

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Videójáték-áruház

Készítette: **Keresztes Iulia**

Neptunkód: **ULA7Z2**

Dátum: **2023. 12. 03.**

Tartalomjegyzék

Bevezetés.....	3
1. feladat.....	3
a) Az adatbázis ER modell tervezése	3
b) Az adatbázis konvertálása XDM modellre.....	5
c) Az XDM modell alapján XML dokumentum készítése	6
d) Az XML dokumentum alapján XMLSchema készítése	29
2. feladat.....	39
a) adatolvasás.....	39
b) adatlekérdezés	42
c) adatmódosítás	48
d) adatírás	60

Bevezetés

A feladat egy Steamhez hasonló videójáték-áruház adatbázis adatainak tárolása XML formátumban, és azok feldolgozásának lehetőségének kialakítása. Olyan adatok kerülnek tárolásra, amelyek segítségével nyomon követhetőek a lényeges információk a webáruházon keresztül elérhető játékokról és a platformra feljelentkezett felhasználókról.

A játékok fontosabb adatai közé tartozik a játék ára, esetleges leírások, műfajok, támogatott nyelvek, a fejlesztő és kiadó stúdió és a játékhoz tartozó elérhető teljesítmények. A felhasználókról tároljuk a felhasználónevet és a megjelenített nevet és egyéb fiókhoz tartozó adatokat, és számon tartjuk, hogy mely felhasználó mely játékokat vette meg, esetleg tette kosárba vagy kívánságlistára, a megvett játékokkal mennyi időt játszott, és milyen teljesítményeket ért el belőlük.

Az adattárolás előkészítésének lépései: a tárolandó adatok ER modellje, ez alapján XDM modell készítése, XMLSchema file megírása, majd XML megírása, feltöltése adatokkal.

Az adatkiolvasás és -feldolgozás: Java környezetben megírt adatfeldolgozási funkciók DOM parser segítségével. A funkciók segítségével az adatok beolvasása, módosítása, kiírása és információk lekérdezése lehetséges.

1. feladat

a) Az adatbázis ER modell tervezése

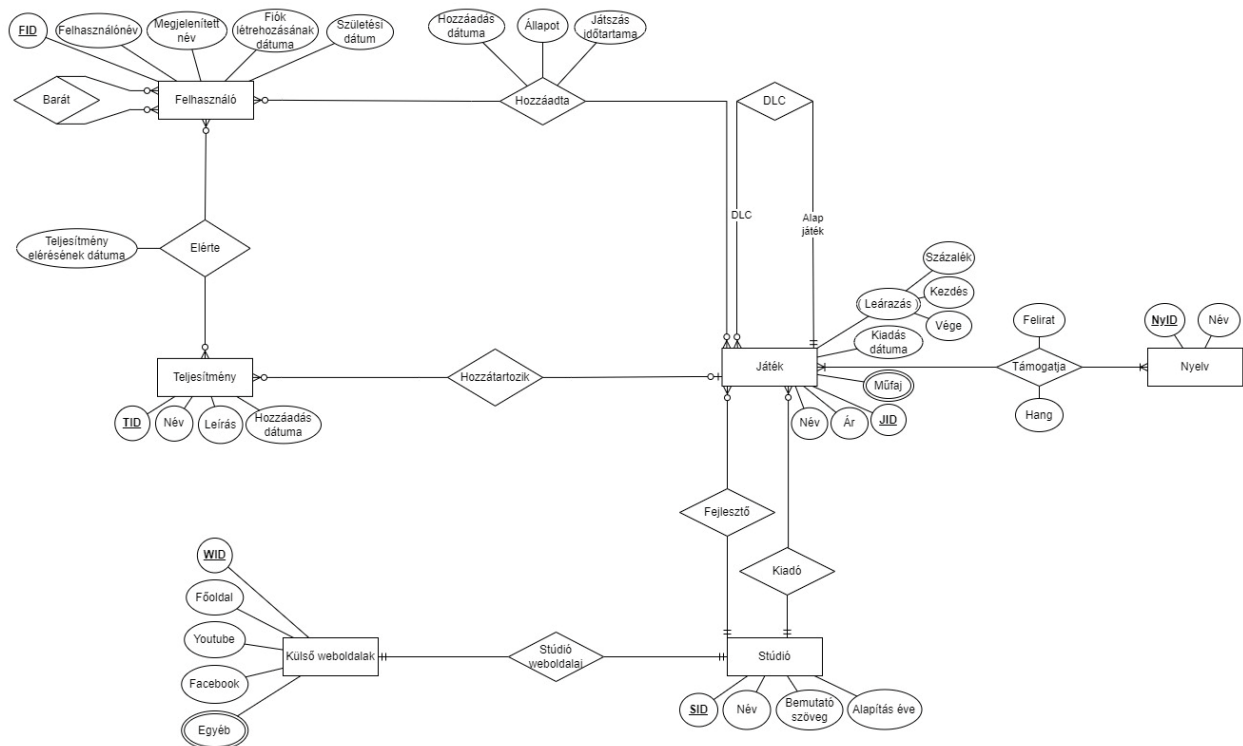
A tárolandó adatokhoz a következő egyedekre volt szükség: játék, stúdió, külső weboldalak, nyelv, teljesítmény, felhasználó.

Egyedek közötti kapcsolatok:

- DLC: a játék 1:N kapcsolata önmagával, ami azt írja le, hogy egy játék nem különálló, hanem egy másik játékhoz tartozó kiegészítő tartalom
- Támogatja: a játék és nyelv egyedek közötti N:M kapcsolat, azt írja le, hogy egy játék támogat-e egy nyelvet (akár adott nyelvű feliraton, akár adott nyelvű beszéden keresztül)
- Fejlesztő, Kiadó: a játék és stúdió közötti 1:N kapcsolatok, azt adják meg, hogy egy játékot mely stúdió fejlesztett, és melyik adta ki

- Hozzá tartozik: a játék és teljesítmény egyedek közötti 1:N kapcsolat, azt adja meg, hogy mely teljesítmények melyik játékhoz tartoznak (ha tartoznak valamelyikhez)
- Stúdió weboldalai: 1:1 kapcsolat a külső weboldalak és a stúdió egyed között, megadja az egyes stúdiókhoz tartozó külső weboldalakat, mint pl. a stúdió főoldalát és az esetlegesen hozzá tartozó szociális média felhasználóneveket és oldalakat
- Hozzáadta: a felhasználó és játék egyedek közötti N:M kapcsolat, azt írja le, hogy egy felhasználó kívánságlistára vagy kosárba tett-e egy játékot, vagy esetleg megvette; ha megvette, az is tárolásra kerül, hogy mennyit játszott már vele
- Elérte: a felhasználó és teljesítmény közötti N:M kapcsolat; megadja a felhasználó által elért teljesítményeket
- Barát: a felhasználó egyed N:M kapcsolata önmagával, megadja, hogy mely két felhasználók barátok a platformon

Az ER modell draw.io-ban készült.



b) Az adatbázis konvertálása XDM modellre

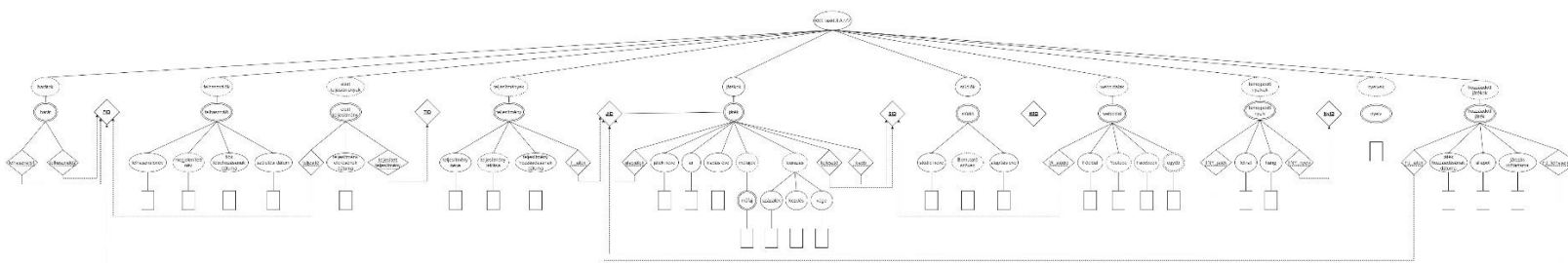
Konveráláskor az egyedekből többszörös előfordulású elemek lettek, a tulajdonságok pedig az egyes egyedekből készült elemek gyerekelei. A kulcs tulajdonságok elsődleges kulcs attribútumokká alakultak. Az 1:1 és 1:N kapcsolatok esetén egy idegenkulcs attribútum került a megfelelő elemhez (az 1:1 kapcsolat esetén a külső weboldal elemhez, 1:N kapcsolatok esetén az N oldalra), N:M kapcsolatokból pedig egy külön elem lett, amelyben két idegenkulcs attribútum mutat a két elemre, amelyek között a kapcsolat áll. A kapcsolathoz tartozó tulajdonságok is ehhez a plusz elemhez tartoznak, mint gyerekelemek.

Ahhoz, hogy a majd készülő XML-ben valamilyen strukturális tagolás legyen (a kommenteken túl), mindegyik egyedekből és tulajdonságokból létrehozott elem egy gyűjtőelem alá került (pl. a „felhasználó” elem egy „felhasználók” elem gyerekelemeként).

Ilyen módon könnyebben és átláthatóbban korlátozni tudtam azt, hogy a videójáték-áruház alapvető működéséhez milyen adatok szükségesek már a létrehozáskor, és melyek azok, amik később, idővel is hozzáadhatóak. Itt arra kell gondolni, hogy „játék” elemek nélkül semmi értelme nincs az áruháznak, a játékoknak pedig szükségképpen van kiadó és fejlesztő stúdiójuk, így a „játékok” és „stúdiók” elemek kötelezőek, míg minden más (például „felhasználók”) később kerülnének be (hiszen az áruház indításakor, logikailag, még egyetlen felhasználó sem lenne). Nyilvánvalóan az XML-ben mindegyik elemből jelenik meg példány, hogy használható legyen, de az elméleti folyamatot nézve így tartottam logikusnak magának az XDM modellnek a felépítését.

Értelemszerűen, ha már létrehoztuk valamely gyűjtőelemet, akkor legalább egy példánynak kell lennie a gyerekeleméből (azaz pl. nem lehet egy „felhasználók” elemünk „felhasználó” példányok nélkül).

Az XDM modell draw.io-ban készült.



c) Az XDM modell alapján XML dokumentum készítése

Az XML struktúrája az XDM modell alapján készült: a gyökérelem az XMLTaskULA7Z2, majd sorra következnek a gyűjtőelemek, bennük a hozzájuk tartozó példányokkal. A gyűjtőelemek sorrendje eltér az XDM modellen láthatótól. Ez azért van, mert az XDM-ben a sorrendezésben a szempont inkább az volt, hogy átlátható legyen az ábra, és a vonalak ne keresztezzék egymást, míg az XML-ben inkább az adatbevitel logikai sorrendjét próbáltam létrehozni. Így az elemek sorrendje: játékok, stúdiók, weboldalak, teljesítmények, nyelvek, támogatottNyelvek, felhasználók, barátok, hozzáadottJátékok, elértTeljesítmények.

Az XML tartalma:

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

<?xml-model href="XMLSchemaULA7Z2.xsd" type="application/xml"
schematypens="http://www.w3.org/2001/XMLSchema"?>

<XMLTaskULA7Z2 xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Játékok -->
  <játékok>
    <játék JID="AC3" fejlesztő="Ubi" kiadó="Ubi">
      <játékNeve>Assassin's Creed III</játékNeve>
      <ár>35</ár>
      <kiadásDátuma>2012-10-30</kiadásDátuma>
      <műfajok>
        <műfaj>akció</műfaj>
        <műfaj>singleplayer</műfaj>
        <műfaj>kaland</műfaj>
      </műfajok>
    </játék>

    <játék JID="TESVDB" alapjáték="TESV" fejlesztő="BethGS" kiadó="BethSW">
      <játékNeve>Dragonborn</játékNeve>
      <ár>20</ár>
      <kiadásDátuma>2012-12-04</kiadásDátuma>
      <műfajok>
        <műfaj>akció</műfaj>
        <műfaj>singleplayer</műfaj>
        <műfaj>fantasy</műfaj>
        <műfaj>kaland</műfaj>
        <műfaj>RPG</műfaj>
      </műfajok>
    </játék>
  </játékok>
</XMLTaskULA7Z2>
```

```
<játék JID="EF" fejlesztő="UbiMtl" kiadó="Ubi">
  <játékNeve>Eagle Flight</játékNeve>
  <ár>20</ár>
  <kiadásDátuma>2019-10-18</kiadásDátuma>
  <műfajok>
    <műfaj>akció</műfaj>
    <műfaj>multiplayer</műfaj>
  </műfajok>
</játék>
```

```
<játék JID="GoD" fejlesztő="BT34" kiadó="RG">
  <játékNeve>Goat of Duty</játékNeve>
  <ár>10</ár>
  <kiadásDátuma>2019-07-10</kiadásDátuma>
  <műfajok>
    <műfaj>akció</műfaj>
    <műfaj>FPS</műfaj>
    <műfaj>multiplayer</műfaj>
  </műfajok>
</játék>
```

```
<játék JID="GoW" fejlesztő="SMS" kiadó="PSS">
  <játékNeve>God of War</játékNeve>
  <ár>50</ár>
  <kiadásDátuma>2018-04-20</kiadásDátuma>
  <műfajok>
    <műfaj>akció</műfaj>
    <műfaj>singleplayer</műfaj>
    <műfaj>fantasy</műfaj>
    <műfaj>kaland</műfaj>
    <műfaj>RPG</műfaj>
  </műfajok>
</játék>
```

```
<játék JID="TESVHF" alapjáték="TESV" fejlesztő="BethGS" kiadó="BethSW">
  <játékNeve>Hearthfire</játékNeve>
  <ár>5</ár>
  <kiadásDátuma>2012-09-04</kiadásDátuma>
  <műfajok>
    <műfaj>akció</műfaj>
    <műfaj>singleplayer</műfaj>
    <műfaj>fantasy</műfaj>
    <műfaj>kaland</műfaj>
    <műfaj>RPG</műfaj>
```

```
</műfajok>
<leárazás>
  <százalék>5</százalék>
  <kezdés>2023-11-25T00:00:00</kezdés>
  <vége>2023-12-06T23:59:59</vége>
</leárazás>
</játék>

<játék JID="R6" fejlesztő="UbiMtl" kiadó="Ubi">
  <játékNeve>Rainbow Six Siege</játékNeve>
  <ár>20</ár>
  <kiadásDátuma>2015-12-01</kiadásDátuma>
  <műfajok>
    <műfaj>akció</műfaj>
    <műfaj>FPS</műfaj>
    <műfaj>multiplayer</műfaj>
  </műfajok>
  <leárazás>
    <százalék>25</százalék>
    <kezdés>2023-12-01T09:00:00</kezdés>
    <vége>2023-12-15T00:00:00</vége>
  </leárazás>
</játék>

<játék JID="TESV" fejlesztő="BethGS" kiadó="BethSW">
  <játékNeve>The Elder Scrolls V: Skyrim</játékNeve>
  <ár>20</ár>
  <kiadásDátuma>2011-11-11</kiadásDátuma>
  <műfajok>
    <műfaj>akció</műfaj>
    <műfaj>singleplayer</műfaj>
    <műfaj>fantasy</műfaj>
    <műfaj>kaland</műfaj>
    <műfaj>RPG</műfaj>
  </műfajok>
</játék>

<játék JID="AC3TKW" alapjáték="AC3" fejlesztő="Ubi" kiadó="Ubi">
  <játékNeve>The Tyranny of King Washington</játékNeve>
  <ár>10</ár>
  <kiadásDátuma>2013-03-19</kiadásDátuma>
  <műfajok>
    <műfaj>akció</műfaj>
    <műfaj>singleplayer</műfaj>
    <műfaj>kaland</műfaj>
```



```
</műfajok>
<leárazás>
  <százalék>15</százalék>
  <kezdés>2023-12-05T10:00:00</kezdés>
  <vége>2023-12-07T10:00:00</vége>
</leárazás>
</játék>

<játék JID="TVEC" fejlesztő="Astro" kiadó="Astro">
  <játékNeve>The Vanishing of Ethan Carter</játékNeve>
  <ár>20</ár>
  <kiadásDátuma>2014-09-25</kiadásDátuma>
  <műfajok>
    <műfaj>singleplayer</műfaj>
    <műfaj>horror</műfaj>
    <műfaj>kaland</műfaj>
    <műfaj>rejtély</műfaj>
  </műfajok>
  <leárazás>
    <százalék>10</százalék>
    <kezdés>2023-12-20T17:00:00</kezdés>
    <vége>2024-01-05T05:00:00</vége>
  </leárazás>
</játék>
</játékok>
```

<!-- Stúdiók -->

```
<stúdiók>
  <stúdió SID="BT34">
    <stúdióNeve>34BigThings srl</stúdióNeve>
    <alapításÉve>2013</alapításÉve>
  </stúdió>

  <stúdió SID="BethGS">
    <stúdióNeve>Bethesda Game Studios</stúdióNeve>
    <bemutatóSzöveg>Bethesda Game Studios bemutatkozó
szövege...</bemutatóSzöveg>
    <alapításÉve>2001</alapításÉve>
  </stúdió>

  <stúdió SID="BethSW">
    <stúdióNeve>Bethesda Softworks</stúdióNeve>
    <bemutatóSzöveg>Bethesda Softworks bemutató szöveg
valami</bemutatóSzöveg>
    <alapításÉve>1986</alapításÉve>
```

