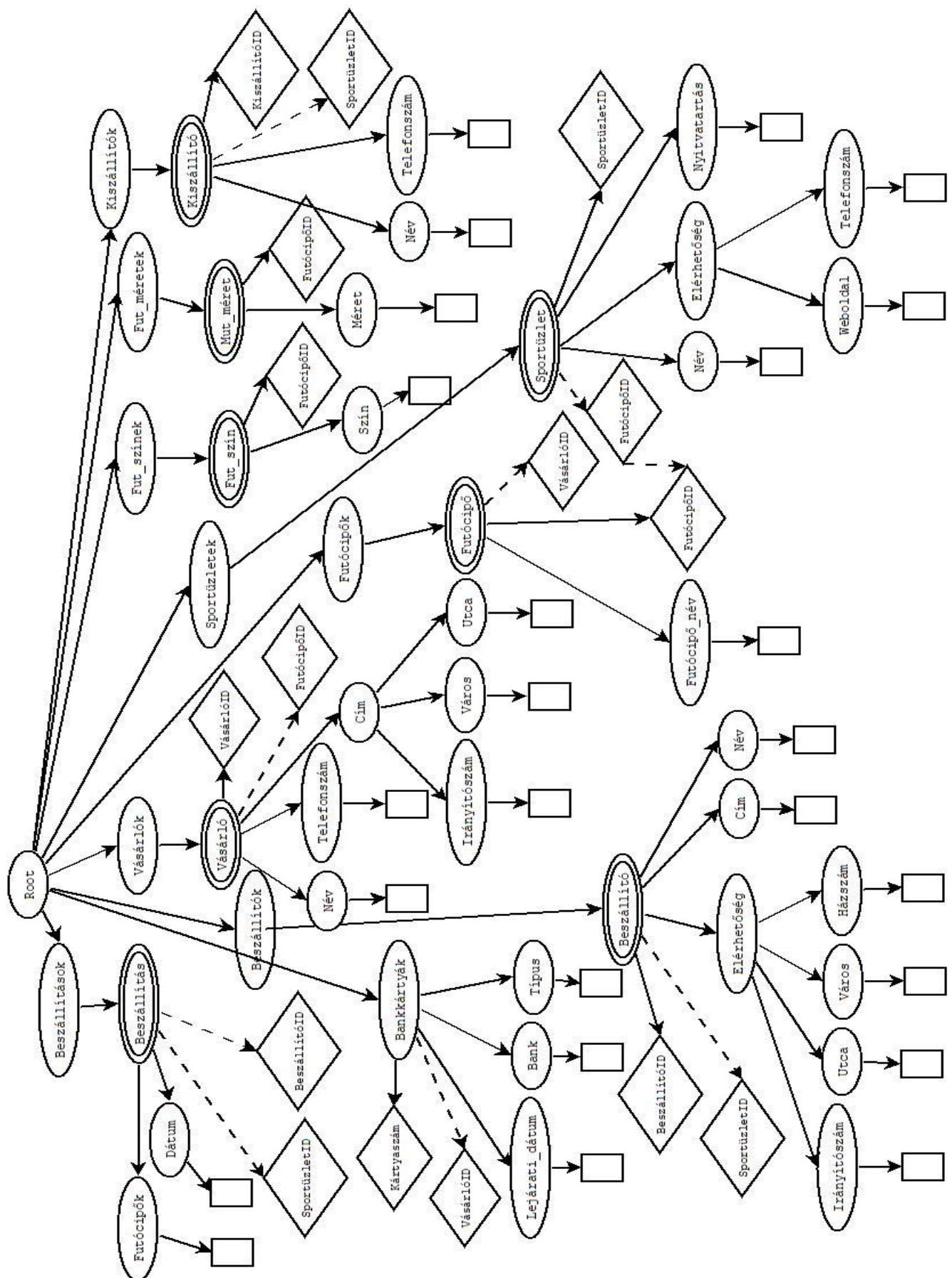
A feladatban egy sportüzlet felépítését fogom bemutatni, hogy miképpen lehet eltárolni mindezt adatbázisban. A sportüzletben a rengeteg vásárlók, termékek, beszállítók, kiszállítók adatait nehéz számon tartani, hisz minden egyes nap rengetegen fordulnak meg egy-egy ilyen cégnél. A sportüzlet közvetlenül négy adatot tart nyilván, egy sportüzletid, a sportüzlet neve, nyitva tartása és elérhetősége, ami egy weboldalból és telefonszámból áll. A sportüzlethez közvetlenül három entitás tartozik, a beszállító, a kiszállító és a termékek, amik jelen esetben futócipők. A beszállítókhoz és kiszállítókhoz is általánosságban tartozik név, cím, elérhetőség, mint telefonszám, irányítószám, város, utca, házszám. Az üzletben lévő futócipőkhöz két fontos adatot rendelünk, a méretet, illetve színt. Minden vásárló vagy vesz futócipőt vagy nem, így őket hozzá rendeljük és nyilvántartjuk az adataikat. Még hozzá, hogy ki vette meg, milyen címen lakik (irányítószám, város, utca), illetve telefonszám. Természetesen minden futócipőnek és vásárlónak egyedi azonosítója van. A futócipőket a vásárlók a bankkártyával vásárolták, akkor azon adatokat is nyilvántartjuk, mint kártyaszám, bankkártya típusa, bank neve, illetve hogy mikor jár le az adott kártya.

