

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Magic the Gathering kártyaverseny

Beadandó féléves feladat.

Készítette: Szalmás Balázs

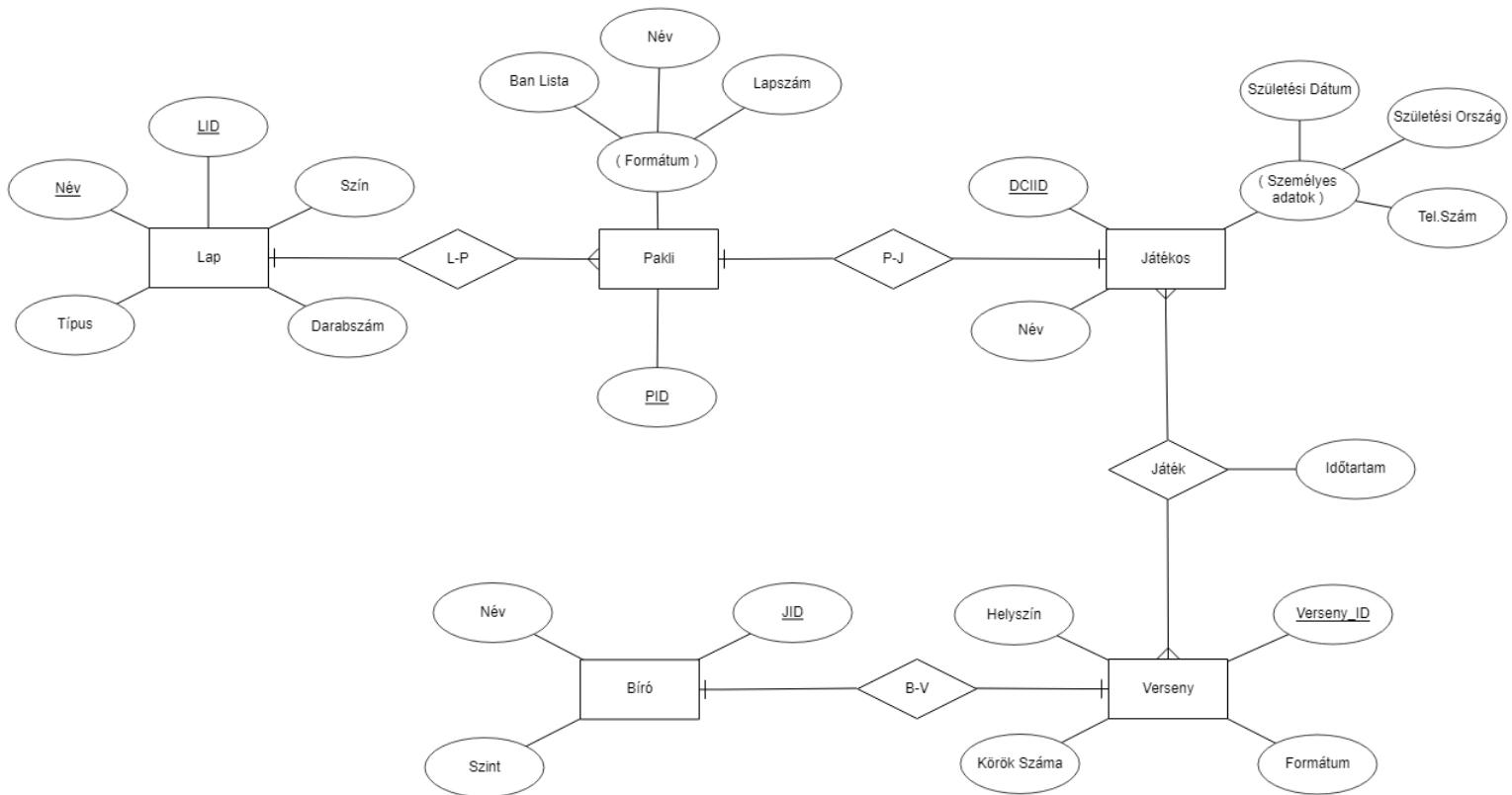
Neptun kód: RMIJO2

1. Feladat

A feladathoz témakörömnek, egy Magic the Gathering (MtG), a világ egyik legismertebb gyűjtögetős kártyajátékából tartott és éppen folyó versenyét választottam.