```
</stúdió>

<stúdió SID="PSS">
  <stúdióNeve>Playstation Studios</stúdióNeve>
  <alapításÉve>2005</alapításÉve>
</stúdió>

<stúdió SID="RG">
  <stúdióNeve>Raiser Games</stúdióNeve>
  <alapításÉve>2016</alapításÉve>
</stúdió>

<stúdió SID="SMS">
  <stúdióNeve>Santa Monica Studios</stúdióNeve>
  <bemutatóSzöveg>Santa Monica Studio bemutató szövege</bemutatóSzöveg>
  <alapításÉve>1999</alapításÉve>
</stúdió>

<stúdió SID="Astro">
  <stúdióNeve>The Astronauts</stúdióNeve>
  <alapításÉve>2012</alapításÉve>
</stúdió>

<stúdió SID="Ubi">
  <stúdióNeve>Ubisoft</stúdióNeve>
  <bemutatóSzöveg>Ubisoft bemutató szöveg</bemutatóSzöveg>
  <alapításÉve>1986</alapításÉve>
</stúdió>

<stúdió SID="UbiMtl">
  <stúdióNeve>Ubisoft Montreal</stúdióNeve>
  <bemutatóSzöveg>Ubi Soft Montreal bemutató szöveg</bemutatóSzöveg>
  <alapításÉve>1997</alapításÉve>
</stúdió>
</stúdiók>

<!-- Weboldalak -->
<weboldalak>
  <weboldal WID="BT34Web" W_stúdió="BT34">
    <főoldal>34bigthings.com</főoldal>
    <Youtube>34BigThings_games</Youtube>
    <Facebook>34bigthings</Facebook>
  </weboldal>

  <weboldal WID="BethGSWeb" W_stúdió="BethGS">
```

```
<főoldal>bethesda.game.studios.com</főoldal>
<egyéb>twitter.com/bethesda.studios</egyéb>
<egyéb>discord.com/invite/bethesda.studios</egyéb>
</weboldal>
```

```
<weboldal WID="BethSWWeb" W_stúdió="BethSW">
  <főoldal>bethesda.net</főoldal>
  <Youtube>bethesda</Youtube>
  <Facebook>bethesda.anz</Facebook>
  <egyéb>twitter.com/bethesda</egyéb>
</weboldal>
```

```
<weboldal WID="PSSWeb" W_stúdió="PSS">
  <főoldal>www.playstation.com</főoldal>
  <Youtube>PlayStation</Youtube>
  <Facebook>playstation</Facebook>
</weboldal>
```

```
<weboldal WID="RGWeb" W_stúdió="RG">
  <főoldal>www.raisergames.com</főoldal>
  <Youtube>RaiserGames</Youtube>
  <Facebook>RaiserGamesPublisher</Facebook>
  <egyéb>instagram.com/raisergames</egyéb>
</weboldal>
```

```
<weboldal WID="SMSWeb" W_stúdió="SMS">
  <főoldal>sms.playstation.com</főoldal>
  <Youtube>SantaMonicaStudio</Youtube>
  <Facebook>santamonicastudio</Facebook>
  <egyéb>twitter.com/sonysantamonica</egyéb>
  <egyéb>instagram.com/santamonicastudio</egyéb>
</weboldal>
```

```
<weboldal WID="AstroWeb" W_stúdió="Astro">
  <főoldal>www.theastronauts.com</főoldal>
  <Youtube>TheAstronauts</Youtube>
  <Facebook>TheAstroCrew</Facebook>
</weboldal>
```

```
<weboldal WID="UbiWeb" W_stúdió="Ubi">
  <főoldal>www.ubisoft.com</főoldal>
  <Youtube>Ubisoft</Youtube>
  <Facebook>ubisoft</Facebook>
  <egyéb>reddit.com/r/ubisoft</egyéb>
  <egyéb>instagram.com/ubisoft</egyéb>
```

```
<egyéb>twitter.com/ubisoft</egyéb>
</weboldal>
```

```
<weboldal WID="UbiMtlWeb" W_stúdió="UbiMtl">
  <főoldal>montreal.ubisoft.com</főoldal>
  <Youtube>UbisoftMtl</Youtube>
  <Facebook>UbisoftMontreal</Facebook>
</weboldal>
</weboldalak>
```

```
<!-- Teljesítmények -->
```

```
<teljesítmények>
  <teljesítmény TID="ACH1">
    <teljesítményNeve>Gyűjtemény</teljesítményNeve>
    <teljesítményLeírása>Szerezz meg 10 játékot.</teljesítményLeírása>
    <teljesítményHozzáadásánakDátuma>2010-05-
25</teljesítményHozzáadásánakDátuma>
  </teljesítmény>
```

```
  <teljesítmény TID="ACH2" T_játék="TESV">
    <teljesítményNeve>Legendás</teljesítményNeve>
    <teljesítményLeírása>Érj el 100-as szintet valamely
képességgel.</teljesítményLeírása>
    <teljesítményHozzáadásánakDátuma>2011-11-
11</teljesítményHozzáadásánakDátuma>
  </teljesítmény>
```

```
  <teljesítmény TID="ACH3" T_játék="TESV">
    <teljesítményNeve>Thu'um mester</teljesítményNeve>
    <teljesítményLeírása>Tanulj meg 20 kiáltást.</teljesítményLeírása>
    <teljesítményHozzáadásánakDátuma>2011-11-
11</teljesítményHozzáadásánakDátuma>
  </teljesítmény>
```

```
  <teljesítmény TID="ACH4" T_játék="TESVHF">
    <teljesítményNeve>Polgár</teljesítményNeve>
    <teljesítményLeírása>Vásárolj egy házat.</teljesítményLeírása>
    <teljesítményHozzáadásánakDátuma>2012-09-
04</teljesítményHozzáadásánakDátuma>
  </teljesítmény>
```

```
  <teljesítmény TID="ACH5" T_játék="TVEC">
    <teljesítményNeve>Befejezetlen történet</teljesítményNeve>
    <teljesítményLeírása>Érj a rejtély végére.</teljesítményLeírása>
```

```
<teljesítményHozzáadásánakDátuma>2014-09-
25</teljesítményHozzáadásánakDátuma>
</teljesítmény>

<teljesítmény TID="ACH6" T_játék="TESVDB">
  <teljesítményNeve>Miraak Temploma</teljesítményNeve>
  <teljesítményHozzáadásánakDátuma>2014-12-
01</teljesítményHozzáadásánakDátuma>
</teljesítmény>

<teljesítmény TID="ACH7" T_játék="AC3TKW">
  <teljesítményNeve>Abszolút hatalom</teljesítményNeve>
  <teljesítményLeírása>Fejezd be a fő történet szálat.</teljesítményLeírása>
  <teljesítményHozzáadásánakDátuma>2015-10-
10</teljesítményHozzáadásánakDátuma>
</teljesítmény>

<teljesítmény TID="ACH8" T_játék="EF">
  <teljesítményNeve>Dögevő</teljesítményNeve>
  <teljesítményLeírása>Pusztíts el 100 keselyűt.</teljesítményLeírása>
  <teljesítményHozzáadásánakDátuma>2017-05-
05</teljesítményHozzáadásánakDátuma>
</teljesítmény>

<teljesítmény TID="ACH9" T_játék="EF">
  <teljesítményNeve>Madárijesztő</teljesítményNeve>
  <teljesítményLeírása>Teljesítsd a Bazilika
fejezetet.</teljesítményLeírása>
  <teljesítményHozzáadásánakDátuma>2017-05-
05</teljesítményHozzáadásánakDátuma>
</teljesítmény>

<teljesítmény TID="ACH10" T_játék="GoW">
  <teljesítményNeve>Befejezetlen ügy</teljesítményNeve>
  <teljesítményLeírása>Segíts mindegyik kóborló
szellemnek.</teljesítményLeírása>
  <teljesítményHozzáadásánakDátuma>2018-04-
20</teljesítményHozzáadásánakDátuma>
</teljesítmény>

<teljesítmény TID="ACH11" T_játék="GoW">
  <teljesítményNeve>Mint otthon</teljesítményNeve>
  <teljesítményHozzáadásánakDátuma>2048-04-
20</teljesítményHozzáadásánakDátuma>
</teljesítmény>
```

```
<teljesítmény TID="ACH12" T_játék="GoD">
  <teljesítményNeve>Cuki és halálos</teljesítményNeve>
  <teljesítményLeírása>Ölj meg 10 ellenfelet anélkül, hogy
meghalj.</teljesítményLeírása>
  <teljesítményHozzáadásánakDátuma>2019-08-
10</teljesítményHozzáadásánakDátuma>
</teljesítmény>
</teljesítmények>

<!-- Nyelvek -->
<nyelvek>
  <nyelv NyID="EN">angol</nyelv>
  <nyelv NyID="DE">német</nyelv>
  <nyelv NyID="IT">olasz</nyelv>
  <nyelv NyID="RU">orosz</nyelv>
  <nyelv NyID="JP">japán</nyelv>
  <nyelv NyID="HU">magyar</nyelv>
  <nyelv NyID="TR">török</nyelv>
</nyelvek>

<!-- Játékok támogatott nyelvei -->
<támogatottNyelvek>
  <!-- Assassin's Creed 3 nyelvei-->
  <támogatottNyelv TNY_játék="AC3" TNY_nyelv="EN">
    <felirat>igen</felirat>
    <hang>igen</hang>
  </támogatottNyelv>
  <támogatottNyelv TNY_játék="AC3" TNY_nyelv="JP">
    <felirat>igen</felirat>
    <hang>igen</hang>
  </támogatottNyelv>
  <támogatottNyelv TNY_játék="AC3" TNY_nyelv="DE">
    <felirat>igen</felirat>
    <hang>igen</hang>
  </támogatottNyelv>
  <támogatottNyelv TNY_játék="AC3" TNY_nyelv="IT">
    <felirat>igen</felirat>
    <hang>igen</hang>
  </támogatottNyelv>
  <támogatottNyelv TNY_játék="AC3" TNY_nyelv="RU">
    <felirat>igen</felirat>
    <hang>igen</hang>
  </támogatottNyelv>
```

```
<!-- Dragonborn nyelvei -->
<támogatottNyelv TNY_játék="TESVDB" TNY_nyelv="EN">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVDB" TNY_nyelv="JP">
    <felirat>igen</felirat>
    <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVDB" TNY_nyelv="DE">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVDB" TNY_nyelv="IT">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVDB" TNY_nyelv="RU">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
```

```
<!-- Eagle Flight nyelvei -->
<támogatottNyelv TNY_játék="EF" TNY_nyelv="EN">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="EF" TNY_nyelv="JP">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="EF" TNY_nyelv="DE">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="EF" TNY_nyelv="IT">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="EF" TNY_nyelv="RU">
    <felirat>igen</felirat>
    <hang>igen</hang>
</támogatottNyelv>
```

```
<!-- Goat of Duty nyelvei -->
```

```
<támogatottNyelv TNY_játék="GoD" TNY_nyelv="EN">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoD" TNY_nyelv="JP">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoD" TNY_nyelv="DE">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoD" TNY_nyelv="IT">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoD" TNY_nyelv="RU">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoD" TNY_nyelv="TR">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>

<!-- God of War nyelvei -->
<támogatottNyelv TNY_játék="GoW" TNY_nyelv="EN">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoW" TNY_nyelv="JP">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoW" TNY_nyelv="HU">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoW" TNY_nyelv="DE">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoW" TNY_nyelv="IT">
  <felirat>igen</felirat>
  <hang>igen</hang>
```



```
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoW" TNY_nyelv="RU">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="GoW" TNY_nyelv="TR">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>

<!-- Hearthfire nyelvei -->
<támogatottNyelv TNY_játék="TESVHF" TNY_nyelv="EN">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVHF" TNY_nyelv="JP">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVHF" TNY_nyelv="DE">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVHF" TNY_nyelv="IT">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESVHF" TNY_nyelv="RU">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>

<!-- Rainbow Six Siege nyelvei -->
<támogatottNyelv TNY_játék="R6" TNY_nyelv="EN">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="R6" TNY_nyelv="JP">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="R6" TNY_nyelv="DE">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
```

```
<támogatottNyelv TNY_játék="R6" TNY_nyelv="IT">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="R6" TNY_nyelv="RU">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>

<!-- The Elder Scrolls V: Skytim nyelvei -->
<támogatottNyelv TNY_játék="TESV" TNY_nyelv="EN">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESV" TNY_nyelv="JP">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESV" TNY_nyelv="DE">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESV" TNY_nyelv="IT">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TESV" TNY_nyelv="RU">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>

<!-- The Tyranny of King Washington nyelvei -->
<támogatottNyelv TNY_játék="AC3TKW" TNY_nyelv="EN">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>

<!-- The Vanishing of Ethan Carter nyelvei -->
<támogatottNyelv TNY_játék="TVEC" TNY_nyelv="EN">
  <felirat>igen</felirat>
  <hang>igen</hang>
</támogatottNyelv>
<támogatottNyelv TNY_játék="TVEC" TNY_nyelv="DE">
  <felirat>igen</felirat>
  <hang>nem</hang>
```

```
</támogatottNyelv>
<támogatottNyelv TNY_játék="TVEC" TNY_nyelv="IT">
  <felirat>igen</felirat>
  <hang>nem</hang>
</támogatottNyelv>
</támogatottNyelvek>

<!-- Felhasználók -->
<felhasználók>
  <felhasználó FID="F1">
    <felhasználónév>toothed2010</felhasználónév>
    <fiókLétrehozásánakDátuma>2010-06-11</fiókLétrehozásánakDátuma>
    <születésiDátum>1982-12-30</születésiDátum>
  </felhasználó>

  <felhasználó FID="F2">
    <felhasználónév>JustAGoat</felhasználónév>
    <megjelenítettNév>Just a goat</megjelenítettNév>
    <fiókLétrehozásánakDátuma>2014-12-01</fiókLétrehozásánakDátuma>
    <születésiDátum>1995-06-06</születésiDátum>
  </felhasználó>

  <felhasználó FID="F3">
    <felhasználónév>sw6767</felhasználónév>
    <megjelenítettNév>Look, Im not your father!</megjelenítettNév>
    <fiókLétrehozásánakDátuma>2015-07-01</fiókLétrehozásánakDátuma>
    <születésiDátum>1999-11-08</születésiDátum>
  </felhasználó>

  <felhasználó FID="F4">
    <felhasználónév>eskill_esq</felhasználónév>
    <megjelenítettNév>SELECT nothing FROM nowhere</megjelenítettNév>
    <fiókLétrehozásánakDátuma>2016-03-05</fiókLétrehozásánakDátuma>
    <születésiDátum>2001-01-01</születésiDátum>
  </felhasználó>

  <felhasználó FID="F5">
    <felhasználónév>snowwingedhunter123</felhasználónév>
    <megjelenítettNév>Snow-winged hunter</megjelenítettNév>
    <fiókLétrehozásánakDátuma>2017-01-15</fiókLétrehozásánakDátuma>
    <születésiDátum>2002-10-04</születésiDátum>
  </felhasználó>

  <felhasználó FID="F6">
    <felhasználónév>smeow3437</felhasználónév>
```

```
<fiókLétrehozásánakDátuma>2017-09-03</fiókLétrehozásánakDátuma>  
<születésiDátum>2002-07-08</születésiDátum>  
</felhasználó>
```

```
<felhasználó FID="F7">  
  <felhasználónév>Dand3lion</felhasználónév>  
  <megjelenítettNév>Jaskier the bard</megjelenítettNév>  
  <fiókLétrehozásánakDátuma>2019-10-20</fiókLétrehozásánakDátuma>  
  <születésiDátum>2000-09-23</születésiDátum>  
</felhasználó>
```

```
<felhasználó FID="F8">  
  <felhasználónév>LBlackWood55</felhasználónév>  
  <fiókLétrehozásánakDátuma>2020-01-01</fiókLétrehozásánakDátuma>  
  <születésiDátum>2006-12-01</születésiDátum>  
</felhasználó>
```

```
<felhasználó FID="F9">  
  <felhasználónév>bbBbeo</felhasználónév>  
  <megjelenítettNév>Beo</megjelenítettNév>  
  <fiókLétrehozásánakDátuma>2020-01-25</fiókLétrehozásánakDátuma>  
  <születésiDátum>2004-11-19</születésiDátum>  
</felhasználó>
```

```
<felhasználó FID="F10">  
  <felhasználónév>eleMentalSnake</felhasználónév>  
  <megjelenítettNév>Mad Hatter</megjelenítettNév>  
  <fiókLétrehozásánakDátuma>2021-02-02</fiókLétrehozásánakDátuma>  
  <születésiDátum>2003-05-25</születésiDátum>  
</felhasználó>
```

```
<felhasználó FID="F11">  
  <felhasználónév>hooty3</felhasználónév>  
  <megjelenítettNév>Hootsifer</megjelenítettNév>  
  <fiókLétrehozásánakDátuma>2022-08-31</fiókLétrehozásánakDátuma>  
  <születésiDátum>2000-05-18</születésiDátum>  
</felhasználó>
```

```
<felhasználó FID="F12">  
  <felhasználónév>wWeki9ii</felhasználónév>  
  <megjelenítettNév>a-Woo</megjelenítettNév>  
  <fiókLétrehozásánakDátuma>2022-09-30</fiókLétrehozásánakDátuma>  
  <születésiDátum>2007-09-15</születésiDátum>  
</felhasználó>
```

```
<felhasználó FID="F13">
  <felhasználónév>corgiii89</felhasználónév>
  <fiókLétrehozásánakDátuma>2022-11-05</fiókLétrehozásánakDátuma>
  <születésiDátum>1989-03-10</születésiDátum>
</felhasználó>

