

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Teremfoglalás

Készítette: **Szabó Alexandra Marianna**

Neptunkód: **R65UKG**

Dátum: 2023.12.03.

Tartalomjegyzék:

Bevezetés	3
A feladat leírása	3
1. feladat	4
1a) Az adatbázis ER modell tervezése	4
1b) Az adatbázis konvertálása XDM modellre	5
1c) Az XDM modell alapján XML dokumentum készítése	5
1d) Az XML dokumentum alapján XMLSchema készítése	14
2. feladat	18
2a) adatolvasás	18
2b) adatmódosítás.....	19
2c) adatlekérdezés	21
2d) adatírás.....	25

Bevezetés

A feladat leírása

A beadandóm témája egy adatbázis, mely segítségével tárolni tudjuk, hogy az egyetemen melyik terem foglalt és melyik terem szabad. Ha foglalt, akkor ki foglalta le és meddig lesz foglalva, vagy éppen milyen órára foglalták le az adott termet. A terem adatait például terem férőhelyét is letudjuk kérdezni belőle vagy hogy az adott terembe van-e működőképes vetítő. Akár azt is megtudhatjuk belőle, hogy hány hallgatónak van adott órája és ahhoz mérten termet választani. Emellett a karokhoz tartozó oktatókat vagy a karok dékánjait és az ő adataikat is megtekinthetjük.

Ezt az adatbázist XML segítségével valósítottam meg. Ehhez készült egy ER modell és egy XDM modell is, amin láthatjuk az XML fájlunk vázát. Majd ennek segítségével létrehoztam egy XML fájlt, és ahhoz egy XMLSchema-t.

A feladat második felében Java programozási nyelv segítségével műveleteket, lekérdezéseket hajtottam végre rajta.

-Beolvastam az XML fájlt és kiírtam konzolra.

-Beolvastam az XML fájlt, módosításokat hajtottam rajta végbe, majd kiírtam a konzolra és egy fájlba

-Beolvastam az XML fájlt lekérdezéseket írtam a fájlhoz és az eredményeket kiírtam a konzolra

-Beolvastam az XML fájlt és kiírtam egy másik fájlba.

Számomra kézenfekvő volt egyetemistaként ezt a témát választani, mivel napiszinten az egyetemen vagyok különböző órákon, különböző termekben. Egy ilyen adatbázis nélkül elég nagy zűrzavar lenne a termekkel és elég bonyolult lenne a terem kiosztás.

1. feladat

1a) Az adatbázis ER modell tervezése

Az ER modellben láthatjuk a különböző egyedeket és az egyedek tulajdonságait, adatait.

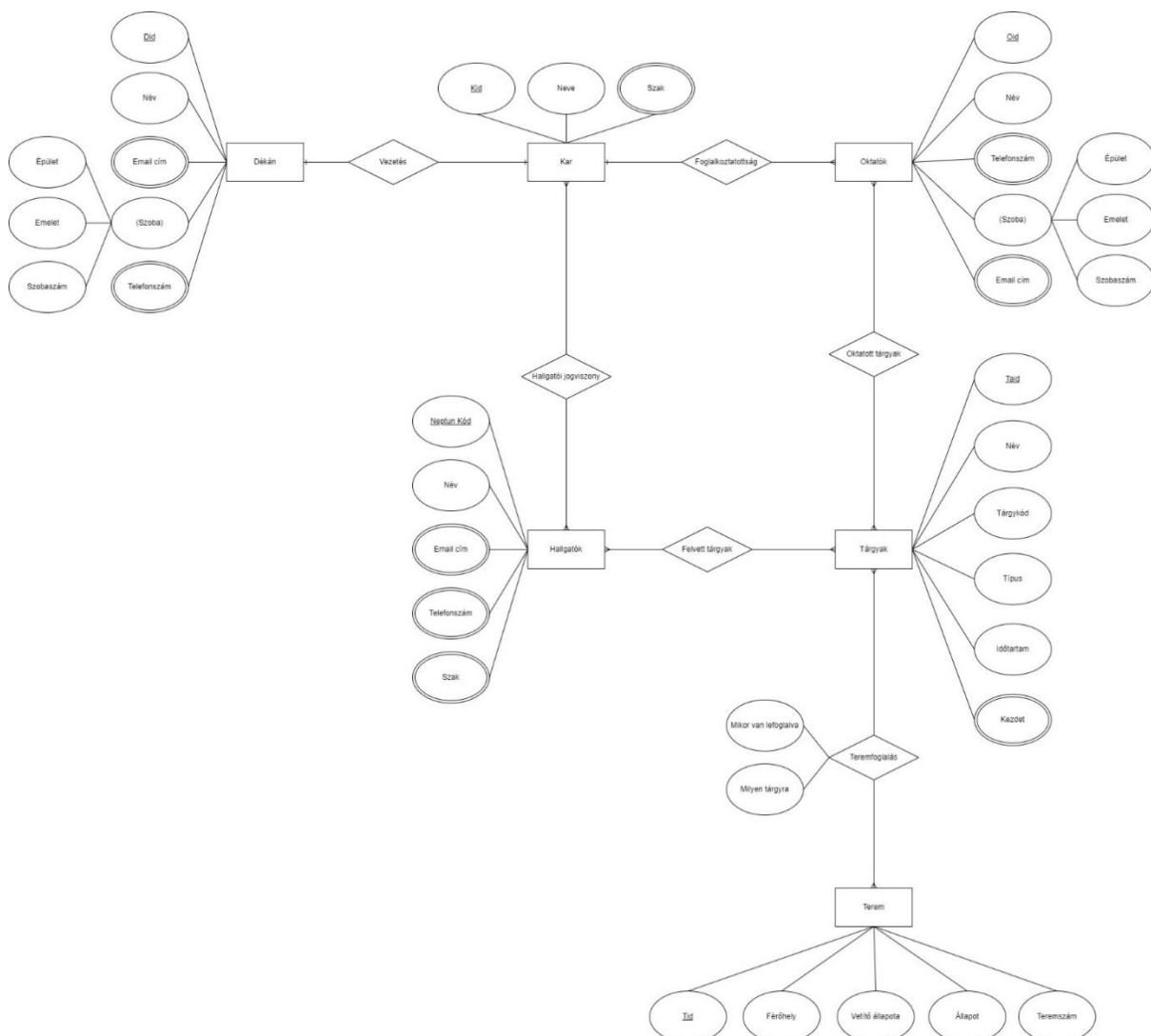
Ahogy láthatjuk a Kar és a Dékán egyedek között egy-egy kapcsolat van, mivel egy karnak csak egy dékánja lehet.

