

Objektum-kapcsolatok fajtái

- ❑ Amikor **objektumok egymással kommunikálnak** (szinkron vagy aszinkron módon egymás metódusait hívják, egyik a másiknak szignált küld, esetleg közvetlenül a másik adattagjain végeznek műveletet), akkor kapcsolat alakul ki közöttük.
- ❑ A kapcsolat fajtája lehet:
 - **Függőség** (*dependency*)
 - **Asszociáció** (*association*) vagy társítás
 - **Aggregáció** (*shared aggregation*) vagy tartalmazás
 - **Kompozíció** (*composite aggregation*) vagy szigorú tartalmazás
 - **Származtatás** vagy öröklődés (*inheritence*)
- ❑ Az objektumok közötti kapcsolatokat az osztályaik szintjén ábrázolhatjuk (**az osztálydiagram az objektumdiagram absztrakciója**).

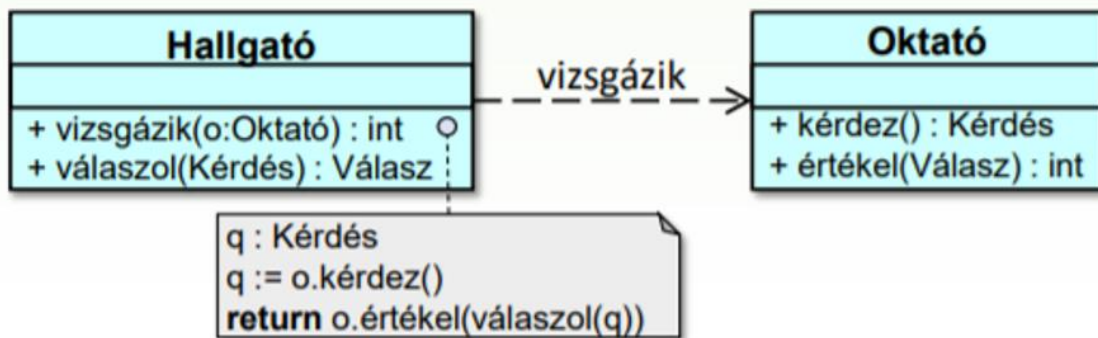
Függőség

<osztály>

≤függőség>

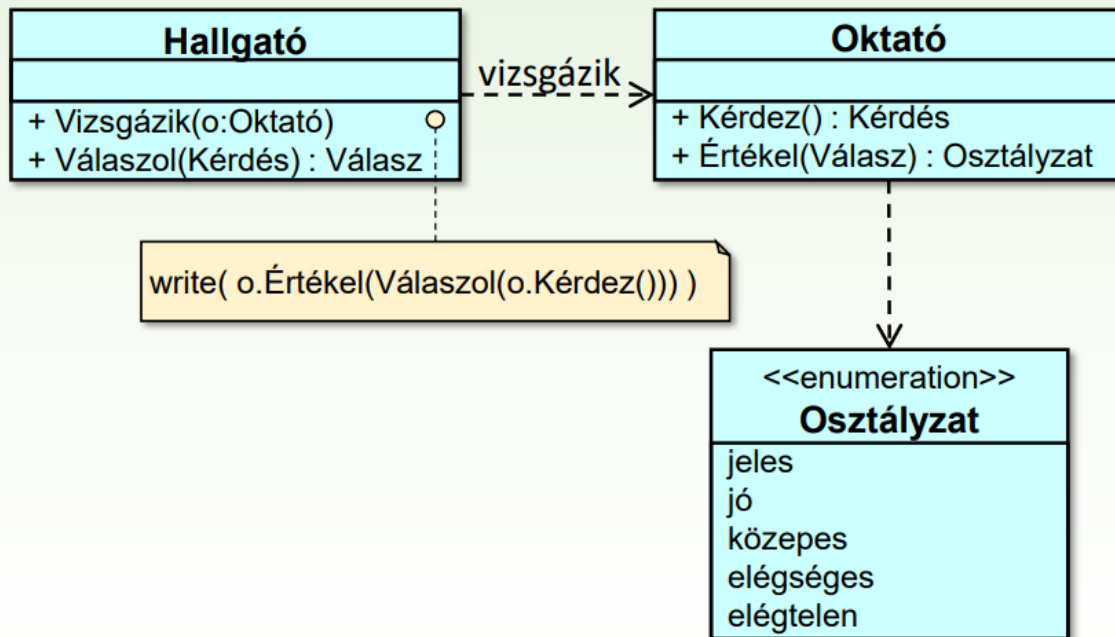
<osztály>

- Egy objektum **epizódszerűen** (rövid ideig) kerül kapcsolatba egy másik objektummal (amelyet paraméterként kap, vagy maga példányosít), mert
 - annak egyik **metódusát hívja** a saját metódusából,
 - **szignált** (üzenetet) **küld** neki,
 - **továbadja** a hivatkozását (pl. kivételkezelésnél),