<felhasználó FID="F14">
  <felhasználónév>julienmoveit</felhasználónév>
  <fiókLétrehozásánakDátuma>2023-12-01</fiókLétrehozásánakDátuma>
  <születésiDátum>2007-02-16</születésiDátum>
</felhasználó>

<felhasználó FID="F15">
  <felhasználónév>bzrk781</felhasználónév>
  <megjelenítettNév>berZerk</megjelenítettNév>
  <fiókLétrehozásánakDátuma>2023-12-01</fiókLétrehozásánakDátuma>
  <születésiDátum>2008-04-27</születésiDátum>
</felhasználó>
</felhasználók>
```

```
<!-- Barátok -->
```

```
<barátok>
  <barát felhasználó1="F2" felhasználó2="F7" />
  <barát felhasználó1="F2" felhasználó2="F13" />
  <barát felhasználó1="F8" felhasználó2="F11" />
  <barát felhasználó1="F8" felhasználó2="F1" />
  <barát felhasználó1="F9" felhasználó2="F7" />
  <barát felhasználó1="F9" felhasználó2="F10" />
  <barát felhasználó1="F9" felhasználó2="F6" />
  <barát felhasználó1="F13" felhasználó2="F9" />
  <barát felhasználó1="F10" felhasználó2="F13" />
  <barát felhasználó1="F4" felhasználó2="F7" />
  <barát felhasználó1="F4" felhasználó2="F10" />
  <barát felhasználó1="F4" felhasználó2="F3" />
  <barát felhasználó1="F4" felhasználó2="F1" />
  <barát felhasználó1="F11" felhasználó2="F7" />
  <barát felhasználó1="F11" felhasználó2="F2" />
  <barát felhasználó1="F11" felhasználó2="F9" />
  <barát felhasználó1="F11" felhasználó2="F13" />
  <barát felhasználó1="F11" felhasználó2="F10" />
  <barát felhasználó1="F11" felhasználó2="F4" />
  <barát felhasználó1="F11" felhasználó2="F5" />
  <barát felhasználó1="F11" felhasználó2="F3" />
  <barát felhasználó1="F11" felhasználó2="F1" />
  <barát felhasználó1="F6" felhasználó2="F7" />
```

```
<barát felhasználó1="F6" felhasználó2="F8" />
<barát felhasználó1="F6" felhasználó2="F13" />
<barát felhasználó1="F6" felhasználó2="F4" />
<barát felhasználó1="F6" felhasználó2="F11" />
<barát felhasználó1="F6" felhasználó2="F3" />
<barát felhasználó1="F6" felhasználó2="F1" />
<barát felhasználó1="F6" felhasználó2="F12" />
<barát felhasználó1="F5" felhasználó2="F6" />
<barát felhasználó1="F12" felhasználó2="F7" />
<barát felhasználó1="F12" felhasználó2="F8" />
<barát felhasználó1="F12" felhasználó2="F9" />
<barát felhasználó1="F12" felhasználó2="F4" />
<barát felhasználó1="F12" felhasználó2="F11" />
<barát felhasználó1="F12" felhasználó2="F7" />
<barát felhasználó1="F12" felhasználó2="F1" />
</barátok>
```

<!-- Felhasználók által kívánságlistába vagy kosárba hozzáadott, vagy megvett játékok-->

<hozzáadottJátékok>

<!-- Dand3lion felhasználó játécai -->

<hozzáadottJáték HJ_játék="AC3" HJ_felhasználó="F7">

<játékHozzáadásánakDátuma>2021-10-29</játékHozzáadásánakDátuma>

<állapot>kívánságlistán</állapot>

</hozzáadottJáték>

<hozzáadottJáték HJ_játék="R6" HJ_felhasználó="F7">

<játékHozzáadásánakDátuma>2023-12-01</játékHozzáadásánakDátuma>

<állapot>megvéve</állapot>

<játszásIdőtartama>0</játszásIdőtartama>

</hozzáadottJáték>

<hozzáadottJáték HJ_játék="AC3TKW" HJ_felhasználó="F7">

<játékHozzáadásánakDátuma>2021-11-01</játékHozzáadásánakDátuma>

<állapot>kívánságlistán</állapot>

</hozzáadottJáték>

<!-- JustAGoat felhasználó játécai -->

<hozzáadottJáték HJ_játék="AC3" HJ_felhasználó="F2">

<játékHozzáadásánakDátuma>2014-12-25</játékHozzáadásánakDátuma>

<állapot>megvéve</állapot>

<játszásIdőtartama>23</játszásIdőtartama>

</hozzáadottJáték>

<hozzáadottJáték HJ_játék="TESVDB" HJ_felhasználó="F2">

<játékHozzáadásánakDátuma>2017-09-16</játékHozzáadásánakDátuma>

<állapot>megvéve</állapot>

<játszásIdőtartama>5</játszásIdőtartama>

```
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="GoD" HJ_felhasználó="F2">
  <játékHozzáadásánakDátuma>2018-04-12</játékHozzáadásánakDátuma>
  <állapot>kosárban</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="GoW" HJ_felhasználó="F2">
  <játékHozzáadásánakDátuma>2018-04-01</játékHozzáadásánakDátuma>
  <állapot>kosárban</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESVHF" HJ_felhasználó="F2">
  <játékHozzáadásánakDátuma>2017-09-16</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>11</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESV" HJ_felhasználó="F2">
  <játékHozzáadásánakDátuma>2017-08-16</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>57</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="AC3TKW" HJ_felhasználó="F2">
  <játékHozzáadásánakDátuma>2014-12-25</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>8</játszásIdőtartama>
</hozzáadottJáték>

<!-- LBlackWood55 felhasználó játéakai -->
<hozzáadottJáték HJ_játék="AC3" HJ_felhasználó="F8">
  <játékHozzáadásánakDátuma>2020-01-01</játékHozzáadásánakDátuma>
  <állapot>kívánságlistán</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="EF" HJ_felhasználó="F8">
  <játékHozzáadásánakDátuma>2020-01-01</játékHozzáadásánakDátuma>
  <állapot>kívánságlistán</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESV" HJ_felhasználó="F8">
  <játékHozzáadásánakDátuma>2021-05-24</játékHozzáadásánakDátuma>
  <állapot>kívánságlistán</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="AC3TKW" HJ_felhasználó="F8">
  <játékHozzáadásánakDátuma>2020-01-01</játékHozzáadásánakDátuma>
  <állapot>kívánságlistán</állapot>
</hozzáadottJáték>

<!-- bbBbeo felhasználó játéakai -->
<hozzáadottJáték HJ_játék="TESVDB" HJ_felhasználó="F9">
```

```
<játékHozzáadásánakDátuma>2021-06-02</játékHozzáadásánakDátuma>
<állapot>kívánságlistán</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESVHF" HJ_felhasználó="F9">
  <játékHozzáadásánakDátuma>2021-06-02</játékHozzáadásánakDátuma>
  <állapot>kívánságlistán</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESV" HJ_felhasználó="F9">
  <játékHozzáadásánakDátuma>2021-05-13</játékHozzáadásánakDátuma>
  <állapot>kosárban</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TVEC" HJ_felhasználó="F9">
  <játékHozzáadásánakDátuma>2020-01-27</játékHozzáadásánakDátuma>
  <állapot>kívánságlistán</állapot>
</hozzáadottJáték>

<!-- corgiii89 felhasználó játéakai -->
<hozzáadottJáték HJ_játék="EF" HJ_felhasználó="F13">
  <játékHozzáadásánakDátuma>2022-11-05</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>6</játszásIdőtartama>
</hozzáadottJáték>

<!-- eleMentalSnake felhasználó játéakai -->
<hozzáadottJáték HJ_játék="TESVDB" HJ_felhasználó="F10">
  <játékHozzáadásánakDátuma>2023-12-01</játékHozzáadásánakDátuma>
  <állapot>kosárban</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="EF" HJ_felhasználó="F10">
  <játékHozzáadásánakDátuma>2021-02-28</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>109</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESVHF" HJ_felhasználó="F10">
  <játékHozzáadásánakDátuma>2023-12-01</játékHozzáadásánakDátuma>
  <állapot>kosárban</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="R6" HJ_felhasználó="F10">
  <játékHozzáadásánakDátuma>2021-02-02</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>17</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESV" HJ_felhasználó="F10">
  <játékHozzáadásánakDátuma>2022-10-30</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
```



```
<játszásIdőtartama>31</játszásIdőtartama>
</hozzáadottJáték>

<!-- eskell_esq felhasználó játéakai -->
<hozzáadottJáték HJ_játék="GoD" HJ_felhasználó="F4">
    <játékHozzáadásánakDátuma>2019-10-09</játékHozzáadásánakDátuma>
    <állapot>megvéve</állapot>
    <játszásIdőtartama>5</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="GoW" HJ_felhasználó="F4">
    <játékHozzáadásánakDátuma>2022-02-04</játékHozzáadásánakDátuma>
    <állapot>kosárban</állapot>
</hozzáadottJáték>

<!-- hooty3 felhasználó játéakai -->
<hozzáadottJáték HJ_játék="AC3" HJ_felhasználó="F11">
    <játékHozzáadásánakDátuma>2022-10-01</játékHozzáadásánakDátuma>
    <állapot>kívánságlistán</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESVHF" HJ_felhasználó="F11">
    <játékHozzáadásánakDátuma>2022-10-01</játékHozzáadásánakDátuma>
    <állapot>kívánságlistán</állapot>
</hozzáadottJáték>

<!-- smeow3437 felhasználó játéakai -->
<hozzáadottJáték HJ_játék="TESVDB" HJ_felhasználó="F6">
    <játékHozzáadásánakDátuma>2021-07-18</játékHozzáadásánakDátuma>
    <állapot>megvéve</állapot>
    <játszásIdőtartama>10</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="EF" HJ_felhasználó="F6">
    <játékHozzáadásánakDátuma>2021-12-14</játékHozzáadásánakDátuma>
    <állapot>megvéve</állapot>
    <játszásIdőtartama>51</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESVHF" HJ_felhasználó="F6">
    <játékHozzáadásánakDátuma>2019-02-12</játékHozzáadásánakDátuma>
    <állapot>megvéve</állapot>
    <játszásIdőtartama>37</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESV" HJ_felhasználó="F6">
    <játékHozzáadásánakDátuma>2018-11-29</játékHozzáadásánakDátuma>
    <állapot>megvéve</állapot>
    <játszásIdőtartama>256</játszásIdőtartama>
</hozzáadottJáték>
```

```
<!-- snowwingedhunter123 felhasználó játéakai -->
<hozzáadottJáték HJ_játék="AC3" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2018-05-28</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>103</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESVDB" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2021-11-11</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>9</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="EF" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2018-10-26</játékHozzáadásánakDátuma>
  <állapot>kívánságlistán</állapot>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="GoD" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2017-09-09</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>13</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="GoW" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2022-10-29</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>36</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESVHF" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2021-11-11</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>31</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="TESV" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2021-11-11</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>174</játszásIdőtartama>
</hozzáadottJáték>
<hozzáadottJáték HJ_játék="AC3TKW" HJ_felhasználó="F5">
  <játékHozzáadásánakDátuma>2018-05-28</játékHozzáadásánakDátuma>
  <állapot>megvéve</állapot>
  <játszásIdőtartama>25</játszásIdőtartama>
</hozzáadottJáték>

<!-- sw6767 felhasználó játéakai -->
<hozzáadottJáték HJ_játék="GoW" HJ_felhasználó="F3">
  <játékHozzáadásánakDátuma>2022-10-01</játékHozzáadásánakDátuma>
```

```

        <állapot>kívánságlistán</állapot>
    </hozzáadottJáték>
    <hozzáadottJáték HJ_játék="R6" HJ_felhasználó="F3">
        <játékHozzáadásánakDátuma>2022-10-22</játékHozzáadásánakDátuma>
        <állapot>kívánságlistán</állapot>
    </hozzáadottJáték>

    <!-- toothed2010 felhasználó játéakai -->
    <hozzáadottJáték HJ_játék="Gow" HJ_felhasználó="F1">
        <játékHozzáadásánakDátuma>2018-05-20</játékHozzáadásánakDátuma>
        <állapot>megvéve</állapot>
        <játszásIdőtartama>77</játszásIdőtartama>
    </hozzáadottJáték>
    <hozzáadottJáték HJ_játék="TESVHF" HJ_felhasználó="F1">
        <játékHozzáadásánakDátuma>2018-05-23</játékHozzáadásánakDátuma>
        <állapot>kívánságlistán</állapot>
    </hozzáadottJáték>
    <hozzáadottJáték HJ_játék="R6" HJ_felhasználó="F1">
        <játékHozzáadásánakDátuma>2010-06-11</játékHozzáadásánakDátuma>
        <állapot>kívánságlistán</állapot>
    </hozzáadottJáték>
    <hozzáadottJáték HJ_játék="TESV" HJ_felhasználó="F1">
        <játékHozzáadásánakDátuma>2010-06-19</játékHozzáadásánakDátuma>
        <állapot>kívánságlistán</állapot>
    </hozzáadottJáték>

    <!-- wWeki9ii felhasználó játéakai -->
    <hozzáadottJáték HJ_játék="R6" HJ_felhasználó="F12">
        <játékHozzáadásánakDátuma>2023-12-01</játékHozzáadásánakDátuma>
        <állapot>kívánságlistán</állapot>
    </hozzáadottJáték>
    <hozzáadottJáték HJ_játék="TVEC" HJ_felhasználó="F12">
        <játékHozzáadásánakDátuma>2022-09-30</játékHozzáadásánakDátuma>
        <állapot>megvéve</állapot>
        <játszásIdőtartama>17</játszásIdőtartama>
    </hozzáadottJáték>
</hozzáadottJátékok>

<!-- Felhasználók által elért teljesítmények -->
<elértTeljesítmény>
    <!-- JustAGoat felhasználó által elért teljesítmények -->
    <elértTeljesítmény teljesítő="F2" teljesítettTeljesítmény="ACH7">
        <teljesítményElérésénekDátuma>2015-02-26</teljesítményElérésénekDátuma>
    </elértTeljesítmény>
    <elértTeljesítmény teljesítő="F2" teljesítettTeljesítmény="ACH6">

```

```
<teljesítményElérésénekDátuma>2017-09-24</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F2" teljesítettTeljesítmény="ACH3">
  <teljesítményElérésénekDátuma>2017-12-11</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F2" teljesítettTeljesítmény="ACH4">
  <teljesítményElérésénekDátuma>2018-01-09</teljesítményElérésénekDátuma>
</elértTeljesítmény>

<!-- eleMentalSnake felhasználó által elért teljesítmények -->
<elértTeljesítmény teljesítő="F10" teljesítettTeljesítmény="ACH9">
  <teljesítményElérésénekDátuma>2021-03-03</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F10" teljesítettTeljesítmény="ACH8">
  <teljesítményElérésénekDátuma>2021-05-12</teljesítményElérésénekDátuma>
</elértTeljesítmény>

<!-- smeow3437 felhasználó által elért teljesítmények -->
<elértTeljesítmény teljesítő="F6" teljesítettTeljesítmény="ACH2">
  <teljesítményElérésénekDátuma>2018-01-15</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F6" teljesítettTeljesítmény="ACH3">
  <teljesítményElérésénekDátuma>2018-11-30</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F6" teljesítettTeljesítmény="ACH6">
  <teljesítményElérésénekDátuma>2019-05-22</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F6" teljesítettTeljesítmény="ACH9">
  <teljesítményElérésénekDátuma>2022-01-06</teljesítményElérésénekDátuma>
</elértTeljesítmény>

<!-- snowwingedhunter123 felhasználó által elért teljesítések -->
<elértTeljesítmény teljesítő="F5" teljesítettTeljesítmény="ACH5">
  <teljesítményElérésénekDátuma>2018-06-10</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F5" teljesítettTeljesítmény="ACH3">
  <teljesítményElérésénekDátuma>2021-11-15</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F5" teljesítettTeljesítmény="ACH2">
  <teljesítményElérésénekDátuma>2021-12-02</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F5" teljesítettTeljesítmény="ACH4">
  <teljesítményElérésénekDátuma>2021-12-11</teljesítményElérésénekDátuma>
</elértTeljesítmény>
<elértTeljesítmény teljesítő="F5" teljesítettTeljesítmény="ACH6">
```

```

        <teljesítményElérésénekDátuma>2022-04-13</teljesítményElérésénekDátuma>
    </elértTeljesítmény>
    <elértTeljesítmény teljesítő="F5" teljesítettTeljesítmény="ACH11">
        <teljesítményElérésénekDátuma>2022-11-07</teljesítményElérésénekDátuma>
    </elértTeljesítmény>

    <!-- toothed2010 felhasználó által elért teljesítmények -->
    <elértTeljesítmény teljesítő="F1" teljesítettTeljesítmény="ACH11">
        <teljesítményElérésénekDátuma>2018-05-24</teljesítményElérésénekDátuma>
    </elértTeljesítmény>
    <elértTeljesítmény teljesítő="F1" teljesítettTeljesítmény="ACH10">
        <teljesítményElérésénekDátuma>2018-05-31</teljesítményElérésénekDátuma>
    </elértTeljesítmény>

    <!-- wWeki9ii felhasználó által elért teljesítmények -->
    <elértTeljesítmény teljesítő="F12" teljesítettTeljesítmény="ACH5">
        <teljesítményElérésénekDátuma>2022-10-03</teljesítményElérésénekDátuma>
    </elértTeljesítmény>
</elértTeljesítmények>

