JEGYZŐKÖNYV

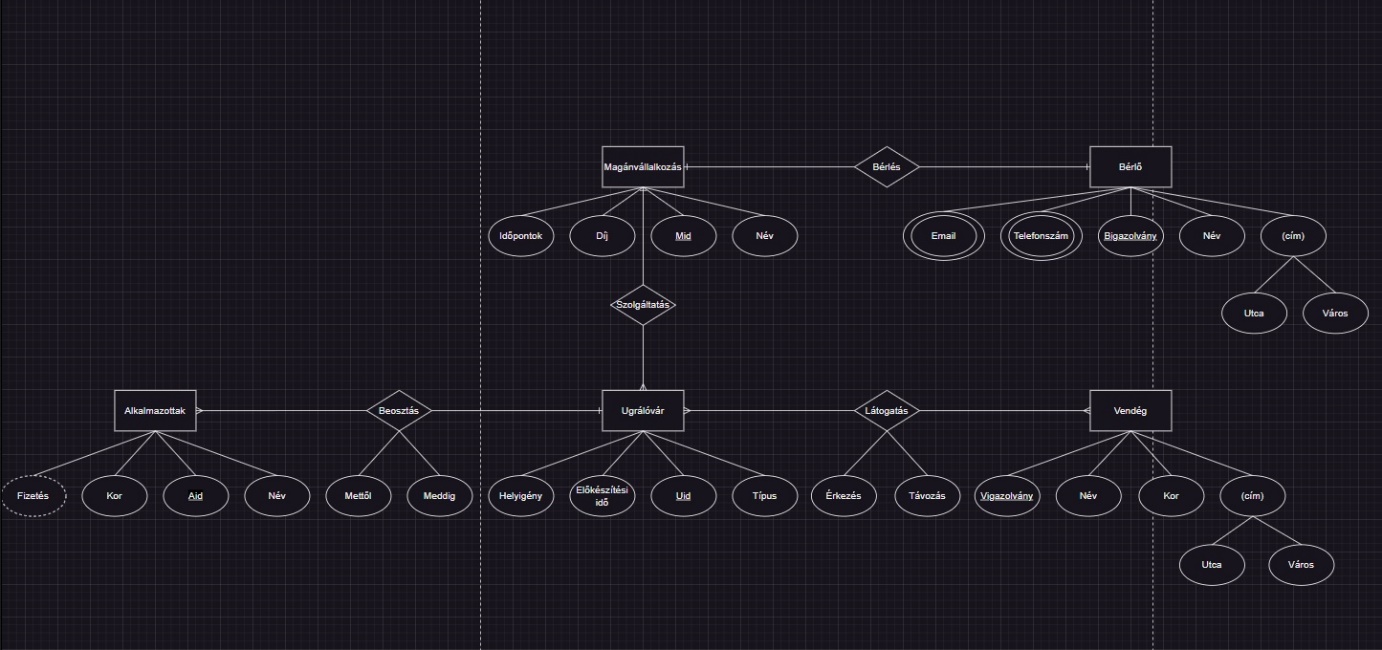
Adatkezelés XML környezetben  
  
Féléves feladat  
  
Ugrálóvár Kölcsönzés

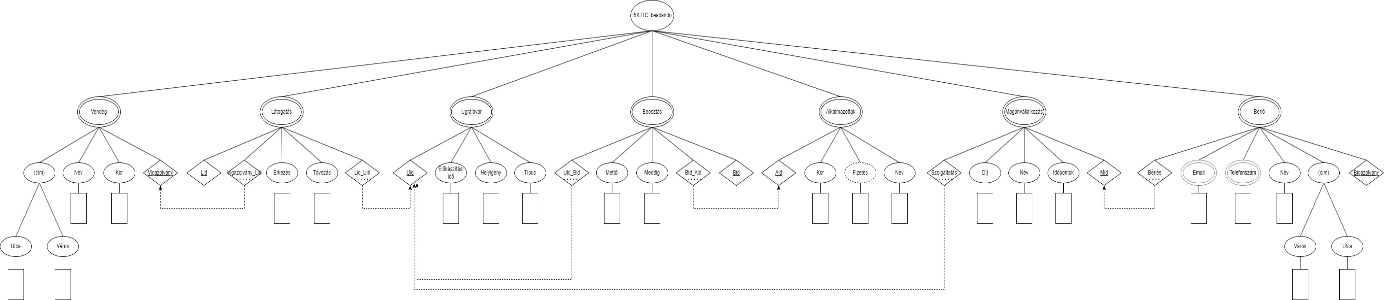
Készítette: Katona Bence

