**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

[**Algoritmusok**](http://www.inf.elte.hu/karunkrol/szervezet/tanszekek/Lapok/Inform%C3%A1ci%C3%B3sRendszerekTansz%C3%A9k.aspx) **és Alkalmazásaik****Tanszék**

**UML diagram szerkesztése Excel fájlon keresztül, és DDL scriptek generálása**

|  |  |
| --- | --- |
| **Témavezető:**  Nagy Sára  Programtervező Matematikus  Mesteroktató | **Szerző:**  Horváth Tibor  Programtervező informatikus  Nappali tagozat |

Budapest, 2021

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

SZAKDOLGOZAT TÉMABEJELENTŐ

**Hallgató adatai:**

Név: Horváth Tibor

Neptun kód: F6ZQMY

**Képzési adatok:**

**Szak:** programtervező informatikus, alapképzés (BA/BSc/BProf)

**Tagozat:** Nappali

Belső témavezetővel rendelkezem

***Témavezető neve:*** *Nagy Sára*

*munkahelyének neve, tanszéke: ELTE-IK, Algoritmusok és Alkalmazásaik Tanszék*

*munkahelyének címe: 1117, Budapest, Pázmány Péter sétány 1/C.*

*beosztás és iskolai végzettsége: mesteroktató, programtervező matematikus*

**A szakdolgozat címe:** UML diagram szerkesztése Excel fájlon keresztül, és DDL scriptek generálása.

**A szakdolgozat témája:**

*(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben szakdolgozat témájának leírását )*

Egy jó szoftver megvalósításának elengedhetetlen követelménye egy jó terv. Ehhez számtalan tervezőeszköz áll rendelkezésünkre ingyenes és vársárlást igénylők egyaránt. Ugyanakkor az ingyenes szoftverek gyakran nem rendelkeznek ugyanazzal az eszközkészlettel, vagy a kevésbé fontos funkciók nem felhasználóbarát módon kerültek megvalósításra.

A dolgozatom célja egy ilyen, nyílt forráskódú modellező eszköz funkcióinak kibővítése, és ezzel párhuzamosan használatának megkönnyítése. Mindezt, egy szintén nyílt forráskódú programozási környezetben. Ezt több módon is szeretném elérni.

Egyfelől a modell szerkesztőből exportálható egy osztálydiagram modellje egy Excel fájlba, ahol az adatok könnyebben szerkeszthetővé válnak, illetve az elemek felvétele is egyszerűsödik. Ezenfelül egy Excel fájl importálásával lehetőség nyílik a teljes modell szöveges létrehozására.

Másfelől egy UML osztálydiagram adatbáziskezelő szoftverbe való átvezetését megkönnyítendő. DDL scriptet generáló funkció megvalósítása, amely többféle SQL implementációra is képes generálni.

Budapest, 2021.06.13.

Tartalomjegyzék

[1 Bevezetés 5](#_Toc91677138)

[2 Felhasználói dokumentáció 6](#_Toc91677139)

[2.1 Felhasznált eszközök 6](#_Toc91677140)

[2.1.1 Papyrus 6](#_Toc91677141)

[2.1.2 Eclipse Modeling Framework 6](#_Toc91677142)

[2.1.3 SWT – Standard Widget Toolkit 6](#_Toc91677143)

[2.1.4 Apache POI 7](#_Toc91677144)

[2.2 Rendszerkövetelmények 7](#_Toc91677145)

[2.3 Telepítés 8](#_Toc91677146)

[2.4 Használat 9](#_Toc91677147)

[2.4.1 Excel – modell konvertáló 9](#_Toc91677148)

[2.4.2 DDL script generátor 16](#_Toc91677149)

[3 Fejlesztői dokumentáció 18](#_Toc91677150)

# Bevezetés

Egy szoftver megalkotásának elengedhetetlen követelménye egy jó terv. A hatékony tervezéshez pedig egy jó tervező eszközre van szükség. Az elérhető eszközök többsége vásárlást vagy előfizetést igényel, amit nem mindenki engedhet meg magának. Ugyanakkor az ingyenesen elérhető szoftverek gyakran nem rendelkeznek ugyanazzal az eszközkészlettel mint versenytársaik.

A szakdolgozatom egy ilyen ingyenesen elérhető eszköz a Papyrus modellező funkcionalitásának kibővítését célozza meg.

Egyfelől lehetővé téve, hogy UML osztálydiagram exportálását egy egyedi szabvány szerint felépített Excel fájlba, így akár bármilyen egyéb eszköz használata nélkül módosíthatóvá téve a modell szerkezetét. Illetve egy megfelelő szerkezettel rendelkező Excel fájl importálása a projektbe így létrehozva vagy módosítva az UML elemeket.

Másfelől egy a Papyrus modellező felületével összekötött képernyő, amin DDL script generáláshoz szükséges többlet adatokkal lehet ellátni az Interfészek, és Osztályok adattagjait. A felvehető adatok tartalmazzák az adattípust, és opcionálisan annak paramétereit. Az attribútum jelölhető elsődleges kulcsnak, idegen kulcsnak megadható a referált Entitás és az azon belül referált attribútum.

# Felhasználói dokumentáció

## Felhasznált eszközök

### Papyrus

A Papyrus [1] egy nyílt forráskódú modell-alapú tervező eszköz, amely használható mind Eclipse bővítményként, mind pedig önálló alkalmazásként. Dolgozatomban az Eclipse bővítmény változatot használtam.

A Papyrus egy grafikus modellező eszköz UML2 modellek szerkesztéséhez, a grafikus részek mind támogatják a kibővítést.

### Eclipse Modeling Framework

Az Eclipse Modeling Framework [2] egy modellezési keretrendszer, amely alkalmas egyéb eszközök és alkalmazások készítésére strukturált adatmodell alapján. Egy XMI-ben leírt modell specifikáció kezeléséhez az EMF számos eszközt kínál, mind Java kód generáláshoz, a modell grafikus szerkesztéséhez, illetve egy alap grafikus szerkesztő felületet is.

Az EMF egy gyakran használt standard adatok modellezésére, számos keretrendszer erre épít, többek között a Papyrus is.

### SWT – Standard Widget Toolkit

A felhasználó felülte megvalósítására az SWT-t használtam fel, amely egy nyílt forráskódú widget fejlesztő eszközkészlet.