</XMLTaskULA7Z2>

```

d) Az XML dokumentum alapján XMLSchema készítése

Az XML és XDM szerkezete alapján létrehoztam az XSD file-t. A gyerekelemekkel rendelkező elemek számára saját komplex típusokat hoztam létre. A „nyelv” elem megvalósításához egy saját egyszerű típust hoztam létre, mivel annak csak egy attribútuma van és szöveges tartalma.

Az elsődleges és idegenkulcsok megvalósításához ID és IDREF típusú attribútumokat használtam, majd xs:key és xs:keyref segítségével létrehoztam a tényleges megszorításokat is hozzájuk. A kötelező kapcsolatok megvalósításához a use=”required” megkötést használtam az attribute-ok esetében, az elemeknél pedig a kötelező, opcionális és többszörös előfordulású elemeknél a minOccurs és maxOccurs értékeket állítottam be. Az 1:1 kapcsolat megvalósításához unique és keyref megkötést tettem a megfelelő IDREF attribute-ra.

Az XMLSchema kódja:

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

  <!-- Egyszerű típusú elemek -->
  <xs:element name="felhasználónév" type="xs:string" />
  <xs:element name="megjelenítettNév" type="xs:string" />
  <xs:element name="fiókLétrehozásánakDátuma" type="xs:date" />
  <xs:element name="születésiDátum" type="xs:date" />

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

  <xs:element name="teljesítményNeve" type="xs:string" />
  <xs:element name="teljesítményLeírása" type="xs:string" />
  <xs:element name="teljesítményHozzáadásánakDátuma" type="xs:date" />

  <xs:element name="játékNeve" type="xs:string" />
  <xs:element name="ár" type="xs:float" />
  <xs:element name="kiadásDátuma" type="xs:date" />
  <xs:element name="műfaj" type="műfajTípus" />
  <xs:element name="százalék" type="xs:int" />
  <xs:element name="kezdés" type="xs:dateTime" />
  <xs:element name="vége" type="xs:dateTime" />

  <xs:element name="stúdióNeve" type="xs:string" />
  <xs:element name="bemutatóSzöveg" type="xs:string" />
  <xs:element name="alapításÉve" type="xs:gYear" />

  <xs:element name="főoldal" type="xs:string" />
  <xs:element name="Youtube" type="xs:string" />
  <xs:element name="Facebook" type="xs:string" />
  <xs:element name="egyéb" type="xs:string" />

  <xs:element name="felirat" type="igazságértékTípus" />
  <xs:element name="hang" type="igazságértékTípus" />

  <xs:element name="játékHozzáadásánakDátuma" type="xs:date" />
  <xs:element name="állapot" type="állapotTípus" />
  <xs:element name="játékszásIdőtartama" type="xs:int" />

  <!-- Saját egyszerű típus deklarációk -->
```

```

<xs:simpleType name="műfajTípus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="fantasy" />
    <xs:enumeration value="horror" />
    <xs:enumeration value="akció" />
    <xs:enumeration value="kaland" />
    <xs:enumeration value="rejtély" />
    <xs:enumeration value="romantikus" />
    <xs:enumeration value="RPG" />
    <xs:enumeration value="shooter" />
    <xs:enumeration value="multiplayer" />
    <xs:enumeration value="singleplayer" />
    <xs:enumeration value="FPS" />
    <xs:enumeration value="belső-nézetes" />
    <xs:enumeration value="külső-nézetes" />
    <xs:enumeration value="indie" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="igazságértékTípus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="igen" />
    <xs:enumeration value="nem" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="állapotTípus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="kívánságlistán" />
    <xs:enumeration value="kosárban" />
    <xs:enumeration value="megvéve" />
  </xs:restriction>
</xs:simpleType>

<!-- Elsődleges kulcsok -->
<xs:attribute name="FID" type="xs:ID" />
<xs:attribute name="TID" type="xs:ID" />
<xs:attribute name="JID" type="xs:ID" />
<xs:attribute name="SID" type="xs:ID" />
<xs:attribute name="WID" type="xs:ID" />
<xs:attribute name="NyID" type="xs:ID" />

<!-- Idegenkulcsok -->

```

```

<xs:attribute name="felhasználó1" type="xs:IDREF" />
<xs:attribute name="felhasználó2" type="xs:IDREF" />
<xs:attribute name="teljesítő" type="xs:IDREF" />
<xs:attribute name="teljesítettTeljesítmény" type="xs:IDREF" />
<xs:attribute name="T_játék" type="xs:IDREF" />
<xs:attribute name="alapjáték" type="xs:IDREF" />
<xs:attribute name="fejlesztő" type="xs:IDREF" />
<xs:attribute name="kiadó" type="xs:IDREF" />
<xs:attribute name="W_stúdió" type="xs:IDREF" />
<xs:attribute name="TNY_játék" type="xs:IDREF" />
<xs:attribute name="TNY_nyelv" type="xs:IDREF" />
<xs:attribute name="HJ_játék" type="xs:IDREF" />
<xs:attribute name="HJ_felhasználó" type="xs:IDREF" />

<!-- Komplex típusú elemek -->
<xs:element name="barátok" type="barátokTípus" />
<xs:element name="barát" type="barátTípus" />

<xs:element name="felhasználók" type="felhasználókTípus">
  <xs:key name="felhasználóID">
    <xs:selector xpath="felhasználó" />
    <xs:field xpath="@FID" />
  </xs:key>
</xs:element>
<xs:element name="felhasználó" type="felhasználóTípus" />

<xs:element name="elértTeljesítmények" type="elértTeljesítményekTípus" />
<xs:element name="elértTeljesítmény" type="elértTeljesítményTípus" />

<xs:element name="teljesítmények" type="teljesítményekTípus">
  <xs:key name="teljesítményID">
    <xs:selector xpath="teljesítmény" />
    <xs:field xpath="@TID" />
  </xs:key>
</xs:element>
<xs:element name="teljesítmény" type="teljesítményTípus" />

<xs:element name="játékok" type="játékokTípus">
  <xs:key name="játékID">
    <xs:selector xpath="játék" />
    <xs:field xpath="@JID" />
  </xs:key>
</xs:element>
<xs:element name="játék" type="játékTípus" />

```



```

<xs:element name="műfajok" type="műfajokTípus" />
<xs:element name="leárazás" type="leárazásTípus" />

<xs:element name="stúdiók" type="stúdiókTípus">
  <xs:key name="stúdióID">
    <xs:selector xpath="stúdió" />
    <xs:field xpath="@SID" />
  </xs:key>
</xs:element>
<xs:element name="stúdió" type="stúdióTípus" />

<xs:element name="weboldalak" type="weboldalakTípus">
  <xs:key name="weboldalID">
    <xs:selector xpath="weboldal" />
    <xs:field xpath="@WID" />
  </xs:key>
  <xs:unique name="W_stúdióUNIQUE">
    <xs:selector xpath="weboldal" />
    <xs:field xpath="@W_stúdió" />
  </xs:unique>
</xs:element>
<xs:element name="weboldal" type="weboldalTípus" />

<xs:element name="támogatottNyelvek" type="támogatottNyelvekTípus" />
<xs:element name="támogatottNyelv" type="támogatottNyelvTípus" />

<xs:element name="nyelvek" type="nyelvekTípus">
  <xs:key name="nyelvID">
    <xs:selector xpath="nyelv"/>
    <xs:field xpath="@NyID" />
  </xs:key>
</xs:element>
<xs:element name="nyelv" type="nyelvTípus" />

<xs:element name="hozzáadottJátékok" type="hozzáadottJátékokTípus" />
<xs:element name="hozzáadottJáték" type="hozzáadottJátékTípus" />

<!-- Komplex típus deklarációk -->
<xs:complexType name="barátokTípus">
  <xs:sequence>
    <xs:element ref="barát" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="barátTípus">

```

```
<xs:attribute ref="felhasználó1" use="required" />
<xs:attribute ref="felhasználó2" use="required" />
</xs:complexType>

<xs:complexType name="felhasználókTípus">
  <xs:sequence>
    <xs:element ref="felhasználó" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="felhasználóTípus">
  <xs:sequence>
    <xs:element ref="felhasználónév" />
    <xs:element ref="megjelenítettNév" minOccurs="0" />
    <xs:element ref="fiókLétrehozásánakDátuma" />
    <xs:element ref="születésiDátum" />
  </xs:sequence>
  <xs:attribute ref="FID" use="required"/>
</xs:complexType>

<xs:complexType name="elértTeljesítményekTípus">
  <xs:sequence>
    <xs:element ref="elértTeljesítmény" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="elértTeljesítményTípus">
  <xs:sequence>
    <xs:element ref="teljesítményElérésénekDátuma" />
  </xs:sequence>
  <xs:attribute ref="teljesítő" use="required"/>
  <xs:attribute ref="teljesítettTeljesítmény" use="required" />
</xs:complexType>

<xs:complexType name="teljesítményekTípus">
  <xs:sequence>
    <xs:element ref="teljesítmény" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="teljesítményTípus">
  <xs:sequence>
    <xs:element ref="teljesítményNeve" />
    <xs:element ref="teljesítményLeírása" minOccurs="0" />
    <xs:element ref="teljesítményHozzáadásánakDátuma" />
  </xs:sequence>
  <xs:attribute ref="TID" use="required" />
  <xs:attribute ref="T_játék" use="optional" />
</xs:complexType>
```

```
</xs:complexType>

<xs:complexType name="játékokTípus">
  <xs:sequence>
    <xs:element ref="játék" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="játékTípus">
  <xs:sequence>
    <xs:element ref="játékNeve" />
    <xs:element ref="ár" />
    <xs:element ref="kiadásDátuma" />
    <xs:element ref="műfajok" minOccurs="0" />
    <xs:element ref="leírás" minOccurs="0" />
  </xs:sequence>
  <xs:attribute ref="JID" use="required" />
  <xs:attribute ref="alapjáték" use="optional" />
  <xs:attribute ref="fejlesztő" use="required" />
  <xs:attribute ref="kiadó" use="required" />
</xs:complexType>

<xs:complexType name="műfajokTípus">
  <xs:sequence>
    <xs:element ref="műfaj" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="leírásTípus">
  <xs:sequence>
    <xs:element ref="százalék" />
    <xs:element ref="kezdés" />
    <xs:element ref="vége" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="stúdiókTípus">
  <xs:sequence>
    <xs:element ref="stúdió" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="stúdióTípus">
  <xs:sequence>
    <xs:element ref="stúdióNeve" />
    <xs:element ref="bemutatóSzöveg" minOccurs="0" />
    <xs:element ref="alapításÉve" />
  </xs:sequence>
  <xs:attribute ref="SID" use="required" />
</xs:complexType>
```

```
</xs:complexType>

<xs:complexType name="weboldalTípus">
  <xs:sequence>
    <xs:element ref="weboldal" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="weboldalTípus">
  <xs:sequence>
    <xs:element ref="főoldal" />
    <xs:element ref="Youtube" minOccurs="0" />
    <xs:element ref="Facebook" minOccurs="0" />
    <xs:element ref="egyéb" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute ref="WID" use="required" />
  <xs:attribute ref="W_stúdió" use="required" />
</xs:complexType>

<xs:complexType name="támogatottNyelvekTípus">
  <xs:sequence>
    <xs:element ref="támogatottNyelv" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="támogatottNyelvTípus">
  <xs:sequence>
    <xs:element ref="felirat" />
    <xs:element ref="hang" />
  </xs:sequence>
  <xs:attribute ref="TNY_játék" use="required" />
  <xs:attribute ref="TNY_nyelv" use="required" />
</xs:complexType>

<xs:complexType name="nyelvekTípus">
  <xs:sequence>
    <xs:element ref="nyelv" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="nyelvTípus">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="NyID" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

```

<xs:complexType name="hozzáadottJátékokTípus">
  <xs:sequence>
    <xs:element ref="hozzáadottJáték" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="hozzáadottJátékTípus">
  <xs:sequence>
    <xs:element ref="játékHozzáadásánakDátuma" />
    <xs:element ref="állapot" />
    <xs:element ref="játékszámIdőtartama" minOccurs="0" />
  </xs:sequence>
  <xs:attribute ref="HJ_játék" use="required" />
  <xs:attribute ref="HJ_felhasználó" use="required" />
</xs:complexType>

<!-- Gyökérelem -->
<xs:complexType name="XMLTASKULA7Z2Típus">
  <xs:sequence>
    <xs:element ref="játékok" />
    <xs:element ref="stúdiók" />
    <xs:element ref="weboldalak" minOccurs="0" />
    <xs:element ref="teljesítmények" minOccurs="0" />
    <xs:element ref="nyelvek" minOccurs="0" />
    <xs:element ref="támogatottNyelvek" minOccurs="0" />
    <xs:element ref="felhasználók" minOccurs="0" />
    <xs:element ref="barátok" minOccurs="0" />
    <xs:element ref="hozzáadottJátékok" minOccurs="0" />
    <xs:element ref="elértTeljesítmények" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:element name="XMLTaskULA7Z2" type="XMLTASKULA7Z2Típus">
  <!-- idegenkulcs megkötések -->
  <xs:keyref name="felhasználó1REF" refer="felhasználóID">
    <xs:selector xpath="barátok/barát" />
    <xs:field xpath="@felhasználó1" />
  </xs:keyref>
  <xs:keyref name="felhasználó2REF" refer="felhasználóID">
    <xs:selector xpath="barátok/barát" />
    <xs:field xpath="@felhasználó2" />
  </xs:keyref>

  <xs:keyref name="teljesítőREF" refer="felhasználóID">
    <xs:selector xpath="elértTeljesítmények/elértTeljesítmény" />
    <xs:field xpath="@teljesítő" />
  </xs:keyref>

```

```
</xs:keyref>
<xs:keyref name="teljesítettTeljesítményREF" refer="teljesítményID">
  <xs:selector xpath="elértTeljesítmények/elértTeljesítmény" />
  <xs:field xpath="@teljesítettTeljesítmény" />
</xs:keyref>

<xs:keyref name="T_játékREF" refer="játékID">
  <xs:selector xpath="teljesítmények/teljesítmény" />
  <xs:field xpath="@T_játék" />
</xs:keyref>

<xs:keyref name="fejlesztőREF" refer="stúdióID">
  <xs:selector xpath="játékok/játék" />
  <xs:field xpath="@fejlesztő" />
</xs:keyref>
<xs:keyref name="kiadóREF" refer="stúdióID">
  <xs:selector xpath="játékok/játék" />
  <xs:field xpath="@kiadó" />
</xs:keyref>

<xs:keyref name="W_stúdióREF" refer="stúdióID">
  <xs:selector xpath="weboldalak/weboldal" />
  <xs:field xpath="@W_stúdió" />
</xs:keyref>

<xs:keyref name="TNY_játékREF" refer="játékID">
  <xs:selector xpath="támogatottNyelvek/támogatottNyelv" />
  <xs:field xpath="@TNY_játék" />
</xs:keyref>
<xs:keyref name="TNY_nyelvREF" refer="nyelvID">
  <xs:selector xpath="támogatottNyelvek/támogatottNyelv" />
  <xs:field xpath="@TNY_nyelv" />
</xs:keyref>

<xs:keyref name="HJ_játékREF" refer="játékID">
  <xs:selector xpath="hozzáadottJátékok/hozzáadottJáték" />
  <xs:field xpath="@HJ_játék" />
</xs:keyref>
<xs:keyref name="HJ_felhasználóREF" refer="felhasználóID">
  <xs:selector xpath="hozzáadottJátékok/hozzáadottJáték" />
  <xs:field xpath="@HJ_felhasználó"></xs:field>
</xs:keyref>
</xs:element>

</xs:schema>
```

2. feladat

a) adatolvasás

Feladat: A teljes dokumentum feldolgozása és kiírása strukturált formában a konzolra, illetve mentés file-ba.

Megvalósítás: A dokumentum beolvasásához a `DocumentBuilderFactory`, `DocumentBuilder` és `Document` osztályokat használtam fel. Az xml-t a `Document.parse` metódussal olvastam be.

Ezután egy rekurzív függvényt használtam (`printChildren`), hogy végigvegyem a gyökércsomóponttól az összes elemet a dokumentumból, és kiírtassam őket a konzolra ugyanazzal a struktúrával, mint az XML-ben volt. Ez a függvény olyan részfüggvényeket tartalmaz, amelyek szétválasztják a kezdőtag, attribútumok, gyerekelemek és zárótag kiíratását (`printNodeInfo`, `printAttributes`, `printChildren` meghívva a gyerekekre, `printNodeEnd`).

A file-ba való kiíratás függvényei lényegileg ugyanazok, mint a konzolra való kiíratás esetén, csak a `System.out.print` helyett a `FileWriter.write` függvényt használtam.

A teljes `DomReadULA7Z2.java` kódja:

```
package hu.domparse.ULA7Z2;

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.*;
import org.xml.sax.SAXException;

public class DomReadULA7Z2 {
    public static void main(String[] args) {
        String readPath = "XMLTaskULA7Z2/DOMParseULA7Z2/src/files/XMLULA7Z2.xml";
        String writePath =
"XMLTaskULA7Z2/DOMParseULA7Z2/src/files/XMLULA7Z2_2a.xml";
        Document dom = ReadXML(readPath);
        Element root = dom.getDocumentElement();
        printChildren(root, 0);
        DomWriteULA7Z2.DOMToXML(dom, writePath);
    }

    // XML beolvasása, DOM parserrel Document létrehozása
    static Document ReadXML(String path){
        Document dom = null;
        try{
```

```

        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        DocumentBuilder db = dbf.newDocumentBuilder();
        dom = db.parse(new File(path));
        dom.normalize();
    }
    catch (ParserConfigurationException pce){
        System.err.println("Parser config error!");
        pce.printStackTrace();
    }
    catch (IOException ioe){
        System.out.println("Parsing error!");
        ioe.printStackTrace();
    }
    catch (SAXException saxe){
        System.out.println("SAX exeption!");
        saxe.printStackTrace();
    }
    return dom;
}