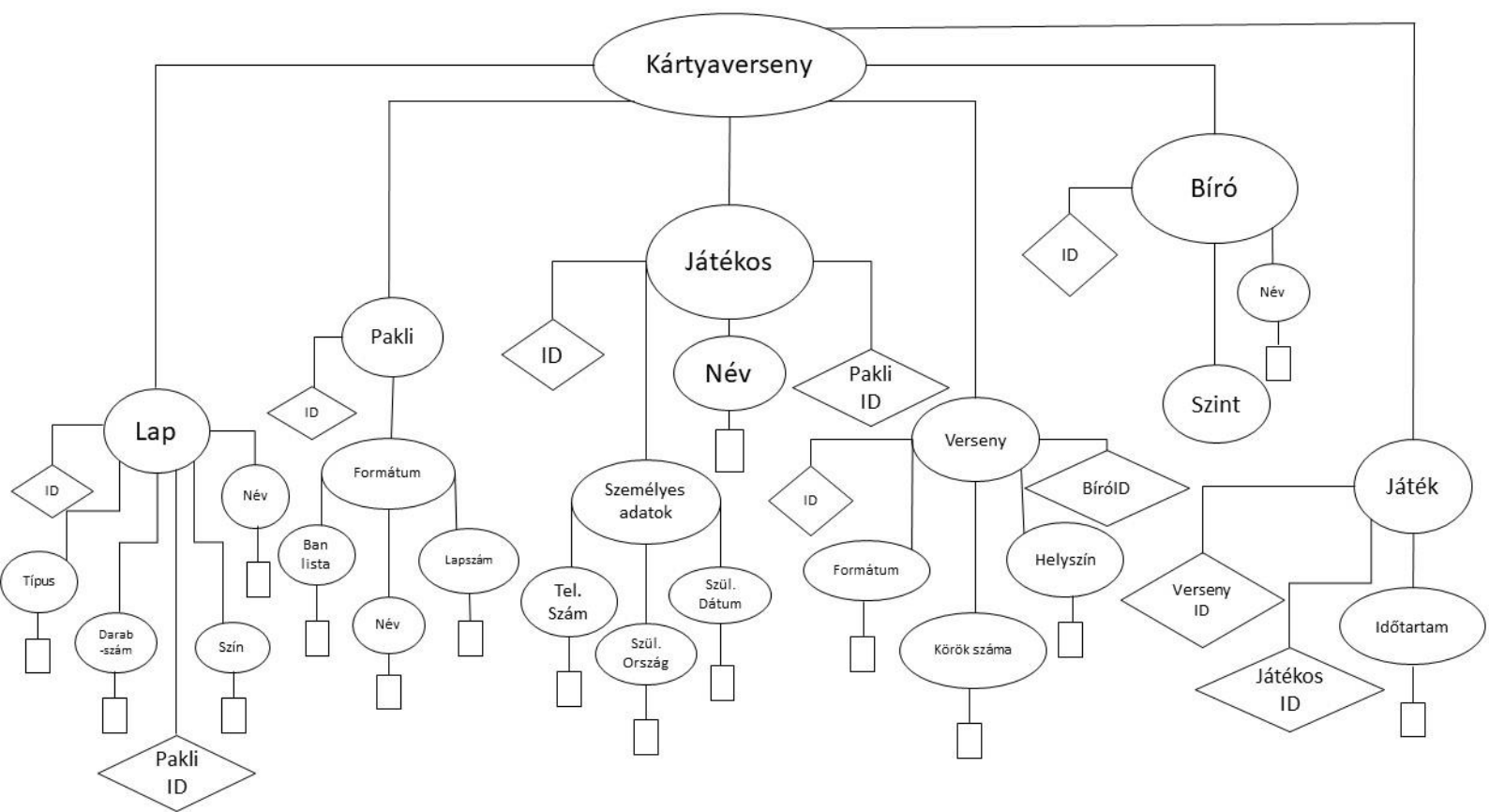
Egy MtG versenynek relatíve egyszerű a felépítése, egy adott rendezvény tarthat akár több versenyt is, amelyre a játékosok jelentkezhetnek. Minden játékos rendelkezik egy azonosító számmal és egy paklival, amellyel részt vehetnek a versenyeken. egy adott verseny minimum 3 körös, azaz minden játékos 3 kört játszik ellenfelével. Minden egyes játéknak van egy maximum játékidéje, mely a szabályzat szerint 50 perc. A játékot formátumtól függően, egy 60 vagy 100 lapos paklival játszá, az általam bemutatott példában ez 60 lap lesz. Minden játékos paklijának fel van tüntetve a neve, a lapszáma és a Ban listája (hivatalosan tiltott lapok listája) melyet egy link formájában elhelyeztem mindegyik paklinál. A bemutató példában természetesen a paklik lapjai is szerepelnek, ezekből csak viszont paklikként 6-6 darab kerül bemutatásra.

a) Az adatbázis ER modellje



b) Az adatbázis konvertálása XDM modellre

A játék több-több kapcsolatból létrejön egy Játék nevű objektum. A konvertálás során idegen kulcsok (foreign key) jönnek létre. A Játékban megjelenik a verseny és a játékos, a játékosban és a lapban, megjelenik a pakli idegen kulcsként. A verseny és a játékos megmutatja, hogy melyik játékos melyik versenyen és éppen melyik játékban vesz részt, a pakli kulcsai pedig megmutatják, hogy egy adott pakli melyik játékosé, és azt, hogy adott lapok melyik pakliba tartoznak.



c) XML dokumentum

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<magicVerseny xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="versenyXMLSchemaRMII02.xsd">
  <verseny biroID="1" id="1">
    <formatum>Modern</formatum>
    <helyszin>Ghent, Belgium, Flanders Expo</helyszin>
    <korok_szama>3</korok_szama>
  </verseny>
  <jatek jatekosID="1" versenyID="1">
    <idotartam>50 perc</idotartam>
  </jatek>
  <jatek jatekosID="2" versenyID="1">
    <idotartam>21 perc</idotartam>
  </jatek>
  <jatek jatekosID="3" versenyID="1">
    <idotartam>13 perc</idotartam>
  </jatek>
  <jatek jatekosID="4" versenyID="1">
    <idotartam>0 perc</idotartam>
  </jatek>
  <jatek jatekosID="5" versenyID="1">
    <idotartam>42 perc</idotartam>
  </jatek>
  <jatek jatekosID="6" versenyID="1">
    <idotartam>7 perc</idotartam>
  </jatek>
  <jatek jatekosID="7" versenyID="1">
    <idotartam>28 perc</idotartam>
  </jatek>
  <jatek jatekosID="8" versenyID="1">
    <idotartam>1 perc</idotartam>
  </jatek>
  <biro id="1">
    <Nev> Jeremy Edimborough</Nev>
    <Szint>3</Szint>
  </biro>
  <jatekos id="1" pakliID="1" versenyID="1">
    <Nev>Karsten Borresen </Nev>
    <szemelyes_adatok>
      <Szul_datum>1990-06-15</Szul_datum>
      <Sul_oroszag>Norvégia</Sul_oroszag>
      <Telefonszam>00471672367</Telefonszam>
    </szemelyes_adatok>
  </jatekos>
  <jatekos id="2" pakliID="2" versenyID="1">
    <Nev>Aleksander Vlakovits</Nev>
    <szemelyes_adatok>
      <Szul_datum>1989-11-25</Szul_datum>
      <Sul_oroszag>Ukrajna</Sul_oroszag>
      <Telefonszam>+36018462471</Telefonszam>
    </szemelyes_adatok>
  </jatekos>
```