A Standard Widget Toolkit vagy SWT [3] a Java programozási nyelv ablakkezelésére és grafikus felhasználói felületek (GUI) létrehozására szolgáló komponensgyűjteménye. Ellentétben az AWT-bel és a Swinggel, az SWT nem része a Java API-nak. Az SWT egy alternatíva az AWT és a Swing helyett.

Az SWT-t eredetileg az IBM fejlesztett ki, de jelenleg az Eclipse Foundation végzi a fejlesztést és a karbantartást. Az első SWT használó alkalmazás az Eclipse volt.

### Apache POI

Az Excel fájlok kezeléséhez az Apache POI eszköz tárát használtam.

Az Apache POI [4] egy az Apache Software Foundation által fejlesztett projekt, amely tisztán Java könyvárakat biztosít a Microsoft Office formátumú fájlok kezelésére.

Komponensei közül az XSSF (XML SpreadSheet Format) került felhasználásra

## Rendszerkövetelmények

Mivel a dolgozat az Eclipse IDE keretrendszer része, illetve Microsoft Office Excel fájlok kezelését is végzi ezért hardveresen elég a két eszköz hardveres követelményinek megfelelnie.

Szoftveresen szükség van:

A Microsoft Office Programcsomagra, amely a Microsoft Store-ból könnyen beszerezhető.

Mivel a program Java nyelv specifikus ezért szükséges a Java Development Kit 11-es verziója. <https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html> [Elérhető 2021.12.10]

Az Eclipse IDE 2021-03 (4.19) verziójának telepítésére, amely itt: <https://www.eclipse.org/downloads/packages/release/2021-03/r> [Elérhető 2021.12.10] megtalálható.

A Papyrus plugin telepítésére az Eclipse-be. Ezt a következőképpen lehet megtenni, ki kell választani a *Help > Install new software…* menüpontot majd az Eclipse update site-on megkeresni az 1. számú ábrán láthatóak szerint:A képen szöveg látható

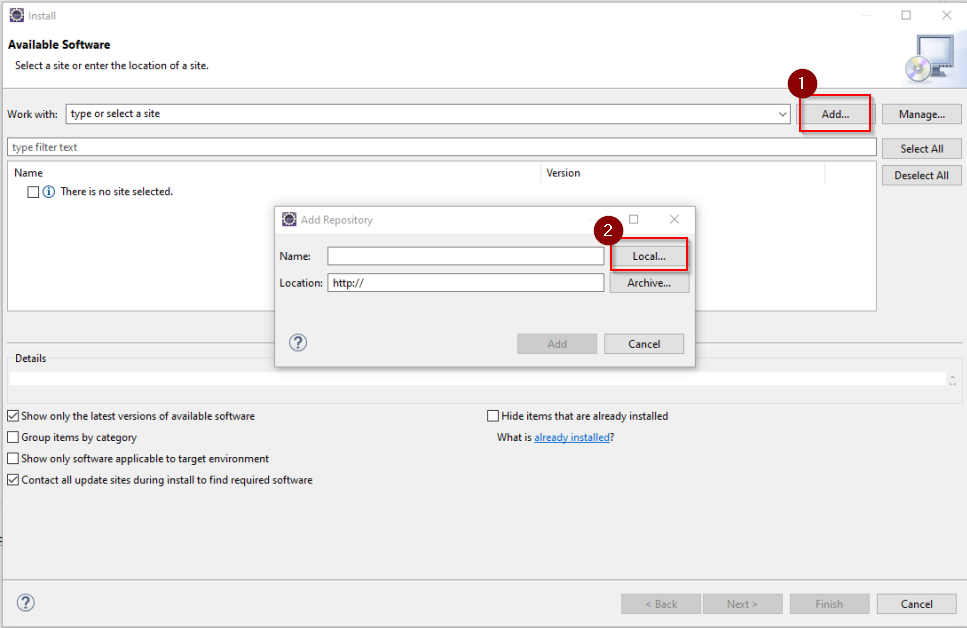
Automatikusan generált leírás

1. ábra Papyrus telepítése Eclipse Update site-on keresztül

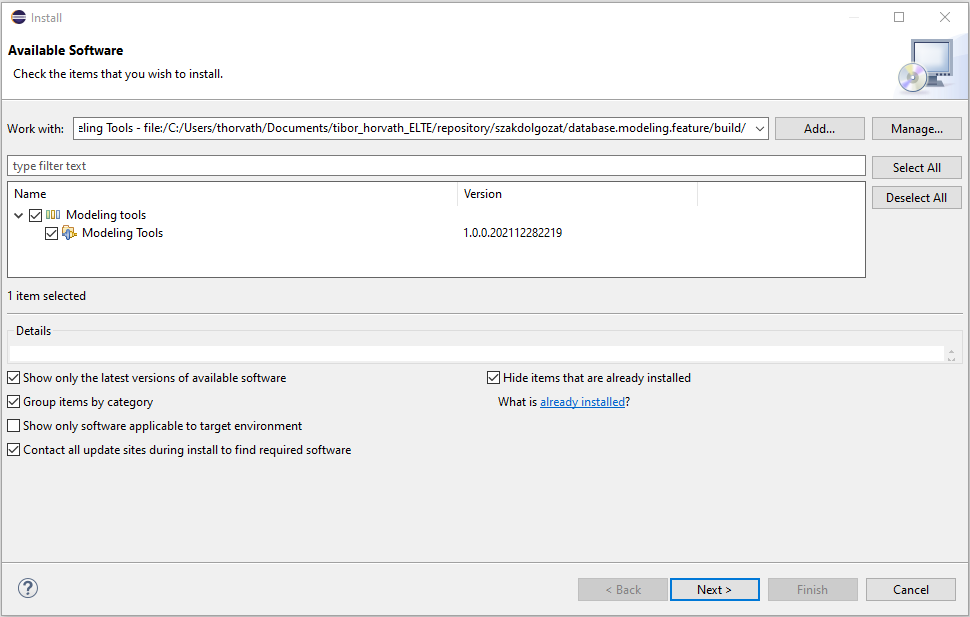
Ezután már csak végig kell lépkedni a telepítő ablakokon és egy újraindítást követően már használatra is kész.

## Telepítés

Ha a Rendszerkövetelményekben bekezdésben leírt eszközök rendelkezésre állnak akkor meg lehet kezdeni a modellező eszköz telepítését, a következőt kell tenni.

1. Eclipse elindítása után a *Help > Install new Software…* menüt kell kiválasztani.
2. Ezután az *Add* majd a *Local* lehetőséget kell választani a 2. ábrán láthatóak szerint, az itt felugró fájlkezelő ablakban meg kell keresni a szoftver build mappáját és azt kiválasztani.