A Kar és az Oktatók egyedek között egy-több kapcsolat van, mivel egy oktató csak egy karhoz tartozhat, de egy karnak több oktatója is van.

A Kar és a Hallgatók egyedek között több-több kapcsolat van, mivel egy karhoz több hallgató is tartozik, és egy hallgató több szakot is elvégezhet egyszerre az egyetemen, így több karhoz is tartozhat egyszerre.

Az Oktatók és a Tárgyak egyedek között is több-több kapcsolat van, mivel egy tárgyat több oktató is taníthat, és egy oktató több tárgyat is taníthat egy félév során.

A Tárgyak és a Terem egyedek között is több-több kapcsolat van, mivel egy terembe több órát is tarthatnak egy hétsorán, és egy adott óra több teremben is lehet. Például az elméleti órák inkább előadókban vannak általában, addig a gyakorlati órák inkább laborokban, vagy kisebb férőhelyű termekben szoktak megvalósulni.

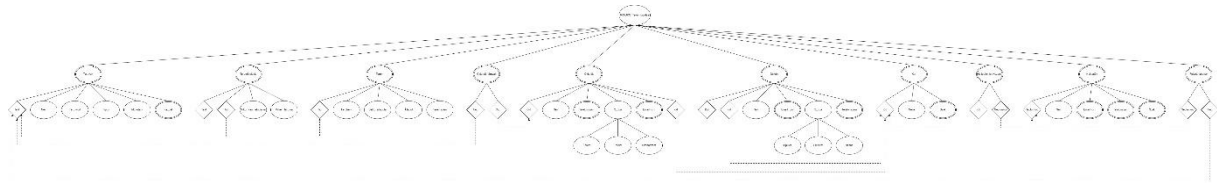


1b) Az adatbázis konvertálása XDM modellre

A konvertálás során lett egy gyöker elemünk R65UKG_Teremfoglalás néven.

A gyöker elem alatt helyezkednek az ER modellben lévő egyedek, és a konvertálás során több-több kapcsolatból keletkezett kapcsoló táblák, amik többértékű tulajdonságok.

Minden tábla alatt láthatjuk az elemeiket és attribútumaikat, helyi- és idegen kulcsaikat.



1c) Az XDM modell alapján XML dokumentum készítése

Az xml dokumentumot feltöltöttem az általam ebben a félévben tanult tárgyak adataival, ebben a félévben engem tanító oktatók adataival, a dékánok adataival, néhány szaktársam adataival, és néhány kezdetleges, valós adattal.

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

<R65UKG_Teremfoglalás xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaR65UKG.xsd">
  <!--Tárgyak tábla adatai-->
  <Tárgyak TaId="01">
    <Név>Idegen nyelv 1.</Név>
    <Tárgykód>MIGENANNY1</Tárgykód>
    <Típus>elmélet</Típus>
    <Időtartam>2</Időtartam>
    <Kezdet>Hétfő 12</Kezdet>
  </Tárgyak>
  <Tárgyak TaId="02">
    <Név>WEB technológiák</Név>
    <Tárgykód>GEIAL331-B</Tárgykód>
    <Típus>elmélet és gyakorlat</Típus>
    <Időtartam>2</Időtartam>
    <Kezdet>Hétfő 12</Kezdet>
    <Kezdet>Hétfő 14</Kezdet>
  </Tárgyak>
  <Tárgyak TaId="03">
    <Név>Mesterséges intelligencia alapok</Név>
    <Tárgykód>GEIAK130-B</Tárgykód>
    <Típus>elmélet és gyakorlat</Típus>
    <Időtartam>2</Időtartam>
    <Kezdet>Kedd 10</Kezdet>
    <Kezdet>Csütörtök 10</Kezdet>
  </Tárgyak>
  <Tárgyak TaId="04">
    <Név>Adatkezelés XML-ben</Név>
    <Tárgykód>GEIAL332-B</Tárgykód>
    <Típus>elmélet és gyakorlat</Típus>
```

```

<Targyak Taid="04">
  <Név>Adatkezelés XML-ben</Név>
  <Targykód>GEIAL332-B</Targykód>
  <Típus>elmélet és gyakorlat</Típus>
  <Időtartam>2</Időtartam>
  <Kezdet>Kedd 12</Kezdet>
  <Kezdet>Szerda 12</Kezdet>
</Targyak>
<Targyak Taid="05">
  <Név>Algoritmusok és vizsgálatuk</Név>
  <Targykód>GEIAK234-B</Targykód>
  <Típus>elmélet</Típus>
  <Időtartam>4</Időtartam>
  <Kezdet>Kedd 14</Kezdet>
</Targyak>
<Targyak Taid="06">
  <Név>Számítógépi hálózatok</Név>
  <Targykód>GEIAL304-B</Targykód>
  <Típus>elmélet</Típus>
  <Időtartam>2</Időtartam>
  <Kezdet>Szerda 8</Kezdet>
  <Kezdet>Csütörtök 12</Kezdet>
</Targyak>
<Targyak Taid="07">
  <Név>Vállalati információs rendszerek fejlesztése</Név>
  <Targykód>GEIAL315-B</Targykód>
  <Típus>elmélet</Típus>
  <Időtartam>2</Időtartam>
  <Kezdet>Szerda 14</Kezdet>
  <Kezdet>Szerda 18</Kezdet>
</Targyak>
<Targyak Taid="08">
  <Név>Játék prototípusok</Név>
  <Targykód>GEIAK310-B</Targykód>
  <Típus>gyakorlat</Típus>
  <Időtartam>4</Időtartam>
  <Kezdet>Csütörtök 16</Kezdet>
</Targyak>

<!--Teremfoglalás tábla adatai-->
<Teremfoglalás REF_Taid="08" REF_Tid="11">
  <Mikor_van_lefoglalva>Csütörtök 16</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Játék prototípusok</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="07" REF_Tid="08">
  <Mikor_van_lefoglalva>Szerda 18</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Vállalati információs rendszerek</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="07" REF_Tid="07">
  <Mikor_van_lefoglalva>Szerda 14</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Vállalati információs rendszerek</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="06" REF_Tid="10">
  <Mikor_van_lefoglalva>Csütörtök 12</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Számítógépi hálózatok</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="06" REF_Tid="06">
  <Mikor_van_lefoglalva>Szerda 8</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Számítógépi hálózatok</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="05" REF_Tid="05">
  <Mikor_van_lefoglalva>Kedd 14</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Algoritmusok és vizsgálatuk</Milyen_tárgyra>

```

