

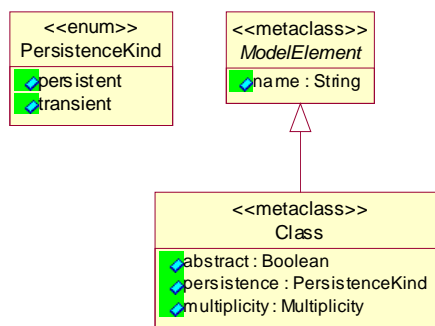
Az UML metamodelje

- Az UML fogalmaiból épített UML modell
 - Osztályai az UML fogalmai
 - Kapcsolatai az UML fogalmai közötti kapcsolatok

Metamodel: Class metaosztály

- Az „osztály” fogalom leírása.
- Háttérben tárolt-e/létrejöhét-e példánya
- Minden osztály a Class metaosztály egy példánya
- (Halmazok halmaza)
- Jelölés: <<metaclass>> sztereotípus
- Többszörösség (multiplicity): [1..3] példányok számára korlát
- Egypéldányú (singleton)

Perzisztens: tartósan fennálló
Transient: átmeneti, ideiglenes



Osztályjellemzők (Feature)

Láthatóság (visibility): „+” nyilvános (public), „#” (protected) védett, „-” rejtett (private)

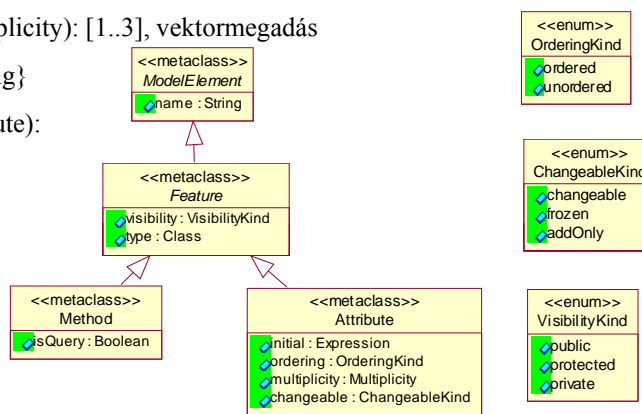
Az osztály szintű attribútumot aláhúzással jelöli az UML (A Rose-ban \$ jel jelöli)

Többszörösség (multiplicity): [1..3], vektormegadás

Rendezettség {ordering}

Tulajdonságok (attribute):

Eljárások (method):



Osztályfüggvények

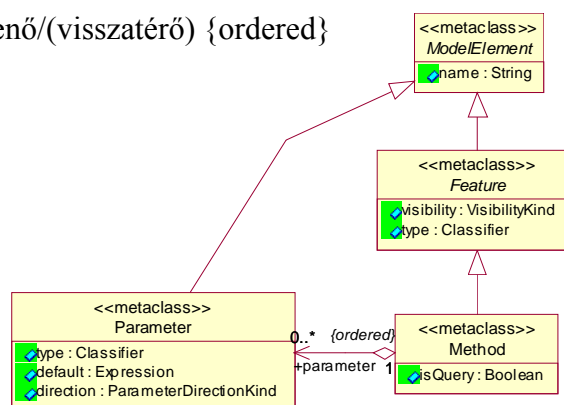
Paraméterek: rendezettek, sorszámmal indexelhetők

Mellékhatása van-e? 'isQuery'

Kimenő/bemenő/átmenő/(visszatérő) {ordered}

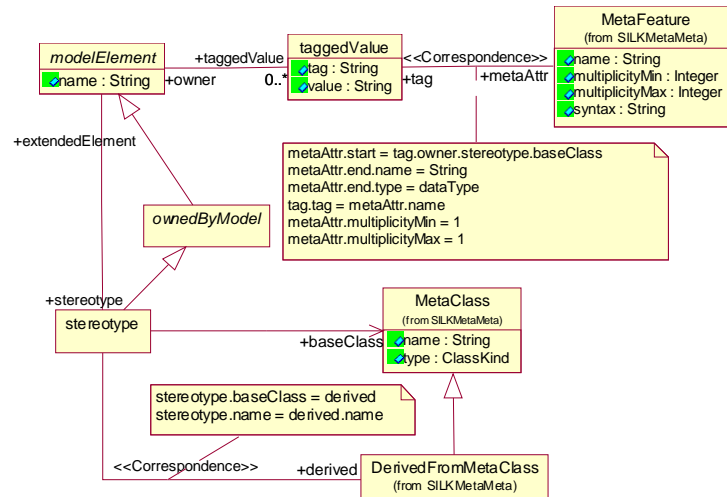
Alapérték (attribute):

Típus (method):