2. ábra Modeling Tools forrás megkeresése

1. A név mezőbe egy tetszés szerinti elnevezés írható, de célszerű olyat választani, amiről később is felismerhető.
2. Ezt követően a listából ki kell jelölni a Modeling tools-t a 3. ábrán látható módon és végig követni a telepítő ablakokat.
3. Az Eclipse újraindítását követően az eszközt már használatba is lehet venni.

3. ábra Modeling tools plugin kiválasztása

## Használat

### Excel – modell konvertáló

Az Excel – modell konvertáló egy eszköz, amelynek célja az osztálydiagrammok modellezésének egyszerűsítése. Azzal, hogy akár a modell teljes tartalma definiálható egy munkafüzetben olyanok is modellezhetnek, akiknek nem áll rendelkezésükre a szükséges eszközkészlet, vagy csak nem ismerik annak használatát.

A program jelen állapotában még nem képes minden elemet kezelni ezért az eszközkészleten kívüli elemek használata esetén hiányos lehet a generált Excel, vagy modell.

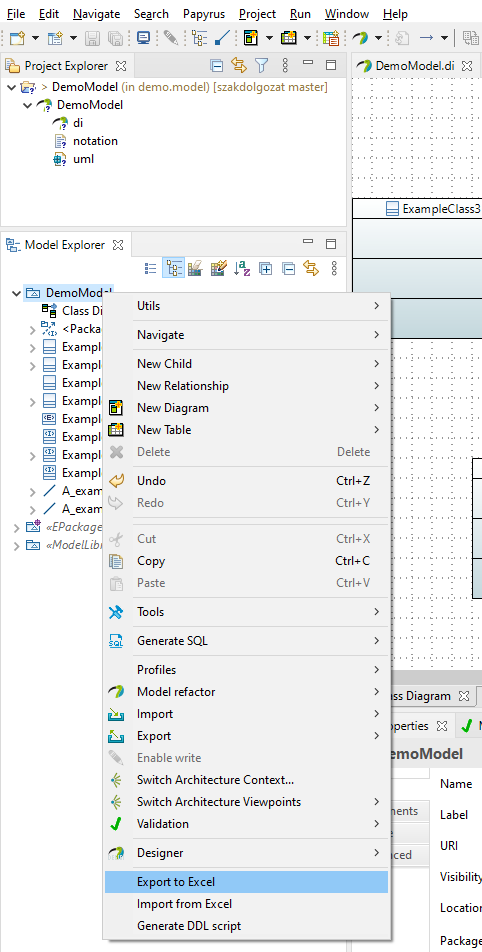
A jelenleg támogatott modell elemek listája:

* Class
* Interface
* Enumeration
* DataType
* Property
* Operation
* Association
* Generalization
* Interface Realization

#### Exportálás

Az exportálás funkció használatához át kell váltani Papyrus perspektívába az Eclipset ezt a *Window > Perspective > Open perspective > Other…* menüben lehet megtenni. Illetve a *Window > Show view > Other…* menüben a **Model Explorer** ablak kinyitása**.**

Ha ez megtörtént akkor ki kell nyitni az exportálni kívánt Papyrus modellt, és a Model Explorer képernyőn a modell egy Package típusú elemére jobb egérgombbal rákattintani. Bármi másra kattintva a menüpont le van tiltva. Az itt felugró menüben található az *Export to Excel* lehetőség.



4. ábra Export to Excel menüpont kiválasztása

A kiválasztása után megnyílik a Windows fájlkezelő és meghatározható a célfájl helye, illetve neve. A sikeres generálás lefutása után automatikusan kinyílik a generált munkafüzet.

Fontos tudni, hogy a generáláshoz kiválasztott Package elemnek van jelentősége mivel csak a hierarchiában a kiválasztott elem alatt szereplő Entitások kerülnek exportálásra.

A generált munkafüzetben az feltűnhet, hogy minden elem mellé generálódik egy **Xmi ID** megnevezésű értékis. Ez egy az EMF által generált egyedi azonosító, amit az objektum semmilyen nemű változtatása nem befolyásol így mindig pontosan beazonosítható marad. Ennek egy meglévő modell Excel-en keresztüli módosítása esetén van jelentősége.

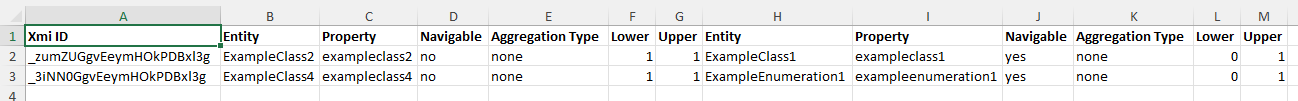
#### Munkafüzet szerkezete

A generált, és az importálásra alkalmas munkafüzeteknek követnie kell egy szabványos szerkezetet.

A következő munkalapoknak kell léteznie:

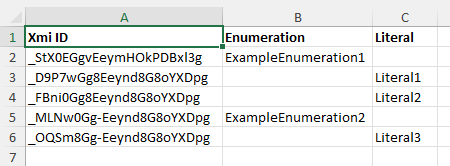
* DataTypes – az adattípusok felsorolásra, oszlopai:
  + Xmi ID – egyedi azonosító,
  + A képen a DataTypes munkalap szerkezete látható
    TypeName – adattípus neve

5. ábra DataTypes munkalap szerkezete

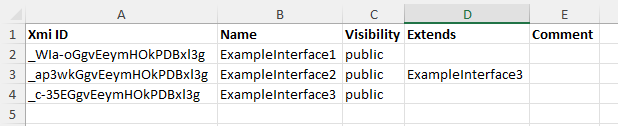
* Associations – az asszociációk felsorolása, oszlopai:
  + Xmi ID – egyedi azonosító
  + Entity – végpont entitás neve
  + Property – végpont attribútum neve
  + Navigable – navigálhatóság
  + Aggregation Type – aggregáció fajtája
  + Upper és Lower – a végpont számossága

6. ábra Asszociációk munkalap szerkezete

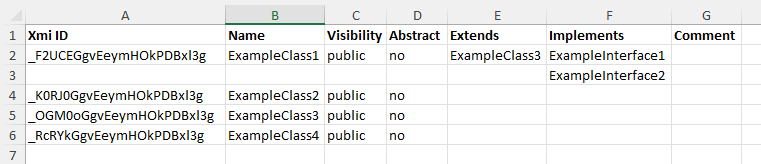
* Enumerations – az enumerációk és elemeik felsorolása, oszlopai:
  + Xmi ID – egyedi azonosító
  + Enumeration – enumeráció neve
  + Literal – enumeráció adattagjai