```

<Teremfoglalás REF_Taid="05" REF_Tid="05">
  <Mikor_van_lefoglalva>Kedd 14 </Mikor_van_lefoglalva>
  <Milyen_tárgyra>Algoritmusok és vizsgálatuk</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="04" REF_Tid="04">
  <Mikor_van_lefoglalva>Kedd 12</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Adatkezelés XML-ben</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="04" REF_Tid="07">
  <Mikor_van_lefoglalva>Szerda 12</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Adatkezelés XML-ben</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="03" REF_Tid="04">
  <Mikor_van_lefoglalva>Kedd 10</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Mesterséges intelligencia alapok</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="03" REF_Tid="09">
  <Mikor_van_lefoglalva>Csütörtök 10</Mikor_van_lefoglalva>
  <Milyen_tárgyra>Mesterséges intelligencia alapok</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="02" REF_Tid="03">
  <Mikor_van_lefoglalva>Hétfő 16</Mikor_van_lefoglalva>
  <Milyen_tárgyra>WEB technológiák</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="02" REF_Tid="02">
  <Mikor_van_lefoglalva>Hétfő 14</Mikor_van_lefoglalva>
  <Milyen_tárgyra>WEB technológiák</Milyen_tárgyra>
</Teremfoglalás>
<Teremfoglalás REF_Taid="01" REF_Tid="01">
  <REF_Taid>01</REF_Taid>
  <REF_Tid>01</REF_Tid>
  <Mikor_van_lefoglalva>Hétfő 12</Mikor_van_lefoglalva>

```

```

<!--Terem tábla adatai-->
<Terem Tid="01">
  <Férőhely>28</Férőhely>
  <Vetítő_állapota>Nincs</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>A5/202</Teremszám>
</Terem>
<Terem Tid="02">
  <Férőhely>100</Férőhely>
  <Vetítő_állapota>Nincs</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>30-as előadó</Teremszám>
</Terem>
<Terem Tid="03">
  <Férőhely>-</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>Inf/202</Teremszám>
</Terem>
<Terem Tid="04">
  <Férőhely>300</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>32-es előadó</Teremszám>
</Terem>
<Terem Tid="05">
  <Férőhely>54</Férőhely>
  <Vetítő_állapota>Nincs</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>A1/320</Teremszám>
</Terem>
<Terem Tid="06">

```

```
<Terem Tid="06">
  <Férőhely>312</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>2-es előadó</Teremszám>
</Terem>
<Terem Tid="07">
  <Férőhely>-</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>Inf/101</Teremszám>
</Terem>
<Terem Tid="08">
  <Férőhely>-</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>Inf/103</Teremszám>
</Terem>
<Terem Tid="09">
  <Férőhely>504</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>1-es előadó</Teremszám>
</Terem>
<Terem Tid="10">
  <Férőhely>384</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>3-as előadó</Teremszám>
</Terem>
<Terem Tid="11">
  <Férőhely>-</Férőhely>
```

```
</Terem>
<Terem Tid="11">
  <Férőhely>-</Férőhely>
  <Vetítő_állapota>Van</Vetítő_állapota>
  <Állapot></Állapot>
  <Teremszám>Inf/15</Teremszám>
</Terem>

<!--Oktatott tárgyak (kapcsoló tábla) tábla adatai-->
<Oktatott_tárgyak REF_Taid="08" REF_Oid="08"></Oktatott_tárgyak>
<Oktatott_tárgyak REF_Taid="07" REF_Oid="07"></Oktatott_tárgyak>
<Oktatott_tárgyak REF_Taid="06" REF_Oid="06"></Oktatott_tárgyak>
<Oktatott_tárgyak REF_Taid="05" REF_Oid="05"></Oktatott_tárgyak>
<Oktatott_tárgyak REF_Taid="04" REF_Oid="04"></Oktatott_tárgyak>
<Oktatott_tárgyak REF_Taid="03" REF_Oid="03"></Oktatott_tárgyak>
<Oktatott_tárgyak REF_Taid="02" REF_Oid="02"></Oktatott_tárgyak>
<Oktatott_tárgyak REF_Taid="01" REF_Oid="01"></Oktatott_tárgyak>

<!--Oktatók tábla adatai-->
<Oktatók Oid="01">
  <Név>Pásztor Krisztina</Név>
  <Telefonszám>+36 46 565-11/17-71</Telefonszám>
  <Szoba>
    <Épület>A1</Épület>
    <Emelet>3</Emelet>
    <Szobaszám>303/a</Szobaszám>
  </Szoba>
  <Email_cím>krisztina.pasztor@uni-miskolc.hu</Email_cím>
</Oktatók>
<Oktatók Oid="02">
  <Név>Agárdi Anita</Név>
  <Telefonszám>17-56</Telefonszám>
  <Szoba>
```



```
</Oktatók>
<Oktatók Oid="02">
  <Név>Agárdi Anita</Név>
  <Telefonszám>17-56</Telefonszám>
  <Szoba>
    <Épület>Informatikai épület</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>111</Szobaszám>
  </Szoba>
  <Email_cím>anita.agardi@uni-miskolc.hu</Email_cím>
</Oktatók>
<Oktatók Oid="03">
  <Név>Fazekas Levente</Név>
  <Telefonszám>-</Telefonszám>
  <Szoba>
    <Épület>Informatikai épület</Épület>
    <Emelet>fsz.</Emelet>
    <Szobaszám>8/b</Szobaszám>
  </Szoba>
  <Email_cím>aitflex@uni-miskolc.hu</Email_cím>
</Oktatók>
<Oktatók Oid="04">
  <Név>Bednarik László</Név>
  <Telefonszám>17-58</Telefonszám>
  <Szoba>
    <Épület>Informatikai épület</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>107/b</Szobaszám>
  </Szoba>
  <Email_cím>bednarik.laszlo@iit.uni-miskolc.hu</Email_cím>
</Oktatók>
<Oktatók Oid="05">
  <Név>Házy Attila</Név>
```

```
<Oktatók Oid="05">
  <Név>Házy Attila</Név>
  <Telefonszám>+36 46 565 111/1890</Telefonszám>
  <Telefonszám>+36 30 466 8317</Telefonszám>
  <Szoba>
    <Épület>A4</Épület>
    <Emelet>3</Emelet>
    <Szobaszám>328</Szobaszám>
  </Szoba>
  <Email_cím>matha@uni-miskolc.hu</Email_cím>
</Oktatók>
<Oktatók Oid="06">
  <Név>Kovács Szilveszter</Név>
  <Telefonszám>+36 46 565 111/2107</Telefonszám>
  <Szoba>
    <Épület>Informatikai épület</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>107/a</Szobaszám>
  </Szoba>
  <Email_cím>kovacsszilveszter@iit.uni-miskolc.hu</Email_cím>
</Oktatók>
<Oktatók Oid="07">
  <Név>Sasvári Péter</Név>
  <Telefonszám>+36 46 565 111/2111</Telefonszám>
  <Szoba>
    <Épület>Informatikai épület</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>109</Szobaszám>
  </Szoba>
  <Email_cím>sasvaripeter@iit.uni-miskolc.hu</Email_cím>
</Oktatók>
<Oktatók Oid="08">
```