```
<jatekos id="3" pakliID="3" versenyID="1">
  <Nev>Szalmás Balázs</Nev>
  <szemelyes_adatok>
    <Szul_datum>2001-02-09</Szul_datum>
    <Sul_oroszag>Magyarország</Sul_oroszag>
    <Telefonszam>+36307273956</Telefonszam>
  </szemelyes_adatok>
</jatekos>
<jatekos id="4" pakliID="4" versenyID="1">
  <Nev>Markus Highley</Nev>
  <szemelyes_adatok>
    <Szul_datum>1995-05-23</Szul_datum>
    <Sul_oroszag>Great Britain</Sul_oroszag>
    <Telefonszam>+4440115679</Telefonszam>
  </szemelyes_adatok>
</jatekos>
<jatekos id="5" pakliID="5" versenyID="1">
  <Nev>Mark Rosewater</Nev>
  <szemelyes_adatok>
    <Szul_datum>1967-05-25</Szul_datum>
    <Sul_oroszag>Egyesült Államok</Sul_oroszag>
    <Telefonszam>+16751639351</Telefonszam>
  </szemelyes_adatok>
</jatekos>
<jatekos id="6" pakliID="6" versenyID="1">
  <Nev>Beringar Werner</Nev>
  <szemelyes_adatok>
    <Szul_datum>2002-07-11</Szul_datum>
    <Sul_oroszag>Németország</Sul_oroszag>
    <Telefonszam>+46112567163</Telefonszam>
  </szemelyes_adatok>
</jatekos>
<jatekos id="7" pakliID="6" versenyID="1">
  <Nev>Ota Hideki</Nev>
  <szemelyes_adatok>
    <Szul_datum>1990-08-15</Szul_datum>
    <Sul_oroszag>Japán</Sul_oroszag>
    <Telefonszam>+81151631643</Telefonszam>
  </szemelyes_adatok>
</jatekos>
<jatekos id="8" pakliID="2" versenyID="1">
  <Nev>Farkas Bence</Nev>
  <szemelyes_adatok>
    <Szul_datum>1994-08-14</Szul_datum>
    <Sul_oroszag>Magyarország</Sul_oroszag>
    <Telefonszam>+364062464365</Telefonszam>
  </szemelyes_adatok>
</jatekos>
<pakli id="1">
  <formatum>
    <Nev>Murktide Tempo</Nev>
    <Ban_lista>https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted</Ban_lista>
    <Lapszam>60</Lapszam>
  </formatum>
</pakli>
```

```
<pakli id="2">
  <formatum>
    <Nev>Death's Shadow</Nev>
    <Ban_lista>https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted</Ban_lista>
    <Lapszam>60</Lapszam>
  </formatum>
</pakli>
<pakli id="3">
  <formatum>
    <Nev>Azorious Control</Nev>
    <Ban_lista>https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted</Ban_lista>
    <Lapszam>60</Lapszam>
  </formatum>
</pakli>
<pakli id="4">
  <formatum>
    <Nev>Amulet Titan</Nev>
    <Ban_lista>https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted</Ban_lista>
    <Lapszam>60</Lapszam>
  </formatum>
</pakli>
<pakli id="5">
  <formatum>
    <Nev>Burn</Nev>
    <Ban_lista>https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted</Ban_lista>
    <Lapszam>60</Lapszam>
  </formatum>
</pakli>
<pakli id="6">
  <formatum>
    <Nev>Jund</Nev>
    <Ban_lista>https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted</Ban_lista>
    <Lapszam>60</Lapszam>
  </formatum>
</pakli>

<lap id="1" pakliID="1">
  <Nev>Raganvan, Nimble Pilferer</Nev>
  <Szin>Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="2" pakliID="1">
  <Nev>Brazen Borrower</Nev>
  <Szin>Kék</Szin>
  <Darabszam>1</Darabszam>
  <Tipus>Creature//Sorcery</Tipus>
</lap>
<lap id="3" pakliID="1">
  <Nev>Consider</Nev>
  <Szin>Kék</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Instant</Tipus>
</lap>
```

```

<lap id="4" pakliID="1">
  <Nev>Expressive Iteration</Nev>
  <Szin>Kék/Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Sorcery</Tipus>
</lap>
<lap id="5" pakliID="1">
  <Nev>Mishra's Bauble</Nev>
  <Szin>Színtelen</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Artifact</Tipus>
</lap>
<lap id="6" pakliID="1">
  <Nev>Spirebluff Canal</Nev>
  <Szin>Színtelen</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Land</Tipus>
</lap>
<lap id="7" pakliID="2">
  <Nev>Death's Shadow</Nev>
  <Szin>Fekete</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="8" pakliID="2">
  <Nev>Kroxa, Titan of Death's Hunger</Nev>
  <Szin>Fekete/Piros</Szin>
  <Darabszam>2</Darabszam>
  <Tipus>Lény</Tipus>
</lap>
<lap id="9" pakliID="2">
  <Nev>Thoughtseize</Nev>
  <Szin>Fekete</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Sorcery</Tipus>
</lap>
<lap id="10" pakliID="2">
  <Nev>Drown in the Loch</Nev>
  <Szin>Kék/Fekete</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Instant</Tipus>
</lap>
<lap id="11" pakliID="2">
  <Nev>Dress Down</Nev>
  <Szin>Kék</Szin>
  <Darabszam>2</Darabszam>
  <Tipus>Enchantment</Tipus>
</lap>
<lap id="12" pakliID="2">
  <Nev>Kolaghan's Command</Nev>
  <Szin>Fekete/Piros</Szin>
  <Darabszam>1</Darabszam>
  <Tipus>Instant</Tipus>
</lap>

```