// adott csomópont strukturált kiírása konzolra rekurzív módon
static void printChildren(Node parent, int indentAmount){
    boolean textOnly = false;
    printNodeInfo(parent, indentAmount);
    for (Node child = parent.getFirstChild(); child != null; child =
child.getNextSibling()) {
        if (child.hasChildNodes()){
            printChildren(child, indentAmount+1);
        }
        else {
            // ha már nincs gyerekeleme, csak szöveges tartalma, az
            kiíratásra kerül
            if (!isRandomNode(child)){
                if (child.hasAttributes()){
                    System.out.println();
                    indent(indentAmount + 1);
                    System.out.print("<" + child.getNodeName());
                    printAttributes(child);
                    System.out.print(" />");
                }
                else{
                    System.out.print(child.getNodeValue());
                    textOnly = true;
                }
            }
        }
    }
}

```



```

    }
}
printNodeEnd(parent, indentAmount, textOnly);
}

// csomópont kezdőtagjének kiírása
private static void printNodeInfo(Node node, int indentAmount){
    System.out.print("\n");
    indent(indentAmount);
    System.out.print("<" + node.getNodeName());
    printAttributes(node);
    System.out.print(">");
}

// csomópont attribútumainak kiírása
private static void printAttributes(Node node){
    if(node.hasAttributes()){
        NamedNodeMap attrs = node.getAttributes();
        for (int i = 0; i < attrs.getLength(); i++){
            System.out.print(" " + attrs.item(i));
        }
    }
}

// csomópont zárótagjének kiírása
private static void printNodeEnd(Node node, int indentAmount, boolean
textOnly){
    if (!textOnly){
        System.out.println();
        indent(indentAmount);
        System.out.print("</" + node.getNodeName() + ">\n");
    }
    else{
        System.out.print("</" + node.getNodeName() + ">");
    }
}

// csak szöveges vagy üres csomópontokat irasson ki
// ez a fölösleges üres sorok elkerülésének érdekében van
private static boolean isRandomNode(Node node){
    if (node.hasAttributes()) return false;
    if (node.getNodeType() != Node.TEXT_NODE){
        return true;
    }
    String value = node.getNodeValue();

```

```

        if (value == null || value.trim().isEmpty() || value == "#text"){
            return true;
        }
        return false;
    }

    // behúzások megvalósítása
    private static void indent(int amount){
        for (int i = 0; i < amount; i++){
            System.out.print("    ");
        }
    }
}

```

b) adatlekérdezés

Feladat: Az XML dokumentum példányaiból legalább 5 módosítás készítése és kiírása a konzolra.

Megvalósítás: Az XML-t ugyanúgy a Document.parse metódussal olvastam be a programba.

A lekérdezéseimben a Document.getElementsByTagName metódust használtam az egyes kiírandó elemek megtalálására, és ha valamilyen feltétel a gyerekelemre vonatkozott, azokat is ezzel találtam meg. Az attribútumok ellenőrzésére Element-té alakítottam az adott Node-ot, és a getAttribute függvénnyel lekértem a szükséges attribútumot.

Öt darab lekérdezést hajtottam végre:

1. **felhasználók platformon megjelenő neve:** azaz ha meg van adva egy megjelenítettNév, akkor az, ha nincs, akkor a felhasználónév
2. **felhasználók által elért teljesítmények:** olyan módon írtam ezt ki, hogy a felhasználó és a teljesítmény nevét lássuk, ne az ID-jüket
3. **azon támogatott nyelvek kiírása, amik esetén hang is támogatva van:** egyes játékok csak feliratot adnak adott nyelvekre, beszédhangot azonban nem; itt azokat a nyelveket kértem le játékonként, amelyek esetén a játéknak hangszínészete is van
4. **adott azonosítójú stúdió weboldalai:** egy egyszerűbb lekérdezés – ha az adott ID-jű stúdiónak vannak weboldalai tárolva, akkor azokat kiírja konzolra
5. **adott azonosítójú felhasználó barátai:** egy listát ír ki a konzolra az adott felhasználó barátainak felhasználónevével

A lekérdezések eredményét nem XML struktúrában írtam ki a konzolra, hanem inkább olyan formában, mint ahogy például egy felhasználó kaphatná vissza a lekérdezett információt.

A DomQueryULA7Z2.java main függvénye:

```
public static void main(String[] args) {
    String readPath = "XMLTaskULA7Z2/DOMParseULA7Z2/src/files/XMLULA7Z2.xml";
    Document dom = DomReadULA7Z2.ReadXML(readPath);

    System.out.println("Lekérdezések:\n");
    // felhasználók platformon megjelenő neve
    // ha meg van adva megjelenített név, akkor az, ha nincs, akkor a
    felhasználónév
    DisplayedName(dom);
    System.out.println();

    // felhasználók által elért teljesítmények
    // olyan módon lekérdezve, hogy a felhasználó és a teljesítmény neve is
    látszodjon
    AchievedAchievements(dom);
    System.out.println();

    // azon támogatott nyelvek kiírása, amik esetén hang is támogatva van
    SupportedSoundLanguages(dom);
    System.out.println();

    // adott azonosítójú stúdió weboldalai
    String SID = "SMS";
    StudioWebsites(dom, SID);
    System.out.println();

    // adott azonosítójú felhasználó barátainak kiírása
    String FID = "F12";
    FriendsList(dom, FID);
}
```

1. lekérdezés: DisplayedName

```
// első lekérdezés: felhasználók platformon megjelenő neve
// ez a megjelenített név, ha meg van adva, vagy a felhasználónév, ha nincs
megadva
private static void DisplayedName(Document dom){
    System.out.println("- felhasználók platformon megjelenő neve");
    NodeList users = dom.getElementsByTagName("felhasználó");
    for (int i = 0; i < users.getLength(); i++){
        Element user = (Element)users.item(i);
```

```

        Node displayName =
user.getElementsByTagName("megjelenítettNév").item(0);
        if (displayName == null){
            displayName =
user.getElementsByTagName("felhasználónév").item(0);
        }
        System.out.println(displayName.getTextContent());
    }
}

```

2. lekérdezés: AchievedAchievements

```

// második lekérdezés: felhasználók által elért teljesítmények
(felhasználónév, teljesítménynév, elért dátuma)
private static void AchievedAchievements(Document dom){
    // felhasználók nevének kigyűjtése ID szerint
    Map<String,String> userNames = new Hashtable<>();
    NodeList users = dom.getElementsByTagName("felhasználó");
    for (int i = 0; i < users.getLength(); i++){
        Element user = (Element)users.item(i);
        String FID = user.getAttribute("FID");
        String userName =
user.getElementsByTagName("felhasználónév").item(0).getTextContent();
        userNames.put(FID, userName);
    }
    // teljesítmények nevének kigyűjtése ID szerint
    Map<String,String> achievementNames = new Hashtable<>();
    NodeList achievements = dom.getElementsByTagName("teljesítmény");
    for (int i = 0; i < achievements.getLength(); i++){
        Element achievement = (Element)achievements.item(i);
        String TID = achievement.getAttribute("TID");
        String achievementName =
achievement.getElementsByTagName("teljesítményNeve").item(0).getTextContent();
        achievementNames.put(TID, achievementName);
    }
    // kiíratás
    System.out.println("- elért teljesítmények olvasható kiíratása");
    System.out.println(" Felhasználó neve      Teljesítmény
neve      Teljesítmény elértének dátuma");
    System.out.println("-----");
    System.out.println("-----");
    NodeList achievedAchievements =
dom.getElementsByTagName("elértTeljesítmény");
    for (int i = 0; i < achievedAchievements.getLength(); i++){

```

```

        Element achAchievement = (Element)achievedAchievements.item(i);
        String FID = achAchievement.getAttribute("teljesítő");
        String TID = achAchievement.getAttribute("teljesítettTeljesítmény");
        String achievedDate =
achAchievement.getElementsByTagName("teljesítményElérésénekDátuma").item(0).getTextContent();
        System.out.println(userNames.get(FID) + " " +
achievementNames.get(TID) + " " + achievedDate);
    }
}

```

3. lekérdezés: SupportedSoundLanguages

```

// harmadik lekérdezés: azon támogatott nyelvek kiírása, amik esetén hang is
támogatva van
private static void SupportedSoundLanguages(Document dom){
    // játék nevének kigyűjtése ID szerint
    Map<String, String> gameNames = new Hashtable<>();
    NodeList games = dom.getElementsByTagName("játék");
    for (int i = 0; i < games.getLength(); i++){
        Element game = (Element)games.item(i);
        String JID = game.getAttribute("JID");
        String gameName =
game.getElementsByTagName("játékNeve").item(0).getTextContent();
        gameNames.put(JID, gameName);
    }

    // nyelvek kigyűjtése ID szerint
    Map<String, String> languages = new Hashtable<>();
    NodeList langs = dom.getElementsByTagName("nyelv");
    for (int i = 0; i < langs.getLength(); i++){
        Element lang = (Element)langs.item(i);
        String NyID = lang.getAttribute("NyID");
        String language = lang.getTextContent();
        languages.put(NyID, language);
    }

    // hanggal támogatott nyelvek kiírása
    System.out.print("- hanggal támogatott nyelvek kiírása játékonként");
    String prevJID = null;
    NodeList supportedLanguages =
dom.getElementsByTagName("támogatottNyelv");
    for (int i = 0; i < supportedLanguages.getLength(); i++){
        Element supportedLanguage = (Element)supportedLanguages.item(i);
    }
}

```

```

        boolean supportsLanguage =
supportedLanguage.getElementsByTagName("hang").item(0).getTextContent().equals("i
gen") ? true : false;
        if (supportsLanguage){
            String JID = supportedLanguage.getAttribute("TNY_játék");
            String NyID = supportedLanguage.getAttribute("TNY_nyelv");
            if (prevJID == null || !prevJID.equals(JID)){
                prevJID = JID;
                System.out.println();
                System.out.println(gameNames.get(JID));
                System.out.println("-----");
            }
            System.out.println(languages.get(NyID));
        }
    }
}

```

4. lekérdezés: StudioWebsites

```

// negyedik lekérdezés: adott azonosítójú stúdió weboldalai
private static void StudioWebsites(Document dom, String SID){
    System.out.println("- " + SID + " azonosítójú stúdió weboldalainak
kiírása (olvasható formátumban, nem XML struktúrálással)");
    // ellenőrizzük, hogy létezik-e adott SID-jű stúdió
    String studioName = null;
    NodeList studios = dom.getElementsByTagName("stúdió");
    for (int i = 0; i < studios.getLength(); i++){
        Element studio = (Element)studios.item(i);
        if (studio.getAttribute("SID").equals(SID)){
            studioName =
studio.getElementsByTagName("stúdióNeve").item(0).getTextContent();
            break;
        }
    }
    if (studioName == null){
        System.err.println("Nincs ilyen ID-jű stúdió!");
        return;
    }

    // a stúdióhoz tartozó weboldalcsoporthoz megkeresése
    NodeList websiteGroups = dom.getElementsByTagName("weboldal");
    Element SIDwebsiteGroup = null;
    for (int i = 0; i < websiteGroups.getLength(); i++){
        Element websiteGroup = (Element)websiteGroups.item(i);
    }
}

```

```

        if (websiteGroup.getAttribute("W_stúdió").equals(SID)){
            SIDwebsiteGroup = websiteGroup;
            break;
        }
    }
    if (SIDwebsiteGroup == null){
        System.out.println("A " + studioName + " stúdióhoz nem tartozik
weboldalcsoporth!");
        return;
    }

    // ha van hozzátartozó weboldalcsoporth, azt kiírjuk
    System.out.println(studioName + " weboldalai/fiókjai");
    System.out.println("-----");
    NodeList websites = SIDwebsiteGroup.getChildNodes();
    for (int i = 0; i < websites.getLength(); i++){
        Node website = websites.item(i);
        String websiteType = website.getNodeName();
        if (!websiteType.equals("#text"))
            System.out.println(website.getNodeName() + ": " +
website.getTextContent());
    }
}

```

5. lekérdezés: FriendsList

```

// ötödik lekérdezés: adott ID-jú felhasználó barátlistája
private static void FriendsList(Document dom, String FID){
    // felhasználónevek összegyűjtése FID szerint
    Map<String, String> usernames = new Hashtable<>();
    NodeList users = dom.getElementsByTagName("felhasználó");
    for (int i = 0; i < users.getLength(); i++){
        Element user = (Element)users.item(i);
        String fid = user.getAttribute("FID");
        String username =
user.getElementsByTagName("felhasználónév").item(0).getTextContent();
        usernames.put(fid, username);
    }

    System.out.println("- adott ID-jü felhasználó barátlistája");
    // ellenőrizzük, hogy van-e adott ID-jú felhasználó
    if (usernames.get(FID) == null){
        System.err.println("Nincs " + FID + " azonosítójú felhasználó!");
        return;
    }
}

```

```

    }

    // ha van ilyen felhasználó, kiírjuk a barátlistáját
    System.out.println(usernames.get(FID) + " barátlistája");
    System.out.println("-----");
    NodeList friends = dom.getElementsByTagName("barát");
    for (int i = 0; i < friends.getLength(); i++){
        Element friend = (Element)friends.item(i);
        String user1 = friend.getAttribute("felhasználó1");
        String user2 = friend.getAttribute("felhasználó2");
        if (user1.equals(FID))
            System.out.println(usernames.get(user2));
        if (user2.equals(FID))
            System.out.println(usernames.get(user1));
    }
}

```

c) adatmódosítás

Feladat: Az XML példányaiból legalább 5 módosítás készítése és kiírása a konzolra.

Megvalósítás: XML beolvasása Document.parse metódussal.

A módosítandó elemek megtalálásához ezúttal is a Document.getElementsByTagName metódust használtam. Csomópont értékének változtatására a Node.setTextContent metódust használtam, attribútum lekérdezéséhez az Element.getAttribute metódust, csomópont létrehozásához Document.createElement metódust.

Hat módosítást végeztem el:

1. **Megjelenített név beállítása adott ID-jű felhasználó számára:** ha már be van állítva neki, akkor azt lecseréljük, ha nincs, hozzáadjuk a megjelenítettNév csomópontot a megfelelő helyre (tehát a felhasználónév után)
2. **Weboldal hozzáadása adott ID-jű stúdióhoz:** ha még nem tartozik weboldalcsoport a stúdióhoz, akkor létrehozunk egy weboldal elemet (de csak akkor, ha főoldal akarunk hozzáadni, hiszen az az egyetlen kötelező gyerekeleme a weboldal elemnek); ha már van weboldalcsoportja, de az adott típusú weboldalból még nincs, akkor létrehozuk a csomópontot, és beszúrjuk a megfelelő helyre (youtube a főoldal után, facebook a youtube és egyéb oldalak közé, egyéb oldal utolsóként); ha már van olyan típusú weboldal, akkor lecseréljük az értékét, kivéve, ha egyéb típusú, ekkor csak beszúrjuk a weboldal elem utolsó gyerekelemként
3. **Teljesítmény hozzáadása adott ID-jű játékhoz (vagy csak a platformhoz):** hozzáadhatunk egy teljesítményt akár egy konkrét játékhoz (ekkor ellenőrizzük, hogy valóban létezik ilyen ID-jű játék), vagy csak általában a platformhoz. A megadott adatok alapján sorra létrehozuk a csomópontokat, a Node.appendChild metódussal hozzáfűzzük az új csomópontokat a DOM-fához a megfelelő helyre (a teljesítmények elem utolsó gyerekeként).

4. **DLC-k neve után zárójelben az alapjáték nevének kiírása:** ha egy játék valamelyik másik játékhoz tartozó kiegészítő tartalom, akkor a nevéhez hozzátesszük az eredeti játék nevét is zárójelben. Mivel egy valid XML-dokumentumot feltételezünk (tehát azt, hogy minden DLC létező játék kiegészítő tartalma), ezért nem ellenőrizzük, hogy létezik-e az alapjáték attribútumba beírt ID.
5. **Lejárt leárazások törlése:** végignézzük a játékokat, és ha valamelyiknek már lejárt a leárazási időszaka (a vége elemében tárolt dátum a jelenlegi dátum előtti), töröljük a csomópontot, a Node.removeChild metódust meghívva a szülőcsomópontban.
6. **Leárazás hozzáadása vagy módosítása adott ID-jű játéknál:** ha a játékhoz már tartozik leárazás, cseréljük annak az adatait. Ha még nem létezik, létrehozuk a szükséges csomópontokat, és hozzáadjuk a DOM-fához

A DomModifyULA7Z2.java main függvénye:

```
public static void main(String[] args) {
    String readPath = "XMLTaskULA7Z2/DOMParseULA7Z2/src/files/XMLULA7Z2.xml";
    Document dom = DomReadULA7Z2.ReadXML(readPath);
    Element root = dom.getDocumentElement();

    // felhasználó megjelenített nevének beállítása FID alapján
    String FID = "F1";
    Element changedUser = AddOrChangeDisplayName(dom, FID, "New Display
Name");

    // stúdióhoz weboldal hozzáadása/módosítás SID alapján
    // jelenleg ez felépíti a teljes weboldal-csoportját ennek a stúdiónak
    String SID = "BethGS";
    AddOrChangeStudioWebsite(dom, SID, "main", "BethGS mainpage");
    AddOrChangeStudioWebsite(dom, SID, "akármii", "BethGS other1");
    AddOrChangeStudioWebsite(dom, SID, "facebook", "BethGS facebook");
    AddOrChangeStudioWebsite(dom, SID, "valami cucc", "BethGS other2");
    Element changedWebsite = AddOrChangeStudioWebsite(dom, SID, "youtube",
"BethGS youtube");

    // Új teljesítmény hozzáadása
    // először egy konkrét játékhoz tartozó teljesítmény
    String achJID = "TESV";
    AddNewAchievement(dom, achJID, "Olvasó", "Olvass el 50 képességek könyvet.",
"2011-11-11");
    // majd egy általános, a platformra vonatkozó teljesítmény
    AddNewAchievement(dom, null, "Barátságos", "Barátkozz össze 10
felhasználóval a platformon.", "2023-12-04");
    NodeList achievementList = dom.getElementsByTagName("teljesítmény");