### 1. feladat

#### 1a) Az adatbázis ER modell:



### 1b) Az adatbázis konvertálása XDM modellre:



1c) Az XDM modell alapján XML dokumentum készítése:

---

```
<?xml version="1.0" encoding="UTF-8"?>

<root>
  <sportuzletek>
    <sportuzlet sportuzletid="s1" futocipoid="f1">
      <nev>Spuri futóbolt</nev>
      <nyitvatartas>H-Szo 8-19</nyitvatartas>
      <elerhetoseg>
        <weboldal>www.spurifutobolt.com</weboldal>
        <telefonszam>06301234567</telefonszam>
      </elerhetoseg>
    </sportuzlet>
    <sportuzlet sportuzletid="s2" futocipoid="f2">
      <nev>Nyulciő futóbolt</nev>
      <nyitvatartas>H-Szo 9-16</nyitvatartas>
      <elerhetoseg>
        <weboldal>www.nyulcipofutobolt.com</weboldal>
        <telefonszam>06302220011</telefonszam>
      </elerhetoseg>
    </sportuzlet>
    <sportuzlet sportuzletid="s3" futocipoid="f3">
      <nev>Fuss futóbolt</nev>
      <nyitvatartas>H-V 8-20</nyitvatartas>
      <elerhetoseg>
        <weboldal>www.fussfutobolt.com</weboldal>
        <telefonszam>+36701234444</telefonszam>
      </elerhetoseg>
    </sportuzlet>
    <sportuzlet sportuzletid="s4" futocipoid="f4">
      <nev>Szaladj futóbolt</nev>
      <nyitvatartas>H-P 8-16</nyitvatartas>
      <elerhetoseg>
        <weboldal>www.szaladjutobolt.com</weboldal>
        <telefonszam>06307653310</telefonszam>
      </elerhetoseg>
    </sportuzlet>
  </sportuzletek>
```

```
<futocipok>
  <futocipo futocipoid="f1" vasarlroid="v1">
    <futocipo_nev>Hoka</futocipo_nev>
  </futocipo>
  <futocipo futocipoid="f2" vasarlroid="v1">
    <futocipo_nev>Asics</futocipo_nev>
  </futocipo>
  <futocipo futocipoid="f3" vasarlroid="v3">
    <futocipo_nev>Adidas</futocipo_nev>
  </futocipo>
  <futocipo futocipoid="f4" vasarlroid="v4">
    <futocipo_nev>Saucony</futocipo_nev>
  </futocipo>
</futocipok>
<fut_szinek>
  <fut_szin futocipoid="f1">
    <szin>Fekete</szin>
  </fut_szin>
  <fut_szin futocipoid="f2">
    <szin>Fehér</szin>
  </fut_szin>
  <fut_szin futocipoid="f3">
    <szin>Piros</szin>
  </fut_szin>
  <fut_szin futocipoid="f4">
    <szin>Kék</szin>
  </fut_szin>
</fut_szinek>
<fut_meretek>
  <fut_meret futocipoid="f1">
    <meret>38</meret>
  </fut_meret>
  <fut_meret futocipoid="f2">
    <meret>42</meret>
  </fut_meret>
  <fut_meret futocipoid="f3">
    <meret>41</meret>
  </fut_meret>
  <fut_meret futocipoid="f4">
    <meret>39</meret>
  </fut_meret>
</fut_meretek>
```

```
<vasarlok>
  <vasarlo vasarloid="vas1" futocipoid="f1">
    <nev>Nagy Lajos</nev>
    <telefonszam>06201112233</telefonszam>
    <cim>
      <iranyitoszam>3780</iranyitoszam>
      <varos>Edelény</varos>
      <utca>Andrásy út 10.</utca>
    </cim>
  </vasarlo>
  <vasarlo vasarloid="vas2" futocipoid="f2">
    <nev>Kovács Géza</nev>
    <telefonszam>06301119999</telefonszam>
    <cim>
      <iranyitoszam>2345</iranyitoszam>
      <varos>Nincslakó</varos>
      <utca>Otthon út 22.</utca>
    </cim>
  </vasarlo>
  <vasarlo vasarloid="vas2" futocipoid="f4">
    <nev>Kovács Géza</nev>
    <telefonszam>06301119999</telefonszam>
    <cim>
      <iranyitoszam>2345</iranyitoszam>
      <varos>Nincslakó</varos>
