JEGYZŐKÖNYV

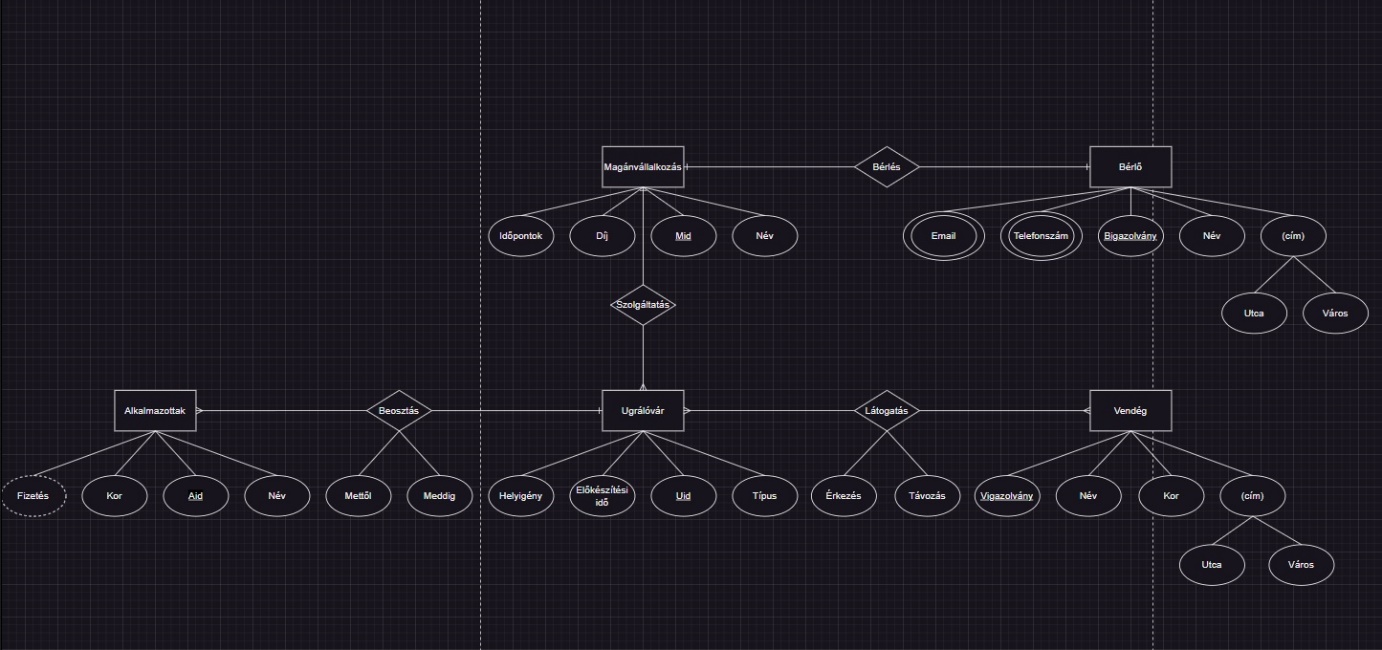
Adatkezelés XML környezetben  
  
Féléves feladat  
  
Ugrálóvár Kölcsönzés

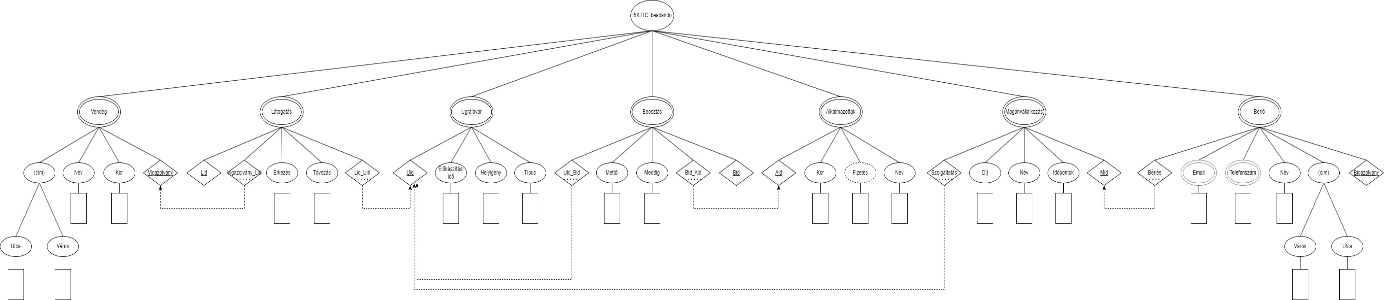
Készítette: Katona Bence

Neptunkód:I5XJTC  
Dátum:2023.12.05

Tartalomjegyzék  
  
Bevezetés  
  
Ugrálóvár kölcsönzés:  
  
Azért ezt a témakört választottam, mert volt már tapasztalatom vele, ráadásul nem tudom más hogy van vele, de én nagyon élveztem az ilyen magasról leugrást anélkül hogy nem halsz bele(magas ugrálóvár). Ugyebár sok féle ugrálóvár létezik különböző méretekkel és funkciókkal, van amit arra terveztek hogy versenyt fussanak benne, egyfajta akadály pálya az egész, van olyan ami meg felvisz a magasba hogy onnan leugorhas, de persze vannak olyanok is amiket gyerekeknek terveztek. Viszont ahhoz hogy egy ilyen élvezetes dologban legyen részünk először ezeket az ugráló várakat ki kell bérelni/ ki kell kölcsönözni. Ez elég könnyen megoldható neten, de szerintem még számos másik módszerrel el lehet érni őket. Általában az árak ilyen 30.000Ft-tól szoktak kezdődni, de ha nekik is ott van hogy benzin költség, és a munkásaik kifizetése. A munkások legtöbbször ilyen 20 év körüli fiatalok szoktak lenni, főleg azért mert őket még jobban szeretik a gyerekek mint az öregebb felnőtteket. Vendégként, ha elmegyünk ilyen ugrálóváras rendezvényre akkor legtöbbször egyből fel is lehet mászni rá és szórakozni, de vannak olyan esetek is amikor kikérik személyes adatainkat, nevet, kort, személyit hasonlókat, bár ezek legtöbbször felmérések miatt történik.