```

<lap id="13" pakliID="3">
  <Nev>Solitude</Nev>
  <Szin>Fehér</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="14" pakliID="3">
  <Nev>Teferi, Time Raveler</Nev>
  <Szin>Fehér/Kék</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Planeswalker - Teferi</Tipus>
</lap>
<lap id="15" pakliID="3">
  <Nev>Teferi, Hero of Dominaria</Nev>
  <Szin>Fehér/Kék</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Planeswalker - Teferi</Tipus>
</lap>
<lap id="16" pakliID="3">
  <Nev>Prismatic Ending</Nev>
  <Szin>Fehér</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Sorcery</Tipus>
</lap>
<lap id="17" pakliID="3">
  <Nev>Fire//Ice</Nev>
  <Szin>Kék/Piros</Szin>
  <Darabszam>2</Darabszam>
  <Tipus>Instant//Instant</Tipus>
</lap>
<lap id="18" pakliID="3">
  <Nev>Counterspell</Nev>
  <Szin>Kék</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Instant</Tipus>
</lap>
<lap id="19" pakliID="4">
  <Nev>Primeval Titan</Nev>
  <Szin>Zöld</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="20" pakliID="4">
  <Nev>Amulet of Vigor</Nev>
  <Szin>Színtelen</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Artifact</Tipus>
</lap>
<lap id="21" pakliID="4">
  <Nev>Summoner's Pact</Nev>
  <Szin>Zöld</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Instant</Tipus>
</lap>

```



```

<lap id="22" pakliID="4">
  <Nev>Dryad of the Ilysan Grove</Nev>
  <Szin>Zöld</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Enchantment Creature</Tipus>
</lap>
<lap id="23" pakliID="4">
  <Nev>Azusa, Lost but Seeking</Nev>
  <Szin>Zöld</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="24" pakliID="4">
  <Nev>Explore</Nev>
  <Szin>Zöld</Szin>
  <Darabszam>3</Darabszam>
  <Tipus>Sorcery</Tipus>
</lap>
<lap id="25" pakliID="5">
  <Nev>Goblin Guide</Nev>
  <Szin>Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="26" pakliID="5">
  <Nev>Monastery Swiftspear</Nev>
  <Szin>Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="27" pakliID="5">
  <Nev>Eidolon of the Great Revel</Nev>
  <Szin>Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Enchantment Creature</Tipus>
</lap>
<lap id="28" pakliID="5">
  <Nev>Lava Spike</Nev>
  <Szin>Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Sorcery</Tipus>
</lap>
<lap id="29" pakliID="5">
  <Nev>Lightning Bolt</Nev>
  <Szin>Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Instant</Tipus>
</lap>
<lap id="30" pakliID="5">
  <Nev>Boros Charm</Nev>
  <Szin>Piros/Fehér</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Sorcery</Tipus>
</lap>

```

```

<lap id="31" pakliID="6">
  <Nev>Tarmogoyf</Nev>
  <Szin>Zöld</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Creature</Tipus>
</lap>
<lap id="32" pakliID="6">
  <Nev>Iquisition of Kozilek</Nev>
  <Szin>Fekete</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Sorcery</Tipus>
</lap>
<lap id="33" pakliID="6">
  <Nev>Unholy Heat</Nev>
  <Szin>Piros</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Instant</Tipus>
</lap>
<lap id="34" pakliID="6">
  <Nev>Abrupt Decay</Nev>
  <Szin>Fekete/Zöld</Szin>
  <Darabszam>1</Darabszam>
  <Tipus>Instant</Tipus>
</lap>
<lap id="35" pakliID="6">
  <Nev>Expedition Map</Nev>
  <Szin>Színtelen</Szin>
  <Darabszam>1</Darabszam>
  <Tipus>Artifact</Tipus>
</lap>
<lap id="36" pakliID="6">
  <Nev>Wooded Foothill</Nev>
  <Szin>Színtelen</Szin>
  <Darabszam>4</Darabszam>
  <Tipus>Land</Tipus>
</lap>
</magicVerseny>

```

d) XMLSchema készítése

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  <xs:element name="magicVerseny">
    <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="verseny" type="versenyType"/>
          <xs:element name="jatek" type="jatekType"/>
          <xs:element name="biro" type="biroType"/>
          <xs:element name="jatekos" type="jatekosType"/>
          <xs:element name="pakli" type="pakliType"/>
          <xs:element name="lap" type="lapType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>

    <xs:key name="versenyId_PK">
      <xs:selector xpath="verseny"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="biroId_PK">
      <xs:selector xpath="biro"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="jatekosId_PK">
      <xs:selector xpath="jatekosok/jatekos"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="pakliId_PK">
      <xs:selector xpath="paklik/pakli"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="lapId_PK">
      <xs:selector xpath="lapok/lap"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:keyref name="versenyId_FK" refer="versenyId_PK">
      <xs:selector xpath="jatekosok/jatekos"/>
      <xs:field xpath="@versenyID"/>
    </xs:keyref>
    <xs:keyref name="biroId_FK" refer="biroId_PK">
      <xs:selector xpath="verseny"/>
      <xs:field xpath="@biroID"/>
    </xs:keyref>
    <xs:keyref name="jatekosId_FK" refer="jatekosId_PK">
      <xs:selector xpath="jatek"/>
      <xs:field xpath="@jatekosID1"/>
    </xs:keyref>
    <xs:keyref name="pakliId_FK" refer="pakliId_PK">
      <xs:selector xpath="jatekosok/jatekos"/>
      <xs:field xpath="@pakliID"/>
    </xs:keyref>
    <xs:keyref name="lapId_FK" refer="lapId_PK">
      <xs:selector xpath="lapok/lap"/>
      <xs:field xpath="@pakliID"/>
    </xs:keyref>
  </xs:element>