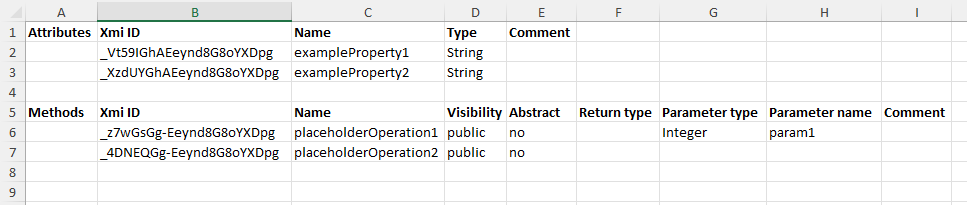
7. ábra Enumerations munkalap

* Interfaces – az interfészek felsorolása, oszlopai:
  + Xmi ID – egyedi azonosító
  + Name – név
  + Visibility – láthatóság
  + Extends – leszármazott interfész neve
  + Comment – komment

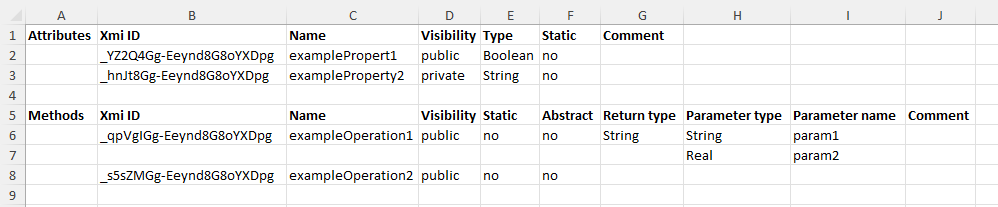
8. ábra Interfaces munkalap

* Classes – az osztályok felsorolása, oszlopai:
  + Xmi ID – egyedi azonosító
  + Name- név
  + Visibility – láthatóság
  + Abstract – absztrakt-e az osztály
  + Extends – leszármazott osztály
  + Implements – implementált interfész
  + Comment – komment

9. ábra Classes munkalap

* Minden az Interfaces vagy Classes lapon definiált entitáshoz tartoznia kell egy azonos névvel létrehozott munkalapnak, ide kerülnek az attribútumok és a metódusok.
* Interfész attribútumait és metódusait tartalmazó munkalap, oszlopai:
  + Attributes – semmilyen érték nem kerül ide, csak arra szolgál, hogy jelölje az attribútumok kezdetét
  + Xmi ID – egyedi azonosító
  + Name – név
  + Type – típus
  + Comment – komment
  + Methods - semmilyen érték nem kerül ide, csak arra szolgál, hogy jelölje a metódusok kezdetét
  + Visibility – láthatóság
  + Abstract – absztrakt-e a metódus
  + Return type – visszatérési érték típusa ha van
  + Parameter type – paraméter típusa
  + Parameter name – paraméter neve

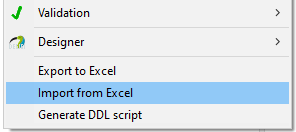
10. ábra Interfész részletező munkalap

* Osztály attribútumait és metódusait tartalmazó munkalap, oszlopai:
  + Attributes – semmilyen érték nem kerül ide, csak arra szolgál, hogy jelölje az attribútumok kezdetét
  + Xmi ID – egyedi azonosító
  + Name – név
  + Visibility - láthatóság
  + Type – típus
  + Static – statikus-e az attribútum
  + Comment – komment
  + Methods - semmilyen érték nem kerül ide, csak arra szolgál, hogy jelölje a metódusok kezdetét
  + Static – statikus-e a metódus
  + Abstract – absztrakt-e a metódus
  + Return type – visszatérési érték típusa, ha van
  + Parameter type – paraméter típusa
  + Parameter name – paraméter neve

11. ábra Osztály részletező munkalap

#### Importálás

Abban az esetben, ha új modellt hozunk létre importálással, akkor szükség van egy Papyrus modell projektre létrehozására.

Az exportáláshoz hasonlóan a Model Explorer képernyőn a modell egy Package típusú elemére jobb egérgombbal kell rákattintani. Bármi másra kattintva a menüpont le van tiltva. Az itt felugró menüben található az *Import from Excel* lehetőség.

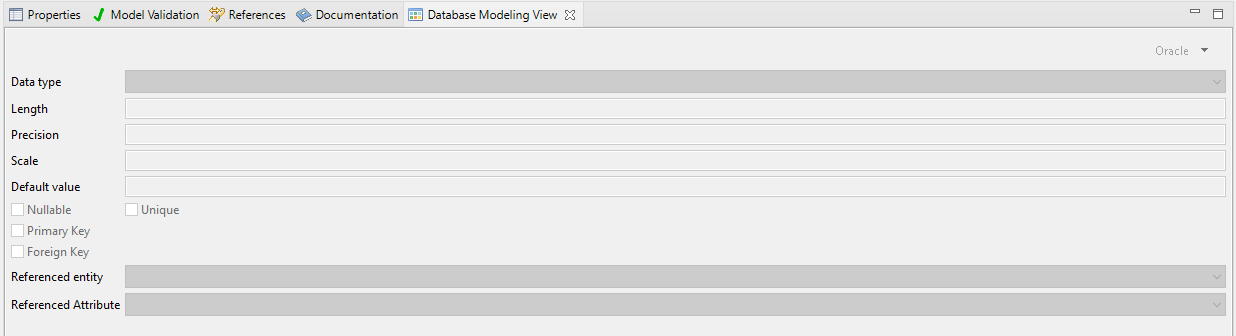
12. ábra Import from excel menüpont

A kiválasztása után megnyílik a Windows fájlkezelő, ahol kiválasztható az importálni kívánt Excel fájl. A fájlkezelő szűrést végez és csak a „.xlsx” kiterjesztésű fájlokat jeleníti meg. Sikeres importálás esetén a *Model Explorer* nézetben a kijelölt Package alatt megjelennek az importált elemek.

Abban az esetben, ha a kiválasztott fájl szerkezete importálásra nem alkalmas hibaüzenetek jelennek meg a képernyőn, amelyek közlik a hiba típusát és felsorolják az érintett elemeket. Ezek az *Error Log* nézetben is megjelennek.