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

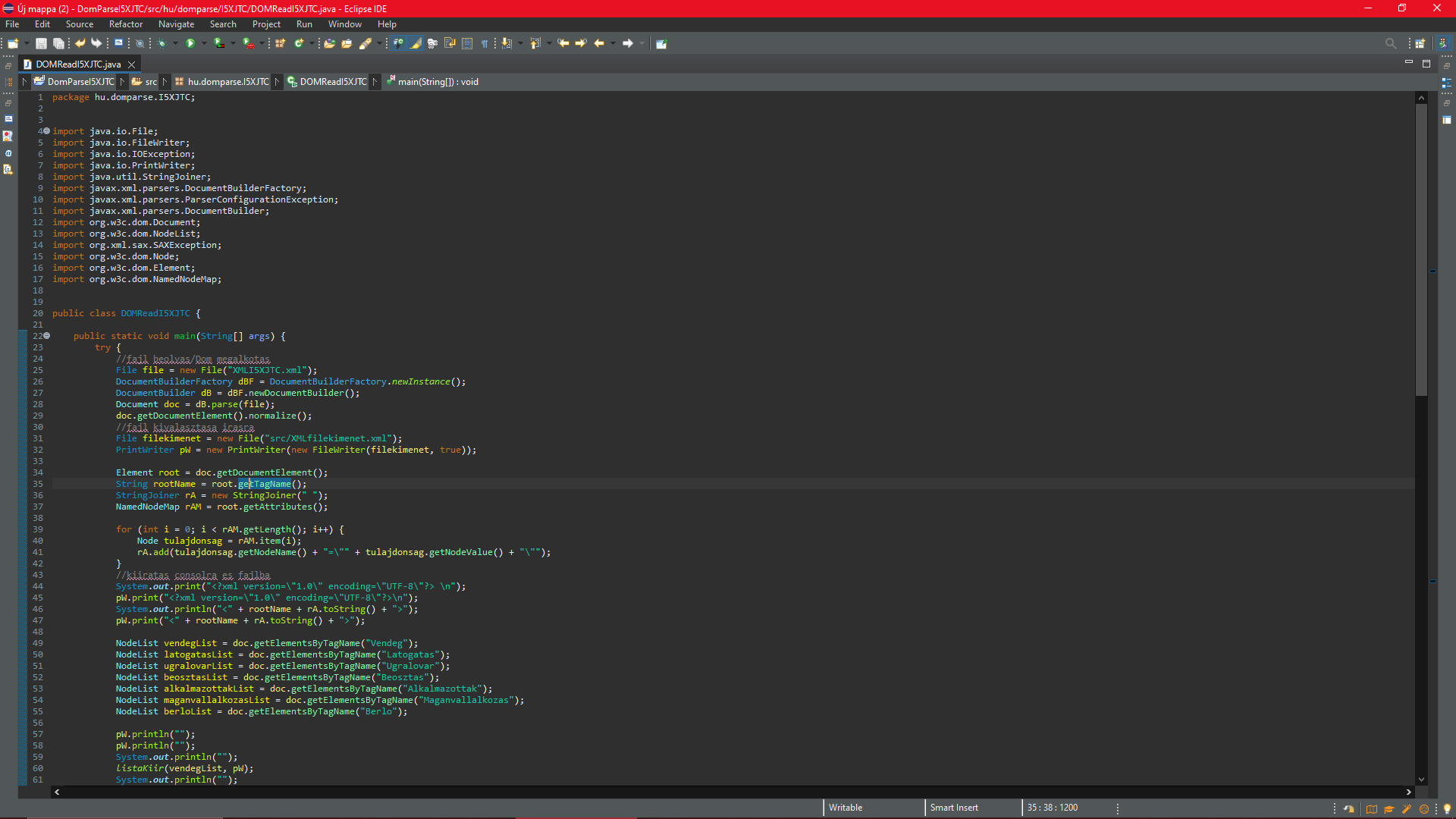
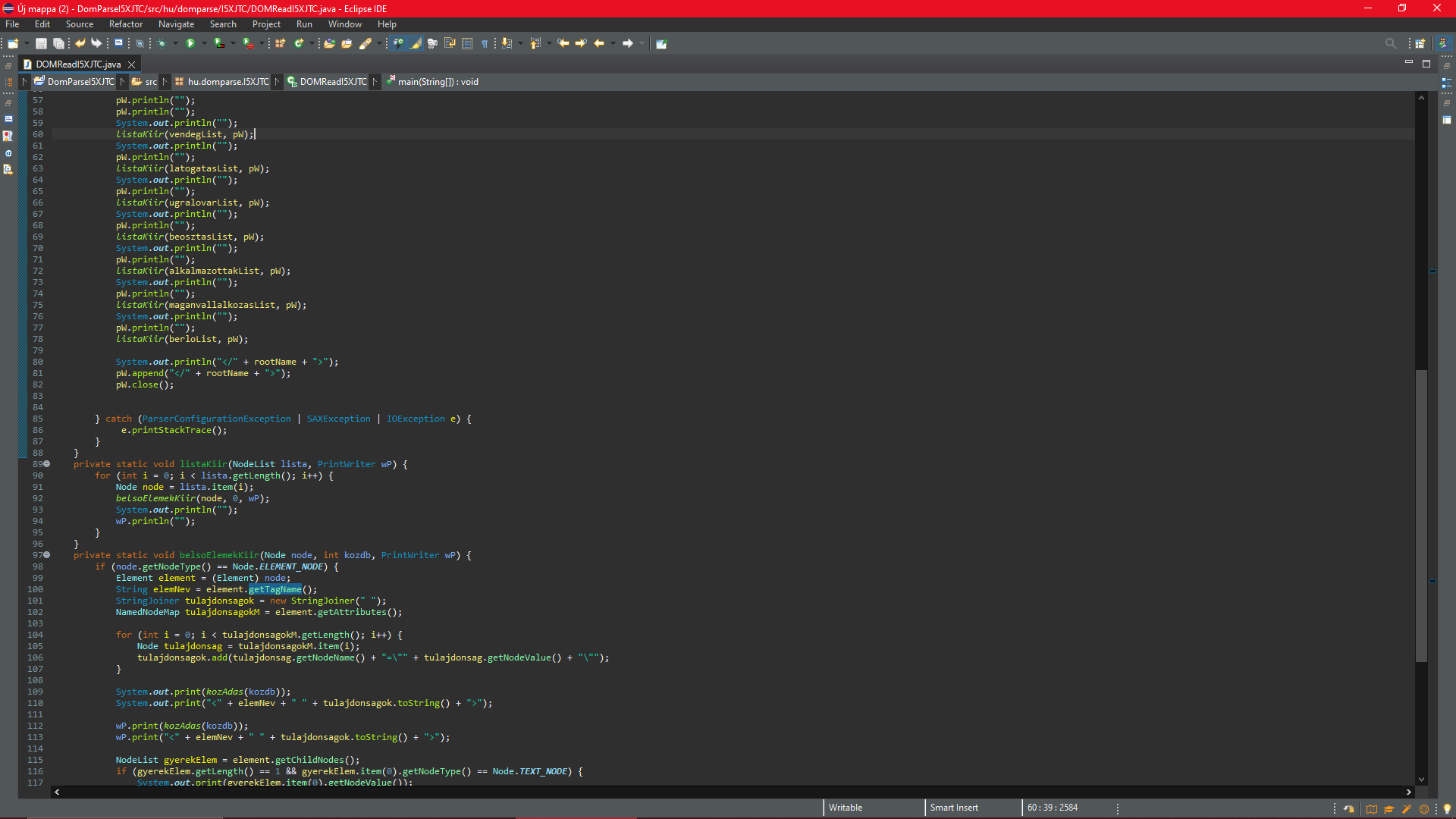
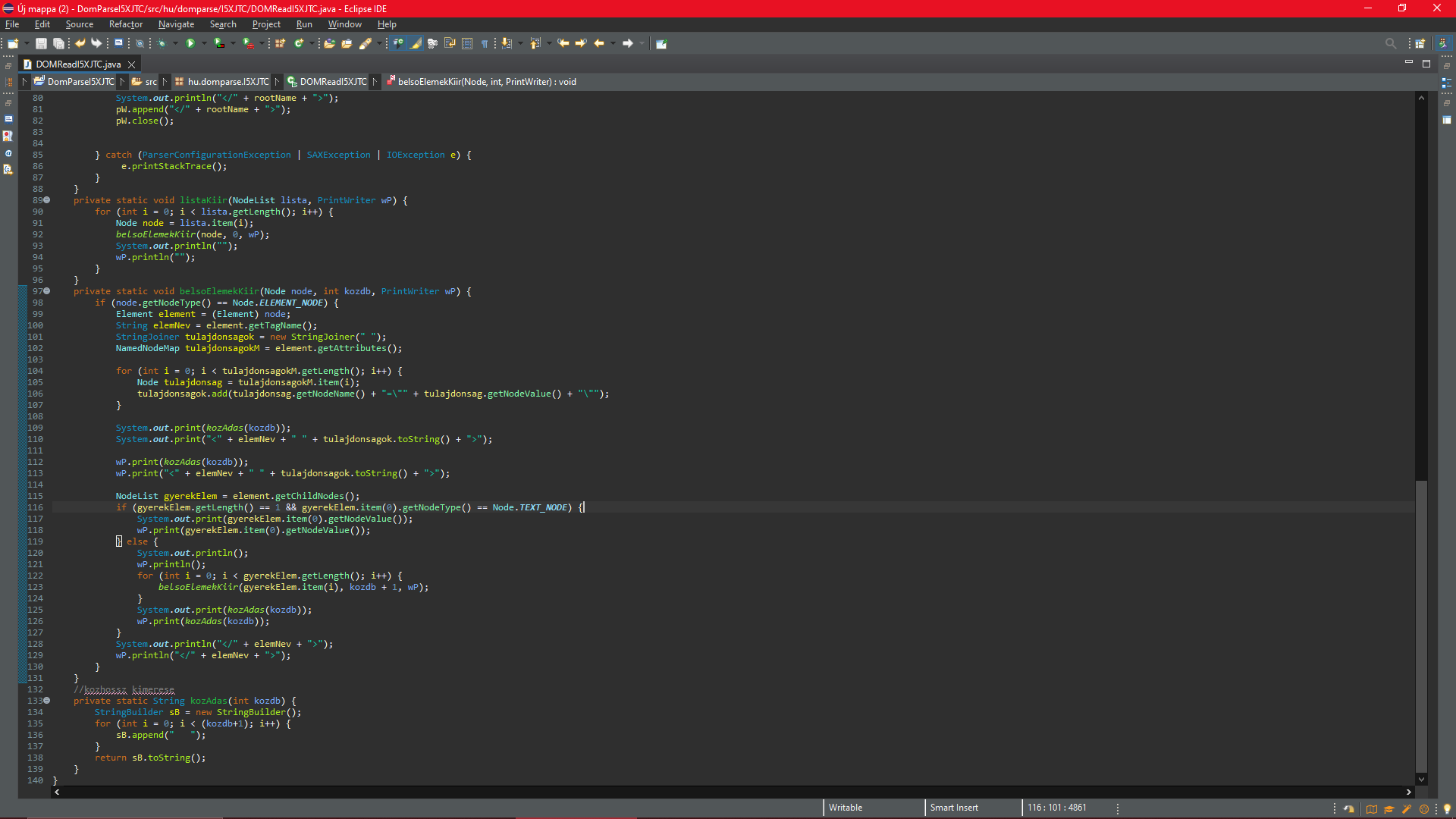
1d) Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek.   
  