```

```

<xs:complexType name="versenyType">
  <xs:sequence>
    <xs:element name="formatum" type="xs:string"/>
    <xs:element name="helyszin" type="xs:string"/>
    <xs:element name="korok_szama" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="biroID" type="xs:unsignedByte" use="required"/>
  <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
</xs:complexType>

<xs:complexType name="jatekType">
  <xs:sequence>
    <xs:element name="idotartam" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="jatekosID" type="xs:unsignedByte" use="required"/>
  <xs:attribute name="versenyID" type="xs:unsignedByte" use="required"/>
</xs:complexType>

<xs:complexType name="biroType">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Szint" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
</xs:complexType>

<xs:complexType name="jatekosType">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="szemelyes_adatok"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
  <xs:attribute name="pakliID" type="xs:unsignedByte" use="required"/>
  <xs:attribute name="versenyID" type="xs:unsignedByte" use="required"/>
</xs:complexType>

<xs:complexType name="pakliType">
  <xs:sequence>
    <xs:element name="formatum" type="formatumType"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
</xs:complexType>

<xs:complexType name="lapType">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Szin" type="xs:string"/>
    <xs:element name="Darabszam" type="xs:integer"/>
    <xs:element name="Tipus" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
  <xs:attribute name="pakliID" type="xs:unsignedByte" use="required"/>
</xs:complexType>

```

```

<xs:complexType name="formatumType">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Ban_lista" type="xs:string"/>
    <xs:element name="Lapszam" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="szemelyes_adatok">
  <xs:sequence>
    <xs:element name="Szul_datum" type="xs:integer"/>
    <xs:element name="Szul_oroszag" type="xs:string"/>
    <xs:element name="Telefonszam" type="telType"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="telType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{2}-\d{3}-\d{2}-\d{2}"/>
    <xs:pattern value="\d{2}-\d{3}-\d{3}"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

2.Feladat

a) DOM olvasás + Kommentek

Ebben a feladatrészben, DOM segítségével, beolvasásokat hajtok végre az előző feladatban elkészített egyedeimen.

```

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 DomReadRMIJ02 {

    public static void main(String[] args) {
        try {
            File xmlFile = new File("XMLRMIJ02.xml"); //beolvasandó file.
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //Az XML dokumentumból DOM objektumot csinálunk.

            DocumentBuilder dBuilder = factory.newDocumentBuilder(); //Document lekéréséhez XML file.
            Document doc = dBuilder.parse(xmlFile); //Dokumentum lekérése.
            doc.getDocumentElement().normalize();
            System.out.println("Magic Verseny információk lekérése.");
            Read(doc);
        } catch (ParserConfigurationException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (SAXException e) {
            e.printStackTrace();
        }
    }

    public static void Read(Document doc) {
        NodeList nList = doc.getElementsByTagName("verseny"); //verseny tagú elemek lekérése egy listába.
        for(int i = 0; i < nList.getLength(); i++) { //listaiterátor
            Node nNode = nList.item(i);
            Element element = (Element) nNode; // A lista aktuális elemét,lekérése után, Elementé konvertáljuk.
            if(nNode.getNodeType() == Node.ELEMENT_NODE) {
                String formatum = element.getElementsByTagName("formatum").item(0).getTextContent();
                String helyszin = element.getElementsByTagName("helyszin").item(0).getTextContent();
                String korszam = element.getElementsByTagName("korok_szama").item(0).getTextContent();
                String versenyID = element.getAttribute("versenyID");
                System.out.println("Verseny információk: \tFormátum:\t"+ formatum+
                    "\n\tHelyszín:\t"+helyszin+
                    "\n\tKörök Száma:\t"+korszam);
                ReadJatekByID(doc, versenyID);
            }
        }
    }
}

```

```

public static void ReadJatekByID(Document doc, String versenyID) {
    NodeList nList = doc.getElementsByTagName("jatek");
    for(int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if(nNode.getNodeType() == Node.ELEMENT_NODE) {
            String idotartam = element.getElementsByTagName("idotartam").item(0).getTextContent();
            String jatekosID = element.getAttribute("jatekosID");
            System.out.println("\n" + (i+1) + ". versenyző \n\tAz aktuális kör hátralévő ideje:\t"+idotartam);