```
<Oktatók Oid="08">
  <Név>Kiss Áron</Név>
  <Telefonszám>-</Telefonszám>
  <Szoba>
    <Épület>Informatikai épület</Épület>
    <Emelet>fsz.</Emelet>
    <Szobaszám>8/b</Szobaszám>
  </Szoba>
  <Email_cím>aikiss@uni-miskolc.hu</Email_cím>
</Oktatók>

<!--Foglalkoztatottság (kapcsoló tábla) tábla adatai-->
<Foglalkoztatottság REF_Oid="01" REF_Kid="03"></Foglalkoztatottság>
<Foglalkoztatottság REF_Oid="02" REF_Kid="03"></Foglalkoztatottság>
<Foglalkoztatottság REF_Oid="03" REF_Kid="03"></Foglalkoztatottság>
<Foglalkoztatottság REF_Oid="04" REF_Kid="03"></Foglalkoztatottság>
<Foglalkoztatottság REF_Oid="05" REF_Kid="03"></Foglalkoztatottság>
<Foglalkoztatottság REF_Oid="06" REF_Kid="03"></Foglalkoztatottság>
<Foglalkoztatottság REF_Oid="07" REF_Kid="03"></Foglalkoztatottság>
<Foglalkoztatottság REF_Oid="08" REF_Kid="03"></Foglalkoztatottság>

<!--Dékán tábla adatai-->
<Dékán Did="01" REF_Kid="01">
  <Név>Prof. Dr. Mucsi Gábor</Név>
  <Email_cím>rekbdhiv@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>A4</Épület>
    <Emelet>fsz.</Emelet>
    <Szobaszám>26</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 565 111/1020</Telefonszám>
</Dékán>
```

```
<Dékán Did="02" REF_Kid="02">
  <Név>Prof. Dr. Palotás Árpád Bence</Név>
  <Email_cím>arpad.palotas@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>C1</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>108</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 565 111/1023</Telefonszám>
  <Telefonszám>+36 46 565 111/2011</Telefonszám>
</Dékán>
<Dékán Did="03" REF_Kid="03">
  <Név>Dr. Siménfalvi Zoltán</Név>
  <Email_cím>gkdh5@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>C1</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>-</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 565 111/2352</Telefonszám>
</Dékán>
<Dékán Did="04" REF_Kid="04">
  <Név>Prof. Dr. Csák Csilla</Név>
  <Email_cím>jogdekan@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>A6</Épület>
    <Emelet>fsz.</Emelet>
    <Szobaszám>4</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 565 111/1350</Telefonszám>
  <Telefonszám>+36 46 565 111/2006</Telefonszám>
</Dékán>
```

```

<Dékán Did="05" REF_Kid="05">
  <Név>Veresné Dr. Somosi Mariann</Név>
  <Email_cím>gazddek@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>A4</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>115</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 565 111/1005</Telefonszám>
  <Telefonszám>+36 46 565 111/1644</Telefonszám>
</Dékán>
<Dékán Did="06" REF_Kid="06">
  <Név>Dr. Illésné Dr. Kovács Mária</Név>
  <Email_cím>boldek@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>A4</Épület>
    <Emelet>4</Emelet>
    <Szobaszám>422</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 565 111/1497</Telefonszám>
</Dékán>
<Dékán Did="07" REF_Kid="07">
  <Név>Dr. Kiss-Tóth Emőke</Név>
  <Email_cím>rekefk@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>B3/B4</Épület>
    <Emelet>1</Emelet>
    <Szobaszám>-</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 366 560</Telefonszám>
</Dékán>
<Dékán Did="08" REF_Kid="08">

```

```

<Dékán Did="08" REF_Kid="08">
  <Név>Dr. Papp Sándor</Név>
  <Email_cím>sandor.papp@uni-miskolc.hu</Email_cím>
  <Szoba>
    <Épület>Bartók tér 1. / Zenepalota</Épület>
    <Emelet>-</Emelet>
    <Szobaszám>-</Szobaszám>
  </Szoba>
  <Telefonszám>+36 46 321 711</Telefonszám>
</Dékán>

<!--Kar tábla adatai-->
<Kar Kid="01">
  <Név>Műszaki Föld- és Környezettudományi Kar</Név>
  <Szak>Földrajz</Szak>
  <Szak>Környezetmérnöki</Szak>
  <Szak>Műszaki földtudományi</Szak>
</Kar>
<Kar Kid="02">
  <Név>Anyag- és Vegyészmérnöki Kar</Név>
  <Szak>Anyagmérnöki</Szak>
  <Szak>Vegyészmérnöki</Szak>
</Kar>

```

```
<Kar Kid="03">
  <Név>Gépészmérnöki és Informatikai Kar</Név>
  <Szak>Energetikai mérnöki</Szak>
  <Szak>Gazdaságinformatikus</Szak>
  <Szak>Gépészmérnöki</Szak>
  <Szak>Ipari termék- és formatervező mérnöki</Szak>
  <Szak>Járműmérnöki</Szak>
  <Szak>Logisztikai mérnöki</Szak>
  <Szak>Mechatronikai mérnöki</Szak>
  <Szak>Mérnök-informatikus</Szak>
  <Szak>Műszaki menedzser</Szak>
  <Szak>Programtervező informatikus</Szak>
  <Szak>Üzemmérnök-informatikus</Szak>
  <Szak>Villamosmérnöki</Szak>
</Kar>
<Kar Kid="04">
  <Név>Állam- és Jogtudományi Kar</Név>
  <Szak>Igazgatásszervező</Szak>
  <Szak>Igazságügyi igazgatási</Szak>
  <Szak>Közigazgatás-szervező</Szak>
  <Szak>Munkaügyi és társadalombiztosítási igazgatási</Szak>
</Kar>
<Kar Kid="05">
  <Név>Gazdaságtudományi Kar</Név>
  <Szak>Emberi erőforrások</Szak>
  <Szak>Gazdálkodási és menedzsment</Szak>
  <Szak>Kereskedelem és marketing</Szak>
  <Szak>Közgazdasági adatelemzés</Szak>
  <Szak>Nemzetközi gazdálkodás</Szak>
  <Szak>Pénzügy és számvitel</Szak>
  <Szak>Sportszervezés</Szak>
  <Szak>Sport- és rekreációs szervezés</Szak>
  <Szak>Turizmus-vendéglátás</Szak>
</Kar>
<Kar Kid="06">
  <Név>Bölcsészeti és Társadalomtudományi Kar</Név>
  <Szak>Anglisztika</Szak>
  <Szak>Germanisztika</Szak>
  <Szak>Gyógyypedagógia</Szak>
  <Szak>Kommunikáció- és médiatudomány</Szak>
  <Szak>Kulturális antropológia</Szak>
  <Szak>Magyar</Szak>
  <Szak>Nemzetközi tanulmányok</Szak>
  <Szak>Pedagógia</Szak>
  <Szak>Politológia</Szak>
  <Szak> Régészet</Szak>
  <Szak>Szabad bölcsészet</Szak>
  <Szak>Szociális munka</Szak>
  <Szak>Szociológia</Szak>
  <Szak>Történelem</Szak>
</Kar>
```