Az xmlschemákhoz híven a szokásos elejét leírtam. Majd mivel hivatkozni kellett az elemekre ezért már a leges legtetején azzal kezdtem, 1 leírtam hogy milyen adatok lesznek itt megjegyzésbe amit 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

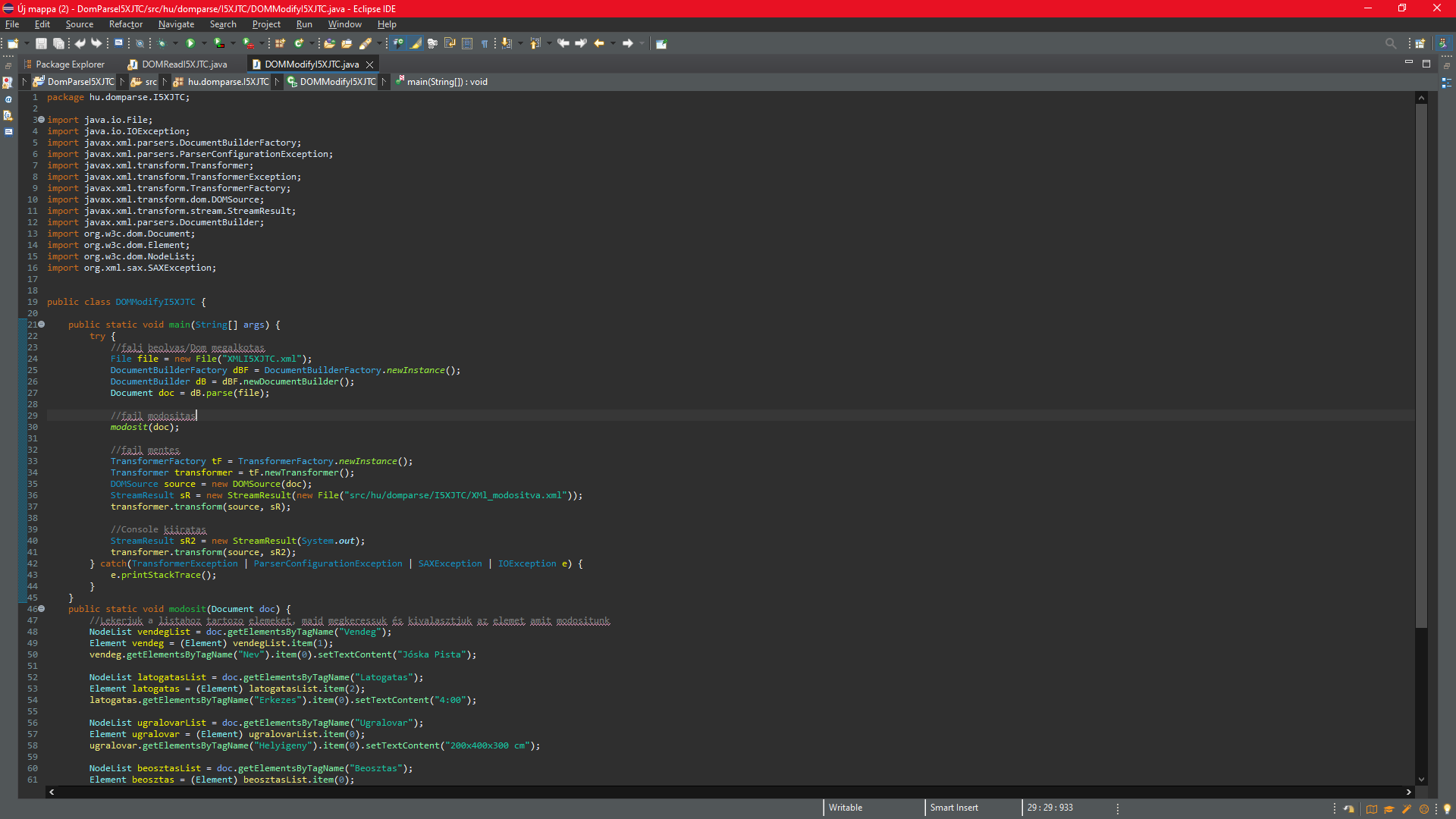
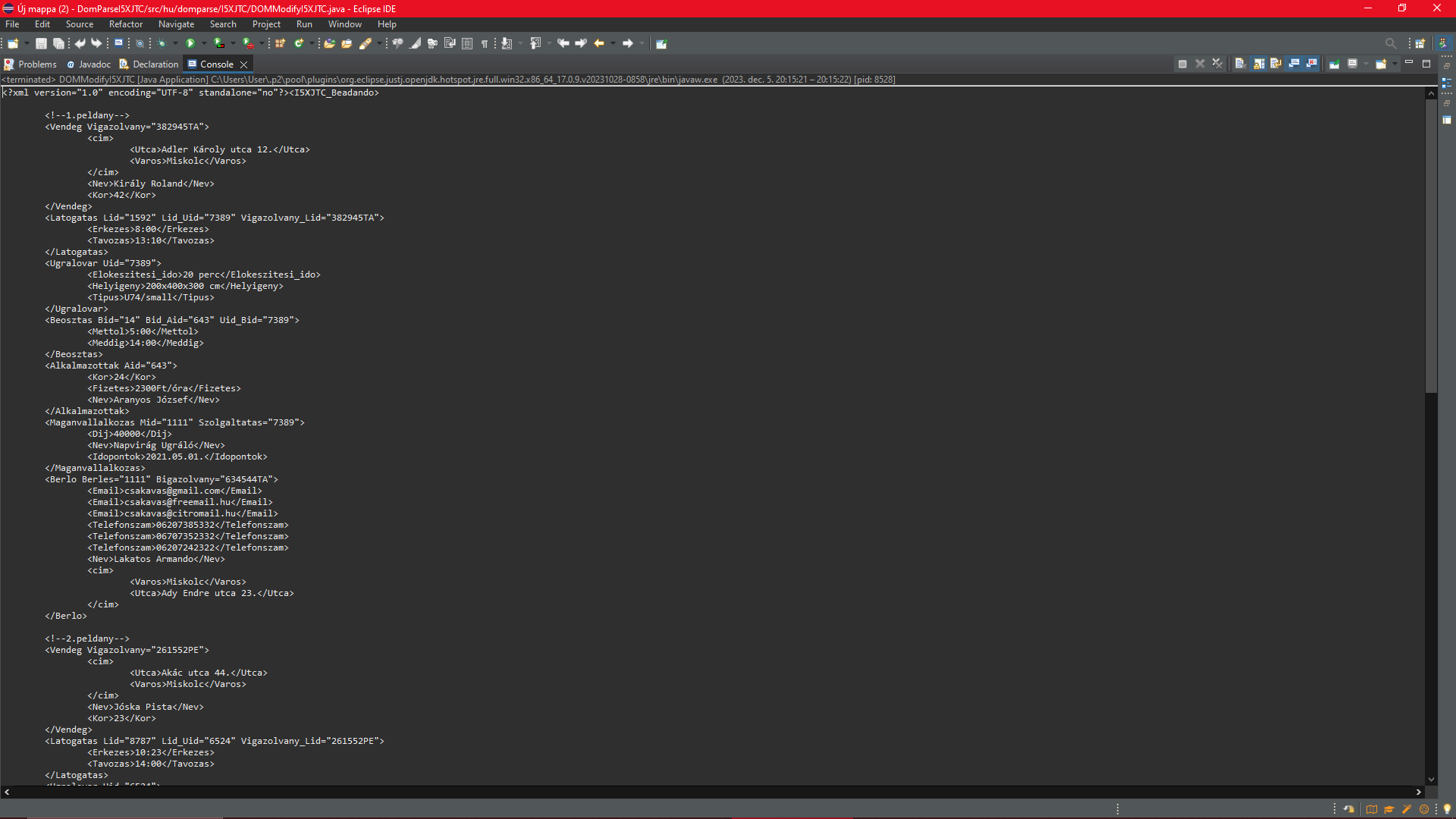
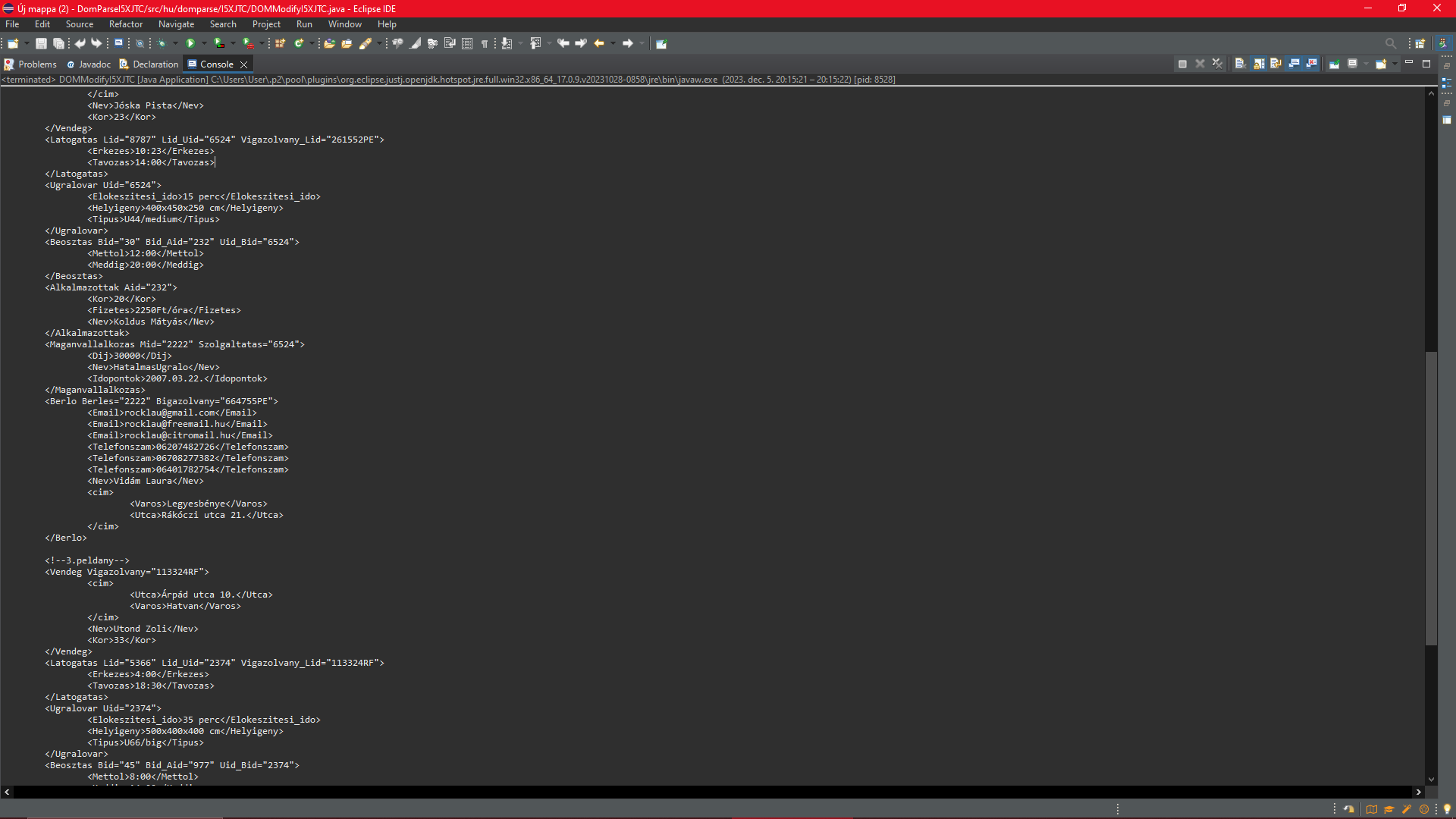
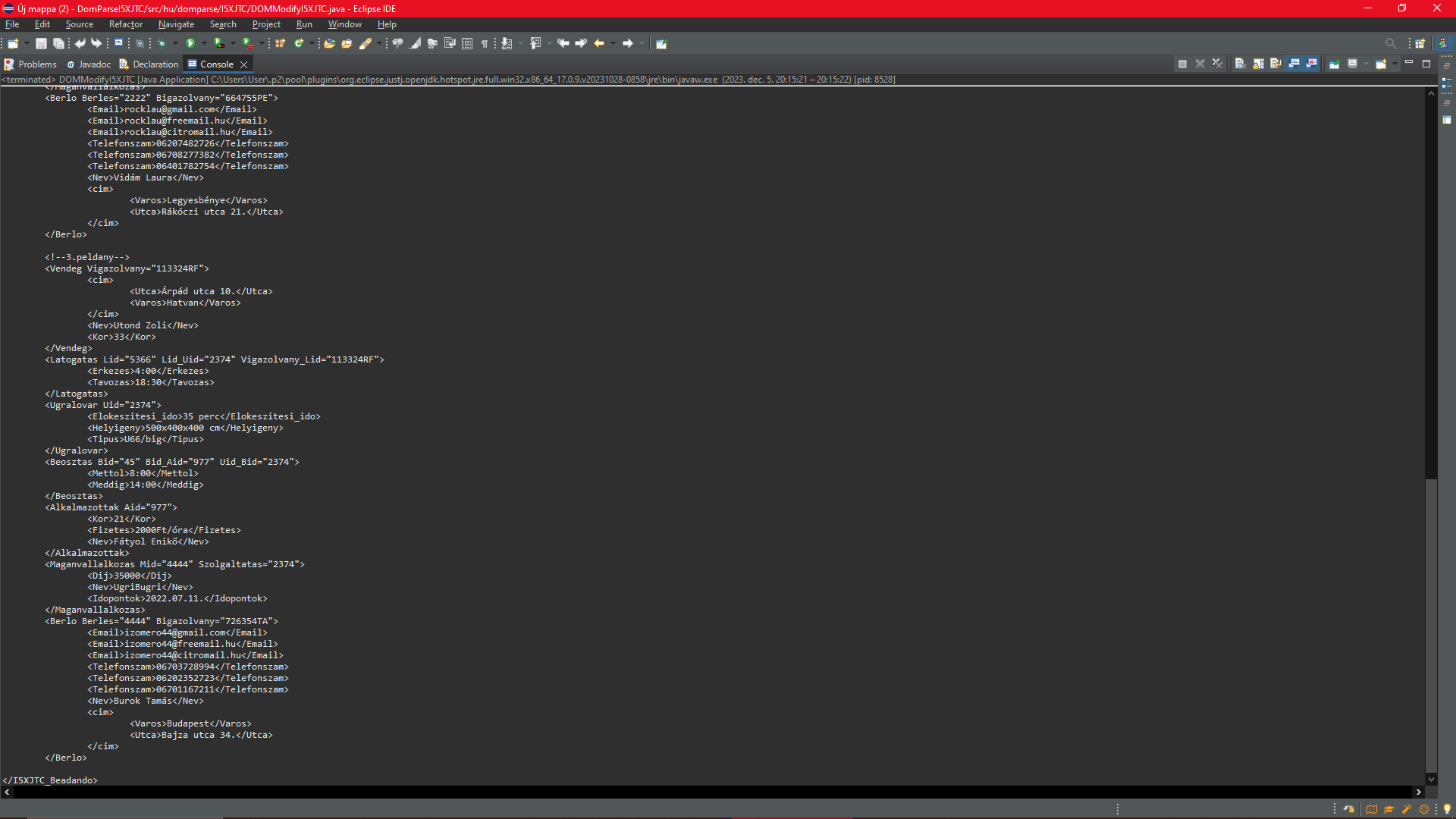
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

