

JEGYZŐKÖNYV

Web technológiák 1 gyakorlat

Féléves feladat

Darts projekt

Készítette: **Juhász Balázs**

Neptunkód: **FIWXDX**

Dátum: **2025. december**

Miskolc, 2025

Tartalomjegyzék

1. Bevezetés.....	2
2. Első feladatok	3
2.1 Az adatbázis ER modell.....	3
2.2 Az adatbázis konvertálása XDM modellre	4
2.3 Az XDM modell alapján XML dokumentum készítése.....	5
2.4 Az XML dokumentum alapján XMLSchema készítése	6
3. Második feladatok	7
3.1 Adatolvasás.....	7
3.2 Adat-lekérdezés.....	8
3.3 Adatmódosítás.....	9

1. Bevezetés

Az adatmodellezés és XML feldolgozás gyakorlati alkalmazása jól szemléltethető egy konkrét példán keresztül. A jelen projektben egy **darts versenyrendszer** adatait dolgoztam fel, amelyben országok, játékosok, felszerelések, versenyek és mérkőzések szerepelnek. A feladat célja az volt, hogy az adatbázis-tervezés teljes folyamatát végig kövessük: az **ER modell** megalkotásától kezdve az **XDM konverzió**n át az **XML dokumentum** és az **XML Schema (XSD)** létrehozásáig, majd a dokumentum feldolgozását a **DOM technológia** segítségével Java nyelven.

Az első lépésben elkészítettük az **ER modellt**, amely legalább öt egyedet tartalmaz valamint többféle kapcsolatot (1:1, 1:N, M:N). Az Egyedekhez kulcsokat, összetett és többértékű attribútumokat rendeltem, így biztosítva az adatbázis konzisztenciáját. Az ER modellből később **XDM modellt** készítettünk, amely már az XML struktúrához illeszkedik, és alkalmas volt arra, hogy szabványos XML dokumentumot generáljak belőle. Az XML-hez kapcsolódóan elkészítettük az **XML Schema (XSD)** fájlt is, amely saját típusokat valamint komplex típusokkal definiálta az összetett elemeket.

A kód minden részében hangsúlyt fektettem a **kommentek használatára**, hogy a megvalósítás átlátható és könnyen követhető legyen. A darts rendszer példája jól mutatja, hogyan lehet egy valós, sporthoz kapcsolódó adatbázist modellezni, majd XML formátumban kezelni és Java programmal feldolgozni.

A jegyzőkönyv célja, hogy lépésről lépésre ismertesse a tervezés és implementáció menetét, kiemelve a fontos kódrészleteket, és bemutassa, hogyan biztosítható az adatok helyessége és konzisztenciája a darts versenyrendszer példáján keresztül.