### DDL script generátor

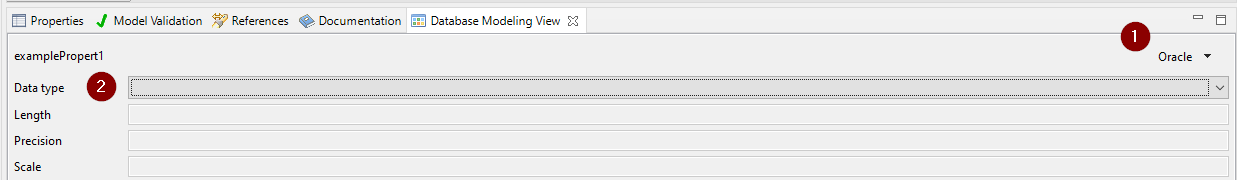
A DDL generáláshoz szükséges extra adatokat a „Database Modeling View” segítségével lehet a modellhez hozzáadni. Ez a *Window > Show view > Other… > Database Modeling Tools > Database Modeling View* menüből nyitható ki.



13. ábra Database Modeling View letiltott elemekkel

Megnyitás után az látható, hogy a képernyő minden eleme le van tiltva, ez mindaddig így is marad ameddig nincs kijelölve egy Property típusú elem a Papyrus diagram szerkesztő vagy a *Model Explorer* felületén.

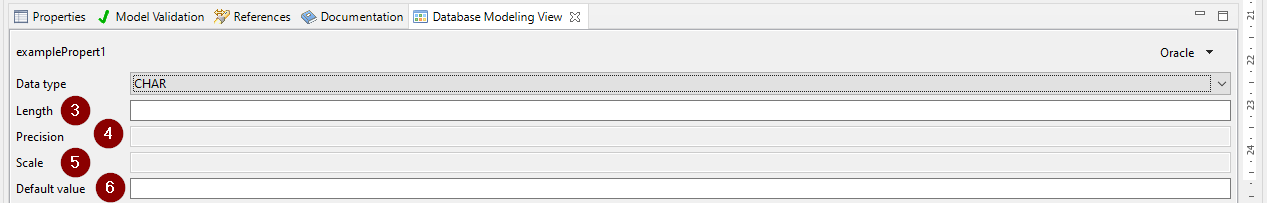
Ha ez megtörtént elérhetővé válik az elemek egy része. Az adatbázis implementáció váltását kezelő legördülő menü *(1)*, és a kiválasztott adatbázis implementáció típusait felsoroló legördülő menü *(2)*.



14. ábra Database Modeling View aktív kiválasztással

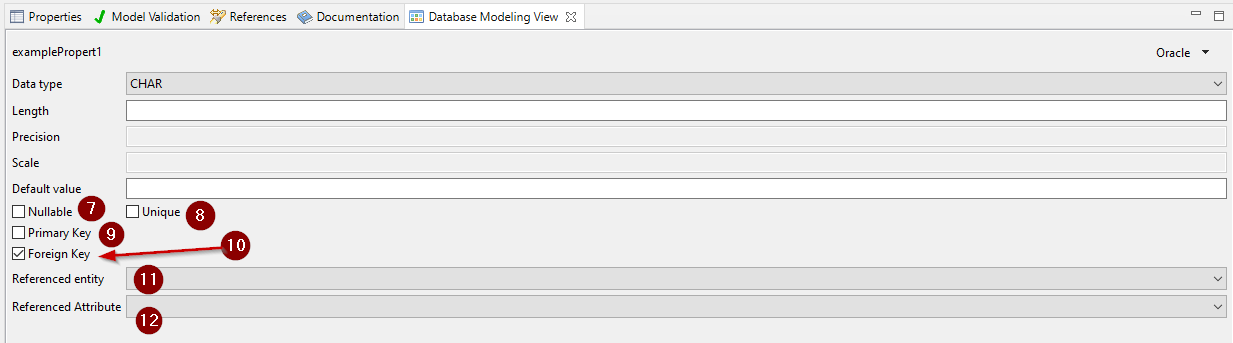
Adatbázis implementáció váltása esetén a modellbe felvett korábbi implementációhoz tartozó adatok mentésre kerülnek, majd az újonnan kiválasztott implementáció adatai kerülnek betöltésre, ha léteznek. Ekkor frissül a „Data Type” *(2)* legördülő menü értékkészlete.

Ha kiválasztásra került egy adattípus akkor az ahhoz tartozó adatok felvitelére alkalmas mezők is elérhetővé válnak. Szöveges típus esetén ez általában a Length *(3)* amellyel a szöveg maximális hosszát és a Default value *(6)* amivel alapértéket állíthatunk be. Szám típusok esetén a Precision *(4)* amivel a szám hosszát, Scale *(5)* amivel a tizedesvessző utáni rész hosszát és Default Value *(6)* amivel alapértéket adhatunk meg.

**

15. ábra Típus és a hozzá tartozó input mezők

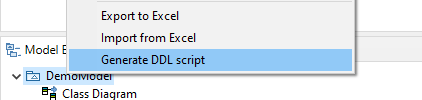
A típus beállításokon kívül a következő megszorítások is alkalmazhatóak az adattagon Nullable *(7)* ezt bejelölve null értéket is felvehet az adatbázisban, Unique *(8)* ezt bejelölve az oszlopban minden érték csak is egyszer szerepelhet, Primary Key *(9)* ezt bejelölve az oszlop elsődleges kulcs lesz. A Foreign Key *(10)* bejelölésével idegenkulcs az oszlopból, illetve elérhetővé válnak a referált entitás *(11)* illetve referált attribútum *(12)* beállítási mezői.

Idegenkulcs definiálásakor a Referenced Entity menüben azok az objektumok kerülnek felsorolásra, amelyekbe vezet az éppen szerkesztett attribútum szülő objektumából asszociáció.

16. ábra Database Modeling View megszorítások magyarázat

Sikeresen kiválasztott referált entitás esetén felsorolásra kerülnek az elem attribútumai a Referenced Attribute mezőben. Fontos, hogy az asszociáció két végén szereplő attribútumok definiált SQL típus megegyezzen, ha ez nem teljesül a program hibaüzenettel jelzi.

Ha egy adott elemhez definiálásra került SQL típus akkor megjelenik rajta a <<Column>> sztereotípus, továbbá, ha elsődleges kulcsnak lett megjelölve akkor a <<PK>>, idegenkulcsnak jelölés esetén a <<FK>> sztereotípus.