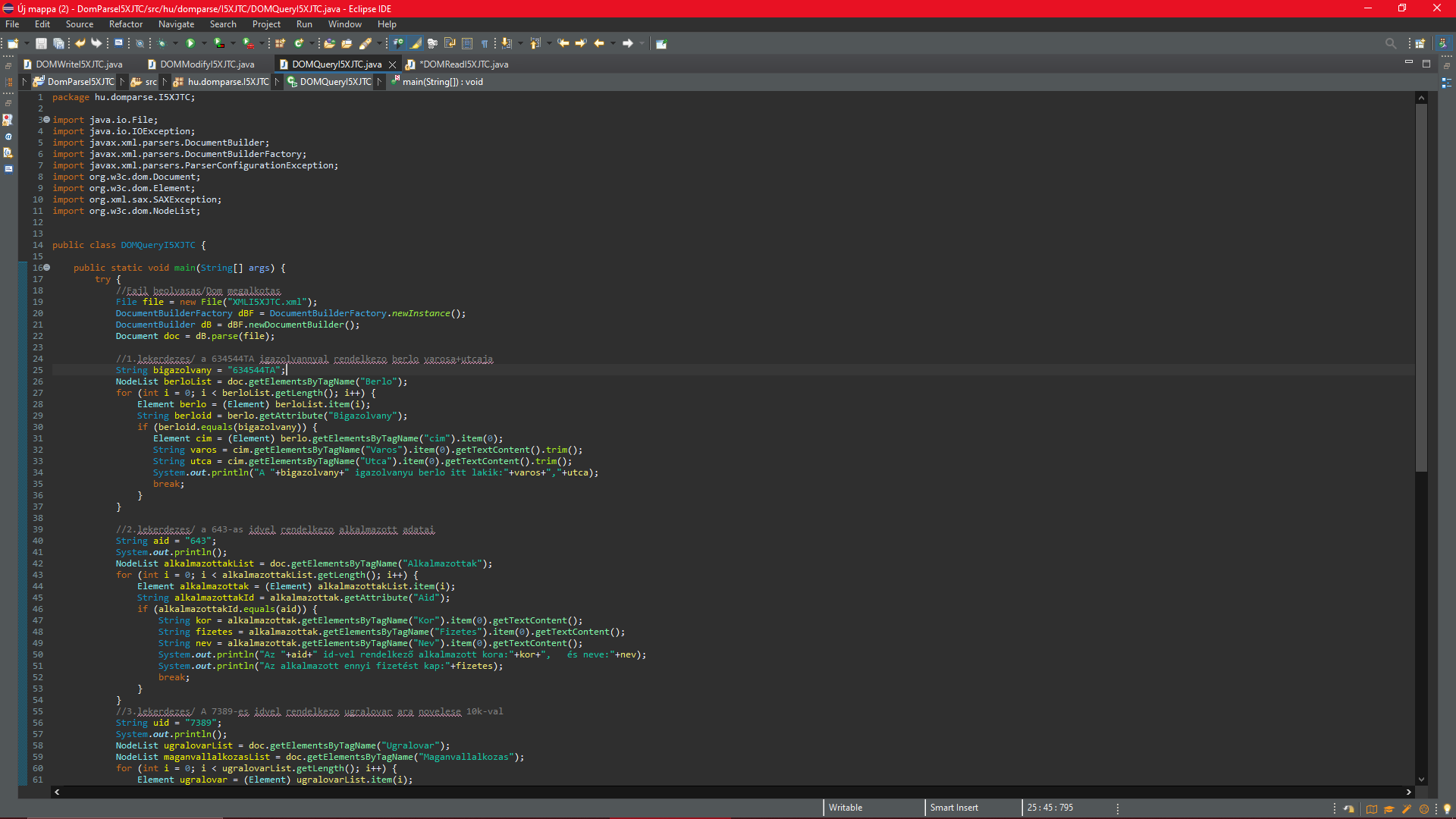
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