1.Feladat  
1a) Az adatbázis ER modell tervezése  
Szerkezete a következőkből tevődik össze: 5 egyed, 4 kapcsolat és 29 tulajdonság, amiben megtalálható atomi, összetett, többértékű, származtatott tulajdonság és minden egyes egyednek van kulcs értéke. Az ER modellben szó van a bérlőről, aki az ugráló várat bérli mellette az adatai: Igazolványszáma, lakcíme alias házszám, utca, város és még a telefonszáma, email címe, neve, szó van arról, hogy mennyit fizet a magánvállalkozásnak/Cégnek, aminek szintúgy megvannak a különböző tulajdonságai, mint például a neve vagy az időpontok hogy mikor lehetne esetleg kikérni és ezekhez előre megszabott költségek társulnak + Pk. Kihagyhatatlan módon az ugráló várak is kaptak egy egyedet és tulajdonságokat, e-mögött a típust és mondjuk egyes ugrálóvárak felállításának idejét értem, de a legfontosabb hogy meglegyen egy biztos hely ahova le lehet tenni, erre szolgál a helyigény + Pk. Az alkalmazottak pedig kötelezőek egyes ugrálóvárakhoz, néhánynál nem is elég 1 ember, a hibák észlelése és a vendégek biztonsága érdekében. Ez az egyed rendelkezik névvel, korral és egy származtatott tulajdonsággal ami a fizetés + Pk. Az ő fizetésük nagy mértékben függ a bérlő által fizetett összegtől és legtöbbször meg van szabva, hogy mettől meddig kell bizonyos ugráló váraknál lenni. Az emberek csak úgy jelentkezhetnek az ugróvárak kipróbálására, ha előtte kitöltenek egy kis kérdőívet ebben megtalálható a neve, igazolvány száma, kora és a lakcíme (a Bérlő egyednél hasonlóan). Ez függ a bérlő hozzáállásától is, de vannak olyan esetek, hogy ugrálóvár után felírnak, hogy mettől meddig voltál azon az ugráló váron. Az elsődleges kulcsok leginkább az egyed kezdőbetűjéből és egy id-ből állnak, de vannak olyanok is amik igazolvánnyal végződnek(pl:Uid,Mid,Aid,Vigazolvány,Bigazolvány). A kapcsolatok név szerint:   
Bérlés a Magánvállalkozást és a Bérlőt köti össze 1:1 kapcsolattal, Szolgáltatás a Magánvállalkozást és az ugráló várat köti össze 1:N kapcsolattal, egy magánvállalkozásnál több ugrálóvár lehet, de egy ugrálóvár csak egy Magánvállalkozáshoz/Céghez tartozhat,  
Látogatás az Ugrálóvárat és a Vendéget köti össze N:M kapcsolattal,  
Beosztás pedig az Ugrálóvárat és az Alkalmazottakat köti össze 1:N kapcsolattal úgy hogy egy ugrálóvárnál több ember is lehet, de egy ember csak egy ugrálóvárnál tud felügyelni.



1b) Az adatbázis konvertálása XDM modellre  
Az Egyedek dupla karikát kaptak, ahogy a kapcsolatok is amik elemet tartalmaztak, vagy N:M kapcsolat volt két egyed közt. Minden tulajdonságnál megmaradt a kör ábrázolás, még az egyéb változatainál is és azok a tulajdonságok amik rendes elemet tartalmaztak és nem másik tulajdonságot azok kaptak egy téglalapot saját maguk alá. A Primary Keys megtartották a formai alakjukat viszont összekötésüket már szaggatott vonallal oldjuk meg, ha egy Foreign Key van szóban. Az alakzatokban lévő nevüket meg aláhúzzuk a megfelelő vonalfajtával attól függően hogy melyik csoportba tartozik.  
  


1c) Az XDM modell alapján XML dokumentum készítése:  
Ugye ahogy elkezdünk minden xml modellt beírtam hogy az xml 1.0-ás verziójával dolgozzon és átállítottam a kódolását UTF-8 ra  
Majd létrehoztam az XDM alapján szépen sorban az elemeket:  
Először leírtam a gyökér elemet majd mind a 3 megjegyzést ami a 3 darab példányra utal. Ezután jöttek a nagyobb tárolók és tulajdonságaik és a tárolók elemei. Mielőtt kitöltöttem volna elemekkel, először összekapcsoltam a táblákat a tulajdonságaikat használva, és ez után töltöttem ki adatokkal az elemeket. Ahogy látszik ahol Összetett volt ott most már 3 adat van megadva és a több értékűt is még leágaztattam a leírt egyedeire.   
A képen szöveg, elektronika, képernyőkép, képernyő látható

Automatikusan generált leírás   
A képen szöveg, elektronika, képernyőkép, képernyő látható

Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

Automatikusan generált leírás

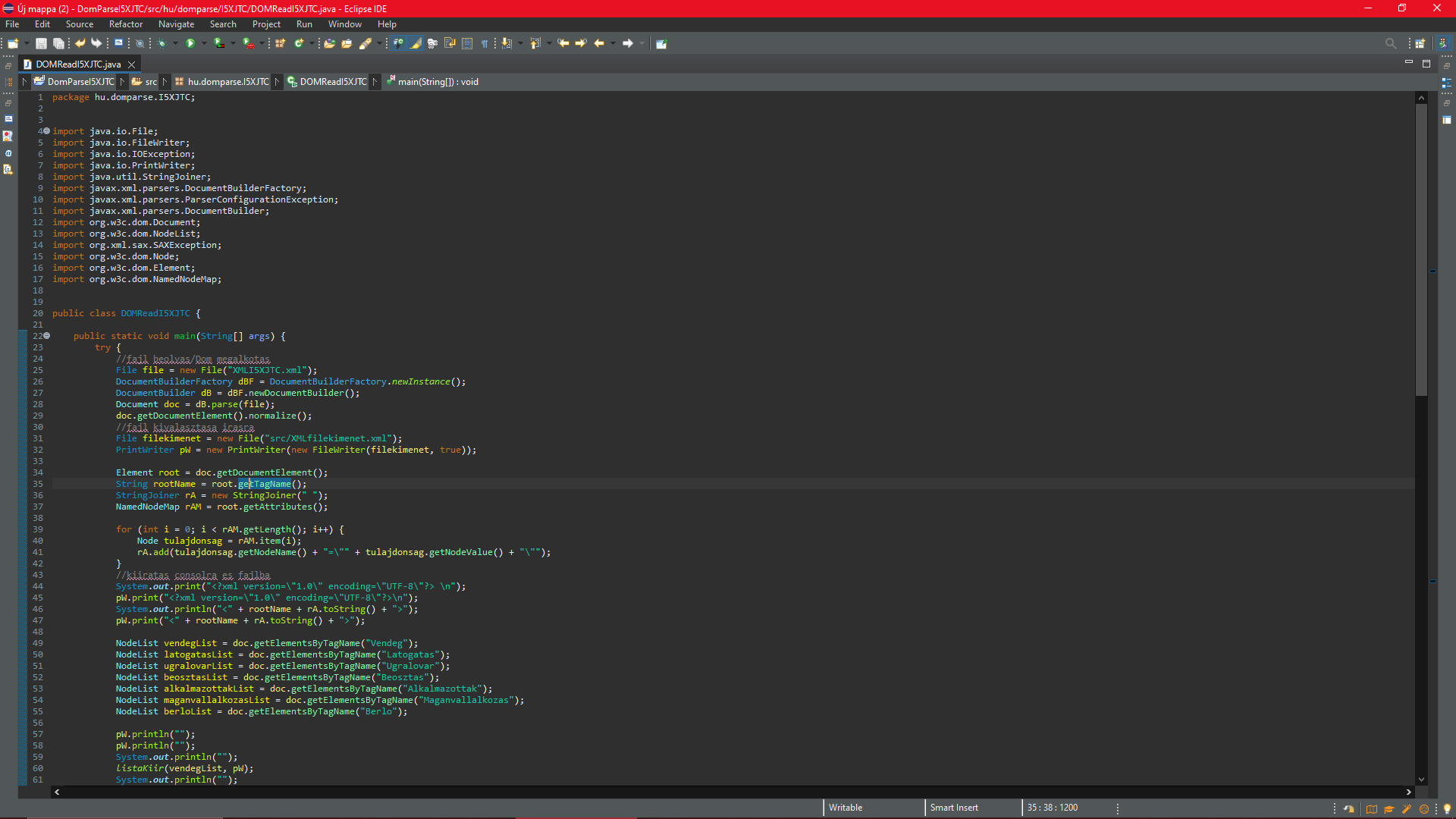
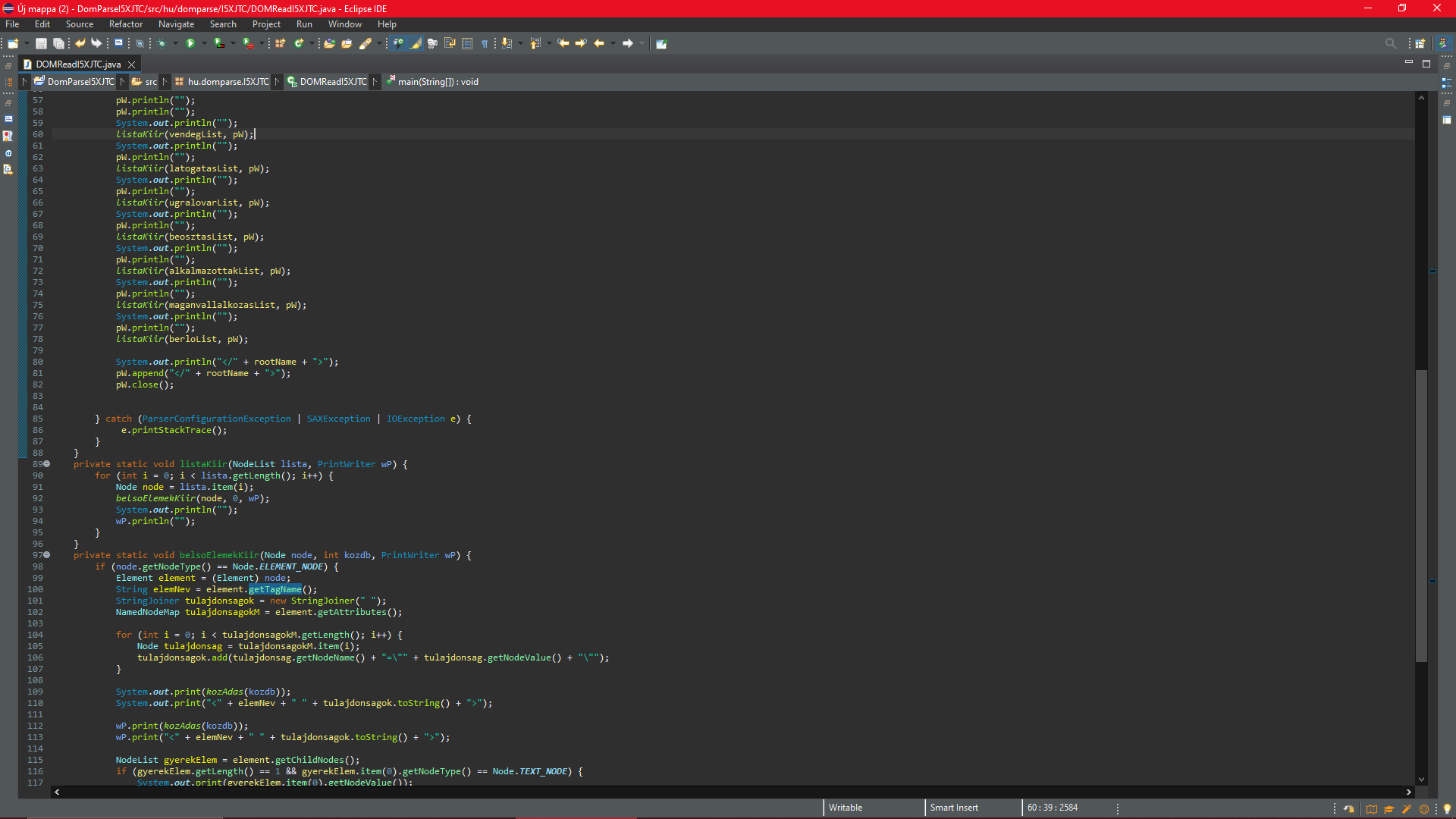
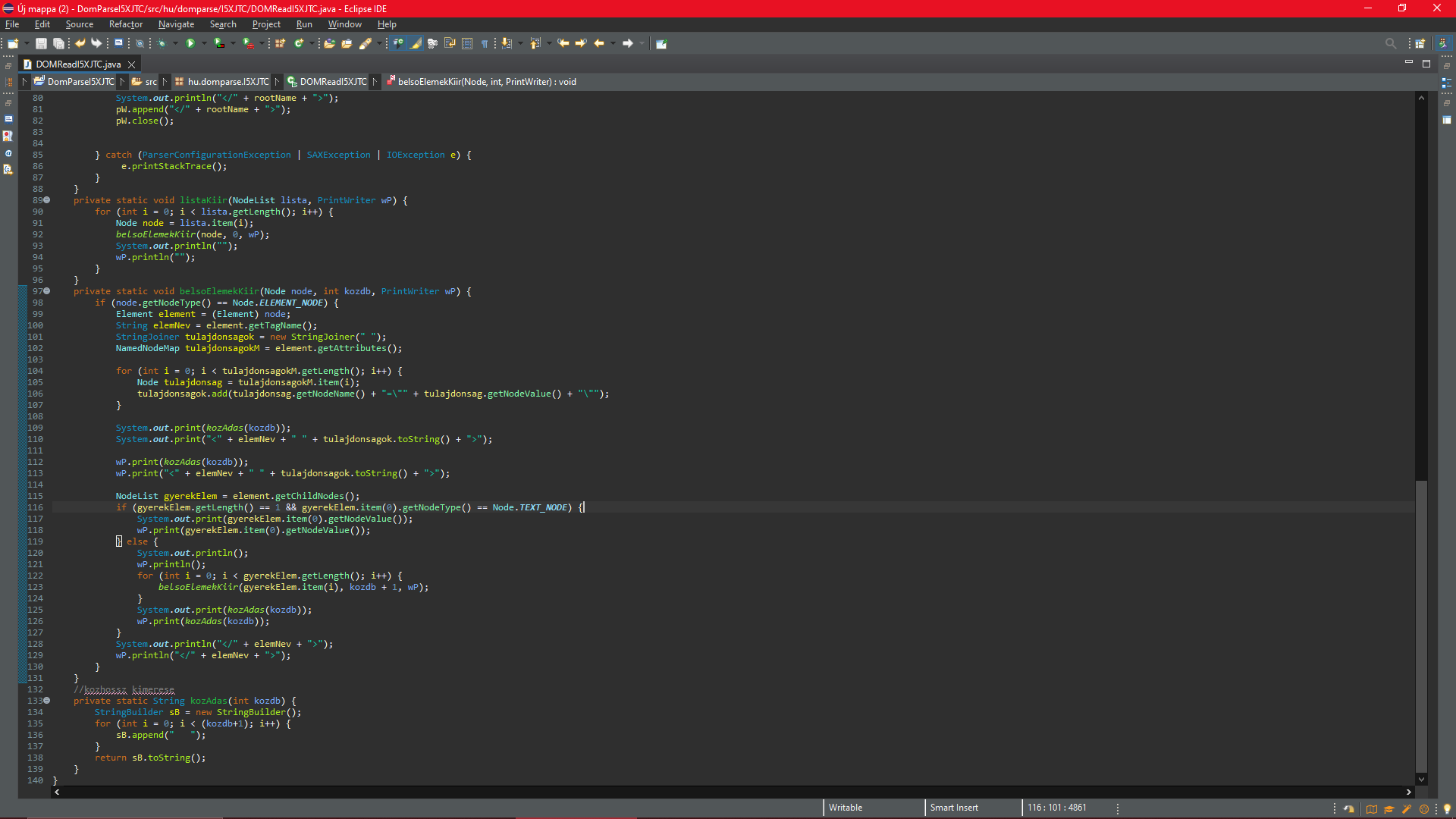
1d) Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek.   
  