**Miután a modell minden Property eleméhez meg lett definiálva legalább egy adattípus lehetőség van DDL script generálására a modell gyökérelemére vagy valamelyik Pacakge típusú elemre jobb egérgombbal rákattintva és a *Generate DDL script* lehetőséget kiválasztva.

17. ábra Generate DDl script menüpont

A korábbiakhoz hasonlóan a felugró fájlkezelő ablakban ki kell választani a cél könyvtárat és nevet adni a generált fájlnak.

# Fejlesztői dokumentáció

## Megoldási terv

A projekt két fő részre osztható az egyik, ami a modell és az Excel fájl közötti konverziót végzi, és a másik, ami a modell kiegészítése SQL típus adatokkal és a script generálása.

A modell – Excel transzformálásához az Apache POI könyvtár került felhasználásra, ezzel könnyen megvalósítható a két irányú fájlkezelés.

A modell SQL specifikus adatokkal való kiegészítéséhet egy Papyrusz profil elkészítésére van szükség, amibe definiálásra kerültek a *DataModel, Column, PK, FK* sztereotípusok. Ezek kerülnek a helyzetnek megfelelően alkalmazásra a modell elemein. továbbá mivel több SQL implementáció kezelésére is szükség van ezért az egyes implementációk adatait el kell tárolni úgy, hogy az a modellből könnyen elérhető legyen és ne bonyolítsa fölöslegesen a kódot. Így minden adatbázis váltáskor a korábbi implementációhoz tartozó adatokkal ellátott elemek szerializálásra kerülnek Gson segítségével és a modell könyvtárában eltároljuk, az implementáció nevével megegyező kiterjesztésű fájlban. Ennek megfelelően az újonnan választott implementációhoz, ha létezik korábban eltárolt adat azt alkalmazni kell a modellen.

Az adatok SQL specifikus adatok felviteléhez szükséges interaktív felületet kell készíteni ehhez az Eclipse – Standard Widget Toolkit megfelelő eszközöket ad.

## Fejlesztői környezet

A fejlesztéshez elengedhetetlen eszközök megegyeznek a szoftver használatához szükséges elemekkel. Telepíteni kell a **Microsoft Office** Programcsomagot, a **Java** **Development Kit 11**-es vagy újabb verzióját, az **Eclipse IDE 2021-03** vagy egy újabb kiadását, és a Papyrus modellező eszközt.

Ezeken kívül a következő bővítményekre lehet szükség:

**UML Extender SDK és (Developer resource)** plugin – Az UML elemek programozott kezeléséhez, létrehozásához. A definiált profilhoz tartozó kód generálásához.

**EMF – Eclipse Modeling framework SDK** – Szükséges a plugin projektek fejlesztéséhez.

EMF specifikus eszközök használatához szükséges

**Eclipse e4 Tools Developer Resource** – Szükséges a képernyő Eclipse-be integrálásához, illetve a Papyrus Model Explorer felületének kiterjesztéséhez a generálási menüpontok elhelyezése céljából.

Ezek mind telepíthetőek a *Help > Install new software…* menüből az editorhoz tartozó update site-on keresztül.

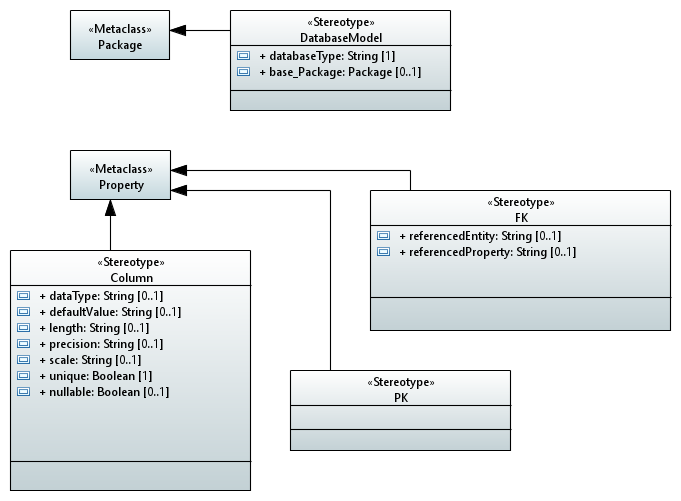
## Komponensek

A plugin projektek kapcsolata
A dolgozatot alkotó plugin projektek egy „custom.modeling.feature” feature projektben vannak összefogva.

18. ábra Pluginek kapcsolata a projektben

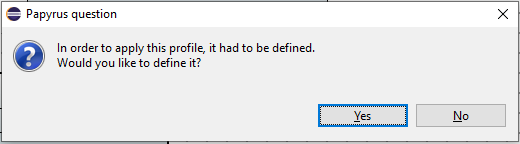
### uml.profile

Az „*uml.profile”* projekt ahol definiálásra kerül a profil és a sztereotípusok. A profilt egy Papyrus „Profile architektúrájú” projektben található a resources mappán belül. Ezt megnyitva egy hétköznapi modellhez hasonlóan szerkeszthető.



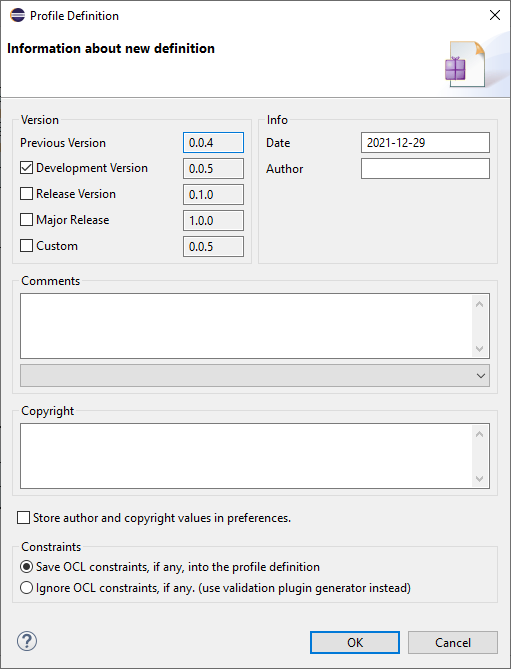
. ábra A profil szerkezete

A <<Metaclass>> elemek határozzák meg hogy a sztereotípus milyen elemekre applikálható. Ebben az esetben *DatabaseModel* csak Package-re, a *Column*, *PK*, és *FK* pedig csak Property-re alkalmazható.