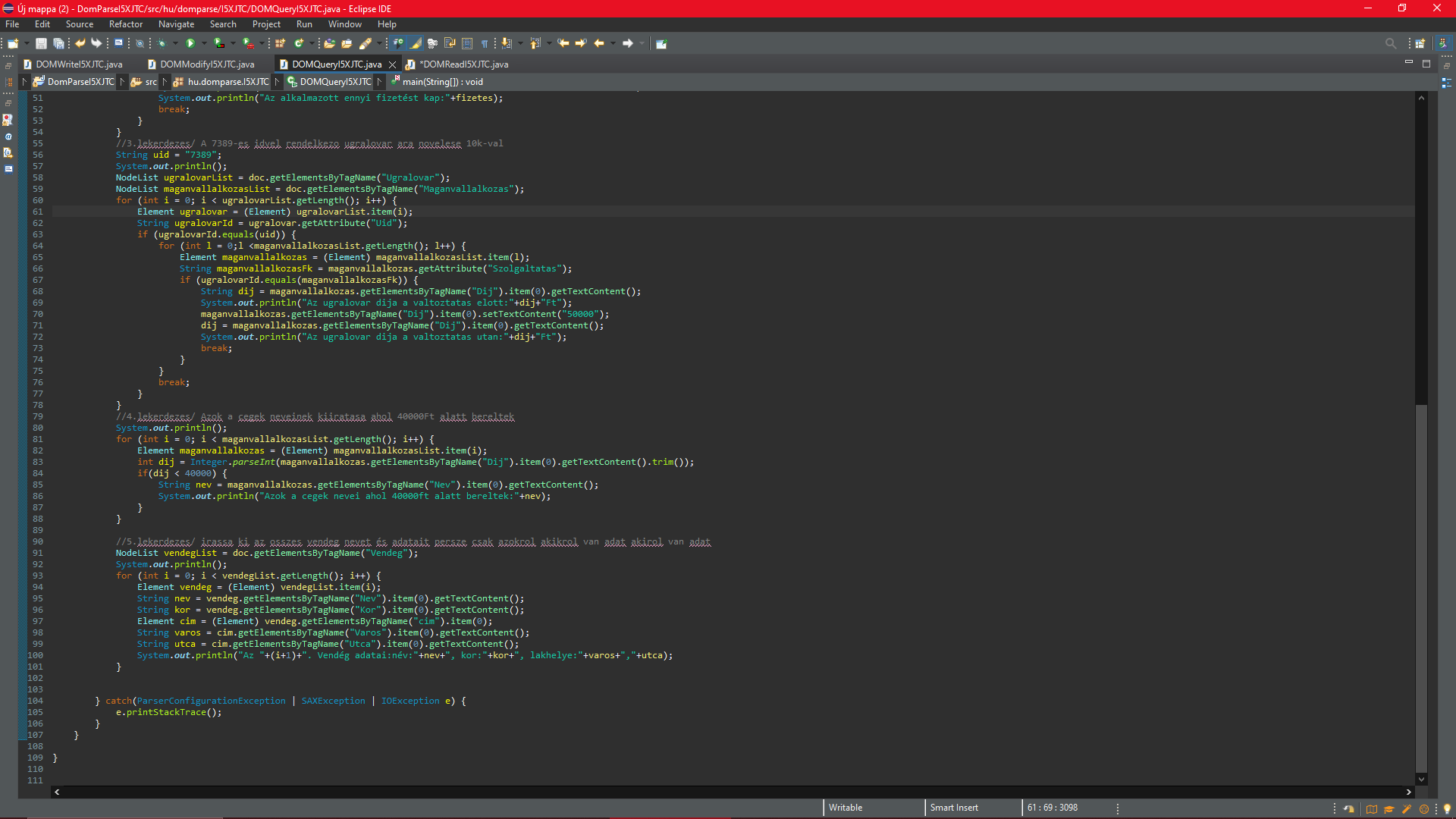
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