            ReadJatekosByID(doc, jatekosID);
        }
    }
}

public static void ReadJatekosByID(Document doc, String jatekosID) {

    NodeList nList = doc.getElementsByTagName("jatekos");
    for(int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if(nNode.getNodeType() == Node.ELEMENT_NODE) {
            if(element.getAttribute("id").equals(jatekosID)) {
                String nev = element.getElementsByTagName("Nev").item(0).getTextContent();
                String szul_datum = element.getElementsByTagName("Szul_datum").item(0).getTextContent();
                String szul_oroszag = element.getElementsByTagName("Sul_oroszag").item(0).getTextContent();
                String telszam = element.getElementsByTagName("Telefonszam").item(0).getTextContent();
                System.out.println("Játekos neve: " + nev +
                    "\nSzemelyes adatok:\t \n\tSzületési Dátum:\t" + szul_datum +
                    "\n\tSzületési ország:\t" + szul_oroszag + "\n\tTelefonszám:\t" + telszam);
                String pakliID = element.getAttribute("pakliID");
                ReadPakliByID(doc, pakliID);
            }
        }
    }
}

public static void ReadPakliByID(Document doc, String pakliID) {

    NodeList nList = doc.getElementsByTagName("pakli");
    for(int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if(nNode.getNodeType() == Node.ELEMENT_NODE) {
            if(element.getAttribute("id").equals(pakliID)) {
                String nev = element.getElementsByTagName("Nev").item(0).getTextContent();
                String banlist = element.getElementsByTagName("Ban_lista").item(0).getTextContent();
                String lapszam = element.getElementsByTagName("Lapszam").item(0).getTextContent();
                System.out.println("Pakli adatok: \t Pakli név: \t"+nev+
                    "\n\tBan Lista: \t"+banlist+
                    "\n\tLapszám: \t"+lapszam);
                ReadLapByPakliID(doc, pakliID);
            }
        }
    }
}
}

```

```

public static void ReadLapByPakliID(Document doc, String pakliID) {
    System.out.println("A pakli fontosabb lapjai:");
    NodeList nList = doc.getElementsByTagName("lap");
    for(int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if(nNode.getNodeType() == Node.ELEMENT_NODE) {
            if(element.getAttribute("pakliID").equals(pakliID)) {
                String nev = element.getElementsByTagName("Nev").item(0).getTextContent();
                String szin = element.getElementsByTagName("Szin").item(0).getTextContent();
                String db = element.getElementsByTagName("Darabszam").item(0).getTextContent();
                String tipus = element.getElementsByTagName("Tipus").item(0).getTextContent();
                System.out.println("\tLap név: \t"+nev+
                    "\n\tSzín: \t"+szin+
                    "\n\tDb: \t"+db+
                    "\n\tTipus: \t"+tipus);
            }
        }
    }
}

public static void ReadLapByID(Document doc, String lapID) {
    NodeList nList = doc.getElementsByTagName("lap");
    for(int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if(nNode.getNodeType() == Node.ELEMENT_NODE) {
            if(element.getAttribute("id").equals(lapID)) {
                String nev = element.getElementsByTagName("Nev").item(0).getTextContent();
                String szin = element.getElementsByTagName("Szin").item(0).getTextContent();
                String db = element.getElementsByTagName("Darabszam").item(0).getTextContent();
                String tipus = element.getElementsByTagName("Tipus").item(0).getTextContent();
                System.out.println("\tLap név: \t"+nev+
                    "\n\tSzín: \t"+szin+
                    "\n\tDb: \t"+db+
                    "\n\tTipus: \t"+tipus);
            }
        }
    }
}

```


Eredmény:

Magic Verseny információk lekérése.

Verseny információk: Formátum: Modern
Helyszín: Ghent, Belgium, Flanders Expo
Körök Száma: 3

1. versenyző

Az aktuális kör hátralévő ideje: 50 perc

Játekos neve: Karsten Borresen

Személyes adatok:

Születési Dátum: 1990-06-15

Születési ország: Norvégia

Telefonszám: 00471672367

Pakli adatok: Pakli név: Murktide Tempo

Ban Lista: <https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted>

Lapszám: 60

A pakli fontosabb lapjai:

Lap név: Raganvan, Nimble Pilferer

Szín: Piros

db: 4

Típus: Creature

Lap név: Brazen Borrower

Szín: Kék

db: 1

Típus: Creature//Sorcery

Lap név: Consider

Szín: Kék

db: 4

Típus: Instant

Lap név: Expressive Iteration

Szín: Kék/Piros

db: 4

Típus: Sorcery

Lap név: Mishra's Bauble

Szín: Színtelen

db: 4

Típus: Artifact

Lap név: Spirebluff Canal

Szín: Színtelen

db: 4

Típus: Land

b) DOM lekérdezések +Kommentek

Ebben a feladatrészben DOM segítségével lekérdezéseket hajtok végre a játékos Egyedeken.

```
package domparse;

import java.io.File;
import java.io.IOException;
import java.util.Scanner;

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

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 DomQueryRMIJ02 {

    public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException, TransformerException {
        File xmlFile = new File("XMLRMIJ02.xml"); //beolvasandó file.

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //Az XML dokumentumból DOM objektumot csinálunk.
        DocumentBuilder dBuilder = factory.newDocumentBuilder(); //Document lekéréséhez XML file.
        Document doc = dBuilder.parse(xmlFile); //Dokumentum lekérése.
        doc.getDocumentElement().normalize();

        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        System.out.println("-----");
        LoadJatekosQuery(doc);
    }

    private static void LoadJatekosQuery(Document doc) throws TransformerException {
        NodeList nodeList = doc.getElementsByTagName("jatekos"); //Lekérjük a játékos elemeket.
        String jatekos;
        Element element = null;
        Node nNode = null;
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            String nev = element.getElementsByTagName("Nev").item(0).getTextContent();
            System.out.println((i + 1) + " " + nev);
        }
    }
}
```