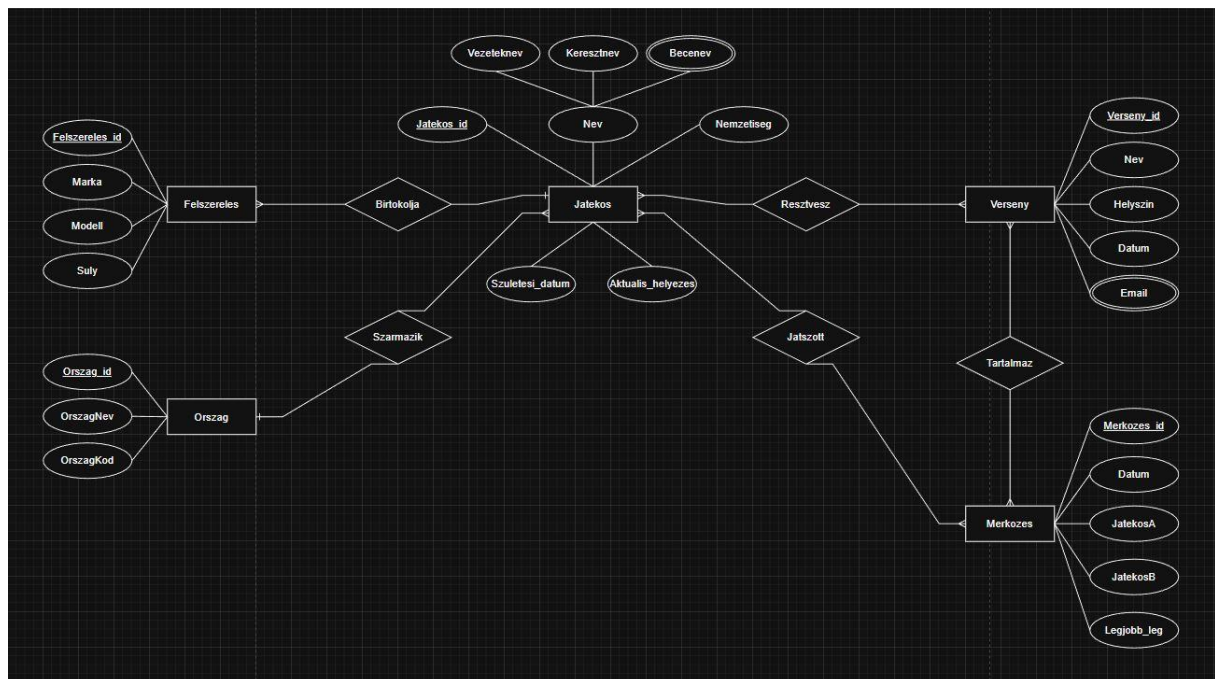
2. Első feladatok

2.1 Az adatbázis ER modell

Az adatbázis a darts versenyek rendszerét modellezi. Legalább 5 egyedet tartalmaz: Ország, Felszerelés, Játékos, Verseny, Mérkőzés. A kapcsolatok között szerepel:

- 1:N kapcsolat (egy országhoz több játékos tartozik),
- M:N kapcsolat (egy versenyen több játékos vesz részt, és egy játékos több versenyen indulhat)

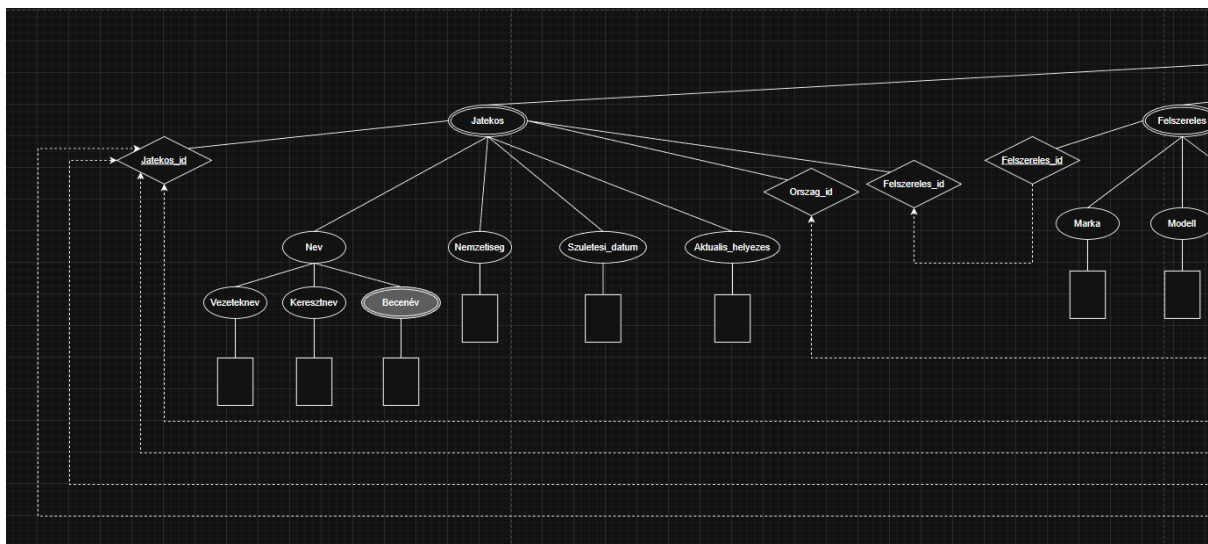
Az egyedekhez kulcsok (PK), idegen kulcsok, normál és összetett attribútumok tartoznak.