      <utca>Otthon út 22.</utca>
    </cim>
  </vasarlo>
  <vasarlo vasarloid="vas3" futocipoid="f3">
    <nev>Erdei Sándor</nev>
    <telefonszam>+36304447768</telefonszam>
    <cim>
      <iranyitoszam>4444</iranyitoszam>
      <varos>Ottlakó</varos>
      <utca>Sehol út 33.</utca>
    </cim>
  </vasarlo>
```

```
<vasarlo vasarloid="vas4" futocipoid="f4">
  <nev>Pingvin Ember</nev>
  <telefonszam>06706669966</telefonszam>
  <cim>
    <iranyitoszam>1111</iranyitoszam>
    <varos>Budapest</varos>
    <utca>Pesti út 05.</utca>
  </cim>
</vasarlo>
</vasarlok>
<bankkartyak>
  <bankkartya kartyaszam="11223344" vasarloid="vas1">
    <lejarati_datum>2025.04.21</lejarati_datum>
    <bank>OTP</bank>
    <tipus>Mastercard</tipus>
  </bankkartya>
  <bankkartya kartyaszam="11227777" vasarloid="vas2">
    <lejarati_datum>2027.11.07</lejarati_datum>
    <bank>Budapest Bank</bank>
    <tipus>Mastercard Gold</tipus>
  </bankkartya>
  <bankkartya kartyaszam="11224545" vasarloid="vas3">
    <lejarati_datum>2022.09.21</lejarati_datum>
    <bank>OTP</bank>
    <tipus>Mastercard Platina</tipus>
  </bankkartya>
  <bankkartya kartyaszam="11223333" vasarloid="vas4">
    <lejarati_datum>2026.03.11</lejarati_datum>
    <bank>Erste Bank</bank>
    <tipus>Mastercard World</tipus>
  </bankkartya>
  <bankkartya kartyaszam="11227777" vasarloid="vas2">
    <lejarati_datum>2027.11.07</lejarati_datum>
    <bank>Budapest Bank</bank>
    <tipus>Mastercard Gold</tipus>
  </bankkartya>
  <bankkartya kartyaszam="11224545" vasarloid="vas3">
    <lejarati_datum>2022.09.21</lejarati_datum>
    <bank>OTP</bank>
    <tipus>Mastercard Platina</tipus>
  </bankkartya>
```

```

    <bankkartya kartyaszam="11223333" vasarloid="vas4">
      <lejarati_datum>2026.03.11</lejarati_datum>
      <bank>Erste Bank</bank>
      <tipus>Mastercard World</tipus>
    </bankkartya>
  </bankkartyak>
  <beszallitasok>
    <beszallitas beszallitoid="bsz1" sportuzletid="s1">
      <datum>2021.11.17</datum>
      <futocipok>Hoka</futocipok>
    </beszallitas>
    <beszallitas beszallitoid="bsz2" sportuzletid="s1">
      <datum>2021.11.17</datum>
      <futocipok>Nike</futocipok>
    </beszallitas>
    <beszallitas beszallitoid="bsz3" sportuzletid="s3">
      <datum>2021.11.17</datum>
      <futocipok>Adidas</futocipok>
    </beszallitas>
    <beszallitas beszallitoid="bsz4" sportuzletid="s4">
      <datum>2021.11.17</datum>
      <futocipok>Saucony</futocipok>
    </beszallitas>
  </beszallitasok>
  <beszallitok>
    <beszallito beszallitoid="bsz1">
      <nev>József Attila</nev>
      <cim>Márnincs út 18.</cim>
      <elérhetoseg>
        <iranyitoszam>1112</iranyitoszam>
        <varos>Pontosvár</varos>
        <utca>Lehetetlen</utca>
        <hazszam>36</hazszam>
      </elérhetoseg>
    </beszallito>
    <beszallito beszallitoid="bsz2">
      <nev>Petőfi Sándor</nev>
      <cim>Kávéház út 12.</cim>
      <elérhetoseg>
        <iranyitoszam>1212</iranyitoszam>
        <varos>Tizenkettő</varos>
        <utca>Pont</utca>
        <hazszam>12</hazszam>
      </elérhetoseg>
    </beszallito>
  </beszallitok>