```

<Kar Kid="07">
  <Név>Egészségtudományi Kar</Név>
  <Szak>Ápolás és betegellátás</Szak>
  <Szak>Egészségügyi gondozás és prevenció</Szak>
  <Szak>Egészségügyi szervező</Szak>
  <Szak>Orvosi diagnosztikai analitikus</Szak>
</Kar>
<Kar Kid="08">
  <Név>Bartók Béla Zeneművészeti Kar</Név>
  <Szak>Alkotóművészet és muzikológia</Szak>
  <Szak>Előadóművészet</Szak>
</Kar>

<!--Hallgatói jogviszony (kapcsoló tábla) tábla adatai-->
<Hallgatói_jogviszony REF_Kid="03" REF_Neptunkód="R65UKG"></Hallgatói_jogviszony>
<Hallgatói_jogviszony REF_Kid="03" REF_Neptunkód="ULA7Z2"></Hallgatói_jogviszony>
<Hallgatói_jogviszony REF_Kid="03" REF_Neptunkód="MPW46D"></Hallgatói_jogviszony>

<!--Hallgatók tábla adatai-->
<Hallgatók Neptunkód="R65UKG">
  <Név>Szabó Alexandra Marianna</Név>
  <Email_cím>szaboalexandra2002@freemail.hu</Email_cím>
  <Email_cím>szaboalexandram18@gmail.com</Email_cím>
  <Telefonszám>06 30 285 5820</Telefonszám>
  <Szak>Programtervező informatikus</Szak>
</Hallgatók>
<Hallgatók Neptunkód="ULA7Z2">
  <Név>Keresztes Iulia</Név>
  <Email_cím>101juliakeresztes@gmail.com</Email_cím>
  <Telefonszám>-</Telefonszám>
  <Szak>Programtervező informatikus</Szak>
</Hallgatók>

<Hallgatók Neptunkód="MPW46D">
  <Név>Görög Krisztina Erzsébet</Név>
  <Email_cím>kgorog989@gmail.com</Email_cím>
  <Telefonszám>-</Telefonszám>
  <Szak>Programtervező informatikus</Szak>
</Hallgatók>

<!--Felvett tárgyak (kapcsoló tábla) tábla adatai-->
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="01"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="02"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="03"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="04"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="05"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="06"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="07"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="R65UKG" REF_Taid="08"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="ULA7Z2" REF_Taid="01"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="ULA7Z2" REF_Taid="03"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="ULA7Z2" REF_Taid="04"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="ULA7Z2" REF_Taid="05"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="ULA7Z2" REF_Taid="07"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="ULA7Z2" REF_Taid="08"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="MPW46D" REF_Taid="02"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="MPW46D" REF_Taid="03"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="MPW46D" REF_Taid="04"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="MPW46D" REF_Taid="05"></Felvett_tárgyak>
<Felvett_tárgyak REF_Neptunkód="MPW46D" REF_Taid="07"></Felvett_tárgyak>

</R65UKG_Teremfoglalás>

```

1d) Az XML dokumentum alapján XMLSchema készítése

Az XMLSchema elkészítése során először kigyűjtöttem az egyszerű elemeket. Majd az elsődlegeskulcsokat gyűjtöttem ki. Folytattam az összetett elemek kigyűjtésével. A legkisebb egységekkel kezdtem, majd mentem tovább a nagyobbakra, összetett elemek közül a gyökérelem vázát hoztam létre. Majd az elsődleges kulcsok és az idegen kulcsok hivatkozásait hoztam létre.

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

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

  <!--Egyszerű elemek -->

  <xs:element name="Név" type="xs:string"/>
  <xs:element name="Tárgykód" type="xs:string"/>
  <xs:element name="Típus" type="xs:string"/>
  <xs:element name="Időtartam" type="xs:integer"/>
  <xs:element name="Kezdet" type="xs:integer"/>
  <xs:element name="Mikor_van_lefoglalva" type="xs:string"/>
  <xs:element name="Milyen_tárgyra" type="xs:string"/>
  <xs:element name="Férőhely" type="xs:integer"/>
  <xs:element name="Vetítő állapota" type="xs:string"/>
  <xs:element name="Állapot" type="xs:string"/>
  <xs:element name="Teremszám" type="xs:string"/>
  <xs:element name="Telefonszám" type="xs:integer"/>
  <xs:element name="Épület" type="xs:string"/>
  <xs:element name="Emelet" type="xs:string"/>
  <xs:element name="Szobaszám" type="xs:integer"/>
  <xs:element name="Email_cím" type="xs:string"/>
  <xs:element name="Szak" type="xs:string"/>

  <!--Elsődleges kulcsok-->
  <xs:attribute name="Taid" type="xs:integer"/>
  <xs:attribute name="Tid" type="xs:integer"/>
  <xs:attribute name="Oid" type="xs:integer"/>
  <xs:attribute name="Kid" type="xs:integer"/>
  <xs:attribute name="Neptunkód" type="xs:string"/>
  <xs:attribute name="Did" type="xs:integer"/>

  <!--Összetett elemek-->

  <xs:element name="Szoba">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Épület"/>
        <xs:element ref="Emelet"/>
        <xs:element ref="Szobaszám"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Tárgyak">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Név"/>
        <xs:element ref="Tárgykód"/>
        <xs:element ref="Típus"/>
        <xs:element ref="Időtartam"/>
        <xs:element ref="Kezdet"/>
      </xs:sequence>
    </xs:complexType>
    <xs:attribute ref="Taid" use="required"/>
  </xs:element>
```

```

<xs:element name="Teremfoglalás">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Mikor_van_lefoglalva"/>
      <xs:element ref="Milyen_tárgyra"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute ref="REF_Taid" use="required"/>
  <xs:attribute ref="REF_Tid" use="required"/>
</xs:element>

<xs:element name="Terem">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Férőhely"/>
      <xs:element ref="Vetítő_állapota"/>
      <xs:element ref="Állapot"/>
      <xs:element ref="Teremszám"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute ref="REF_Tid" use="required"/>
</xs:element>

<xs:element name="Oktatott_tárgyak">
  <xs:attribute ref="REF_Taid" use="required"/>
  <xs:attribute ref="REF_Oid" use="required"/>
</xs:element>