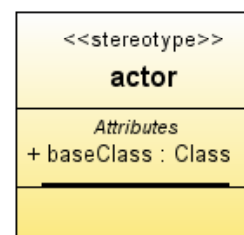
Metamodell kiterjesztése sztereotípussal

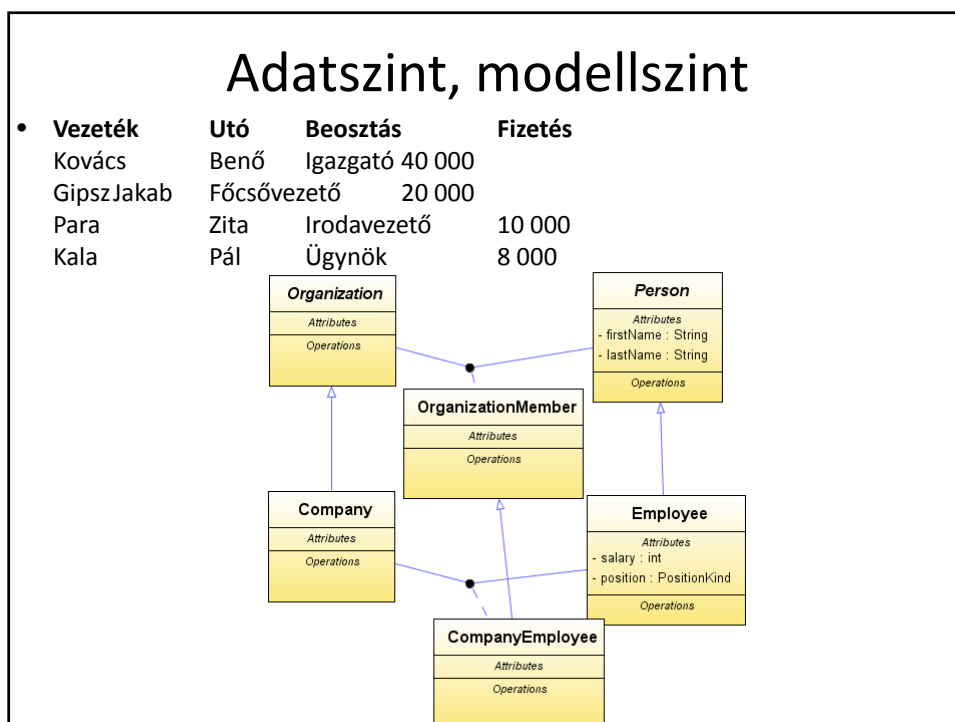
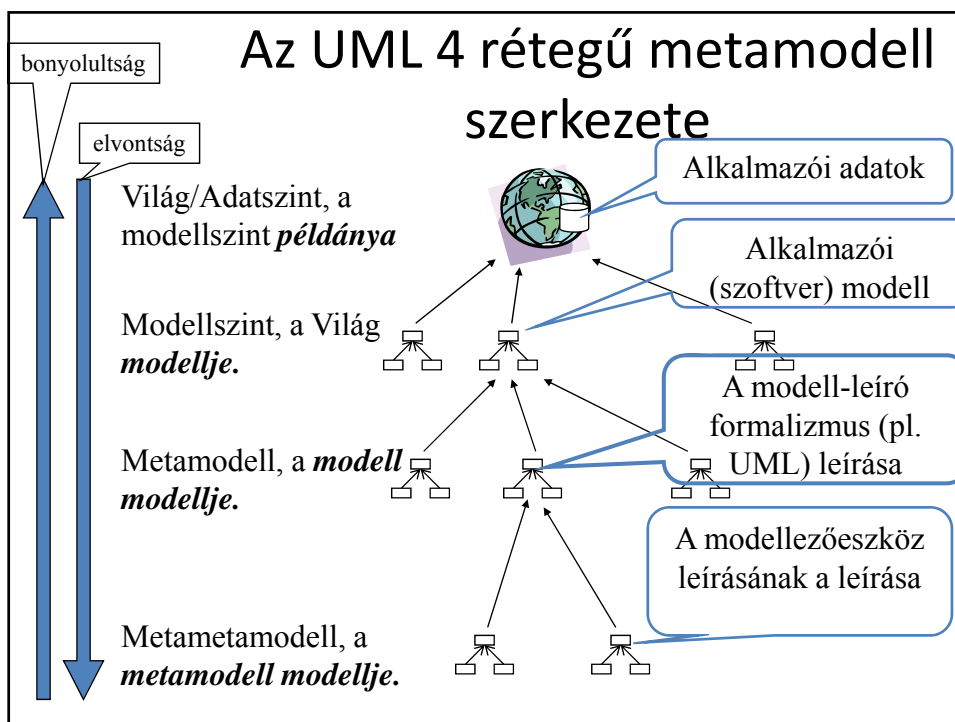
- Egy UML fogalom/metaosztály terjeszthető ki



Hogyan olvassuk ki?