```



```
<beszallito beszallitoid="bsz3">
  <nev>Ady Endre</nev>
  <cim>Őrizem Szemed út 34.</cim>
  <elерhetoseg>
    <iranyitoszam>2345</iranyitoszam>
    <varos>Párizs</varos>
    <utca>Járt</utca>
    <hazszam>29</hazszam>
  </elерhetoseg>
</beszallito>
<beszallito beszallitoid="bsz4">
  <nev>Kosztolányi Dezső</nev>
  <cim>Halotti út 66.</cim>
  <elерhetoseg>
    <iranyitoszam>3210</iranyitoszam>
    <varos>Köszsvár</varos>
    <utca>Nyár</utca>
    <hazszam>55</hazszam>
  </elерhetoseg>
</beszallito>
</beszallitok>
<kiszallitok>
  <kiszallito kiszallitoid="k1" sportuzletid="s1">
    <nev>Még Nincs</nev>
    <telefonszam>+36308887878</telefonszam>
  </kiszallito>
  <kiszallito kiszallitoid="k2" sportuzletid="s2">
    <nev>Majd Lesz</nev>
    <telefonszam>06201472586</telefonszam>
  </kiszallito>
  <kiszallito kiszallitoid="k3" sportuzletid="s3">
    <nev>Pontosan Tudó Nincs</nev>
    <telefonszam>+36708884561</telefonszam>
  </kiszallito>
  <kiszallito kiszallitoid="k4" sportuzletid="s4">
    <nev>Sehol Senki</nev>
    <telefonszam>06303332121</telefonszam>
  </kiszallito>
</kiszallitok>
</root>
```

---

## 1d) Az XML dokumentum alapján XMLSchema készítése:

---

```
<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.example.org/XMLSchemaB7IK21"
  elementFormDefault="qualified">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="sportuzletek">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="sportuzlet" minOccurs="1"
                maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="nev" type="xs:string" />
                    <xs:element name="nyitvatartas" type="xs:string" />
                    <xs:element name="elerhetoseg">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="weboldal" type="xs:string" />
                          <xs:element name="telefonszam"
                            type="telefonTipus" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="sportuzletid" type="xs:string" />
                  <xs:attribute name="futocipoid" type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="futocipok">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="1" maxOccurs="unbounded"
                name="futocipo">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="futocipo_nev" type="xs:string" />
                    <xs:attribute name="futocipoid" type="xs:string" />
                    <xs:attribute name="vasarloid" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="fut_szinek">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="1" maxOccurs="unbounded"
                name="fut_szin">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="szin" type="szinTipus" />
                    <xs:attribute name="futocipoid" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

---

```

</xs:element>
<xs:element name="fut_meretek">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded"
        name="fut_meret">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="meret" type="meretTipus" />
            </xs:sequence>
            <xs:attribute name="futocipoid" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
<xs:element name="vasarlok">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded"
        name="vasarlo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nev" type="xs:string" />
              <xs:element name="telefonszam" type="telefonTipus" />
              <xs:element name="cim">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="iranyitoszam"
                      type="iranyitoszamTipus" />
                    <xs:element name="varos" type="xs:string" />
                    <xs:element name="utca" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="vasarloid" type="xs:string" />
            <xs:attribute name="futocipoid" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
<xs:element name="bankkartyak">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded"
        name="bankkartya">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="lejarati_datum" type="datumTipus" />
              <xs:element name="bank" type="xs:string" />
              <xs:element name="tipus" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="kartyaszam"
              type="xs:positiveInteger" />
            <xs:attribute name="vasarloid" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
<xs:element name="beszallitasok">
  <xs:complexType>

```

```

        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="unbounded"
                name="beszallitas">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="datum" type="datumTipus" />
                        <xs:element name="futocipok" type="xs:string" />
                    </xs:sequence>
                    <xs:attribute name="beszallitoid" type="xs:string" />
                    <xs:attribute name="sportuzletid" type="xs:string" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="beszallitok">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="unbounded"
                name="beszallito">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="nev" type="xs:string" />
                        <xs:element name="cim" type="xs:string" />
                        <xs:element name="elerhetoseg">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="iranyitoszam"
                                        type="iranyitoszamTipus" />
                                    <xs:element name="varos" type="xs:string" />
                                    <xs:element name="utca" type="xs:string" />
                                    <xs:element name="hazszam"
                                        type="xs:positiveInteger" />
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="beszallitoid" type="xs:string" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="kiszallitok">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="unbounded"
                name="kiszallito">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="nev" type="xs:string" />
                        <xs:element name="telefonszam" type="telefonTipus" />
                    </xs:sequence>
                    <xs:attribute name="kiszallitoid" type="xs:string" />
                    <xs:attribute name="sportuzletid" type="xs:string" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:key name="Key_sportuzlet">
    <xs:selector xpath="sportuzlet" />
    <xs:field xpath="@sportuzletid" />
</xs:key>
<xs:key name="Key_futocipo">
    <xs:selector xpath="futocipo" />
    <xs:field xpath="@futocipoid" />
</xs:key>
<xs:key name="Key_vasarlo">
    <xs:selector xpath="vasarlo" />
    <xs:field xpath="@vasarlroid" />
</xs:key>
<xs:key name="Key_bankkartya">
    <xs:selector xpath="bankkartya" />
    <xs:field xpath="@kartyaszam" />
</xs:key>
<xs:key name="Key_beszallito">
    <xs:selector xpath="beszallito" />
    <xs:field xpath="@beszallitoid" />
</xs:key>
<xs:key name="Key_kiszallito">
    <xs:selector xpath="kiszallito" />
    <xs:field xpath="@kiszallitoid" />
</xs:key>
<xs:keyref name="FKey_fut_szin_futocipo"
    refer="Key_futocipo">
    <xs:selector xpath="fut_szin" />
    <xs:field xpath="@futocipoid" />
</xs:keyref>
<xs:keyref name="FKey_fut_meret_futocipo"
    refer="Key_futocipo">
    <xs:selector xpath="fut_meret" />
    <xs:field xpath="@futocipoid" />
</xs:keyref>
<xs:keyref name="FKey_sportuzlet_futocipo"
    refer="Key_futocipo">
    <xs:selector xpath="sportuzlet" />
    <xs:field xpath="@futocipoid" />
</xs:keyref>
<xs:keyref name="FKey_beszallitas_baszallito"
    refer="Key_beszallito">
    <xs:selector xpath="beszallitas" />
    <xs:field xpath="@beszallitoid" />
</xs:keyref>

```