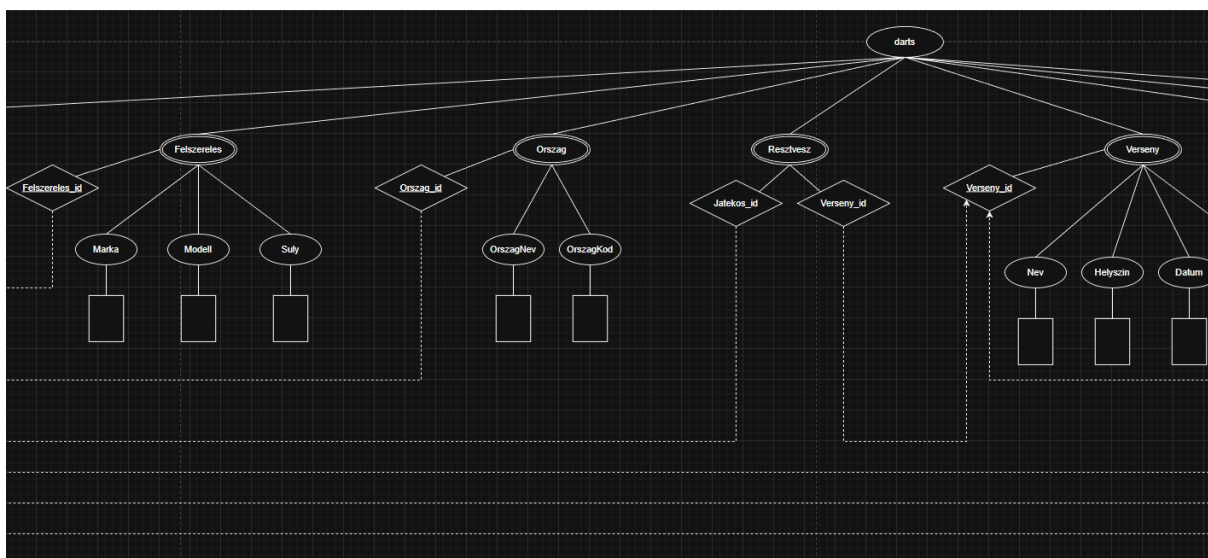
1. ábra: ER modell

2.2 Az adatbázis konvertálása XDM modellre

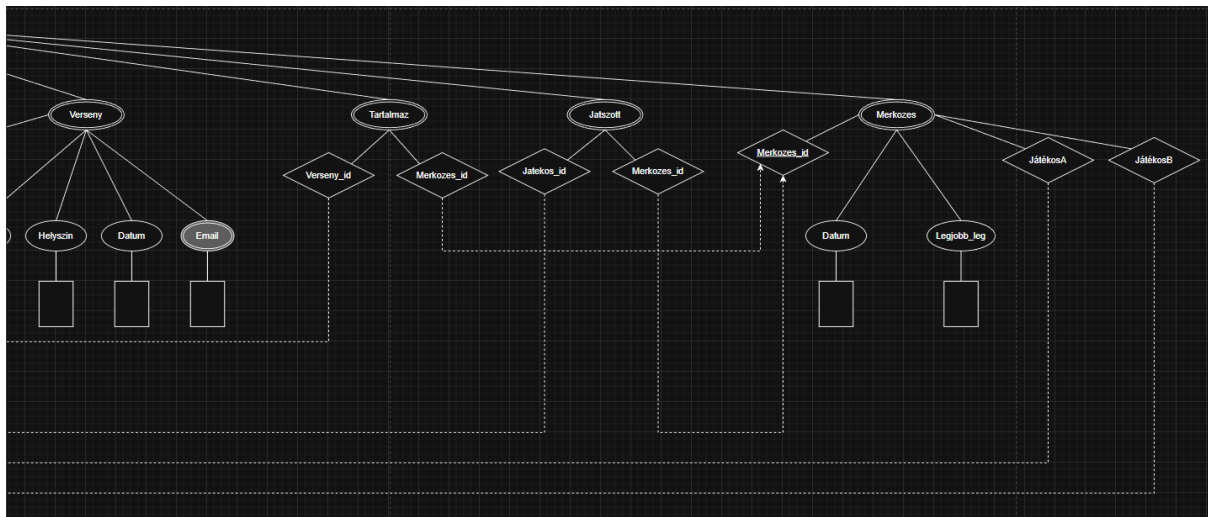
Az ER modellből XDM modellt készítettem, amely az XML struktúrához illeszkedik. Az Egyedekből elemek, a kapcsolatokból beágyazott elemek és attribútumok lettek. Az XDM modellben a PK és FK kapcsolatok vonalakkal jelölve, keresztezés nélkül.



2. ábra: XDM modell 1.



3. ábra: XDM modell 2.



4. ábra: XDM modell 3.

A modell készítése során muszáj voltam, három külön képernyőképet készíteni, mivel nagyon széles lett az elkészített modell! Viszont a külön FIWXDX_XDM.drawio fájl megnézésével meglehetősen lehet nézni.