Ha a profil módosításra kerül mentéskor az editor megkérdezi, hogy definiálni szeretnénk-e a profilt mivel az szükséges az alkalmazásához.

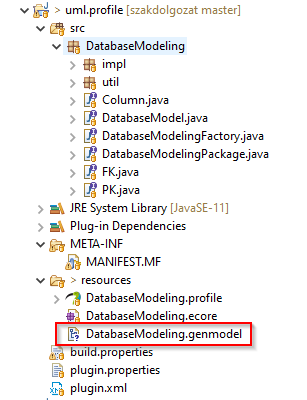
20. ábra Profil definiálás dialógus

Itt az igen opciót kell választani, majd a következő felugró képernyőn meghatározni a verzió adatokat.



21. ábra Profil verzió adatainak definiálása

Ezzel a profil új változata el is készült, és a Papyrus editor grafikus felületén keresztül alkalmazható is a modellen. De a programozott felhasználáshoz frissíteni kell a projekt src mappájában található kódot, ami EMF generátum. Ehhez a profil modell mellett található „.genmodel” kiterjesztésű EMF Generator Model fájlon kell jobkattintás után a *Reload…* menüpontot választani.

Ha ez megtörtént a következő felugró ablakon az UML model lehetőséget kell kiválasztani és továbblépni. A következő oldalon a *Browse Workspace…-*re kattintva meg kell keresni aprofilhoz tartozó UML fájlt majd a *Load*-ra kattintani.

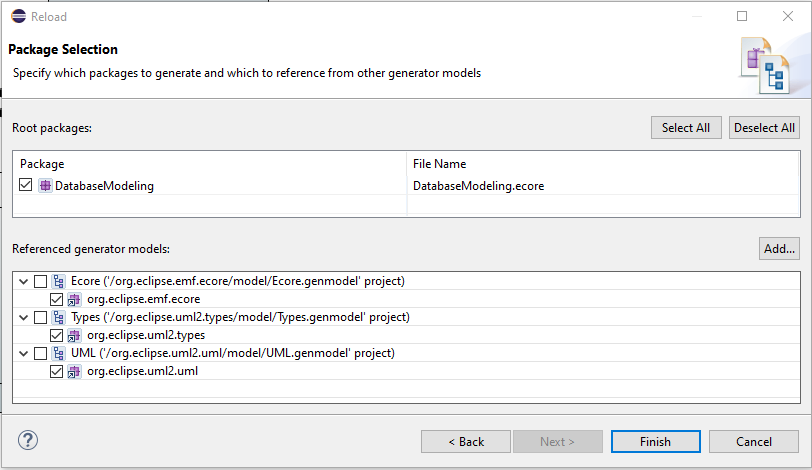
22. ábra uml.profile projekt szerkezet

A képen szöveg látható

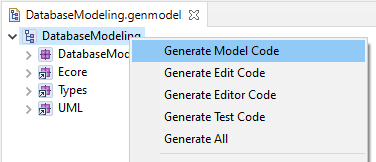
Automatikusan generált leírásEkkor egy figyelmeztető üzenet ablak fog felugrani sok-sok figyelmeztetéssel, de ezek a profil definícióból erednek ezért nem kell foglalkozni vele. Persze módosítások végrehajtása után érdemes áttekinteni mert okozhat hibát.

23. ábra Profil újra töltéskor megjelenő figyelmeztetések

Tovább lépve az alábbi képernyő jelenik meg ahol kiválasztható, hogy mely csomagok kerüljenek generálásra és melyek legyenek csak referálva.



24. ábra Generáladó, és referált csomagok meghatározása

Ezután az editor automatikusan kinyitja a „.genmodel” fájlt, ha ez nem történne meg akkor ki kell nyitni. Majd az editorban megjelenő legfelső elemre jobb egérgombbal kattintva válasszuk a *Generate Model Code* lehetőséget.

25. ábra Genmodel - generate model code menüpont

Ezt követően már programozható a frissített profil.

### modeling.tools.util

A leggyakrabban használt kódok gyűjteménye.

#### database.modeling.util.resource

A „database.modeling.util.resource” csomagban vannak a fájlkezeléssel, illetve az Eclipse felületéről érkező események kezelésével kapcsolatos hasznos függvények.

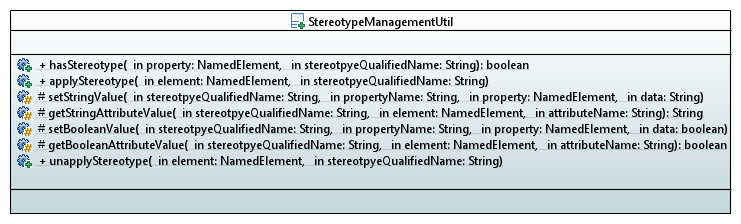
Az *EclipseResourceUtil* osztály tartalamazz minden a fájlkezeléssel kapcsolatos függvényt, az elérési utak kezelésétől a fájlok írásáig.

Az *EclipseSelectionUtil* függvényei kezelik a Model Explorer felületéről beérkező kijelöléseket.

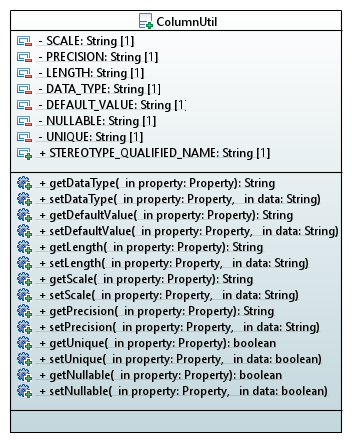
#### database.modeling.util.stereotype

A „database.modeling.util.stereotype” függvényei kezelnek minden a profillal és sztereotípusokkal kapcsolatos műveletet.

A *StereotypeManagementUtil* osztály tartalmazza minden az általánosított függvényeket minden sztereotípus applikálásához, eltávolításához és típus szerint általánosított getter, setter metódusokat is tartalmaz.

A sztereotípus specifikus util osztályok, mint *a ColumnUtil, PKUtil…* tartalmazzák az adott sztereotípus minősített nevét, mint publikus statikus változó, ami a *StereotypeManagementUtil* általánosított függvényeinek használatához szükséges. Illetve ugyanennek az osztálynak a típus szerint általánosított getter setter függvényeit becsomagolva definiálva van az osztály minden attribútumához egy getter és setter függvény. Emiatt fel vannak véve az attribútumok nevei is mint privát statikus változók.