Az xmlschemákhoz híven a szokásos elejét leírtam. Majd mivel hivatkozni kellett az elemekre ezért már a leges legtetején azzal kezdtem, 1 leírtam hogy milyen adatok lesznek itt megjegyzésbe amit a példa jegyzetből másoltam, 2 leírtam az elemeket neveiket és típusaikat egyesével majd írtam hozzájuk korlátozásokat, amik kaptak korlátozásokat azoknak a típusának a nevét megváltoztattam. Miután beírtam a többi megjegyzést egyből a gyökér elemmel kezdtem leírtam az elemeit és az elemeinek a típusait, majd külön külön azoknak az elmeit és tulajdonságait, persze az elemeikre csak hivatkoztam mert azokat már megadtam felül. Amint ezek megvoltak rátértem a kulcsokra ott megadtam a kulcs nevét hogy hol található és azt hogy hogyan milyen néven találhatjuk meg, ezek után meg belekezdtem az idegen kulcsokba ahol megadtam az idegen kulcsok nevét hogy milyen Primary Key-re is hivatkozik pontosabban a neve, ahogy előbb meg lett adva, utána meg lett adva hogy hol helyezkedik el ez az idegen kulcs és legvégül a Primary Key tényleges nevét adtuk meg ahogy valóban kinéz a fájlokban és legvégül maradt egy Egy-Egy kapcsolat, elnevezés, megmondod a helyét és a primary key-t hozzá.  
A képen szöveg, elektronika, képernyőkép, képernyő látható