```

```

<xs:element name="Oktatók">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Név"/>
      <xs:element ref="Telefonszám"/>
      <xs:element ref="Szoba"/>
      <xs:element ref="Email_cím"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute ref="Oid" use="required"/>
</xs:element>

<xs:element name="Foglalkoztatottság">
  <xs:attribute ref="REF_Oid" use="required"/>
  <xs:attribute ref="REF_Kid" use="required"/>
</xs:element>

<xs:element name="Dékán">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Név"/>
      <xs:element ref="Email_cím"/>
      <xs:element ref="Szoba"/>
      <xs:element ref="Telefonszám"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute ref="Did" use="required"/>
  <xs:attribute ref="REF_Kid" use="required"/>
</xs:element>

```



```

<xs:element name="Kar">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Név"/>
      <xs:element ref="Szak"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute ref="Kid" use="required"/>
</xs:element>

<xs:element name="Hallgatói_jogviszony">
  <xs:attribute ref="REF_Kid" use="required"/>
  <xs:attribute ref="REF_Neptunkód" use="required"/>
</xs:element>

<xs:element name="Hallgatók">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Név"/>
      <xs:element ref="Email_cím"/>
      <xs:element ref="Telefonszám"/>
      <xs:element ref="Szak"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute ref="Neptunkód" use="required"/>
</xs:element>

<xs:element name="Felvett_tárgyak">
  <xs:attribute ref="REF_Neptunkód" use="required"/>
  <xs:attribute ref="REF_Taid" use="required"/>
</xs:element>

<xs:element name="R65UKG_Teremfoglalás">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Tárgyak"/>
      <xs:element ref="Teremfoglalás"/>
      <xs:element ref="Terem"/>
      <xs:element ref="Oktatott_tárgyak"/>
      <xs:element ref="Oktatók"/>
      <xs:element ref="Foglalkoztatottság"/>
      <xs:element ref="Dékán"/>
      <xs:element ref="Kar"/>
      <xs:element ref="Hallgatói_jogviszony"/>
      <xs:element ref="Hallgatók"/>
      <xs:element ref="Felvett_tárgyak"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```



```

<!--Elsődleges kulcsok-->

<xs:key name="Neptunkód">
  <xs:selector xpath="Hallgatók"/>
  <xs:field xpath="@Neptunkód"/>
</xs:key>

<xs:key name="Kid">
  <xs:selector xpath="Kar"/>
  <xs:field xpath="@Kid"/>
</xs:key>

<xs:unique name="Did">
  <xs:selector xpath="Dékán"/>
  <xs:field xpath="@Did"/>
</xs:unique>

<xs:key name="Oid">
  <xs:selector xpath="Oktatók"/>
  <xs:field xpath="@Oid"/>
</xs:key>

<xs:key name="Tid">
  <xs:selector xpath="Terem"/>
  <xs:field xpath="@Tid"/>
</xs:key>

<xs:key name="Taid">
  <xs:selector xpath="Tárgyak"/>
  <xs:field xpath="@Taid"/>
</xs:key>

```

```

<!--Idegen kulcsok-->

<xs:keyref name="REF_Neptunkód" refer="Neptunkód">
  <xs:selector xpath="Hallgatók"/>
  <xs:field xpath="@REF_Neptunkód"/>
</xs:keyref>

<xs:keyref name="REF_Taid" refer="Taid">
  <xs:selector xpath="Tárgyak"/>
  <xs:field xpath="@REF_Taid"/>
</xs:keyref>

<xs:keyref name="REF_Kid" refer="Kid">
  <xs:selector xpath="Kar"/>
  <xs:field xpath="@REF_Kid"/>
</xs:keyref>

<xs:keyref name="REF_Oid" refer="Oid">
  <xs:selector xpath="Oktatók"/>
  <xs:field xpath="@REF_Oid"/>
</xs:keyref>

<xs:keyref name="REF_Tid" refer="Tid">
  <xs:selector xpath="Terem"/>
  <xs:field xpath="@REF_Tid"/>
</xs:keyref>

</xs:schema>

```

2. feladat

2a) adatolvasás

Beolvastam az XMLR65UKG.xml fájlt a DocumentBuilder és DocumentBuilderFactory előre definiált osztály segítségével. Majd megvizsgáltam egy if-else elágazás segítségével, hogy tényleg sikerült-e beolvasnom a fájlt. Majd kiíratam az elemeket kacsacsőrrel elválasztva az értékeikkel együtt és a hozzájuk tartozó attribútumokat pedig kötőjellel velük egy sorba.

```
package hu.domparse.R65UKG;

import java.io.File;
import java.io.IOException;

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

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

public class DOMReadR65UKG {

    Run | Debug
    public static void main(String[] args) {
        File xmlFile = new File(pathname: "XMLTaskR65UKG/DOMParserR65UKG/src/hu/domparse/R65UKG/XMLR65UKG.xml");
        Document doc = ReadFile(xmlFile);

        if(doc!=null){
            doc.getDocumentElement().normalize();
            System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        }
        else {
            System.out.println(x:"Document is null");
            System.exit(-1);
        }

        NodeList nodeList = doc.getDocumentElement().getChildNodes();
        String separation = "";
        listData(nodeList, separation);
    }

    public static Document ReadFile(File xmlFile){
        Document doc = null;

        try{
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            doc = dbBuilder.parse(xmlFile);
        } catch (ParserConfigurationException | SAXException | IOException exception) {
            exception.printStackTrace();
        }
        return doc;
    }
}
```

```

public static void listData(NodeList nodeList, String separation){
    separation += "\t";

    if(nodeList!=null){
        for(int i=0; i<nodeList.getLength(); i++){
            Node node = nodeList.item(i);
            if(node.getNodeType()==Node.ELEMENT_NODE && !node.getTextContent().trim().isEmpty()){
                System.out.print(separation + "< " + node.getNodeName() + " >: ");
                NamedNodeMap attribute = node.getAttributes();
                for(int j=0; j<attribute.getLength(); j++){
                    System.out.print(" - " + attribute.item(j));
                }
                System.out.println();
                NodeList newNodeList = node.getChildNodes();
                listData(newNodeList, separation);
            }
            else if(node instanceof Text){
                String value = node.getNodeValue().trim();
                if(value.isEmpty()){
                    continue;
                }
                System.out.println(separation + node.getTextContent());
            }
        }
    }
}

```

2b) adatmódosítás

Beolvastam az XMLR65UKG.xml fájlt a DocumentBuilder és DocumentBuilderFactory előre definiált osztály segítségével. Az első módosításomat külön általam írt függvény használata nélkül oldottam meg. Ott az egyes azonosítójú terem állapotát állítottam be foglaltra. A többi módosításhoz írtam külön függvényeket. Új oktató, új hallgatót és új termet adtam hozzá. Az egyes azonosítóval rendelkező oktatót átállítottam 13-as azonosítójúra. Majd kiírtattam a konzolra és le is mentettem a dokumentumot modify.xml nevű fájlba.