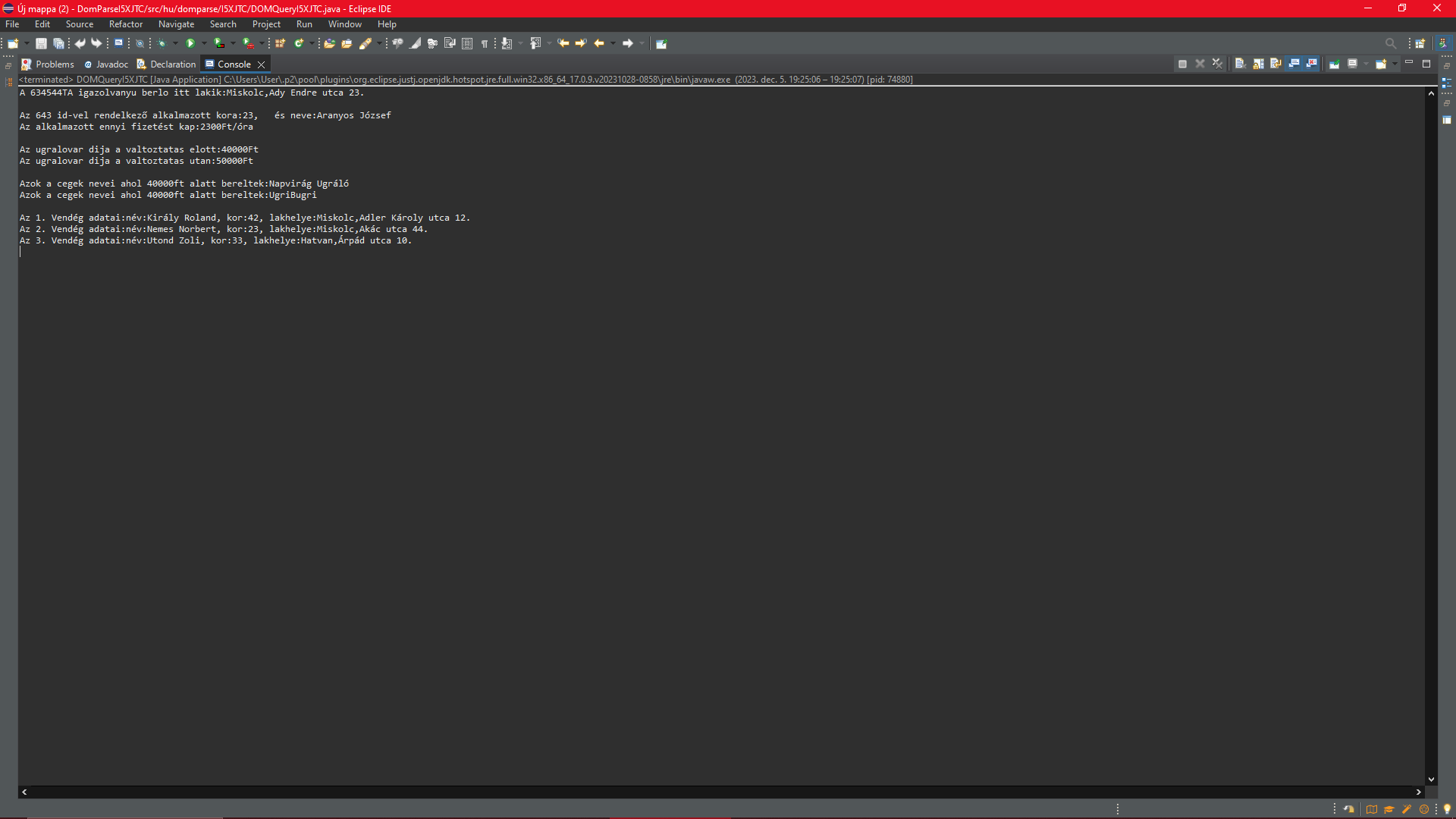
Automatikusan generált leírás  
A képen szöveg, elektronika, képernyőkép, képernyő látható

Automatikusan generált leírás  
  
2. feladat  
  
2a) adatolvasás (kód – comment együtt) – fájlnév: DOMReadNeptunkod.java  
  
Azzal kezdünk hogy behívjuk az XMLI5XJTC.xml fájlt majd a Dom-ot létrehozzuk, normalizáljuk a dokumentumot, majd kiválasztjuk egy fájlt írásra ami most az XMLfilekimenet.xml és lehetővé tesszük az írásra. Létrehozzuk a gyökér elemet, gyökér elem nevét csak úgy, mint a tulajdonságait. Majd a tulajdonsag-ba beletöltjük egy for ciklussal csak hogy a tulajdonságok elé be tudjunk helyezni egy fordított perjelet. Elkezdjük a kiíratást egyszerre konzolra és fájlba is. Legelőször megadjuk neki a verzió számot és hogy UTF-8 kódolást használjon majd Node listákat hozunk létre a gyökér elem gyerek elemeiből. Majd meghívjuk a listaKiir függvényt annyiszor ahány gyerek eleme van a gyökérnek, a gyerek elemeket és az íráshoz szükséges pW-t, rendesen elválasztva elküldjük. A végén pedig lezárjuk a gyökér elem bezárásával és ezt try catch-ben végezzük el ami hibát dob vissza ha nem sikerül. Nézzünk rá a listaKiir függvényre, alapból meghívjuk a for ciklust hogy a lista elemeit node-okban eltároljuk majd ebben a függvényben meghívunk egy másik függvényt a belsoElemekKiir fvg-t, ahol meg kell adni az elemet a szóköz távolságát és a wP PrintWriter-t. Ebben a függvényben rögtön egy if-fel kezdünk, hogy ellenőrizzük az elem fajtáját és ha megegyezik akkor enged csak tovább. Az adatokat elemekként eltároljuk az element-ben, majd eltároljuk a nevüket is, ugyebár itt is a tulajdonságokat külön választjuk és kapnak és fordított per jelet. Majd kiíratjuk a kozAdas függvény eredményét(ami röviden annyit tesz hogy beadunk egy számot és szám\*helyet fogja kihagyni), majd kiíratjuk az elemek nevét és tulajdonságait vagy tulajdonságát, ugyan ezt megtesszük fájlba írásnál, majd létrehozunk egy Listát gyerekElem-ként amibe beletöltjük az elem gyerekelemeit, aztán egy if-ben megkérdezzük hogy gyerekelem hossza megegyezik-e az 1-el és a gyerekelem típusa megegyezik-e egy szöveges típussal  
, majd ezután kiíratjuk a gyerekelemeket, ha meg nem teljesül a feltétel akkor for ciklusban hívja meg a belsoElemKiir függvényt, végig menve a procedúrán még egyszer keresve gyerek elemeket. Majd kiiratunk távolságot hogy meglegyen a formája a dokumentumnak, és lezárjuk a szülő elemet  
  