```

<xs:keyref name="FKey_beszallitas_sportuzlet"
  refer="Key_sportuzlet">
  <xs:selector xpath="beszallitas" />
  <xs:field xpath="@sportuzletid" />
</xs:keyref>
<xs:keyref name="FKey_kiszallito_sportuzlet"
  refer="Key_sportuzlet">
  <xs:selector xpath="kiszallito" />
  <xs:field xpath="@sportuzletid" />
</xs:keyref>
<xs:keyref name="FKey_vasarlo_futocipo" refer="Key_futocipo">
  <xs:selector xpath="vasarlo" />
  <xs:field xpath="@futocipoid" />
</xs:keyref>
<xs:keyref name="FKey_bankkartya_vasarlo"
  refer="Key_vasarlo">
  <xs:selector xpath="bankkartya" />
  <xs:field xpath="@vasarloid" />
</xs:keyref>
</xs:element>

<xs:simpleType name="iranyitoszamTipus">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="1000" />
    <xs:maxInclusive value="9999" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="szinTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="Fehér" />
    <xs:pattern value="Fekete" />
    <xs:pattern value="Piros" />
    <xs:pattern value="Kék" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="meretTipus">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="34" />
    <xs:maxInclusive value="47" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="telefonTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0]{1}[0-9]{2}[0]{1}[0-9]{7}" />
    <xs:pattern value="[+]{1}[0-9]{3}[0]{1}[0-9]{7}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="datumTipus">
  <xs:restriction base="xs:string">

```

```
<!-- 2000.5.1 ,2000.5.01 -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1-9]{1}[.]{1}[1-9]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1-9]{1}[.]{1}[1-2]{1}[0-9]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1-9]{1}[.]{1}[1][0]{1}[1-9]{1}" />

<!-- 2000.10.2,2000.10.02 -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][1]{1}[0-2]{1}[.]{1}[1-9]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][1]{1}[0-2]{1}[.]{1}[1-2]{1}[0-9]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][1]{1}[0-2]{1}[.]{1}[1][0]{1}[1-9]{1}" />

<!-- 2000.05.1 ,2000.05.01 -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[1-9]{1}[.]{1}[1-9]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[1-9]{1}[.]{1}[1-2]{1}[0-9]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[1-9]{1}[.]{1}[1][0]{1}[1-9]{1}" />

<!-- január 31nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][1]{1}[.]{1}[1][3]{1}[0-1]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[1][1]{1}[.]{1}[1][3]{1}[0-1]{1}" />
```

```
<!-- február 29nap -->

<!-- március 31nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][3]{1}[.]{1}[1][3]{1}[0-1]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[3]{1}[.]{1}[1][3]{1}[0-1]{1}" />

<!-- április 30nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][4]{1}[.]{1}[1][3]{1}[0]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[4]{1}[.]{1}[1][3]{1}[0]{1}" />

<!-- május 31nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][5]{1}[.]{1}[1][3]{1}[0-1]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[5]{1}[.]{1}[1][3]{1}[0-1]{1}" />

<!-- június 30nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][6]{1}[.]{1}[1][3]{1}[0]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1][0]{1}[6]{1}[.]{1}[1][3]{1}[0]{1}" />
```

```

<!-- július 31nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[7]{1}[.]{1}[3]{1}[0-1]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[0]{1}[7]{1}[.]{1}[3]{1}[0-1]{1}" />

<!-- augusztus 31nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[8]{1}[.]{1}[3]{1}[0-1]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[0]{1}[8]{1}[.]{1}[3]{1}[0-1]{1}" />