- Egy sztereotípus definíció megfelel egy új metaosztálynak, ami a „baseClass” tulajdonságával megadott UML fogalom kiterjesztése
- Egy név-érték párt egy sztereotipizált osztályhoz adunk.
- Megfelel egy metatulajdonság-értékpárnak, amely a sztereotípus alaposztályához tartozik





Modell

- (Egy szoftver által kezelt) információ szerkezetének adatfüggetlen, pontos leírása
- Az adatok csoportosítása → osztályok, példányaik
- Tulajdonságok és értékeik
- Közös tulajdonságok kiemelése, elvonatkoztatás → általánosítás
- Adatkapcsolatok csoportosítása → kapcsolatok, példányaik

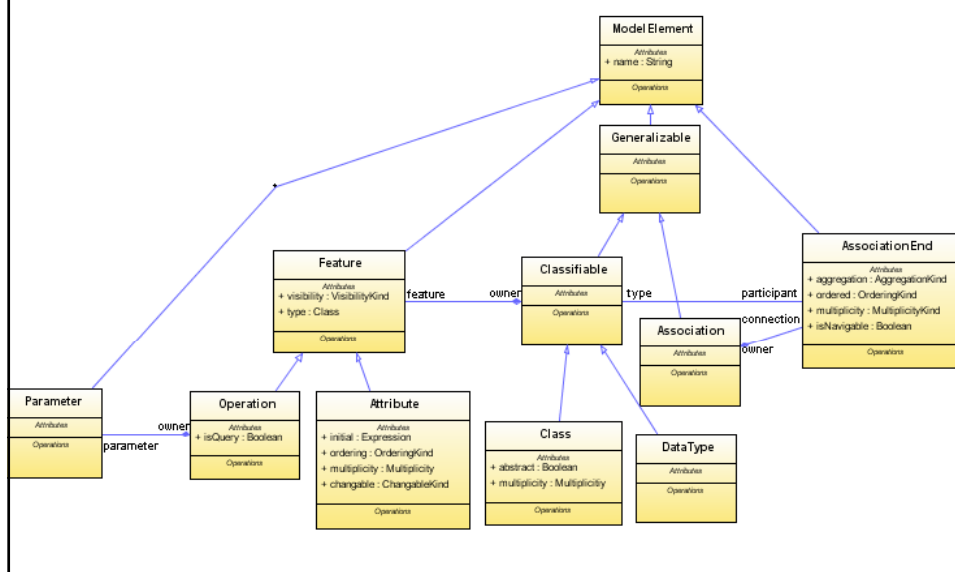
Metamodellek

- Metamodell: egy szemantikus modell készítése során használt eszközök és szabályok pontos definíciója *(Szemantikus modell: A rendszerrel a kezelő természetes nyelven kommunikál, így elvileg bármit kifejezhet, amire gondol.)*
- Metamodellezés: modellezés, melynek tárgya egy modellezőeszköz maga
- (Szemantika \leftrightarrow szintaktika
- Információk, adatok szerkezete $\leftrightarrow \sim$ szöveges megjelenítési formája)

Mire kellenek a metamodellek

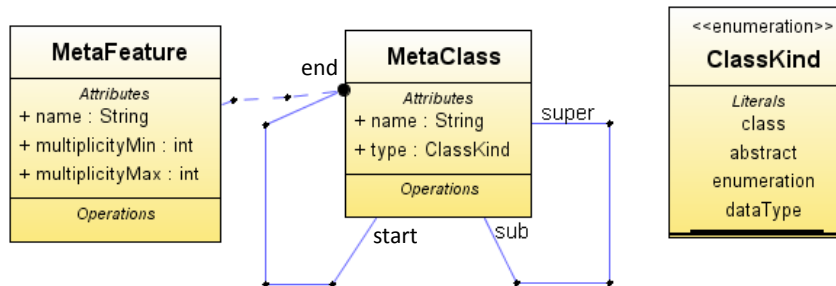
- **CASE/modellező eszköz fejlesztő:** újabb modellemeket is támogatnia kellhet...
- **Adat/információ tárház tervező** (adatbázis, repository(tárház): Adatszerkezet (séma) megadására/módosítására szolgáló részeszközök
- **Rendszerintegrátor:** különböző jellegű adatok integrációjához szükséges azok szerkezetének (metamodelljének) a megalkotása/megértése...
- **Végfelhasználó:** pl. egy új rendszer kiértékelésekor/megismerésekor

Egy UML rész-metamodell



Meta-metamodell

- A metamodell leírása/modellje – a metamodell a meta2modell példánya
- *Enumeration-felsorolás*