```

package hu.domp.parse.R65UKG;

import java.io.*;

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

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

public class DOMModifyR65UKG {

    Run | Debug
    public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException, TransformerException {
        File xmlFile = new File(pathname:"XMLTaskR65UKG/DOMParseR65UKG/src/hu/domp/parse/R65UKG/XMLR65UKG.xml");

        DocumentBuilderFactory dFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dFactory.newDocumentBuilder();

        Document document = dBuilder.parse(xmlFile);
        document.normalize();
    }
}

```

```

//beállítjuk a terem állapotát foglaltra
NodeList nodeList = document.getElementsByTagName(tagname:"Terem");
for(int i=0; i<nodeList.getLength(); i++){
    Node node = nodeList.item(i);
    if(node.getNodeType() == Node.ELEMENT_NODE){
        if(node.getAttributes().getNamedItem(name:"Tid").getTextContent().equals(anObject:"1")){
            NodeList childNodeList = node.getChildNodes();
            for(int j=0; j<childNodeList.getLength(); j++){
                Node childNode = childNodeList.item(j);
                if(childNode.getNodeName().equals(anObject:"Állapot")){
                    childNode.setTextContent(textContent:"Foglalt");
                }
            }
        }
    }
}
}
Element root = document.getDocumentElement();

//új oktatót adunk hozzá
Element oktato = (Element)document.getElementsByTagName(tagname:"Oktatók").item(index:0);
root.insertBefore(createOktato(document,Oid:"09",name:"Tomba Tamás",email:"tomba@iit.uni-miskolc.hu"), oktato);

//új hallgatót adunk hozzá
Element hallgato = (Element)document.getElementsByTagName(tagname:"Hallgatók").item(index:0);
root.insertBefore(createHallgato(document,neptuncode:"G2GNPO",name:"Tucsá Eszter Boglárka",email:"tucsá.eszter@gmail.com", specialise:"n

//új terem adunk hozzá
Element terem = (Element)document.getElementsByTagName(tagname:"Terem").item(index:0);
root.insertBefore(createTerem(document,Tid:"12",seat:"400",room_number:"37-es előadó"), terem);

```

```

//Az 1-es id-vel rendelkező oktatót módosítjuk 13-as id-re
modifyId(document, table:"Oktatók", id:"Oid", old_id:"01", new_id:"13");

//Lemenjük a módosított dokumentumot
saveDocument(document, filename:"modify.xml");
}

public static void saveDocument(Document document, String filename) throws TransformerException {
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transf = transformerFactory.newTransformer();

    transf.setOutputProperty(OutputKeys.ENCODING, value:"UTF-8");
    transf.setOutputProperty(OutputKeys.INDENT, value:"yes");
    transf.setOutputProperty(name:"{https://xml.apache.org/xslt}indent-amount", value:"2");

    DOMSource source = new DOMSource(document);

    File myFile = new File(filename);
    PrintStream p = null;
    try{
        p = new PrintStream(System.out, autoFlush:true,encoding:"UTF-8");
    }
    catch(Exception e){
        System.out.println(x:"Don't make it!");
        return;
    }

    StreamResult console = new StreamResult(p);

    StreamResult file = new StreamResult(myFile);

```

```

    transf.transform(source, console);
    transf.transform(source, file);
}

private static Node createOktato(Document document, String Oid, String name, String email){
    Element element = document.createElement(tagName:"Oktatók");

    element.setAttribute(name:"Oid", Oid);
    element.appendChild(createElement(document,name:"Név", name));
    element.appendChild(createElement(document,name:"Email_cím",email));

    return element;
}

private static Node createHallgato(Document document, String neptuncode, String name, String email, String specialise){
    Element element = document.createElement(tagName:"Hallgatók");

    element.setAttribute(name:"Neptunkód", neptuncode);
    element.appendChild(createElement(document,name:"Név", name));
    element.appendChild(createElement(document,name:"Email_cím",email));
    element.appendChild(createElement(document,name:"Szak",specialise));

    return element;
}

private static Node createTerem(Document document, String Tid, String seat, String room_number){
    Element element = document.createElement(tagName:"Terem");

    element.setAttribute(name:"Tid", Tid);
    element.appendChild(createElement(document,name:"Férőhely", seat));
    element.appendChild(createElement(document,name:"Teremszám", room_number));

    return element;
}

private static Node createElement(Document document, String name, String value){
    Element element = document.createElement(name);
    element.appendChild(document.createTextNode(value));

    return element;
}

public static void modifyId(Document document, String table, String id, String old_id, String new_id){
    NodeList list = document.getElementsByTagName(table);
    for(int i=0; i<list.getLength(); i++){
        Node node = list.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE){
            if(node.getAttributes().getNamedItem(id).getTextContent().equals(old_id)){
                node.getAttributes().getNamedItem(id).setTextContent(new_id);
            }
        }
    }
}
}

```

2c) adatlekérdezés

Beolvastam az XMLR65UKG.xml fájlt a DocumentBuilder és DocumentBuilderFactory előre definiált osztály segítségével. Majd lekérdeztem azokat a tárgyakat, amelyekből elméleti és gyakorlati óra is van egyaránt. Majd azokat az oktatókat, melyeknek az Informatikai épületben van a szobájuk. Majd azokat a termeket, ahol 100-nál nagyobb a férőhely, majd külön azokat a termeket, ahol van vetítő. Végül pedig a saját nevemet kérdeztem le a Neptunkódom alapján. Közben pedig minden kapott eredményt kiírtattam a konzolra.