```

//Játékos kiválasztása.
System.out.println("Melyik játékos pakliját kívánja látni?");
Scanner sc = new Scanner(System.in);
jatekos = sc.nextLine();
for (int i = 0; i < nodeList.getLength(); i++) {
    nNode = nodeList.item(i);
    element = (Element) nNode;
    if (nNode.getNodeType() == Node.ELEMENT_NODE) {
        if(jatekos.equals("Karsten Borresen")) {
            LoadPakliQuery(doc,"1");
            break;
        }
        if(jatekos.equals("Aleksander Vlakovits")) {
            LoadPakliQuery(doc,"2");
            break;
        }
        if(jatekos.equals("Szalmás Balázs")) {
            LoadPakliQuery(doc,"3");
            break;
        }
        if(jatekos.equals("Markus Highley")) {
            LoadPakliQuery(doc,"4");
            break;
        }
        if(jatekos.equals("Mark Rosewater")) {
            LoadPakliQuery(doc,"5");
            break;
        }
        if(jatekos.equals("Beringar Werner")) {
            LoadPakliQuery(doc,"6");
            break;
        }
        if(jatekos.equals("Ota Hideki")) {
            LoadPakliQuery(doc,"7");
            break;
        }
        if(jatekos.equals("Farkas Bence")) {
            LoadPakliQuery(doc,"8");
            break;
        }
    }
}
sc.close();
}

```

```
//A kiválasztott játékos paklijának az adatainak kiírása.
private static void LoadPakliQuery(Document doc, String id) throws TransformerException {
    NodeList nodeList = doc.getElementsByTagName("pakli");
    int pakli = 0;
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        Element element = (Element) nNode;
        String pID = element.getAttribute("id");
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

            if(id.equals(pID)) {
                pakli +=1;
                System.out.println(pakli + ". pakli adatai:");
                String paID = element.getAttribute("id");
                DomReadRMIJO2.ReadPakliByID(doc, paID);
            }
        }
    }
}
```

Eredmény:

```
Root element: magicVerseny
-----
1) Karsten Borresen
2) Aleksander Vlakovits
3) Szalmás Balázs
4) Markus Highley
5) Mark Rosewater
6) Beringar Werner
7) Ota Hideki
8) Farkas Bence
Melyik játékos pakliját kívánja látni?
Szalmás Balázs
1. pakli adatai:
Pakli adatok:   Pakli név:   Azorious Control
Ban Lista:     https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
Lapszám:       60
A pakli fontosabb lapjai:
Lap név:       Solitude
Szín:          Fehér
db:            4
Típus:         Creature
Lap név:       Teferi, Time Raveler
Szín:          Fehér/Kék
db:            4
Típus:         Planeswalker - Teferi
Lap név:       Teferi, Hero of Dominaria
Szín:          Fehér/Kék
db:            4
Típus:         Planeswalker - Teferi
Lap név:       Prismatic Ending
Szín:          Fehér
db:            4
Típus:         Sorcery
Lap név:       Fire//Ice
Szín:          Kék/Piros
db:            2
Típus:         Instant//Instant
Lap név:       Counterspell
Szín:          Kék
db:            4
Típus:         Instant
```

c) DOM módosítás +Kommentek

Ebben a feladatrészen, DOM segítségével módosításokat hajtok végre négy különböző egyed típuson: a verseny, a pakli, a lap és a játékos elemeken.

```
package domparse;

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;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.io.IOException;
import java.util.Scanner;

public class DomModifyRMIJ02 {

    public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException, TransformerException {
        File xmlFile = new File("XMLRMIJ02.xml"); // xml fájl bekérése
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); // olvasás lehetővé tétele
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();
        System.out.println("XML Módosítása");
        System.out.println("Adja meg mit szeretne módosítani: ");
        System.out.println("1 Verseny módosítása\n2 Játékos módosítása\n3 Pakli módosítása\n4 Lap módosítása");
        Modify(doc);
    }

    //A modify metódusban megkérdezzük a felhasználót, melyik adatot kívánja megváltoztatni.
    private static void Modify(Document doc) throws TransformerException {
        int numberOfVerseny = doc.getElementsByTagName("verseny").getLength();
        int numberOfJatekos = doc.getElementsByTagName("jatekos").getLength();
        int numberOfPakli = doc.getElementsByTagName("pakli").getLength();
        int numberOfLap = doc.getElementsByTagName("lap").getLength();

        Scanner scanner = new Scanner(System.in);
        System.out.println("Adja meg a sorszámot: ");
        int readInt = scanner.nextInt();
        switch(readInt) {
            case 1:
                ModifyVerseny(doc, numberOfVerseny);
                break;
            case 2:
                ModifyJatekos(doc, numberOfJatekos);
                break;
            case 3:
                ModifyPakli(doc, numberOfPakli);
                break;
            case 4:
                ModifyLap(doc, numberOfLap);
                break;
        }
        scanner.close();
    }
}
```

```

private static void ModifyJatekos(Document doc, int numberOfJatekos) throws TransformerException {
    System.out.println("Melyik játékos adatait kívánja módosítani?");
    for (int i = 0; i < numberOfJatekos + 1; i++) {
        System.out.println(i + ". jatekos");
        DomReadRMIJ02.ReadJatekosByID(doc, String.valueOf(i));
    }
    String id = ReadID();
    Scanner sc = new Scanner(System.in);
    System.out.println("Név: ");
    String nev = sc.nextLine();
    System.out.print("Születési dátum: ");
    String szul_datum = sc.nextLine();
    System.out.print("Születési ország: ");
    String szul_oroszag = sc.nextLine();
    System.out.print("Telefonszám: ");
    String telszam = sc.nextLine();
    sc.close();
    NodeList nodeList = doc.getElementsByTagName("jatekos");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;
            String sid = element.getAttribute("id");
            if (sid.equals(id)) {
                Node node1 = element.getElementsByTagName("Nev").item(0);
                node1.setTextContent(nev);
                Node node2 = element.getElementsByTagName("Szul_datum").item(0);
                node2.setTextContent(szul_datum);
                Node node3 = element.getElementsByTagName("Sul_oroszag").item(0);
                node3.setTextContent(szul_oroszag);
                Node node4 = element.getElementsByTagName("Telefonszam").item(0);
                node4.setTextContent(telszam);
                System.out.println("Sikeres módosítás");
            }
        }
    }
    ModifyXML(doc);
}

```