26. ábra StereotypeManagementUtil osztály diagram



27. ábra Példa egy sztereotípus specifikus util osztály szerkezetére

A *ProfileUtil* a korábbiakhoz hasonlóan az elemeken való alkalmazást hivatott megkönnyíteni.

#### database.modeling.util.uml

A csomag mindössze egyetlen *ModelObjectUtil* osztállyal rendelkezik, amely a modellben található elemek típus vagy azonosító szerinti szűrését segíti. Metódusai mind publikusak és statikusak.

### tools.apache-poi

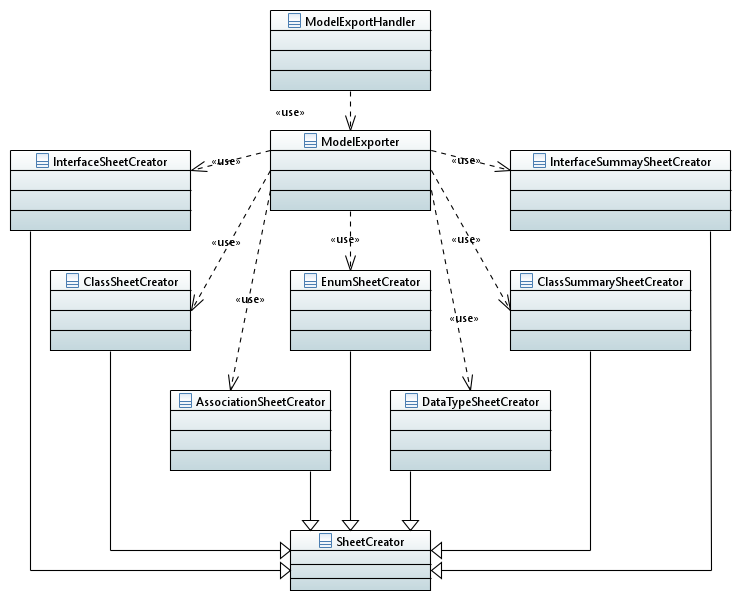
A projekt nem tartalmaz kódot mindössze az Apache-POI library tárolására céljából készült.

### uml.excel.transfer

#### Excel exportálás

A „*model.transfer.export*” csomagban található minden a modell fájllá alakításához szükséges objektum konverzió. A végrehajtás a *ModelExportHandler* osztályból indul ki, ez kezeli a képernyőn megjelenő menüpontot, és annak elérhetőségét. Mielőtt a menüpont megjelenne lefut a @CanExecute **boolean** canExecute(@Named(IServiceConstants.***ACTIVE\_SELECTION***) ISelection selection) függvényami a paraméterén keresztül megkapja az aktuális kijelölést majd eldönti hogy az adott elem megfelelő-e (Pacakage típusú). Igaz értékkel való visszatérés esetén lefut a @Execute **public** **void** execute(@Named(IServiceConstants.***ACTIVE\_SELECTION***) ISelection selection) amiben a *ModelExporter* osztályon keresztül elindul a generálás.

A *ModelExporter* export függvényén belül típusok (munkalapok) szerint kerülnek az objektumok összeszedésre a *ModelObjectUtil* osztályait felhasználva.



28. ábra model.transfer.export csomag osztályainak a kapcsolata

*Az AssociationSheetCreator és DataTypeSheetCreator* osztályok működési elve nagyon hasonló. A modellből típus szerint összegyűjtésre kerülnek a munkalaphoz szükséges elemek majd ezeken végig iterálva minden elem adatait beleírjuk a munkalapba.

A *ClassSheetCreator* és az *InterfaceSheetCreator* esetén osztályok esetén a Property és Operation elemeket is ki kell írni. Így a többi munkalaphoz képest az a változás történt itt, hogy a munkalap vízszintesen két szegmensre lett osztva Attribútumok és Metódusok névvel. A két szegmens első sorába a többi laphoz hasonlóan a fejléc kerül kiírásra, de itt az első oszlopba nem az „Xmi ID” kerül, hanem a szegmens megnevezése, így ez az oszlop a két fejléc elemen kívül más adatot nem is tartalmazhat.

A *ClassSummaryShett, InterfaceSummarySheet, EnumerationSheetCreator, ClassSheetCreator* és az *InterfaceSheetCreator* osztályokban felmerülhet még az az eset amikor egy elemhez több rész-elem is tartozik, pl.: Enumerációk esetén a literálok, metódusok esetén a paraméterek, osztályok esetén a megvalósított interfészek. (Ahogy az a 7. és 9. ábrán is látható) Ezekben az esetekben, az algoritmus az enumerációktól eltekintve mindenhol az első ilyen rész-elem adatait az objektummal egy sorba írja majd minden további elem új sorba kerül, ahol ezen kívül más adat nem kerül rögzítésre.

#### Excel importálás

A „*model.transfer.importer*” tartalma felelős az Excel fájl tartalmának UML elemekké alakításáért. A végrehajtás a *ExcelImportHandler* osztályból indul ki, ez kezeli a képernyőn megjelenő menüpontot, és annak elérhetőségét. Az importálás elindításához szükséges, hogy a jobb egérgombbal kattintással egy megfelelő elem kerüljön kijelölésre, majd az *Import from Excel* menüpontot választvaa fájlkezelővel egy „.xlsx” kiterjesztésű fájlt válasszunk ki. Ha mindez teljesült a tényleges importálás elindulása előtt az *ExcelStructureValidator* osztály a következő ellenőrzéseket végzi el a fájl szerkezetén:

* Létezik-e minden munkalap. Ezt az jelenti egy Excel generálásakor minden esetben megszülető lapok, mint a DataType, Associations, Enumerations, Interfaces, Classes lapoknak jelen kell lennie, illetve az Interfaces, Classes lapon definiált entitások mindegyikéhez egy a nevével megegyező lapnak kell léteznie.

**Irodalomjegyzék**

[1] Papyrus modellező eszköz [Online] <https://en.wikipedia.org/wiki/Papyrus_(software)> [Elérhető 2021.12.28]

[2] EMF <https://www.eclipse.org/modeling/emf/> [Elérhető 2021.12.28]

[3] SWT <https://hu.wikipedia.org/wiki/Standard_Widget_Toolkit>

[4] Apache POI <https://en.wikipedia.org/wiki/Apache_POI>