    // DLC-k nevéhez hozzáírni, hogy melyik játék DLC-je
```

```

AddBasegameName(dom);

// lejárt leárazás törlése
RemoveEndedSales(dom);

// leárazás megadása/módosítása JID alapján
String JID = "TESV";
AddOrChangeSale(dom, JID, 40, "2023-12-05T05:00:00", "2023-12-
10T10:00:00");

// módosított dokumentum kiírása konzolra
System.out.println("Módosítások:");

System.out.print("- megjelenített név beállítása az " + FID + "
azonosítójú játékos számára");
DomReadULA7Z2.printChildren(changedUser, 0);
System.out.println();

System.out.print("- " + SID + "-hez tartozó weboldal megváltoztatása");
DomReadULA7Z2.printChildren(changedWebsite, 0);
System.out.println();

System.out.print("- teljesítmény hozzáadása a " + achJID + " azonosítójú
játékhoz");
DomReadULA7Z2.printChildren(achievementList.item(achievementList.getLength()
- 2), 0);
System.out.println();

System.out.print("- platformra vonatkozó teljesítmény hozzáadása");
DomReadULA7Z2.printChildren(achievementList.item(achievementList.getLength()
- 1), 0);
System.out.println();

System.out.println("- DLC-k neve után zárójelben az alapjáték nevének
kiírása");
System.out.println("- lejárt leárazások törlése");
System.out.print("- leárazás hozzáadása vagy módosítása a " + JID + "
azonosítójú játékban");
DomReadULA7Z2.printChildren(root.getElementsByTagName("játékok").item(0),
0);

}

```

1. módosítás: AddOrChangeDisplayName

```
// első módosítás: felhasználó megjelenített nevének beállítása
private static Element AddOrChangeDisplayName(Document dom, String FID,
String displayName){
    // megadott azonosítójú felhasználó megtalálása
    NodeList users = dom.getElementsByTagName("felhasználó");
    Element FIDuser = null;
    Node displayNameNode = null;
    for (int i = 0; i < users.getLength(); i++){
        Element user = (Element)users.item(i);
        if (user.getAttribute("FID").equals(FID)){
            displayNameNode =
user.getElementsByTagName("megjelenítettNév").item(0);
            FIDuser = user;
        }
    }
    if (FIDuser == null){
        System.err.println("Nincs " + FID + " azonosítójú felhasználó!");
        return FIDuser;
    }

    // ha a felhasználónak még nincs megjelenített név beállítva, létrehozuk
a csomópontot
    if (displayNameNode == null){
        // csomópont létrehozása
        displayNameNode = dom.createElement("megjelenítettNév");
        displayNameNode.setTextContent(displayName);

        // megjelenítettNév csomópont beszúrása a felhasználónév csomópont
után
        Node registrationDate =
FIDuser.getElementsByTagName("fiókLétrehozásánakDátuma").item(0);
        FIDuser.insertBefore(displayNameNode, registrationDate);
    }
    // ha már van megjelenített neve a felhasználónak
    else {
        displayNameNode.setTextContent(displayName);
    }
    return FIDuser;
}
```

2. módosítás: AddOrChangeStudioWebsite

```
// második módosítás: stúdióhoz weboldal hozzáadása SID alapján
private static Element AddOrChangeStudioWebsite(Document dom, String SID,
String websiteType, String value){
    // ellenőrizzük, hogy létezik-e adott azonosítójú stúdió
    NodeList studios = dom.getElementsByTagName("stúdió");
    boolean SIDstudio = false;
    for (int i = 0; i < studios.getLength() && SIDstudio == false; i++){
        Element studio = (Element)studios.item(i);
        if (studio.getAttribute("SID").equals(SID)){
            SIDstudio = true;
        }
    }
    if (!SIDstudio){
        System.err.println("A weboldalnak valamilyen stúdióra mutatnia
kell!");
        return null;
    }

    // a weboldal típusát átalakítjuk olyanra, amilyen gyerekcsomópont
    lehetséges egy weboldal elemben
    websiteType = websiteType.toLowerCase();
    if (websiteType.equals("main") || websiteType.equals("főoldal") ||
websiteType.equals("mainpage")){
        websiteType = "főoldal";
    }
    else if (websiteType.equals("youtube") || websiteType.equals("yt")){
        websiteType = "Youtube";
    }
    else if (websiteType.equals("facebook") || websiteType.equals("fb")){
        websiteType = "Facebook";
    }
    else{
        websiteType = "egyéb";
    }

    // megadott azonosítójú weboldalcsoporthoz megtalálása
    NodeList websiteGroups = dom.getElementsByTagName("weboldal");
    Element SIDwebsiteGroup = null;
    Node website = null;
    for (int i = 0; i < websiteGroups.getLength(); i++){
        Element websiteGroup = (Element)websiteGroups.item(i);
        if (websiteGroup.getAttribute("W_stúdió").equals(SID)){
            website = websiteGroup.getElementsByTagName(websiteType).item(0);
        }
    }
}
```

```

        SIDwebsiteGroup = websiteGroup;
    }
}
if (SIDwebsiteGroup == null && !websiteType.equals("főoldal")){
    System.err.println("Nincs " + SID + "-hez tartozó weboldalcsoporthoz! Ha létre szeretnél hozni egyet, először főoldalt kell hozzáadni!");
    return null;
}

// Ha még nem tartozik weboldalcsoporthoz a stúdióhoz, és főoldalt akarunk
// módosítani/hozzáadni, akkor létrehozunk
// a weboldalcsoporthoz is
if (SIDwebsiteGroup == null){
    // weboldalcsoporthoz csomópont létrehozása
    Element websiteGroup = dom.createElement("weboldal");
    websiteGroup.setAttribute("WID", SID + "Web");
    websiteGroup.setAttribute("W_stúdió", SID);

    // főoldal csomópont létrehozása és hozzáadása az új
    weboldalcsoporthoz
    Node mainpageNode = dom.createElement("főoldal");
    mainpageNode.setTextContent(value);
    websiteGroup.appendChild(mainpageNode);

    // weboldalcsoporthoz hozzáadása a weboldalak listájához
    // ha még nincs ilyen lista, létrehozunk
    Node websites = dom.getElementsByTagName("weboldalak").item(0);
    if (websites == null){
        websites = dom.createElement("weboldalak");
        Node gamesList = dom.getElementsByTagName("játékok").item(0);
        dom.insertBefore(websites, gamesList);
    }
    websites.appendChild(websiteGroup);
}
else{
    // ha már van kellő típusú csomópont, akkor csak egyszerűen
    lecseréljük az értékét
    if (website != null && !websiteType.equals("egyéb")){
        website.setTextContent(value);
        return SIDwebsiteGroup;
    }
    // ha nincs ilyen csomópont, akkor biztos, hogy nem főoldalról van
    szó, ekkor létrehozunk a megfelelő csomópontot
    website = dom.createElement(websiteType);
    website.setTextContent(value);
}

```

```

        // és beszúrjuk a megfelelő helyre
        if (websiteType.equals("Youtube")){
            Node mainpage =
SIDwebsiteGroup.getElementsByTagName("főoldal").item(0);
            SIDwebsiteGroup.insertBefore(website, mainpage.getNextSibling());
        }
        else if (websiteType.equals("Facebook")){
            Node otherWebsites =
SIDwebsiteGroup.getElementsByTagName("egyéb").item(0);
            SIDwebsiteGroup.insertBefore(website, otherWebsites);
        }
        else{
            SIDwebsiteGroup.appendChild(website);
        }
    }
    return SIDwebsiteGroup;
}

```

3. módosítás: AddNewAchievement

```

// harmadik módosítás: új teljesítmény hozzáadása
private static void AddNewAchievement(Document dom, String JID, String name,
String description, String addDate){
    // a kötelező megadású mezők nem lehetnek null-ok
    if (name == null){
        System.err.println("A teljesítmény nevét kötelező megadni!");
        return;
    }
    if (addDate == null){
        System.err.println("A teljesítmény hozzáadási dátumát kötelező
megadni!");
        return;
    }
    try{
        LocalDate addDateLD = LocalDate.parse(addDate);
        if (addDateLD.compareTo(LocalDate.now()) > 0){
            System.err.println("A hozzáadási dátum nem lehet későbbi a
mainál!");
            return;
        }
    }catch (Exception e){
        System.err.println("A hozzáadási dátum nem megfelelő formátumú! (A
formátum: ÉÉÉÉ-HH-NN, pl. 2022-04-25)");
        e.printStackTrace();
    }
}

```

```

        return;
    }

    // ha játékhoz adunk hozzá teljesítményt, ellenőrizni kell, hogy létezik-e ilyen játék
    if (JID != null){
        boolean JIDgame = false;
        NodeList games = dom.getElementsByTagName("játék");
        for (int i = 0; i < games.getLength(); i++){
            Element game = (Element)games.item(i);
            if (game.getAttribute("JID").equals(JID)){
                JIDgame = true;
                break;
            }
        }
        if (!JIDgame){
            System.err.println("Nincs ilyen azonosítójú játék!");
            return;
        }
    }

    // teljesítmény csomópont létrehozása
    int achievementIndex =
dom.getElementsByTagName("teljesítmény").getLength() + 1;
    Element newAchievement = dom.createElement("teljesítmény");
    newAchievement.setAttribute("TID", "ACH" +
String.valueOf(achievementIndex));
    if (JID != null)
    {
        newAchievement.setAttribute("T_játék", JID);
    }

    // gyerekcsomópontok létrehozása és hozzáadása a teljesítmény
    csomópontokhoz
    Node achievementName = dom.createElement("teljesítményNeve");
    achievementName.setTextContent(name);
    newAchievement.appendChild(achievementName);

    if (description != null){
        Node achievementDescription =
dom.createElement("teljesítményLeírása");
        achievementDescription.setTextContent(description);
        newAchievement.appendChild(achievementDescription);
    }

```

```

        Node achievementAddDate =
dom.createElement("teljesítményHozzáadásánakDátuma");
        achievementAddDate.setTextContent(addDate);
        newAchievement.appendChild(achievementAddDate);

        // új teljesítmény hozzáadása a teljesítménylistához
        // ellenőrizzük, hogy van-e már teljesítménylistánk
        Node achievementList =
dom.getElementsByTagName("teljesítmények").item(0);
        // ha nincs, hozzuk létre, és szűrjük be a megfelelő helyre
        if (achievementList == null)
        {
            achievementList = dom.createElement("teljesítmények");
            achievementList.appendChild(newAchievement);

            // a "teljesítmények" csomópontnak a "játékok" csomópont után kell
jönnie
            // helyes dokumentumfelépítést feltételezve ennek már kötelezően
léteznie kell, mielőtt a teljesítményeket létrehozzuk
            Node insertAfterNode = dom.getElementsByTagName("játékok").item(0);
            dom.insertBefore(achievementList, insertAfterNode.getNextSibling());
        }
        achievementList.appendChild(newAchievement);
    }
}

```

4. módosítás: AddBasegameName

```

// negyedik módosítás: DLC-k neve után odaírni, hogy melyik játék DLC-i
private static void AddBasegameName(Document dom){
    NodeList games = dom.getElementsByTagName("játék");

    // játék nevének gyors elérése érdekében JID-játékNév párosokat hozunk
létre
    Map<String, String> gameNames = new Hashtable<>();
    for (int i = 0; i < games.getLength(); i++){
        Element game = (Element)games.item(i);
        if (!game.hasAttribute("alapjáték")){
            String gameName =
game.getElementsByTagName("játékNeve").item(0).getTextContent();
            gameNames.put(game.getAttribute("JID"), gameName);
        }
    }
}

```



```

        // mivel kézzel bevitt dokumentumrész esetén megtörténhet, hogy
        // valamelyik DLC azelőtt legyen felvive a file-ba, hogy
        // az alapjátékának az adatai bekerüljenek, így biztosabb kétszer
        // végigfutni a játékokon, és csak második alkalommal
        // hozzátenni az alapjáték-neveket a DLC-khez
        // Ha biztosra vehetnénk, hogy teljesen program által lett felvive
        // minden, akkor valószínűleg egyből az első for-ciklus
        // keretén belül el lehetett volna intézni ezt is
        for (int i = 0; i < games.getLength(); i++){
            Element game = (Element)games.item(i);
            if (game.hasAttribute("alapjáték")){
                String baseGameName =
gameNames.get(game.getAttribute("alapjáték"));
                Node gameName = game.getElementsByTagName("játékNeve").item(0);
                gameName.setTextContent(gameName.getTextContent() + " (" +
baseGameName + " DLC)");
            }
        }
    }
}

```

5. módosítás: RemoveEndedSales

```

// ötödik módosítás: lejárt leárazások törlése
public static void RemoveEndedSales(Document dom){
    NodeList salesEndDates = dom.getElementsByTagName("vége");
    for (int i = 0; i < salesEndDates.getLength(); i++){
        Node saleEndDate = salesEndDates.item(i);
        LocalDateTime endTime =
LocalDateTime.parse(saleEndDate.getTextContent());
        if (endTime.compareTo(LocalDateTime.now()) <= 0){
            Node sale = saleEndDate.getParentNode();
            sale.getParentNode().removeChild(sale);
        }
    }
}
}

```

6. módosítás: AddOrChangeSale

```

// hatodik módosítás: leárazás hozzáadása játékhoz vagy létező módosítása
private static void AddOrChangeSale(Document dom, String JID, int percent,
String startTime, String endTime) {
    // adatok helyességének ellenőrzése
    if (percent < 0 || percent > 100){

```

```

        System.err.println("A leárazási százaléknak 0 és 100 közötti értéknek
kell lennie!");
        return;
    }
    LocalDateTime startTimeLDT;
    LocalDateTime endTimeLDT;
    try{
        startTimeLDT = LocalDateTime.parse(startTime);
    } catch(Exception e){
        System.err.println("A leárazás kezdési ideje nem megfelelő
formátumban van. (A helyes formátum: ÉÉÉÉ-HH-NNT00:PP:MM, pl 2023-03-
11T11:12:12)");
        e.printStackTrace();
        return;
    }
    try{
        endTimeLDT = LocalDateTime.parse(endTime);
    } catch(Exception e){
        System.err.println("A leárazás lejárási ideje nem megfelelő
formátumban van. (A helyes formátum: ÉÉÉÉ-HH-NNT00:PP:MM, pl 2023-03-
11T11:12:12)");
        e.printStackTrace();
        return;
    }
    if (startTimeLDT.compareTo(endTimeLDT) > 0 ||
startTimeLDT.equals(endTimeLDT)){
        System.err.println("A leárazás kezdési idejének korábbiak kell
lennie, mint a lejárási időnek!");
        return;
    }
    if (endTimeLDT.compareTo(LocalDateTime.now()) < 0){
        System.err.println("Nem lehet már lejárt leárazást hozzáadni!");
    }

    // megadott ID-jú játék megtalálása
    NodeList games = dom.getElementsByTagName("játék");
    Element JIDgame = null;
    Node sale = null;
    for (int i = 0; i < games.getLength(); i++){
        Element game = (Element)games.item(i);
        if (game.getAttribute("JID").equals(JID)){
            sale = game.getElementsByTagName("leárazás").item(0);
            JIDgame = game;
            break;
        }
    }

```

```

    }
    if (JIDgame == null){
        System.err.println("Nincs " + JID + " azonosítójú játék!");
        return;
    }

    // ha a játékhoz nem tartozik leárazás, hozzáadjuk
    if (sale == null){
        // leárazás csomópont létrehozása
        sale = dom.createElement("leárazás");
        Node salePercent = dom.createElement("százalék");
        salePercent.setTextContent(String.valueOf(percent));

        Node saleStartTime = dom.createElement("kezdés");
        saleStartTime.setTextContent(startTime);

        Node saleEndTime = dom.createElement("vége");
        saleEndTime.setTextContent(endTime);

        sale.appendChild(salePercent);
        sale.appendChild(saleStartTime);
        sale.appendChild(saleEndTime);