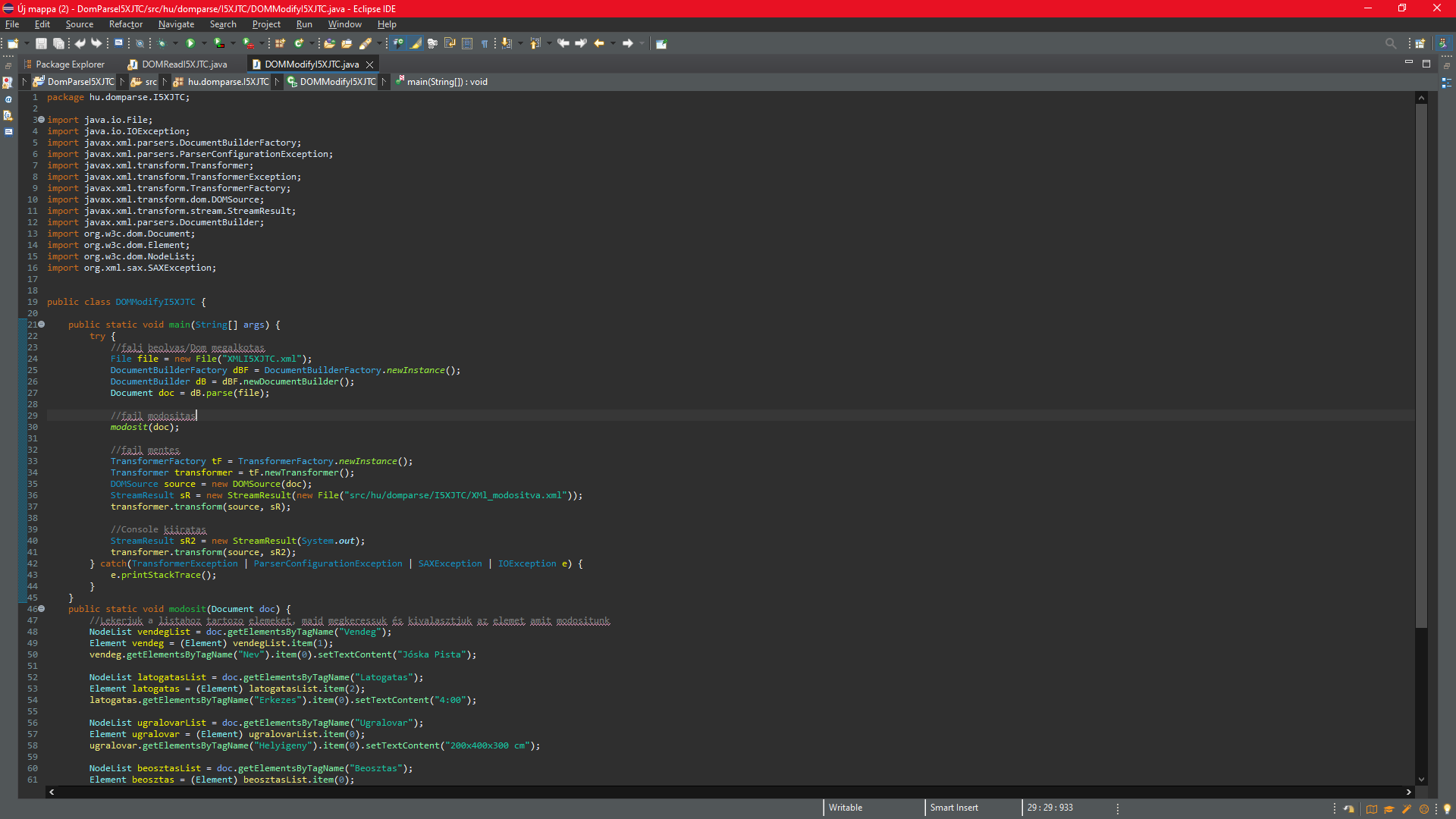
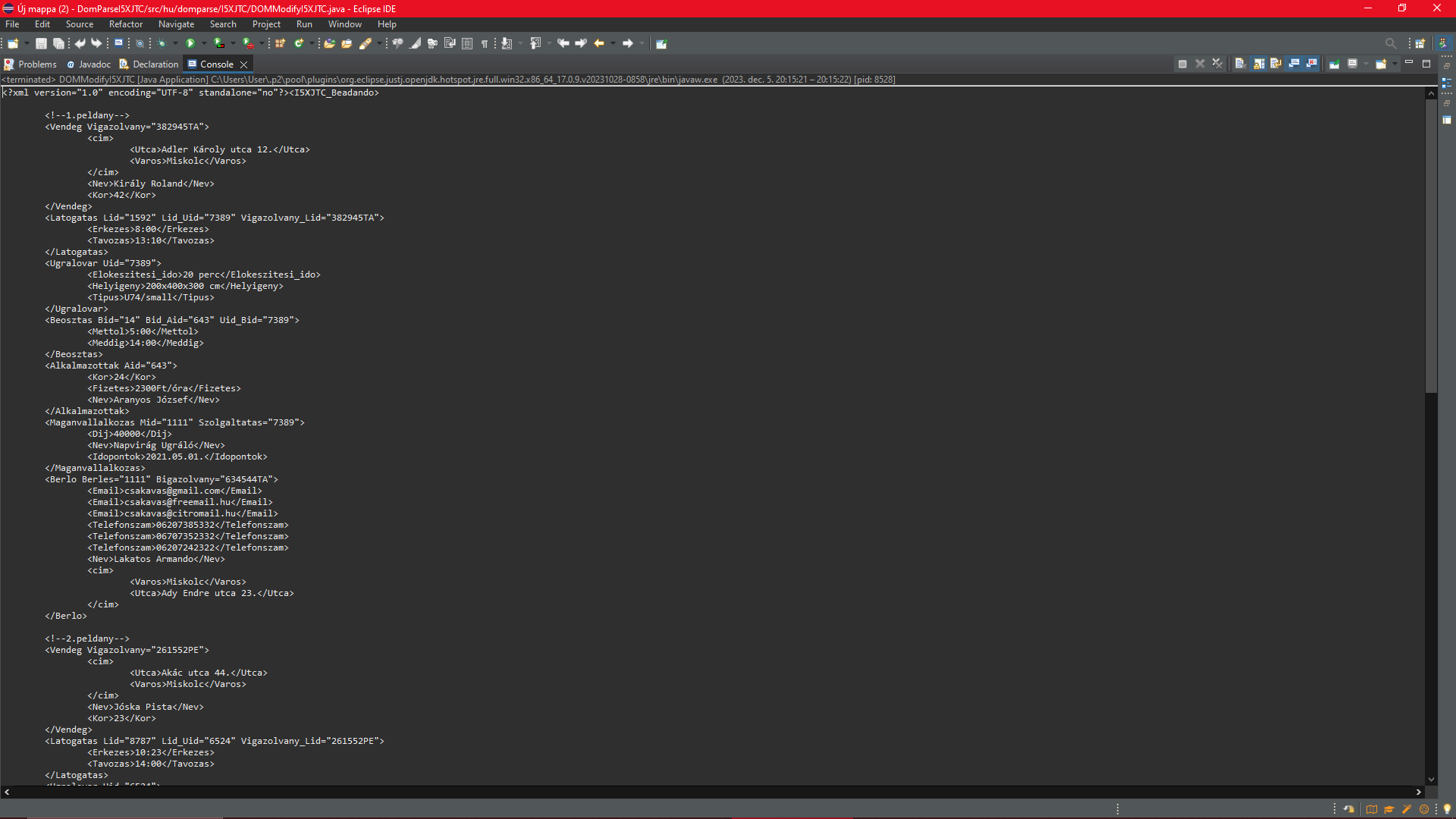
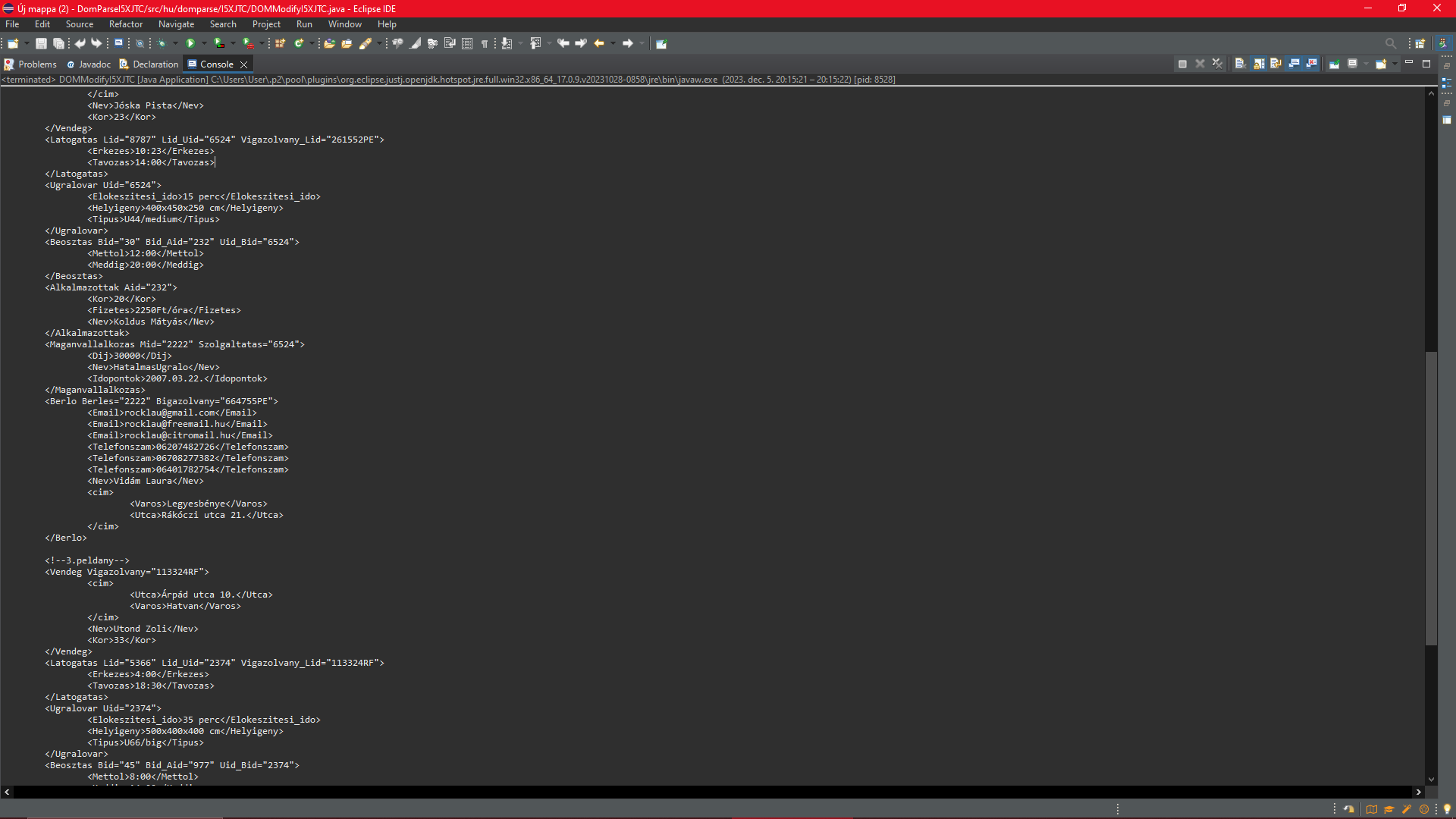
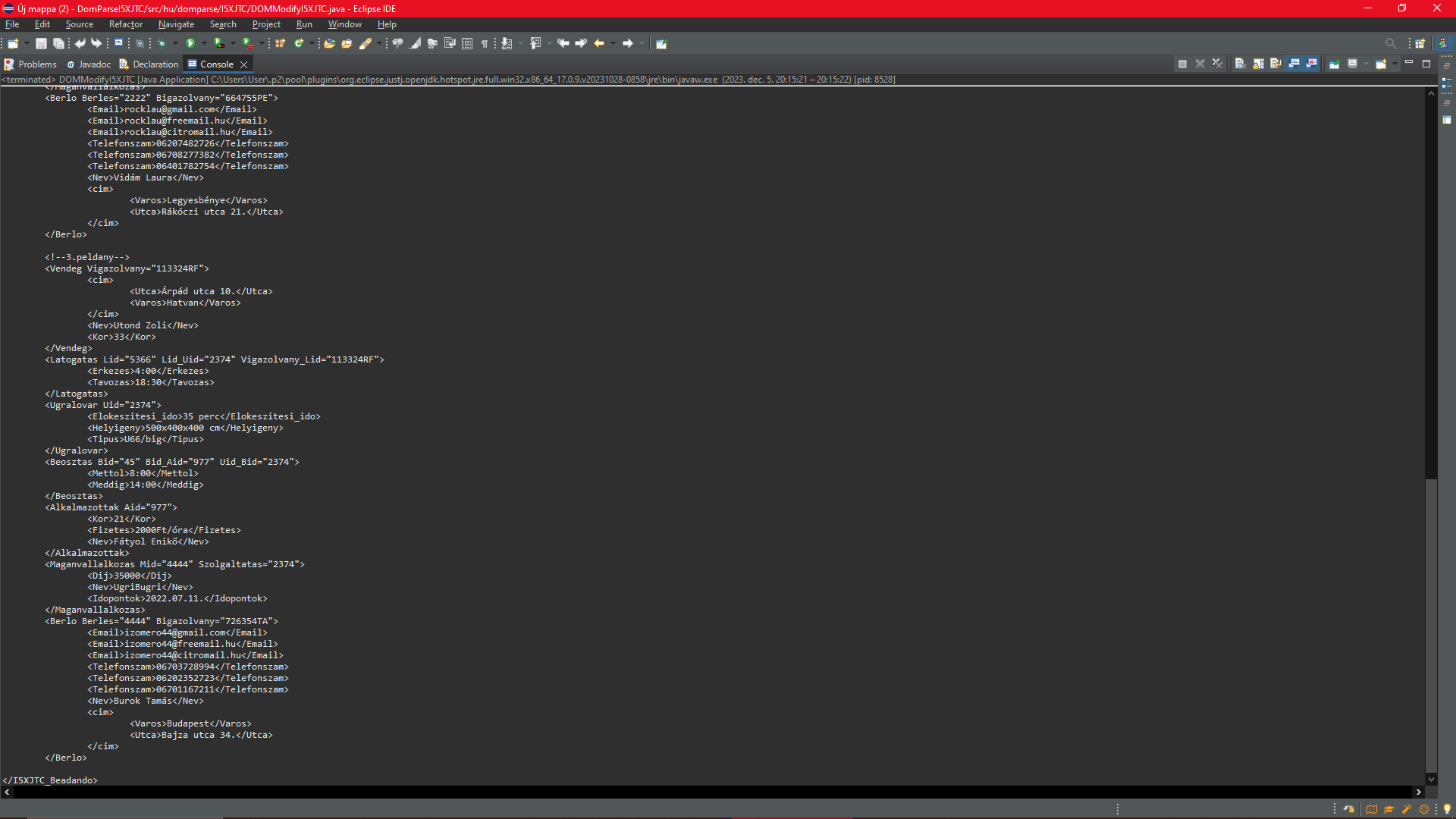
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