2.3 Az XDM modell alapján XML dokumentum készítése

Az XDM modellből XML dokumentumot generáltam. Minden többszörös előfordulású elemhez legalább 2 példányt készítettem (pl. több ország, több játékos, több verseny). Az XML-ben attribútumokkal jelöltem az azonosítókat, és beágyazott elemekkel a részleteket.

```
<jatekos jatekos_id="1" orszag_id="3" felszereles_id="1">
  <nev>
    <vezeteknev>John</vezeteknev>
    <keresztnev>Smith</keresztnev>
    <becenev>Bully Boy</becenev>
  </nev>
  <nemzetiseg>Brit</nemzetiseg>
  <szuletesi_datum>1985-04-12</szuletesi_datum>
  <aktualis_helyezes>5</aktualis_helyezes>
</jatekos>
```

Magyarázat: A játékos elem attribútumai kapcsolódnak más entitásokhoz (ország, felszerelés), míg a gyermekelemek a játékos adatait tartalmazzák.

2.4 Az XML dokumentum alapján XMLSchema készítése

Az XML dokumentumhoz XSD sémát készítettem. Saját típusokat definiáltam az elemekhez, valamint komplex típusokat használtam az összetett elemekhez.

```
<xs:element name="jatekos">

  <xs:complexType>

    <xs:sequence>

      <xs:element name="nev" type="nevType"/>

      <xs:element name="nemzetiseg" type="xs:string"/>

      <xs:element name="szuletesi_datum" type="xs:date"/>

      <xs:element name="aktualis_helyezes" type="xs:integer"/>

    </xs:sequence>

    <xs:attribute name="jatekos_id" type="xs:integer" use="required"/>

    <xs:attribute name="orszag_id" type="xs:integer" use="required"/>

    <xs:attribute name="felszereles_id" type="xs:integer" use="required"/>

  </xs:complexType>

</xs:element>
```

Magyarázat: A jatekos elem komplex típust kapott, attribútumokkal az idegen kulcsokra, és gyermekelemekkel a részletekre.

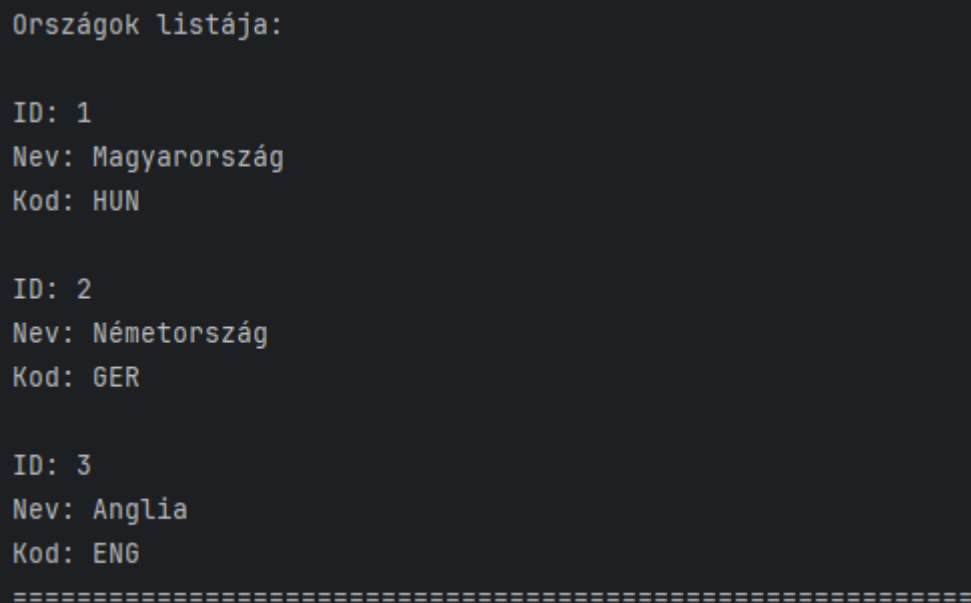
3. Második feladatok

3.1 Adatolvasás

A NeptunkodDOMRead.java osztály beolvassa az XML dokumentumot, normalizálja, majd kiírja a teljes tartalmat blokk formában a konzolra.

```
NodeList orszagList = doc.getElementsByTagName("ország");
for (int i = 0; i < orszagList.getLength(); i++) {
    Node nNode = orszagList.item(i);
    if (nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element e = (Element) nNode;
        System.out.println("ID: " + e.getAttribute("ország_id") +
            ", Név: " + e.getElementsByTagName("nev").item(0).getTextContent());
    }
}
```

Magyarázat: Az országok listáját bejárjuk, és minden elem attribútumait és gyermekelemeit kiírjuk.



```
Országok listája:

ID: 1
Név: Magyarország
Kod: HUN

ID: 2
Név: Németország
Kod: GER

ID: 3
Név: Anglia
Kod: ENG

=====
```