<!-- szeptember 30nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[9]{1}[.]{1}[3]{1}[0]{1}" />
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[0]{1}[9]{1}[.]{1}[3]{1}[0]{1}" />

<!-- október 31nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1]{1}[0]{1}[.]{1}[3]{1}[0-1]{1}" />

<!-- november 30nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1]{1}[1]{1}[.]{1}[3]{1}[0]{1}" />

<!-- december 31nap -->
<xs:pattern
  value="[1-9]{1}[0-9]{3}[.]{1}[1]{1}[2]{1}[.]{1}[3]{1}[0-1]{1}" />

</xs:restriction>
</xs:simpleType>

</xs:schema>

```



## 2. feladat

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

**2a) Adatolvasás:** Beolvassuk az összes adatot az XML modellből, ami jelen esetben a sportüzletek kapcsolódó elemeket és azok adatait jelenti, majd kiíratjuk a consola az adatokat.

```
package hu.domparse.b7ik21;

import java.io.File;
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;

public class DomReadB7IK21 {

    public static void main(String[] args) {
        NodeList list;
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            // fájl beolvasása
            Document document = builder.parse(new File("XMLB7IK21.xml"));
            document.getDocumentElement().normalize();
            // Gyökér elem megkeresése
            System.out.println("Root element : " + document.getDocumentElement().getNodeName());
            System.out.println("-----");
            // Aktuális elem meghatározása
            list = document.getElementsByTagName("sportuzlet");

            for (int i = 0; i < list.getLength(); i++) {
                Node node = list.item(i);
                System.out.println("\nCurrent element: " + node.getNodeName());
                // Sportüzlet adatainak kiírása
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element element = (Element) node;

                    System.out.println("Sportüzlet id: " + element.getAttribute("sportuzletid"));
                    System.out.println("Futócipő id: " + element.getAttribute("futocipoid"));

                    System.out.println("Név: " + element.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("Nyitvatartás: " + element.getElementsByTagName("nyitvatartas").item(0).getTextContent());

                    // Sportüzleten belüli elérhetőség adatok kiírása
                    Node elerhetoseg = list.item(i);
                    if (elerhetoseg.getNodeType() == Node.ELEMENT_NODE) {
                        Element elementelerhetoseg = (Element) node;
                        System.out.println("Weboldal: "
                            + elementelerhetoseg.getElementsByTagName("weboldal").item(0).getTextContent());
                        System.out.println("Telefonszám: "
                            + elementelerhetoseg.getElementsByTagName("telefonszam").item(0).getTextContent());
                    }
                }
            }
        }
    }
}
```

```

// Aktuális elem meghatározása
list = document.getElementsByTagName("futocipo");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Alkapcs adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Futócipő id: " + element.getAttribute("futocipoid"));
        System.out.println("Vásárló id: " + element.getAttribute("vasarloid"));

        System.out.println(
            "Futócipő név: " + element.getElementsByTagName("futocipo_nev").item(0).getTextContent());
    }
}
// Aktuális elem meghatározása
list = document.getElementsByTagName("fut_szin");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Alkapcs adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Futócipő id: " + element.getAttribute("futocipoid"));

        System.out.println("Szín: " + element.getElementsByTagName("szin").item(0).getTextContent());
    }
}
// Aktuális elem meghatározása
list = document.getElementsByTagName("fut_meret");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Alkapcs adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Futócipő id: " + element.getAttribute("futocipoid"));

        System.out.println("Méret: " + element.getElementsByTagName("meret").item(0).getTextContent());
    }
}
// Aktuális elem meghatározása
list = document.getElementsByTagName("vasarlo");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Vásárló adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Vásárló id: " + element.getAttribute("vasarloid"));
        System.out.println("Futócipő id: " + element.getAttribute("futocipoid"));

        System.out.println("Név: " + element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println(
            "Telefonszám: " + element.getElementsByTagName("telefonszam").item(0).getTextContent());
        // Vásárló cím adatainak kiírása
        Node nodecim = list.item(i);
        if (nodecim.getNodeType() == Node.ELEMENT_NODE) {
            Element elementcim = (Element) node;
            System.out.println("Írányítószám: "
                + elementcim.getElementsByTagName("iranyitoszam").item(0).getTextContent());
            System.out

```

```

        .println("Város: " + elementcim.getElementsByTagName("varos").item(0).getTextContent());
        System.out.println("Utca: " + elementcim.getElementsByTagName("utca").item(0).getTextContent());
    }
}
// Aktuális elem meghatározása
list = document.getElementsByTagName("bankkartya");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Bankkártya adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Kártyaszám: " + element.getAttribute("kartyaszam"));
        System.out.println("Vásárló id: " + element.getAttribute("vasarloid"));