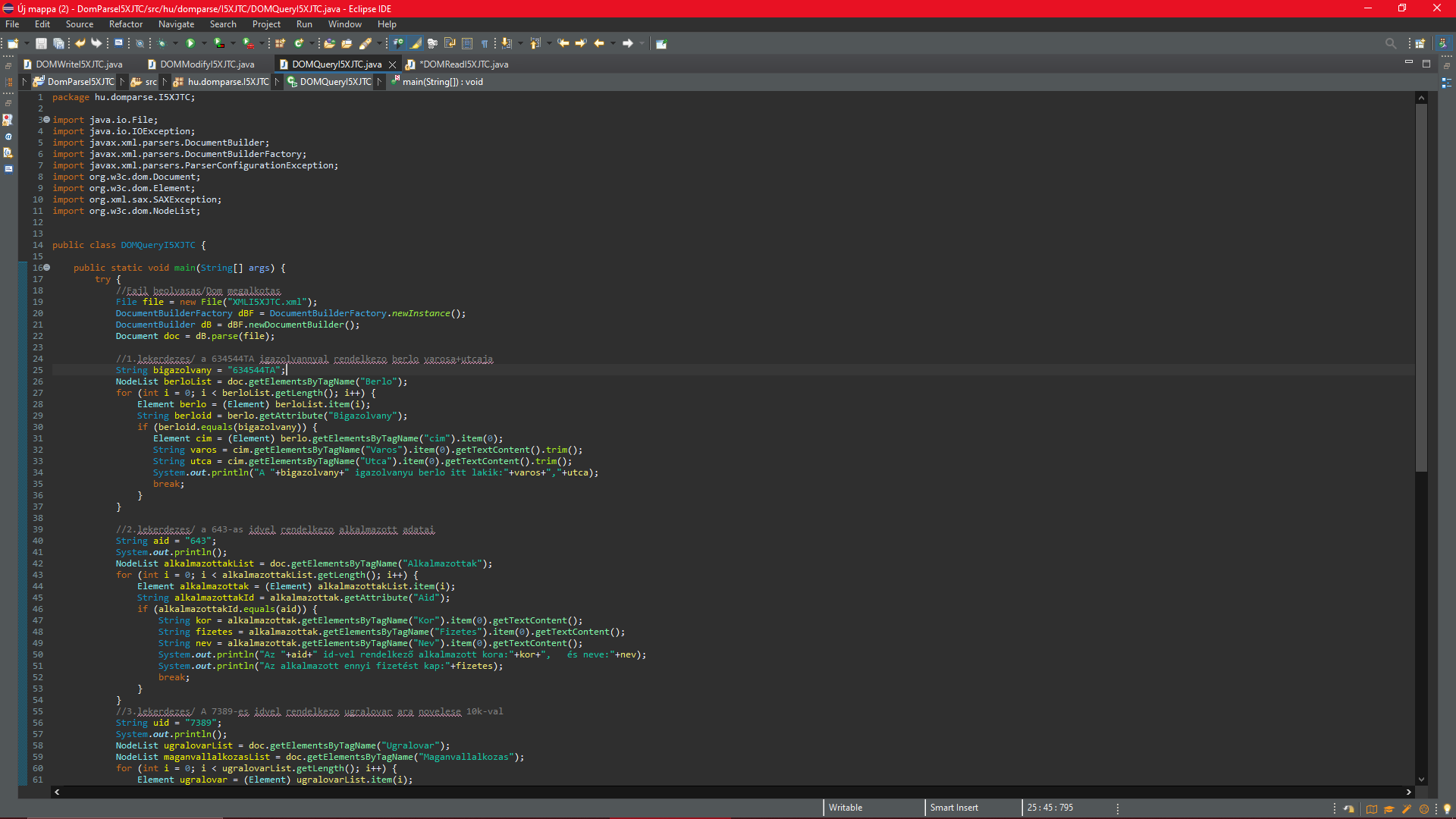
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