```

package hu.domparse.R65UKG;

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.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.Text;
import org.xml.sax.SAXException;

public class DOMQueryR65UKG {

    Run | Debug
    public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException{
        File xmlFile = new File(pathname:"XMLTaskR65UKG/DOMParserR65UKG/src/hu/domparse/R65UKG/XMLR65UKG.xml");

        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();

        Document doc = dbBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        System.out.println(x:"1. Azok a tárgyak, melyekből elmélet és gyakorlat is van egyaránt: ");

        NodeList targyList= doc.getElementsByTagName(tagname:"Tárgyak");

        NodeList targyList= doc.getElementsByTagName(tagname:"Tárgyak");
        for(int i=0; i<targyList.getLength(); i++){
            Node node = targyList.item(i);
            printTargy(node, condition:"elmélet és gyakorlat");
        }

        System.out.println(x:"");
        System.out.println(x:"2. Azok az oktatók, melyeknek az Informatikai épületben van a szobájuk: ");

        NodeList oktatoList = doc.getElementsByTagName(tagname:"Oktatók");
        for(int i=0; i<oktatoList.getLength(); i++){
            Node node = oktatoList.item(i);
            printOktato(node, condition:"Informatikai épület");
        }

        System.out.println(x:"");
        System.out.println(x:"3. Azok a termek, ahol 100-nál nagyobb a férőhely: ");

        NodeList teremList = doc.getElementsByTagName(tagname:"Terem");
        for(int i=0; i<teremList.getLength(); i++){
            Node node = teremList.item(i);
            printTerem(node, condition:100);
        }

        System.out.println(x:"");
        System.out.println(x:"4. Azok a termek, ahol van vetítő: ");

        for(int i=0; i<teremList.getLength(); i++){
            Node node = teremList.item(i);
            printTerem2(node,condition:"Van");
        }
    }
}

```

```

System.out.println(x:"");
System.out.println(x:"5. Annak a diáknak a neve, akihez az R65UKG Neptunkód tartozik: ");

NodeList hallgatoList = doc.getElementsByTagName(tagname:"Hallgatók");
for(int i=0; i<hallgatoList.getLength(); i++){
    Node node = hallgatoList.item(i);
    printHallgato(node, condition:"R65UKG");
}

```

```

private static void printTargy(Node node, String condition){
    if(node.getNodeType() == Node.ELEMENT_NODE){
        Element element = (Element) node;
        String Taid = element.getAttribute(name:"Taid");

        Node nname = element.getElementsByTagName(name:"Név").item(index:0);
        String name = nname.getTextContent();

        Node nsubject_code = element.getElementsByTagName(name:"Tárgykód").item(index:0);
        String subject_code = nsubject_code.getTextContent();

        Node ntype = element.getElementsByTagName(name:"Típus").item(index:0);
        String type = ntype.getTextContent();

        if(type.equals(condition)){
            System.out.printf("Tárgyneve: " + name + ", tárgykódja: " + subject_code);
            System.out.println(x:"");
        }
    }
}

```

```

private static void printOktato(Node node, String condition){
    if(node.getNodeType() == Node.ELEMENT_NODE){
        Element element = (Element) node;
        String Oid = element.getAttribute(name:"Oid");

        Node nname = element.getElementsByTagName(name:"Név").item(index:0);
        String name = nname.getTextContent();

        Node nbuilding = element.getElementsByTagName(name:"Épület").item(index:0);
        String building = nbuilding.getTextContent();

        if(building.equals(condition)){
            System.out.printf(name);
            System.out.println(x:"");
        }
    }
}

```

```

private static void printTerem(Node node, int condition){
    if(node.getNodeType() == Node.ELEMENT_NODE){
        Element element = (Element) node;
        String Tid = element.getAttribute(name:"Tid");

        Node nroom_number= element.getElementsByTagName(name:"Teremszám").item(index:0);
        String room_number = nroom_number.getTextContent();

        Node nseat = element.getElementsByTagName(name:"Férőhely").item(index:0);
        int seat = 0;
        try{
            seat = Integer.parseInt(nseat.getTextContent());
        }
        catch(Exception e){
            seat=0;
        }

        if(seat > condition){
            System.out.printf("Teremszáma: " + room_number + ", férőhely: " + seat);
            System.out.println(x:"");
        }
    }
}

```

```

private static void printTerem2(Node node, String condition){
    if(node.getNodeType() == Node.ELEMENT_NODE){
        Element element = (Element) node;
        String Tid = element.getAttribute(name:"Tid");

        Node nroom_number= element.getElementsByTagName(name:"Teremszám").item(index:0);
        String room_number = nroom_number.getTextContent();

        Node nprojector = element.getElementsByTagName(name:"Vetítő_állapota").item(index:0);
        String projector = nprojector.getTextContent();

        if(projector.equals(condition)){
            System.out.printf("Teremszáma: " + room_number);
            System.out.println(x:"");
        }
    }
}

```

```

private static void printHallgato(Node node, String condition){
    if(node.getNodeType() == Node.ELEMENT_NODE){
        Element element = (Element) node;
        String Neptun_code = element.getAttribute(name:"Neptunkód");

        Node nname = element.getElementsByTagName(name:"Név").item(index:0);
        String name = nname.getTextContent();

        if(Neptun_code.equals(condition)){
            System.out.printf(name);
            System.out.println(x:"");
        }
    }
}

```


2d) adatírás

Beolvastam az XMLR65UKG.xml fájlt a DocumentBuilder és DocumentBuilderFactory előre definiált osztály segítségével, majd kiírtam a konzolra és egy XMLR65UKG1.xml fájlba.

```
package hu.domp.parse.R65UKG;

import java.io.File;
import java.io.IOException;
import java.io.PrintStream;

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

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

public class DOMWriteR65UKG {
    Run | Debug
    public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException, Trans
        File xmFile = new File(pathname:"XMLTaskR65UKG/DOMParseR65UKG/src/hu/domp.parse/R65UKG/XMLR65UKG.xml");

        DocumentBuilderFactory dFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dFactory.newDocumentBuilder();

        Document document = dBuilder.parse(xmFile);
        document.normalize();

        //Lemenjünk a dokumentumot
        saveDocument(document, filename:"XMLR65UKG1.xml");

    public static void saveDocument(Document document, String filename) throws TransformerException {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transf = transformerFactory.newTransformer();

        transf.setOutputProperty(OutputKeys.ENCODING, value:"UTF-8");
        transf.setOutputProperty(OutputKeys.INDENT, value:"yes");
        transf.setOutputProperty(name:"{https://xml.apache.org/xslt}indent-amount", value:"2");

        DOMSource source = new DOMSource(document);

        File myFile = new File(filename);
        PrintStream p = null;
        try{
            p = new PrintStream(System.out, autoFlush:true,encoding:"UTF-8");
        }
        catch(Exception e){
            System.out.println(x:"Don't make it!");
            return;
        }

        StreamResult console = new StreamResult(p);

        StreamResult file = new StreamResult(myFile);

        transf.transform(source, console);
        transf.transform(source, file);
    }
}
```