        System.out.println("Lejárat dátum: "
            + element.getElementsByTagName("lejarati_datum").item(0).getTextContent());
        System.out.println("Bank: " + element.getElementsByTagName("bank").item(0).getTextContent());
        System.out.println("Típus: " + element.getElementsByTagName("tipus").item(0).getTextContent());
    }
}
// Aktuális elem meghatározása
list = document.getElementsByTagName("beszallitas");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Bszállító adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Bszállító id: " + element.getAttribute("beszallitoid"));

        System.out.println("Sportüzlet id: " + element.getAttribute("sportuzletid"));

        System.out.println("Dátum: " + element.getElementsByTagName("datum").item(0).getTextContent());
        System.out.println("Futócipők: " + element.getElementsByTagName("futocipok").item(0).getTextContent());
    }
}
// Aktuális elem meghatározása
list = document.getElementsByTagName("beszallito");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Bszállító adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Bszállító id: " + element.getAttribute("beszallitoid"));

        System.out.println("Név: " + element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Cím: " + element.getElementsByTagName("cim").item(0).getTextContent());
        // Vásárló cím adatainak kiírása
        Node nodeelerhetoseg = list.item(i);
        if (nodeelerhetoseg.getNodeType() == Node.ELEMENT_NODE) {
            Element elementelerhetoseg = (Element) node;
            System.out.println("Irányítószám: "
                + elementelerhetoseg.getElementsByTagName("iranyitoszam").item(0).getTextContent());
            System.out.println("Város: " + elementelerhetoseg.getElementsByTagName("varos").item(0).getTextContent());
            System.out.println("Utca: " + elementelerhetoseg.getElementsByTagName("utca").item(0).getTextContent());
            System.out.println("Házszám: "
                + elementelerhetoseg.getElementsByTagName("hazzsam").item(0).getTextContent());
        }
    }
}

```

```

    }
}
// Aktuális elem meghatározása
list = document.getElementsByTagName("kiszallito");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Kiszállító adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Kiszállító id: " + element.getAttribute("kiszallitoid"));
        System.out.println("Sportüzlet id: " + element.getAttribute("sportuzletid"));

        System.out.println("Név: " + element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Telefonszám: " + element.getElementsByTagName("telefonszam").item(0).getTextContent());
    }
}

} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}

}
}

```

**2b) Adatmódosítás:** Három adatmódosítást hajtunk végre, először a vásárlókhöz tartozó címek (irányítószám, város, utca) adatait, majd a bankkártya adatokat ( lejárat dátum, bank, típus) és végül a futócipőköz tartozó színek adatait módosítjuk.

```
package hu.domparse.b7ik21;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Result;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DomModifyB7IK21 {

    public static void main(String[] args) {
        NodeList list;
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            // fájl beolvasása
            Document document = builder.parse(new File("XMLB7IK21.xml"));
            document.getDocumentElement().normalize();

            // Aktuális elem meghatározása
            list = document.getElementsByTagName("vasarlo");
```

```

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Vásárló adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;

        // Vásárlóid eltárolása egy stringbe
        String id = element.getAttribute("vasarloid");

        System.out.println("Vásárló id: " + element.getAttribute("vasarloid"));
        System.out.println("Futócipő id: " + element.getAttribute("futocipoid"));

        System.out.println("Név: " + element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Telefonszám: " + element.getElementsByTagName("telefonszam").item(0).getTextContent());
        // Vásárló cím adatainak kiírása
        Node nodecim = list.item(i);
        if (nodecim.getNodeType() == Node.ELEMENT_NODE) {
            Element elemcim = (Element) node;
            // id vizsgálata és ha megegyezzik vas1-el akkor irányítószám módosítás
            if (id.equals("vas1")) {
                elemcim.getElementsByTagName("iranyitoszam").item(0).setTextContent("3333");
                elemcim.getElementsByTagName("varos").item(0).setTextContent("Miskolc");
                elemcim.getElementsByTagName("utca").item(0).setTextContent("Új utca 33.");
            }
            System.out.println("Irányítószám: "
                + elemcim.getElementsByTagName("iranyitoszam").item(0).getTextContent());
            System.out.println("Város: " + elemcim.getElementsByTagName("varos").item(0).getTextContent());
            System.out.println("Utca: " + elemcim.getElementsByTagName("utca").item(0).getTextContent());
        }
    }
}

// Aktuális elem meghatározása
list = document.getElementsByTagName("bankkartya");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Bankkártya adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elementBankkartya = (Element) node;

        // Kártyaszám eltárolása egy stringbe
        String id = elementBankkartya.getAttribute("kartyaszam");

        if (id.equals("11224545")) {
            elementBankkartya.getElementsByTagName("lejarati_datum").item(0).setTextContent("2025.05.05");
            elementBankkartya.getElementsByTagName("bank").item(0).setTextContent("Budapest Bank");
            elementBankkartya.getElementsByTagName("tipus").item(0).setTextContent("Mastercard World");
        }