```

private static void ModifyVerseny(Document doc, int numberOfVerseny) throws TransformerException {
    System.out.println("Melyik verseny adatait kívánja módosítani?");
    for (int i = 0; i < numberOfVerseny + 1; i++) {
        System.out.println(i + ". verseny");
        DomReadRMIJ02.Read(doc);
    }
    String id = ReadID();
    Scanner sc = new Scanner(System.in);
    System.out.println("Formátum: ");
    String format = sc.nextLine();
    System.out.print("Helyszín: ");
    String helyszin = sc.nextLine();
    System.out.print("Körök száma: ");
    String korszam = sc.nextLine();
    sc.close();
    NodeList nodeList = doc.getElementsByTagName("verseny");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;
            String sid = element.getAttribute("id");
            if (sid.equals(id)) {
                Node node1 = element.getElementsByTagName("formatum").item(0);
                node1.setTextContent(format);
                Node node2 = element.getElementsByTagName("helyszin").item(0);
                node2.setTextContent(helyszin);
                Node node3 = element.getElementsByTagName("korok_szama").item(0);
                node3.setTextContent(korszam);
                System.out.println("Sikeres módosítás");
            }
        }
    }
    ModifyXML(doc); // Létrehozzuk a módosított XML-t.
}

```

```

private static void ModifyLap(Document doc, int numberOfLap) throws TransformerException
{
    System.out.println("Melyik lap adatait kívánja módosítani?");
    for (int i = 0; i < numberOfLap + 1; i++) {
        System.out.println(i + ". lap");
        DomReadRMIJ02.ReadLapByID(doc, String.valueOf(i));
    }
    String id = ReadID();
    Scanner sc = new Scanner(System.in);
    System.out.println("Név: ");
    String nev = sc.nextLine();
    System.out.print("Szín: ");
    String szin = sc.nextLine();
    System.out.print("Darabszám:");
    String db = sc.nextLine();
    System.out.print("Típus: ");
    String tipus = sc.nextLine();
    sc.close();
    NodeList nodeList = doc.getElementsByTagName("lap");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;
            String sid = element.getAttribute("id");
            if (sid.equals(id)) {
                Node node1 = element.getElementsByTagName("Nev").item(0);
                node1.setTextContent(nev);
                Node node2 = element.getElementsByTagName("Szin").item(0);
                node2.setTextContent(szin);
                Node node3 = element.getElementsByTagName("Darabszam").item(0);
                node3.setTextContent(db);
                Node node4 = element.getElementsByTagName("Tipus").item(0);
                node4.setTextContent(tipus);

                System.out.println("Sikeres módosítás");
            }
        }
    }
    ModifyXML(doc);
}

private static String ReadID() {
    Scanner sc = new Scanner(System.in);
    System.out.print("\nid:");
    String id = sc.nextLine();
    return id;
}

public static void ModifyXML(Document doc) throws TransformerException {
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();
    DOMSource source = new DOMSource(doc);
    StreamResult result = new StreamResult(new File("XMLRMIJ02.xml"));
    transformer.transform(source, result);
}
}

```

```

private static void ModifyPakli(Document doc, int numberOfPakli) throws TransformerException {
    System.out.println("Melyik Pakli adatait kívánja módosítani?");
    for (int i = 0; i < numberOfPakli + 1; i++) {
        System.out.println(i + ". pakli");
        DomReadRMIJ02.ReadPakliByID(doc, String.valueOf(i));
    }
    String id = ReadID();
    Scanner sc = new Scanner(System.in);
    System.out.println("Név: ");
    String nev = sc.nextLine();
    System.out.print("Ban lista: ");
    String banlist = sc.nextLine();
    System.out.print("Lapszam: ");
    String lapszam = sc.nextLine();
    sc.close();
    NodeList nodeList = doc.getElementsByTagName("jatekos");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;
            String sid = element.getAttribute("id");
            if (sid.equals(id)) {
                Node node1 = element.getElementsByTagName("Nev").item(0);
                node1.setTextContent(nev);
                Node node2 = element.getElementsByTagName("Ban_lista").item(0);
                node2.setTextContent(banlist);
                Node node3 = element.getElementsByTagName("Lapszam").item(0);
                node3.setTextContent(lapszam);
                System.out.println("Sikeres módosítás");
            }
        }
    }
    ModifyXML(doc);
}

```

Eredmény:

```

XML Módosítása
Adja meg mit szeretne módosítani:
1 Verseny módosítása
2 Játékos módosítása
3 Pakli módosítása
4 Lap módosítása
Adja meg a sorszámot:
3
Melyik Pakli adatait kívánja módosítani?
0. pakli
1. pakli
Pakli adatok:   Pakli név:      Murktide Tempo
               Ban Lista:      https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
               Lapszám:        60
2. pakli
Pakli adatok:   Pakli név:      Death's Shadow
               Ban Lista:      https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
               Lapszám:        60
3. pakli
Pakli adatok:   Pakli név:      Azorious Control
               Ban Lista:      https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
               Lapszám:        60
4. pakli
Pakli adatok:   Pakli név:      Amulet Titan
               Ban Lista:      https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
               Lapszám:        60
5. pakli
Pakli adatok:   Pakli név:      Burn
               Ban Lista:      https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
               Lapszám:        60
6. pakli
Pakli adatok:   Pakli név:      Jund
               Ban Lista:      https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
               Lapszám:        60

id:3
Név:
Jeska Control
Ban lista: https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/banned-restricted
Lapszam: 60
Sikeres módosítás

```