  
  
  
2b) adatmódosítás (kód – comment együtt) – fájlnév: DOMModifyNeptunkod.java  
  
Meghívjuk az XMLI5XJTC.xml fájlt, létrehozzuk a Dom-ot, ugyebár ezt a fájlt fogjuk módosítani egy függvény segítségével, szóval ezután meghívjuk a függvényt a modosit-tal csak a dokumentumot átadva, ezután lementjük a módósításokat az XML\_modositva.xml fájlba majd kiírhatjuk a konzolra is és ha ez nem megy a try parancs hibát dob, de nézzünk is rá a modosit függvényre. Lekreálunk egy Vendég listát a Vendeg szülő elemből. Majd a második alkalomnál a vendég nevét kicseréljük Jóska Pistára. A másodiknál is hasonló a történet csak ott a harmadik megjelenésnél cseréljük ki az Erkezest a Latogatas elemben 4:00-re, majd a 3.-nál az Ugralovar elem Helyigeny-ének 1. elemét 200x400x300 cm-esre. A 4.-nél a Beosztas-nak a Mettol elemjét 5:00-ra. Az 5.-nél az Alkalmazottak Kor elemjét 24-re. A 6.-nál a Maganvallalkozas Nev-ét a 2. alkalomnál HatalmasUgralo-ra. A 7.-nél pedig a Berlo-nek a Nev-ét a 3. alkalomnál Burok Tamásra.

  
  
  
  
  
  
2c) adatlekérdezés (kód – comment együtt) – fájlnév: DOMQueryNeptunkod.java  
  
Beolvassuk a fájlt, majd megalkotjuk a Dom-ot az XMLI5XJTC.xml fájlt felhasználva, ezekkel együtt a következő tartalmakat egy try-ba lett rakva ami megvizsgálja a fájl beolvasás, xml parser hibák, dokumentum hibák, konfigurációs hibák. Ezután 5 lekérdezéssel fogunk szemben állni mindegyik megoldásai között külön bekezdés hogy elválassza őket, 1. lekérdezés feladat: a 634544TA igazolvannyal rendelkezo berlo varosa+utcaja  
1. feladat magyarázat: eltároljuk a megadott igazolvány adatot, majd létrehozunk egy listát ami megkapja a Berlo-nek az elérését, majd sorra eltároljuk elemként a berlo gyerekeit, majd eltároljuk a Berloből kiolvasott id-t és ha megegyezik az eleinte megadott id-vel akkor eltároljuk a hozzá köthető várost és utcát, ezután meg kiíratjuk őket. Majd kilépünk a for ciklusból hogy ne menjen tovább.  
2. lekérdezés feladat: a 643-as idvel rendelkezo alkalmazott adatai  
2. feladat magyarázat: eltároljuk a megadott id-t majd létrehozzuk a listát ami megkapja az Alkalmazottak-nak az elérését, majd sorra eltároljuk elemként az alkalmazottak gyerekeit(nem valóságos rendes gyerekeket) majd eltároljuk az alkalmazottakból kiolvasott id-t és ha megegyezik az elején, kérdésben szereplő id-vel akkor kiíratja a korát fizetését és nevét az alkalmazottnak.  
3. lekérdezés feladat: a 7389-es idvel rendelkezo ugralovar ara novelese 10k-val  
3. feladat magyarázat: a feladatban megadott id-t eltároljuk majd létrehozunk két listát amik amit a fájlban lévő Maganvallalkozas-hoz és az Ugralovar-hoz csatolunk, aztán mindent ami az ugrálóvárban van eltárolunk elemként, majd eltároljuk az ugrálóvár Uid-jét és az összes ilyet megvizsgáljuk hogy egyezik-e a feladatban megadott id-vel, ha megegyezik akkor elkezdjük a magán vállalkozásnak is eltárolni az adatait elemként, és eltároljuk az idegen kulcsot is majd összevetjük az uid-vel, ha megegyezik akkor kiíratjuk a mostani árát az ugrálóvárnak majd hozzáadjuk 10k-t, eltároljuk és végül kiíratjuk.  
A break-ekkel kilépünk a for ciklusokból.  
4. lekérdezés feladat: azok a cegek neveinek kiiratasa ahol 40000Ft alatt bereltek  
4. feladat magyarázat: Mivel már a Maganvallalkozas elemeihez már csináltunk listát ezért már csak az elemeit el kell tárolni, majd a magánvállalkozás Dij elemeit átalakítjuk int-é és eltároljuk egy dij változóban, amit if-be rakunk és összevetjük hogy kisebb-e 40000-nél, ha kisebb akkor a magánvállalkozás nevét kiíratjuk.  
5. lekérdezés feladat: irassa ki az osszes vendeg nevet és adatait persze csak azokrol akikrol van adat  
5. feladat magyarázat: Egy vendég lista létrehozása a fájl Vendeg eleméből majd a vendég lista adatainak eltárolása elemként, majd a vendégből a cimet is eltároljuk elemként és változókként kiíratjuk a vendégek nevét, korát, városát, utcáját.





  
  
2d) adatírás (kód – comment együtt) – fájlnév: DOMWriteNeptunkod.java  
  
Dom létrehozása új dokumentumhoz, majd a gyökér elemet hozzuk létre, létrehozzuk a gyökér tulajdonságait, függvény hívásokat hozunk létre amikben megadjuk a dokumentumba kívánt adatokat, majd megadunk pár beállítást az elemek helyes beviteléhez. Ezután jön a kiírtatás, amit az XMLI5XJTC.xml-be fog elvégezni és a parancssorra. Ha a beállítások nem sikerültek dobja vissza a hibát. Ezután a legvégén van egy függvény amit minden main-en kívüli függvény meghív a neve: createElement, ez azért felelős hogy létrehozzon egy szülő elemet. Ezután már csak a main függvényben meghívott függvények(függvényekneve = gyökér elem gyerek elemi) maradnak. Ezekben a függvényekben a gyökér elem gyerek elemei és azok gyerek elemei és tulajdonságai és így tovább vannak. Ahogy a meghívásban megadott értékeket átadta a függvénynek úgy megalkotja az elemeket, tulajdonságokat és egymáshoz láncolja őket(szülő gyerek kapcsolatok megalkotása) ahogy saját magát is a gyökér elemhez.