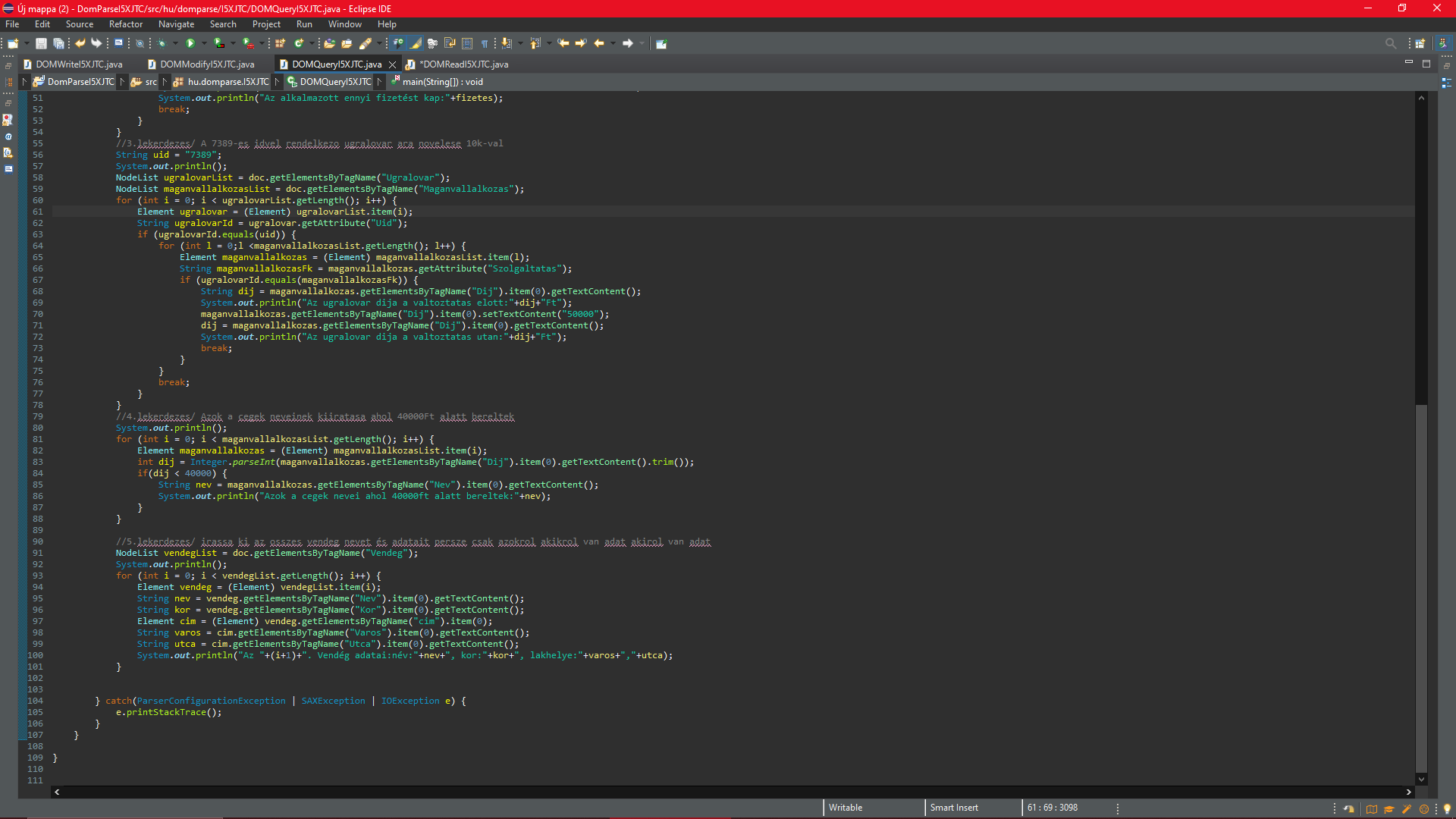
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