5. ábra: Adatolvasás példa

3.2 Adat-lekérdezés

A NeptunkodDOMQuery.java osztály célzott lekérdezéseket végez az XML dokumentumon. Legalább 4 lekérdezést készítettünk: országok listázása, magyar játékosok szűrése, versenyek dátumai, mérkőzések legjobb leg értékei.

```
String nemzetiseg =  
e.getElementsByTagName("nemzetiseg").item(0).getTextContent();  
  
if (nemzetiseg.equalsIgnoreCase("Magyar")) {  
  
    String nev = e.getElementsByTagName("vezeteknev").item(0).getTextContent() + "  
    "+  
  
        e.getElementsByTagName("keresztnev").item(0).getTextContent();  
  
    System.out.println("Név: " + nev + ", Nemzetiség: " + nemzetiseg);  
}
```

Magyarázat: A játékosok közül csak a magyar nemzetiségűeket szűrjük ki, és kiírjuk a nevüket.

```
--- Magyar játékosok ---  
Név: Nagy Gábor, Nemzetiség: Magyar
```

6. ábra: Lekérdezés példa

3.3 Adatmódosítás

A NeptunkodDOMModify.java osztályban az XML dokumentumot beolvastuk, majd legalább 4 módosítást hajtottunk végre rajta. A módosítások között szerepel attribútum átírása NamedNodeMap segítségével, gyermekelemek módosítása NodeList bejárással, dátum átírása, valamint új attribútum hozzáadása. A módosított dokumentumot a konzolra írtuk ki Transformer segítségével.

```
Node jatekos1 = doc.getElementsByTagName("jatekos").item(0);
```

```
NamedNodeMap attr = jatekos1.getAttributes();
```

```
Node orszagAttr = attr.getNamedItem("orszag_id");
```

```
orszagAttr.setTextContent("99"); // új ország ID
```

```
Node jatekos2 = doc.getElementsByTagName("jatekos").item(1);
```

```
NodeList childNodes = jatekos2.getChildNodes();
```

```
for (int i = 0; i < childNodes.getLength(); i++) {
```

```
    Node node = childNodes.item(i);
```

```
    if (node.getNodeType() == Node.ELEMENT_NODE) {
```

```
        Element elem = (Element) node;
```

```
        if ("becenev".equals(elem.getNodeName())) {
```

```
            elem.setTextContent("Lightning");
```

```
        }
```

```
    }
```

```
}
```

Magyarázat: Az első játékos attribútumát módosítjuk NamedNodeMap segítségével, míg a második játékos becenevét NodeList bejárással változtatjuk meg. Ez mutatja, hogy mindkét DOM eszközt tudatosan használjuk.

```

--- Módosított XML fájl ---

<?xml version="1.0" encoding="UTF-8" standalone="no"?><darts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="FIWXDX_XMLSchema.xsd">

<!-- Országok listája, azonosítóval és kóddal -->
<országok>
  <ország ország_id="1">
    <nev>Magyarország</nev>
    <kod>HUN</kod>
  </ország>
  <ország ország_id="2">
    <nev>Németország</nev>
    <kod>GER</kod>
  </ország>
  <ország ország_id="3">
    <nev>Anglia</nev>
    <kod>ENG</kod>
  </ország>
</országok>

```

7. ábra: Módosítás példa 1.

```

<!-- Játékosok, kapcsolódnak országhoz és felszereléshez -->
<jatekosok>
  <jatekos felszereles_id="1" jatekos_id="1" ország_id="99">
    <nev>
      <vezeteknev>John</vezeteknev>
      <keresztnev>Smith</keresztnev>
      <becenev>Bully Boy</becenev>
    </nev>
    <nemzetiseg>Brit</nemzetiseg>
    <szuletesi_datum>1985-04-12</szuletesi_datum>
    <aktualis_helyezes>5</aktualis_helyezes>
  </jatekos>

```

8. ábra: Módosítás példa 2.

```

<jatekos felszereles_id="2" jatekos_id="2" ország_id="2">
  <nev>
    <vezeteknev>Pietreczko</vezeteknev>
    <keresztnev>Ricardo</keresztnev>
    <becenev>Lightning</becenev>
  </nev>
  <nemzetiseg>Német</nemzetiseg>
  <szuletesi_datum>1990-11-23</szuletesi_datum>
  <aktualis_helyezes>12</aktualis_helyezes>
</jatekos>

```

9. ábra: Módosítás példa 3.