        System.out.println("Kártyaszám: " + elementBankkartya.getAttribute("kartyaszam"));
        System.out.println("Vásárló id: " + elementBankkartya.getAttribute("vasarloid"));

        System.out.println("Lejárati dátum: "
            + elementBankkartya.getElementsByTagName("lejarati_datum").item(0).getTextContent());
        System.out.println("Bank: " + elementBankkartya.getElementsByTagName("bank").item(0).getTextContent());
        System.out.println("Típus: " + elementBankkartya.getElementsByTagName("tipus").item(0).getTextContent());
    }
}

```



```

// Aktuális elem meghatározása
list = document.getElementsByTagName("fut_szin");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nCurrent element: " + node.getNodeName());
    // Apcaps adatainak kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elementFutMeret = (Element) node;

        // Futócipő ID eltárolása egy stringbe
        String id = elementFutMeret.getAttribute("futocipoid");

        if (id.equals("f1")) {
            elementFutMeret.getElementsByTagName("szin").item(0).setTextContent("Zöld");
        }

        if (id.equals("f2")) {
            elementFutMeret.getElementsByTagName("szin").item(0).setTextContent("Lila");
        }

        System.out.println("Futócipő id: " + elementFutMeret.getAttribute("futocipoid"));

        System.out
            .println("Szín: " + elementFutMeret.getElementsByTagName("szin").item(0).getTextContent());
    }
}

// létrehozza az xml fájlt módosítva
Transformer transformer = TransformerFactory.newInstance().newTransformer();
Source input = new DOMSource(document);
Result output = new StreamResult(new File("XMLB7IK21Modify.xml"));
transformer.transform(input, output);

} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

### 3c) Adatlekérdezés: Négy lekérdezést hajtok végre.

1. Milyen futócipő méretek vannak
2. A vásárlók neveinek keresése
3. A vásárlók címeinek keresése
4. Kilistázza azon címeket, amelyik a megadott irányítószámnál nagyobbak

```
package hu.domparse.b7ik21;

import java.io.File;

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;

public class DOMQueryB7IK21 {

    public static void main(String[] args) {

        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            // fájl beolvasása
            Document document = builder.parse(new File("XMLB7IK21.xml"));
            document.getDocumentElement().normalize();
            // Gyökér elem megkeresése
            System.out.println("Root element : " + document.getDocumentElement().getNodeName());
            System.out.println("-----");

            // Milyen futócipő méretek vannak
            query(document, "fut_meret", "meret");
            System.out.println("-----");

            // A vásárlók neveinek keresése
            query(document, "vasarlo", "nev");
```



```

        System.out.println("-----");

        // A vásárlók címeinek keresése
        query(document, "vasarlo", "cim");
        System.out.println("-----");

        // Kilistázza azon címeket, amelyik a megadott irányítószámnál nagyobbak
        queryVasatloIranditoszamNagyon(document, "vasarlo", "irandyitoszam", 3000);
        System.out.println("-----");

    } catch (ParserConfigurationException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

public static void query(Document document, String elemKeres, String mitKeres) {
    // Aktuális elem meghatározása
    NodeList list = document.getElementsByTagName(elemKeres);

    for (int i = 0; i < list.getLength(); i++) {
        NodeList nodeList = list.item(i).getChildNodes();

        for (int k = 0; k < nodeList.getLength(); k++) {
            Node gye = nodeList.item(k);
            if (gye.getNodeName().compareTo(mitKeres) == 0) {
                System.out.println(gye.getTextContent());
            }
        }
    }
}

```

```

}

public static void queryVasatloIranditoszamNagyon(Document document, String elemKeres, String irsz, int keresIrsz) {
    // Aktuális elem meghatározása
    NodeList list = document.getElementsByTagName(elemKeres);
    int irszVan;

    for (int i = 0; i < list.getLength(); i++) {
        Node node = list.item(i);
        // Vásárló adatainak kiírása
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            // Vásárló cím adatainak kiírása
            Node nodecim = list.item(i);
            if (nodecim.getNodeType() == Node.ELEMENT_NODE) {
                Element elementIrsz = (Element) node;
                irszVan = Integer.parseInt(elementIrsz.getElementsByTagName(irsz).item(0).getTextContent());
                if (irszVan > keresIrsz) {
                    System.out.println("Irányítószám: "
                        + elementIrsz.getElementsByTagName("irandyitoszam").item(0).getTextContent());
                    System.out.println(
                        "Város: " + elementIrsz.getElementsByTagName("varos").item(0).getTextContent());
                    System.out
                        .println("Utca: " + elementIrsz.getElementsByTagName("utca").item(0).getTextContent());
                    System.out.println();
                }
            }
        }
    }
}
}
}
}
}

```