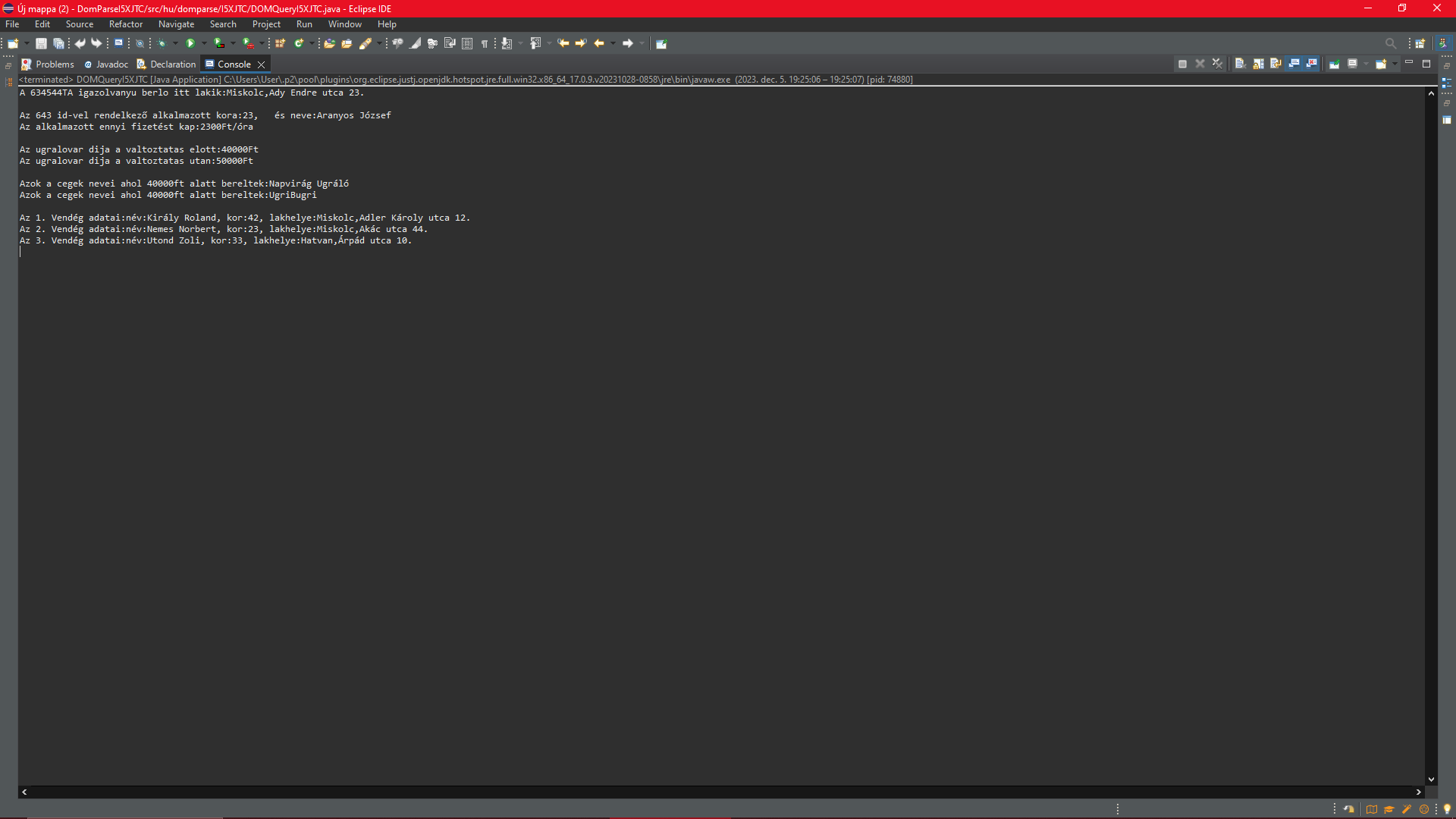
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

Automatikusan generált leírás  
  
2. feladat  
  
2a) adatolvasás (kód – comment együtt) – fájlnév: DOMReadNeptunkod.java  
  
  
  
  
2b) adatmódosítás (kód – comment együtt) – fájlnév: DOMModifyNeptunkod.java

  
  
  
  
  
  
2c) adatlekérdezés (kód – comment együtt) – fájlnév: DOMQueryNeptunkod.java





  
  
2d) adatírás (kód – comment együtt) – fájlnév: DOMWriteNeptunkod.java