MetaFeature

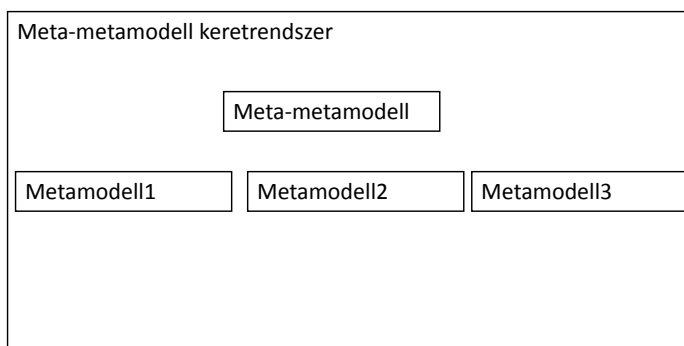
START	NAME	END	MIN	MAX
Feature	owner	Classifiable	1	1
Classifiable feature	Feature	1	*	
Classifiable participant	AssociationEnd	0	*	
AssociationEnd	type	Classifiable	1	1
Association connection	AssociationEnd	2	*	
AssociationEnd	owner	Association	1	1
Parameter	owner	Operation	1	1
Operation	parameter	Parameter	0	*

- Mit kellene még modellezni? MetaGeneralization, MetaClass

Mire jók a meta-metamodellek?

- Ugyanahhoz a rendszerhez többféle metamodel is felépíthető (különböző célokra)
- Mikor? Ha modelltárházat akarunk építeni, és:
 - Többféle metamodelre akarjuk alkalmazni
 - Módosítani, fejleszteni akarjuk a metamodelt
- Mikor? Különböző metamodelű rendszerek között intelligens információ integrációra
 - Megadjuk az egyik rendszer metamodeljét (pl. UML)
 - Megadjuk a másik rendszer metamodeljét
 - Megadjuk a metamodellek közötti átalakítási szabályokat (pl. UML asszociációkkal+OCL megszorításokkal)

Metamodel keretrendszer



Pl: az OCL metamodelje UML-ben

Készítsük el az OCL metamodeljét UML-ben.

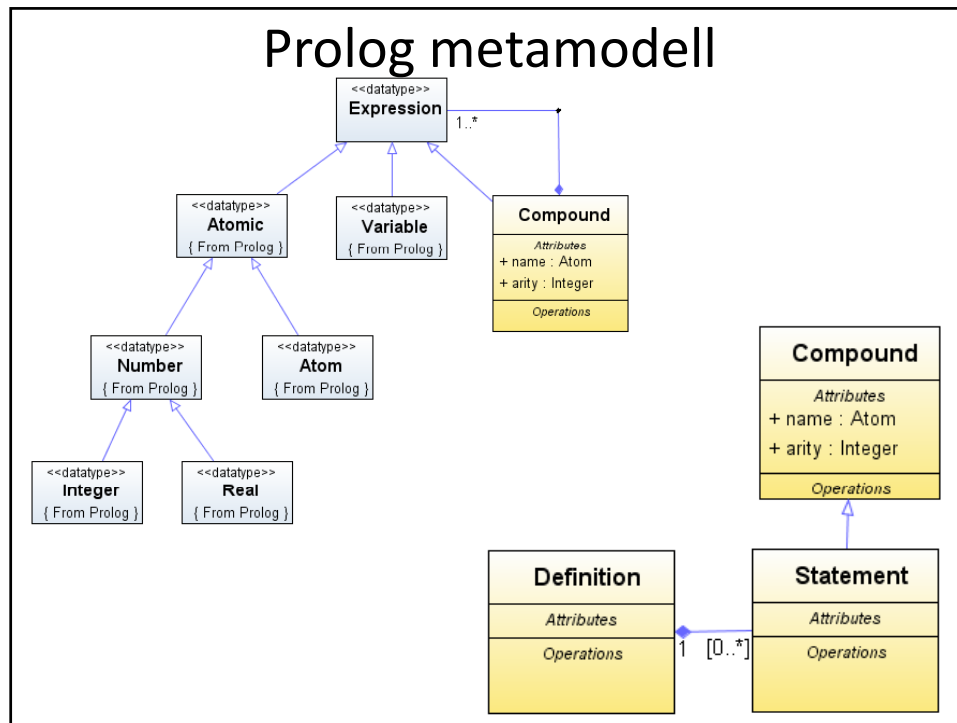
Használjuk a következő fogalmakat:

Kifejezés, Literál, ÖsszetettKifejezés, Feature,
Navigáció, Művelet, Paraméterek

Egyéb metamodellek

- Software Process Engineering Metamodel (SPEM)
- XML metamodel
- Common Warehouse Metamodel (CWM): szabvány
segítségével egységesen leírható egy adatbázis szerkezete az XML nyelvvel.
- Prolog metamodel
- Relációs metamodel
- SQL metamodel

Prolog metamodel



WHILE nyelv