        // leárazás csomópont hozzáadása a játék elemhez
        JIDgame.appendChild(sale);
    }
    // ha már létezik a leárazás, módosítjuk
    else {
        Node saleData =
((Element)sale).getElementsByTagName("százalék").item(0);
        saleData.getFirstChild().setNodeValue(String.valueOf(percent));

        saleData = ((Element)sale).getElementsByTagName("kezdés").item(0);
        saleData.getFirstChild().setNodeValue(startTime);

        saleData = ((Element)sale).getElementsByTagName("vége").item(0);
        saleData.getFirstChild().setNodeValue(endTime);
    }
}

```

d) adatírás

Feladat: készítsen egy DOM API programot, amely egy XMLNeptunkod.xml dokumentum tartalmát fa struktúra formában kiírja a konzolra és egy XMLNeptunkod1.xml fájlba.

Megvalósítás: A nagyobb csomópontokhoz (stúdió, játék, felhasználó, támogatottNyelv stb.) egy-egy metódust írtam, ami paraméterként megkapja a csomópontához tartozó lehetséges adatokat (attribútumok és gyerekelemek értéke). Mindegyik metódusban ellenőrizve van, hogy minden szükséges adat meg legyen adva (ne legyen null), és az értékek megfeleljenek a tőlük elvárt feltételeknek (pl. az ID-k ne ismétlődjenek, az IDREF-ek létező elemre mutassanak, a százalék 0 és 100 között legyen, az ár nem lehet negatív stb.). Ezek a függvények visszaadják az elkészült csomópontot, vagy hiba, hiányosság esetén null-t (ebben az esetben egy hibaüzenetet is kiírnak).

Az ID-vel rendelkező elemek létrehozásakor az elkészült csomópont egy Map változóba kerül (ahol ID-csomópont párosokat tárolunk mindegyik nagy csomópont esetén). Ezek a Map változók globális változók, mivel a különböző elemlelétrehozási metódusoknál ezek segítenek gyorsan ellenőrizni, hogy létezik-e adott ID-jű, adott típusú elem, és így meghatározható legyen, hogy egy IDREF értéke megfelelő-e.

Az egyszerű típusú csomópontoknál a különböző típusokra (int, string, enumeration valamely eleme stb.) külön függvényeket készítettem, hogy az ezekre vonatkozó ellenőrzések egy részét leegyszerűsítsem (pl. számadat esetén egyből szám lehessen a paraméter, ne kelljen Integer.parse-ot használni) vagy elkülönítsem (pl. dátum típusú adatnál a formátumellenőrzés), illetve a játékműfajra vonatkozó csomópontoknál csak egy indexre van szükség, amely a globális genres változóból rendeli majd hozzá a megfelelő műfaj stringet a csomópontához.

A DomWriteULA7Z2 által kiírt XML tartalma nem egyezik teljes mértékben az eredeti XML-ével. Ez azért van, mert az eredeti file-ba rengeteg adat került be, hogy a lekérdezésekhez és módosításokhoz legyen miből válogatni, de ennél a feladatnál inkább az volt a lényeg, hogy egy valid, eredetivel megegyező struktúrájú XML-t fel tudunk építeni, ezért szűkítettem a példányok halmazát valami elfogadható időben begépelhető méretűre.

Az adatbeviteli részt úgy írtam meg, hogy viszonylag átlátható és könnyen bővíthető/módosítható legyen, illetve ha valahol becsúszna egy hiba, akkor könnyű legyen beazonosítani (azáltal, hogy a felépült dokumentum csak akkor kerüljön kiírásra, ha minden csomópontot sikerült megfelelően létrehozni, ellenkező esetben pedig csak a hibaüzeneteket látjuk, és azt, hogy melyik csomópontnál következtek be).

A file-ba való kiírás metódusainak működése az a) alpontban vannak leírva, mivel a konkrét file-ba írás lépésén kívül minden ugyanaz, mint a DomReadULA7Z2.java-ban leírt printChildren és hozzátartozó metódusok.

A DomWriteULA7Z2.java globális változói:

```
// néhány gyakran szükséges változó
private static FileWriter xml;
private static Document dom;

// ID-Csomópont párosok mentése gyors IDREF-ellenőrzés érdekében
private static Map<String, Node> studios = new Hashtable<>();
private static Map<String, Node> websiteGroups = new Hashtable<>(); // itt a
W_stúdió szerint tároljuk
private static Map<String, Node> games = new Hashtable<>();
private static Map<String, Node> achievements = new Hashtable<>();
private static Map<String, Node> languages = new Hashtable<>();
private static Map<String, Node> users = new Hashtable<>();
private static List<String> friends = new ArrayList<>(); // a két felhasználó
FID-jét fűzzük össze egy stringként, mindkét sorrendben

// minden ID mentése gyors ismétlődésellenőrzés érdekében
private static List<String> IDs = new ArrayList<>();

// műfajok és állapotok nevei
private static String[] genres = {"fantasy", "horror", "akció", "kaland",
"rejtély", "romantikus", "RPG", "shooter", "multiplayer", "singleplayer", "FPS",
"belső-nézetes", "külső-nézetes", "indie"};
```

A main függvény:

```
public static void main(String[] args) {
    String writePath =
"XMLTaskULA7Z2/DOMParseULA7Z2/src/files/XMLULA7Z2_2d.xml";

    // dom létrehozása
    try{
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        DocumentBuilder db = dbf.newDocumentBuilder();
        dom = db.newDocument();
    } catch (Exception e){
        e.printStackTrace();
        return;
    }

    Element root =
(Element)dom.appendChild(dom.createElement("XMLTaskULA7Z2"));
    boolean error = false;
```

```

        // az egyes csomópontok létrehozásához szükséges adatok
        // (mivel az XML-be rengeteg példányt létrehoztam, itt nem fogom az
egészet újra beírni, csak egy részét)
        //      - stúdiók
        String[] SIDs = {"BT34", "BethGS", "BethSW", "RG"};
        String[] studioNames = {"34BigThings srl", "Bethesda Game Studios",
"Bethesda Softworks", "Raiser Games"};
        String[] introTexts = {null, "Bethesda Game Studios bemutatkozó
szövege...", "Bethesda Softworks bemutató szöveg valami", null};
        int[] studioFoundationYears = {2013, 2001, 1986, 2016};

        Node studiosNode = dom.createElement("stúdiók");
        root.appendChild(studiosNode);
        for (int i = 0; i < SIDs.length; i++){
            System.out.println(i + ". stúdió");
            Node studio = CreateStudioNode(SIDs[i], studioNames[i],
introTexts[i], studioFoundationYears[i]);
            if (studio == null){
                error = true;
                continue;
            }
            studiosNode.appendChild(studio);
        }

        //      - weboldalak
        String[] wSIDs = SIDs;
        String[] mainpages = {"34bigthings.com", "bethesdagamestudios.com",
"bethesda.net", "www.raisergames.com"};
        String[] youtubePages = {"34BigThings_games", null, "bethesda",
"RaiserGames"};
        String[] facebookPages = {"34bigthings", null, "bethesda.anz",
"RaiserGamesPublisher"};
        String[][] others = {null, {"twitter.com/bethesdastudios",
"discord.com/invite/bethesdastudios"}, {"twitter.com/bethesda"},
{"instagram.com/raisergames"}};

        if (wSIDs != null && wSIDs.length > 0){
            Node websitesNode = dom.createElement("weboldalak");
            root.appendChild(websitesNode);
            for (int i = 0; i < wSIDs.length; i++){
                System.out.println(i + ". weboldalcsoport");
                Node websiteGroup = CreateWebsiteGroupNode(null, wSIDs[i],
mainpages[i], youtubePages[i], facebookPages[i], others[i]);
                if (websiteGroup == null){

```

```

        error = true;
        continue;
    }
    websitesNode.appendChild(websiteGroup);
}

//      - játékok
String[] JIDs = {"TESV", "TESVDB", "TESVHF", "GoD"};
String[] bjIDs = {null, "TESV", "TESV", null};
String[] devSIDs = {"BethGS", "BethGS", "BethGS", "BT34"};
String[] pSIDs = {"BethSW", "BethSW", "BethSW", "RG"};
String[] gameNames = {"The Elder Scrolls V: Skyrim", "Dragonborn",
"Hearthfire", "Goat of Duty"};
float[] prices = {20, 20, 5, 10};
String[] publishedDates = {"2011-11-11", "2012-12-04", "2012-09-04",
"2019-07-10"};
int[][] gameGenres = {{2, 9, 0, 3, 6}, {2, 9, 0, 3, 6}, {2, 9, 0, 3, 6},
{2, 10, 8}};
String[][] saleDatas = {null, null, {"5", "2023-11-25T00:00:00", "2023-
12-06T23:59:59"}, null};

Node gamesNode = dom.createElement("játékok");
root.appendChild(gamesNode);
for (int i = 0; i < JIDs.length; i++){
    System.out.println(i + ". játék");
    Node game = CreateGameNode(JIDs[i], bjIDs[i], devSIDs[i], pSIDs[i],
gameNames[i], prices[i], publishedDates[i], gameGenres[i], saleDatas[i]);
    if (game == null){
        error = true;
        continue;
    }
    gamesNode.appendChild(game);
}

//      - teljesítmények
String[] TIDs = {"ACH1", "ACH2", "ACH3", "ACH4", "ACH6", "ACH12"};
String[] tJIDs = {null, "TESV", "TESV", "TESVHF", "TESVDB", "GoD"};
String[] aNames = {"Gyűjtemény", "Legendás", "Thu'um mester", "Polgár",
"Miraak Temploma", "Cuki és halálos"};
String[] descriptions = {"Szerezz meg 10 játékot.", "Érj el 100-as
szintet valamely képességeddel.", "Tanulj meg 20 kiáltást.", "Vásárolj egy
házat.", null, "Ölj meg 10 ellenfelet anélkül, hogy meghalj."};
String[] aAddDates = {"2010-05-25", "2011-11-11", "2011-11-11", "2012-09-
04", "2014-12-01", "2019-08-10"};

```

```

if (TIDs != null && TIDs.length > 0){
    Node achievementsNode = dom.createElement("teljesítmények");
    root.appendChild(achievementsNode);
    for (int i = 0; i < TIDs.length; i++){
        System.out.println(i + ". teljesítmény");
        Node achievement = CreateAchievementNode(TIDs[i], tJIDs[i],
aNames[i], descriptions[i], aAddDates[i]);
        if (achievement == null){
            error = true;
            continue;
        }
        achievementsNode.appendChild(achievement);
    }
}

//      - nyelvek
String[] NyIDs = {"EN", "DE", "IT", "RU", "JP", "HU", "TR"};
String[] langs = {"angol", "német", "olasz", "orosz", "japán", "magyar",
"török"};

if (NyIDs != null && NyIDs.length > 0){
    Node languagesNode = dom.createElement("nyelvek");
    root.appendChild(languagesNode);
    for (int i = 0; i < NyIDs.length; i++){
        System.out.println(i + ". nyelv");
        Node language = CreateLanguageNode(NyIDs[i], langs[i]);
        if (language == null){
            error = true;
            continue;
        }
        languagesNode.appendChild(language);
    }
}

//      - támogatott nyelvek
String[] slJIDs = {"TESV", "TESV", "TESV", "TESV", "TESV", "TESVDB",
"TESVDB", "TESVDB", "TESVDB", "TESVHF", "TESVHF", "TESVHF", "TESVHF",
"TESVHF", "GoD", "GoD", "GoD", "GoD", "GoD", "GoD"};
String[] slNyIDs = {"EN", "JP", "DE", "IT", "RU", "EN", "JP", "DE", "IT",
"RU", "EN", "JP", "DE", "IT", "RU", "EN", "JP", "DE", "IT", "RU", "TR"};
boolean[] subtitleSupport = {true, true, true, true, true, true, true,
true, true, true, true, true, true, true, true, true, true, true, true,
true};

```



```

        boolean[] soundSupport = {true, true, true, true, true, true, false,
true, true, true, true, false, true, true, true, false, false, false, false,
false, false};

        if (slJIDs != null && slJIDs.length > 0){
            Node supportedLanguagesNode = dom.createElement("támogatottNyelvek");
            root.appendChild(supportedLanguagesNode);
            for (int i = 0; i < slJIDs.length; i++){
                System.out.println(i + ". támogatott nyelv");
                Node supportedLanguage = CreateSupportedLanguageNode(slJIDs[i],
slNyIDs[i], subtitleSupport[i], soundSupport[i]);
                if (supportedLanguage == null){
                    error = true;
                    continue;
                }
                supportedLanguagesNode.appendChild(supportedLanguage);
            }
        }

        //      - felhasználók
        String[] FIDs = {"F1", "F5", "F7", "F11", "F13", "F14"};
        String[] usernames = {"toothed2010", "snowwingedhunter123", "Dand3lion",
"hooty3", "corgiii89", "julienmoveit"};
        String[] displayNames = {null, "Snow-winged hunter", "Jaskier the bard",
"Hootsiifer", null, null};
        String[] registrationDates = {"2010-06-11", "2017-01-15", "2019-10-20",
"2022-08-31", "2022-11-05", "2023-12-01"};
        String[] birthdates = {"1982-12-30", "2002-10-04", "2000-09-23", "2000-
05-18", "1989-03-10", "2007-02-16"};

        if (FIDs != null && FIDs.length > 0){
            Node usersNode = dom.createElement("felhasználók");
            root.appendChild(usersNode);
            for (int i = 0; i < FIDs.length; i++){
                System.out.println(i + ". felhasználó");
                Node user = CreateUserNode(FIDs[i], usernames[i],
displayNames[i], registrationDates[i], birthdates[i]);
                if (user == null){
                    error = true;
                    continue;
                }
                usersNode.appendChild(user);
            }
        }
    }

```

```

//      - barátok
String[] FID1 = {"F11", "F11", "F11", "F11", "F11", "F1", "F13", "F7",
"F13"};
String[] FID2 = {"F1", "F5", "F7", "F13", "F14", "F7", "F5", "F5",
"F14"};

if (FID1 != null && FID1.length > 0){
    Node friendsNode = dom.createElement("barátok");
    root.appendChild(friendsNode);
    for (int i = 0; i < FID1.length; i++){
        System.out.println(i + ". barát");
        Node friend = CreateFriendNode(FID1[i], FID2[i]);
        if (friend == null){
            error = true;
            continue;
        }
        friendsNode.appendChild(friend);
    }
}

//      - hozzáadott játékok
String[] agJIDs = {"TESV", "TESVHF", "TESV", "TESVDB", "TESVHF", "GoD",
"TESVHF"};
String[] agFIDs = {"F1", "F1", "F5", "F5", "F5", "F5", "F11"};
String[] agAddDates = {"2010-06-19", "2018-05-23", "2021-11-11", "2021-
11-11", "2021-11-11", "2017-09-09", "2022-10-01"};
String[] states = {"kosárban", "kívánságlistán", "megvéve", "megvéve",
"megvéve", "megvéve", "megvéve"};
int[] playtimes = {0, 0, 174, 9, 31, 13, 0};

if (agJIDs != null && agJIDs.length > 0){
    Node addedGamesNode = dom.createElement("hozzáadottJátékok");
    root.appendChild(addedGamesNode);
    for (int i = 0; i < agJIDs.length; i++){
        System.out.println(i + ". hozzáadott játék");
        Node addedGame = CreateAddedGameNode(agJIDs[i], agFIDs[i],
agAddDates[i], states[i], playtimes[i]);
        if (addedGame == null){
            error = true;
            continue;
        }
        addedGamesNode.appendChild(addedGame);
    }
}

```

```

//      - elért teljesítmények
String[] aaFIDs = {"F5", "F5", "F5"};
String[] aaTIDs = {"ACH2", "ACH3", "ACH6"};
String[] achieveDates = {"2021-12-02", "2021-11-15", "2022-04-13"};

if (aaFIDs != null && aaFIDs.length > 0){
    Node achievedAchievementsNode =
dom.createElement("elértTeljesítmények");
    root.appendChild(achievedAchievementsNode);
    for (int i = 0; i < aaFIDs.length; i++) {
        System.out.println(i + ". elért teljesítmény");
        Node achievedAchievement =
CreateAchievedAchievementsNode(aaFIDs[i], aaTIDs[i], achieveDates[i]);
        if (achievedAchievement == null){
            error = true;
            continue;
        }
        achievedAchievementsNode.appendChild(achievedAchievement);
    }
}

if (!error){
    DomReadULA7Z2.printChildren(root, 0);
    DOMToXML(dom, writePath);
}
}

```

Egyszerű csomópontokra vonatkozó függvények:

```

// különböző típusú csomópontok létrehozása
// megspórolja annak az ellenőrzését, hogy megfelelő adattípust adunk-e meg
egy adott csomóponthoz
private static Node createTextNode(String nodeName, String nodeContent){
    if (nodeContent == null || nodeContent.isEmpty()){
        System.err.println("(" + nodeName + ") A csomópontok tartalma nem
lehet üres!");
        return null;
    }
    Node node = dom.createElement(nodeName);
    node.setTextContent(nodeContent);
    return node;
}

```

```
private static Node createIntNode(String nodeName, int nodeContent){  
    return createTextNode(nodeName, String.valueOf(nodeContent));  
}
```

```
private static Node createFloatNode(String nodeName, float nodeContent){  
    return createTextNode(nodeName, String.valueOf(nodeContent));  
}
```

```
private static Node createDateNode(String nodeName, String nodeContent,  
String dateType){  
    // dateType lehet date vagy datetime  
    if (dateType.equals("date")){  
        try{  
            LocalDate.parse(nodeContent);  
            return createTextNode(nodeName, nodeContent);  
        }catch(Exception e){  
            System.err.println("(" + nodeName + ") Dátum formátuma nem  
megfelelő! Helyes formátum: ÉÉÉÉ-HH-NN, pl. 2023-10-12");  
            return null;  
        }  
    }  
    else if (dateType.equals("datetime")){  
        try{  
            LocalDateTime.parse(nodeContent);  
            return createTextNode(nodeName, nodeContent);  
        }catch(Exception e){  
            System.err.println("(" + nodeName + ") Dátum formátuma nem  
megfelelő! Helyes formátum: ÉÉÉÉ-HH-NN'T00:PP:MM, pl. 2023-10-12T10:00:00");  
            return null;  
        }  
    }  
    else{  
        System.err.println("(" + nodeName + ") Két dátumtípus adható meg:  
date, datetime!");  
        return null;  
    }  
}
```

```
private static Node createGenreNode(int genreIndex){  
    if (genreIndex >= genres.length){  
        System.err.println("Az index túl nagy! " + genres.length + " műfaj  
van!");  
    }  
}
```

```

        return null;
    }
    return createTextNode("műfaj", genres[genreIndex]);
}

private static Node createBoolNode(String nodeName, boolean nodeContent){
    if (nodeContent){
        return createTextNode(nodeName, "igen");
    }
    else{
        return createTextNode(nodeName, "nem");
    }
}
}

```

```

private static Node createStateNode(String state){
    // állapot lehet: "kívánságlistán", "kosárban", "megvéve"
    if (!(state.equals("kívánságlistán") || state.equals("kosárban") ||
state.equals("megvéve"))){
        System.err.println("Nem valid állapot! Állapot típusok:
kívánságlistán, kosárban, megvéve");
        return null;
    }
    return createTextNode("állapot", state);
}
}

```

Nagy elemeket létrehozó metódusok:

```

// stúdió csomópont létrehozása
// nem kötelező megadású: introText
private static Node CreateStudioNode(String SID, String name, String
introText, int foundedIn){
    // ellenőrzések
    if (SID == null || SID.isEmpty() || name == null){
        System.err.println("Csak a bemutatószöveg maradhat üresen!");
        return null;
    }
    if (foundedIn < 1900 && foundedIn > LocalDate.now().getYear())
    {
        System.err.println("(" + SID + ") Nem megfelelő évszám!");
        return null;
    }
    if (IDs.contains(SID)){
        System.err.println("(" + SID + ") Már van ilyen ID-jű stúdió!");
        return null;
    }
}

```

```

    }

    // csomópontok létrehozása
    Element studioNode = dom.createElement("stúdió");
    Node nameNode = createTextNode("stúdióNeve", name),
        foundedNode = createIntNode("alapításÉve", foundedIn);

    try{
        studioNode.setAttribute("SID", SID);
        studioNode.appendChild(nameNode);
        if (introText != null){
            Node introTextNode = createTextNode("bemutatóSzöveg", introText);
            studioNode.appendChild(introTextNode);
        }
        studioNode.appendChild(foundedNode);
    } catch (NullPointerException npe){
        return null;
    }

    // csak akkor adjuk hozzá az ID-ket, ha sikerült rendesen létrehozni
    IDs.add(SID);
    studios.put(SID, studioNode);

    return studioNode;
}

```

```

// weboldalcsoport csomópont létrehozása
// nem kötelező megadásúak: WID, youtube, facebook, others
private static Node CreateWebsiteGroupNode(String WID, String SID, String
mainpage, String youtube, String facebook, String[] others){
    // ellenőrzések
    if (SID == null || mainpage == null){
        System.err.println("Az SID és a főoldallink nem maradhatnak
üresen!");
        return null;
    }
    if (studios.get(SID) == null){
        System.err.println("(" + SID + ") Először létre kell hozni az adott
ID-jü stúdiót!");
        return null;
    }
    if (WID == null || WID.isEmpty()){
        WID = SID + "Web";
        if (IDs.contains(WID)){

```

```

        System.err.println("Meg kell adni egy WID-et, mert már létezik "
+ WID + " azonosítójú elem! Nem ajánlott SID+'Web' nevű azonosítót használni a
WID-en kívül!");
        return null;
    }
}
if (IDs.contains(WID)){
    System.err.println("(" + SID + ") Már létezik ilyen azonosítójú
elem!");
    return null;
}
if (websiteGroups.get(SID) != null){
    System.err.println("(" + SID + ") Egy stúdióhoz csak egy
weboldalcsoporthoz tartozhat!");
    return null;
}

// csomópontok létrehozása
Element websiteGroupNode = dom.createElement("weboldal");
Node mainpageNode = createTextNode("főoldal", mainpage);

try{
    websiteGroupNode.setAttribute("WID", WID);
    websiteGroupNode.setAttribute("W_stúdió", SID);
    websiteGroupNode.appendChild(mainpageNode);
} catch (NullPointerException npe){
    return null;
}

if (youtube != null){
    Node youtubeNode = createTextNode("Youtube", youtube);
    if (youtubeNode == null) return null;
    websiteGroupNode.appendChild(youtubeNode);
}
if (facebook != null){
    Node facebookNode = createTextNode("Facebook", facebook);
    if (facebookNode == null) return null;
    websiteGroupNode.appendChild(facebookNode);
}

if (others != null){
    for (String other : others){
        Node otherNode = createTextNode("egyéb", other);
        if (otherNode != null) websiteGroupNode.appendChild(otherNode);
    }
}

```

```

    }

    // csak akkor adjuk hozzá az ID- és weboldalcsoporthoz, ha valóban
    // sikerült létrehozni
    IDs.add(WID);
    websiteGroups.put(SID, websiteGroupNode);

    return websiteGroupNode;
}

```

```

// játék csomópont létrehozása
// nem kötelező megadásúak: baseGame, gameGenres, saleData
private static Node CreateGameNode(String JID, String baseGame, String
developer, String publisher, String name, float price, String publishedDate,
int[] genreIndices, String[] saleData){
    // kötelező megadású elemek ellenőrzése
    if (JID == null || developer == null || publisher == null ||
JID.isEmpty()){
        System.err.println("Csak az alapjáték attribútum lehet üres!");
        return null;
    }
    if (name == null || publishedDate == null){
        System.err.println("Csak a játékműfajok és a leírásinformációk
maradhatnak üresen!");
        return null;
    }

    // attribútumok ellenőrzése
    // (JID-nél egyediség, alapjátéknál, hogy van-e olyan JID-ű játék,
    // fejlesztő és kiadó stúdiónál, hogy van-e ilyen stúdió)
    if (IDs.contains(JID)){
        System.err.println("(" + JID + ") Már van ilyen ID-jű játék!");
        return null;
    }
    if (baseGame != null && games.get(baseGame) == null){
        System.err.println("(" + JID + ") Alapjátéknak csak már bevitt játék
feltethető meg!");
    }
    if (studios.get(developer) == null || studios.get(publisher) == null){
        System.err.println("(" + JID + ") A fejlesztőnek és kiadónak már
létező stúdióknak kell lenniük!");
        return null;
    }
}

```



```

        // elemtartalmak ellenőrzése
        // (a leárazások elemei közül egyik sem lehet üres, ha már meg van adva;
a százalék 0 és 100 között legyen; az ár ne legyen negatív)
        if (price < 0){
            System.err.println("Az ár nem lehet negatív!");
            return null;
        }
        if (saleData != null){
            if (saleData.length != 3){
                System.err.println("Meg kell adni az összes leárazásadatot!");
                return null;
            }
            for (String saleElement : saleData){
                if (saleElement == null){
                    System.err.println("Ha már meg van adva valamilyen
leárazásadat, meg kell adni az egészet!");
                    return null;
                }
            }

            int percent = Integer.parseInt(saleData[0]);
            if (percent <= 0 || percent > 100){
                System.err.println("A leárazási százaléknak 0 és 100 közötti
értéknek kell lennie!");
                return null;
            }
        }

        // csomópontok tényleges létrehozása
        Element gameNode = dom.createElement("játék");
        Node nameNode = createTextNode("játékNeve", name),
            priceNode = createFloatNode("ár", price),
            publishedDateNode = createDateNode("kiadásDátuma", publishedDate,
"date");

        try{
            gameNode.setAttribute("JID", JID);
            if (baseGame != null) gameNode.setAttribute("alapjáték", baseGame);
            gameNode.setAttribute("fejlesztő", developer);
            gameNode.setAttribute("kiadó", publisher);
            gameNode.appendChild(nameNode);
            gameNode.appendChild(priceNode);
            gameNode.appendChild(publishedDateNode);
        } catch (NullPointerException npe){
            return null;
        }
    }
}

```

```

    }

    // műfajok csomópont összeállítása
    if (genreIndices != null){
        Node genresNode = dom.createElement("műfajok");
        for (int genreIndex : genreIndices){
            Node genreNode = createGenreNode(genreIndex);
            if (genreNode == null) return null;
            genresNode.appendChild(genreNode);
        }
        gameNode.appendChild(genresNode);
    }

    // leárazás csomópont összeállítása
    // ami nincs ellenőrizve:
    //     - a leárazás ne legyen máris lejárva (ezek később törölhetőek)
    //     - a leárazás kezdési ideje ne legyen később, mint a befejezési
    idő
    if (saleData != null){
        Node saleDataNode = dom.createElement("leárazás");
        Node salePercent = createIntNode("százalék",
Integer.parseInt(saleData[0]));
        if (salePercent == null){
            System.err.println("Leárazási százalék nem megfelelően volt
megadva!");
            return null;
        }
        Node saleStart = createDateNode("kezdés", saleData[1], "datetime");
        if (saleStart == null){
            System.err.println("Leárazás kezdési időpontja nem megfelelően
volt megadva!");
            return null;
        }
        Node saleEnd = createDateNode("vége", saleData[2], "datetime");
        if (saleEnd == null){
            System.err.println("Leárazás végződési időpontja nem megfelelően
volt megadva!");
            return null;
        }
        saleDataNode.appendChild(salePercent);
        saleDataNode.appendChild(saleStart);
        saleDataNode.appendChild(saleEnd);
        gameNode.appendChild(saleDataNode);
    }

```

```

        // csak akkor adjuk hozzá az ID- és játéklistához, ha valóban sikerült
        létrehozni
        IDs.add(JID);
        games.put(JID, gameNode);

        return gameNode;
    }

```

```

    // teljesítmény csomópont létrehozása
    // nem kötelező megadású: JID, description
    private static Node CreateAchievementNode(String TID, String JID, String
name, String description, String addDate){
        // ellenőrzések
        if (TID == null || TID.isEmpty() || name == null || addDate == null){
            System.err.println("Csak a JID és a teljesítmény leírása maradhat
üresen!");
            return null;
        }
        if (IDs.contains(TID)){
            System.err.println("(" + TID + ") Már van ilyen ID-jü
teljesítmény!");
            return null;
        }
        if (JID != null && games.get(JID) == null){
            System.err.println("(" + TID + ") A teljesítmény előtt a játékot kell
létrehozni!");
            return null;
        }

        // csomópontok létrehozása
        Element achievementNode = dom.createElement("teljesítmény");
        Node nameNode = createTextNode("teljesítményNeve", name),
            addDateNode = createDateNode("teljesítményHozzáadásánakDátuma",
addDate, "date");

        try{
            achievementNode.setAttribute("TID", TID);
            if (JID != null) achievementNode.setAttribute("T_játék", JID);
            achievementNode.appendChild(nameNode);
            if (description != null){
                Node descriptionNode = createTextNode("teljesítményLeírása",
description);
                achievementNode.appendChild(descriptionNode);
            }
        }
    }

```

```

        achievementNode.appendChild(addDateNode);
    } catch (NullPointerException npe){
        return null;
    }

    // csak akkor adjuk hozzá az ID- és teljesítménylistához, ha valóban
    // sikerült létrehozni
    IDs.add(TID);
    achievements.put(TID, achievementNode);

    return achievementNode;
}

```

```

// nyelv csomópont létrehozása
// minden kötelező megadású
private static Node CreateLanguageNode(String NyID, String language){
    // ellenőrzések
    if (NyID == null || NyID.isEmpty() || language == null ||
    language.isEmpty()){
        System.err.println("A nyelv csomópontnak kötelező megadni az ID-jét
és a nevét is!");
        return null;
    }
    if (IDs.contains(NyID)){
        System.err.println("(" + NyID + ") Már van ilyen azonosítójú
nyelv!");
        return null;
    }

    // csomópont létrehozása
    Element languageNode = dom.createElement("nyelv");
    languageNode.setAttribute("NyID", NyID);
    languageNode.setTextContent(language);

    IDs.add(NyID);
    languages.put(NyID, languageNode);
    return languageNode;
}

```

```

// támogatottNyelv csomópont létrehozása
// minden kötelező megadású
private static Node CreateSupportedLanguageNode(String JID, String NyID,
boolean subtitles, boolean sound){
    // ellenőrzések

```

```

        if (JID == null || NyID == null){
            System.err.println("A támogatott nyelveknél mindent meg kell adni!");
            return null;
        }
        if (!(subtitles || sound)){
            System.err.println("(" + NyID + ") Ha nem támogat sem feliratot, sem
hangot egy nyelv esetén, akkor a játék nem támogatja a nyelvet!");
            return null;
        }
        if (games.get(JID) == null){
            System.err.println("(" + JID + ", " + NyID + ") Először létre kell
hozni a játékot!");
            return null;
        }
        if (languages.get(NyID) == null){
            System.err.println("(" + JID + ", " + NyID + ") Először létre kell
hozni a nyelvet!");
            return null;
        }

        // csomópontok létrehozása
        Element supportedLanguageNode = dom.createElement("támogatottNyelv");
        Node subtitlesNode = createBoolNode("felirat", subtitles),
            soundNode = createBoolNode("hang", sound);

        try{
            supportedLanguageNode.setAttribute("TNY_játék", JID);
            supportedLanguageNode.setAttribute("TNY_nyelv", NyID);
            supportedLanguageNode.appendChild(subtitlesNode);
            supportedLanguageNode.appendChild(soundNode);
        } catch (NullPointerException npe){
            return null;
        }

        return supportedLanguageNode;
    }

```

```

    // felhasználó csomópont létrehozása
    // nem kötelező megadású: displayName
    private static Node CreateUserNode(String FID, String username, String
displayName, String registrationDate, String birthdate){
        // ellenőrzések
        if (FID == null || FID.isEmpty() || username == null || registrationDate
== null || birthdate == null){

```

```

        System.err.println("Csak a megjelenített név maradhat üresen!");
        return null;
    }
    if (IDs.contains(FID)){
        System.err.println("(" + FID + ") Már van ilyen ID-jű felhasználó!");
    }

    // csomópontok létrehozása
    Element userNode = dom.createElement("felhasználó");
    Node usernameNode = createTextNode("felhasználónév", username),
        registrationDateNode = createDateNode("fiókLétrehozásánakDátuma",
registrationDate, "date"),
        birthdateNode = createDateNode("születésiDátum", birthdate, "date");

    try{
        userNode.setAttribute("FID", FID);
        userNode.appendChild(usernameNode);
        if (displayName != null){
            Node displayNameNode = createTextNode("megjelentNév",
displayName);
            userNode.appendChild(displayNameNode);
        }
        userNode.appendChild(registrationDateNode);
        userNode.appendChild(birthdateNode);
    } catch (NullPointerException npe){
        return null;
    }

    // ha sikerült létrehozni, hozzáadjuk az ID- és felhasználólistához
    IDs.add(FID);
    users.put(FID, userNode);

    return userNode;
}

```

```

// barát csomópont létrehozása
// minden kötelező megadású
private static Node CreateFriendNode(String FID1, String FID2){
    // ellenőrzések
    if (FID1 == null || FID1.isEmpty() || FID2 == null || FID2.isEmpty()){
        System.err.println("A barát csomópontnál mindkét felhasználó ID-jét
meg kell adni!");
        return null;
    }
}

```

```

        if (users.get(FID1) == null || users.get(FID2) == null){
            System.err.println("(" + FID1 + ", " + FID2 + ") Először a
felhasználókat kell létrehozni!");
            return null;
        }
        if (friends.contains(FID1 + FID2) || friends.contains(FID2 + FID1)){
            System.err.println("(" + FID1 + ", " + FID2 + ") A felhasználók
barátsága már be lett jegyezve!");
            return null;
        }

        // csomópont létrehozása
        Element friendNode = dom.createElement("barát");
        friendNode.setAttribute("felhasználó1", FID1);
        friendNode.setAttribute("felhasználó2", FID2);

        friends.add(FID1 + FID2);
        friends.add(FID2 + FID1);
        return friendNode;
    }

```

```

        // hozzáadottJáték csomópont létrehozása
        // minden kötelező megadású
        private static Node CreateAddedGameNode(String JID, String FID, String
addDate, String state, int playtime){
            // ellenőrzések
            if (JID == null || FID == null || addDate == null || state == null)
            {
                System.err.println("Hozzáadott játék esetén minden adatot meg kell
adni!");
                return null;
            }
            if (state.equals("megvéve") && playtime < 0){
                System.err.println("A játszási idő nem lehet negatív");
                return null;
            }
            if (games.get(JID) == null){
                System.err.println("(" + JID + ", " + FID + ") Először létre kell
hozni a játékot!");
                return null;
            }
            if (users.get(FID) == null){
                System.err.println("(" + JID + ", " + FID + ") Először létre kell
hozni a felhasználót!");
            }

```

```

        return null;
    }

    // csomópontok létrehozása
    Element addedGameNode = dom.createElement("hozzáadottJáték");
    Node addDateNode = createDateNode("játékHozzáadásánakDátuma", addDate,
    "date"),
        stateNode = createStateNode(state);

    try{
        addedGameNode.setAttribute("HJ_játék", JID);
        addedGameNode.setAttribute("HJ_felhasználó", FID);
        addedGameNode.appendChild(addDateNode);
        addedGameNode.appendChild(stateNode);
    } catch (NullPointerException npe){
        return null;
    }

    if (state.equals("megvéve")){
        Node playtimeNode = createIntNode("játékszásIdőtartama", playtime);
        if (playtimeNode == null) return null;
        addedGameNode.appendChild(playtimeNode);
    }

    return addedGameNode;
}

```

```

// elértTeljesítmény csomópont létrehozása
// minden kötelező megadású
private static Node CreateAchievedAchievementsNode(String FID, String TID,
String achieveDate){
    // ellenőrzések
    if (FID == null || TID == null || achieveDate == null){
        System.err.println("Az elértTeljesítmény csomópontnál semmi sem
maradhat üresen!");
        return null;
    }
    if (users.get(FID) == null){
        System.err.println("(" + FID + ", " + TID + ") Először létre kell
hozni a felhasználót!");
        return null;
    }
    if (achievements.get(TID) == null){

```



```
        System.err.println("(" + FID + ", " + TID + ") Először létre kell  
hozni a teljesítményt!");  
        return null;  
    }  
  
    // csomópontok létrehozása  
    Element achievedAchievementNode = dom.createElement("elértTeljesítmény");  
    Node achieveDateNode = createDateNode("teljesítményElérésénekDátuma",  
achieveDate, "date");  
  
    try{  
        achievedAchievementNode.setAttribute("teljesítő", FID);  
        achievedAchievementNode.setAttribute("teljesítettTeljesítmény", TID);  
        achievedAchievementNode.appendChild(achieveDateNode);  
    } catch (NullPointerException npe){  
        return null;  
    }  
  
    return achievedAchievementNode;  
}
```