 - egy **lehetséges állapotát (értékét) használja** (pl. felsorolt típus értékét).
- Osztályok közötti függőség az, amikor egy metódus (amely lehet osztályszintű is) egy másik osztály **osztályszintű metódusát hívja**.



Példa függőségekre

- Egy hallgató vizsgál egy oktatónál, aki kérdést tesz fel neki, majd értékeli a válaszát.

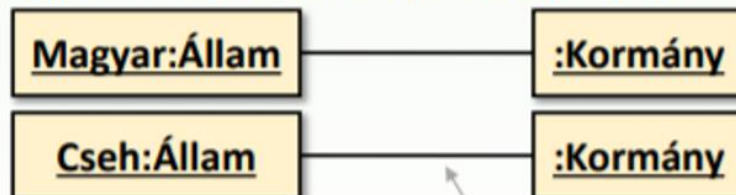


Asszociáció

- ❑ Objektumok között **hosszabb időszakon keresztül** fennálló kapcsolat, amelyben az objektumok sorozatosan küldenek üzenetet egymásnak.
- ❑ Matematikai értelemben az asszociáció egy reláció az adott osztályok példányait (objektumait) tartalmazó halmazok direkt szorzatán:
 - egy asszociáció több objektum-kapcsolatot ír le,
 - egy objektum egy asszociáció több kapcsolatában is megjelenhet, de különböző asszociációk kapcsolataiban is szerepelhet.



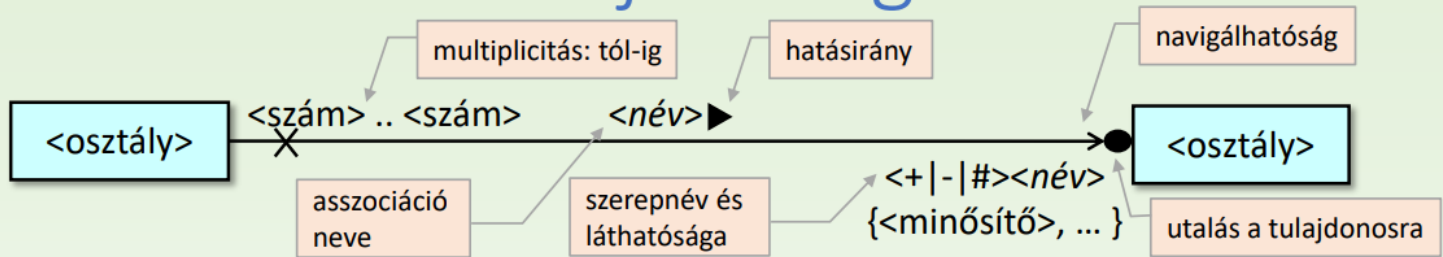
Az osztálydiagram egy lehetséges példányosítása (felpopulálása):



az objektumok (object)
az osztály példányai

a kapcsolatok (link)
az asszociáció példányai

Asszociáció tulajdonságai

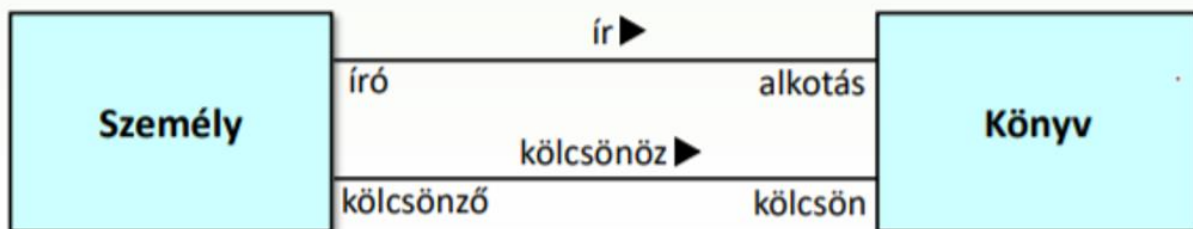


□ Az asszociáció tulajdonságai az általa leírt kapcsolatokat jellemzik:

- **név**: a kapcsolatok közös megnevezése
- **hatásirány**: kapcsolódó objektumok közti fogalmi viszony
- **multiplicitás**: egy objektumhoz kapcsolható objektumok száma
- **aritás**: az egyes kapcsolatokban résztvevő objektumok száma
- **navigálhatóság**: a kapcsolatban melyik objektumot kell gyorsan elérni
- **asszociációvég nevek**: a kapcsolatban álló objektumok hivatkozási nevei
- asszociációvégek neveinek **láthatósága**
- asszociációvégek neveinek **tulajdonosa**
- asszociációvég nevével jellemzett gyűjtemény **minősítése**

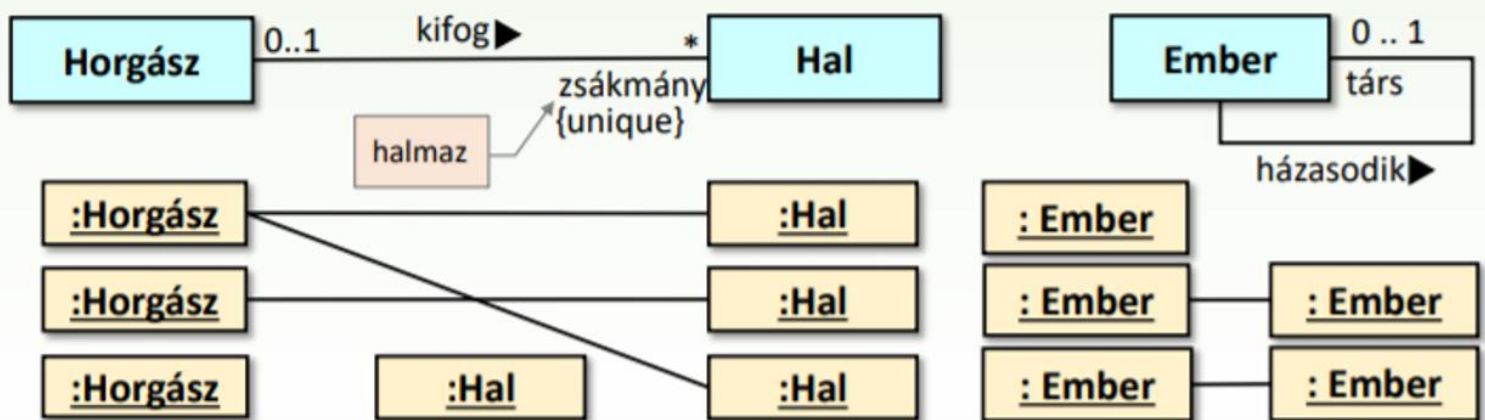
Nevek és a hatásirány

- Egy UML modell kifejezőképességén javít, ha a különböző asszociációkat, és annak kapcsolataiban megjelenő objektumokat nevekkkel látjuk el: az asszociációkra a megoldandó feladat szövege olyan egyszerű bővített mondatokkal utal, amelynek állítmánya (néha a tárgya) az **asszociáció neve**, a mondat többi (nem állítmány, nem jelző) eleme pedig az **asszociációvégek nevei** az ún. **szerepnevek** lesznek.
- A **bináris** (két objektum kapcsolatát leíró) asszociációk neve mellé rajzolt fekete háromszög hegye az asszociáció **hatásiránya**, amely mindig az asszociációt jellemző mondat alanyát adó objektum felől mutat a másik (sokszor ez a mondat tárgya) irányába. Az alany osztályában többnyire megjelenik az asszociáció nevével megegyező nevű metódus is.



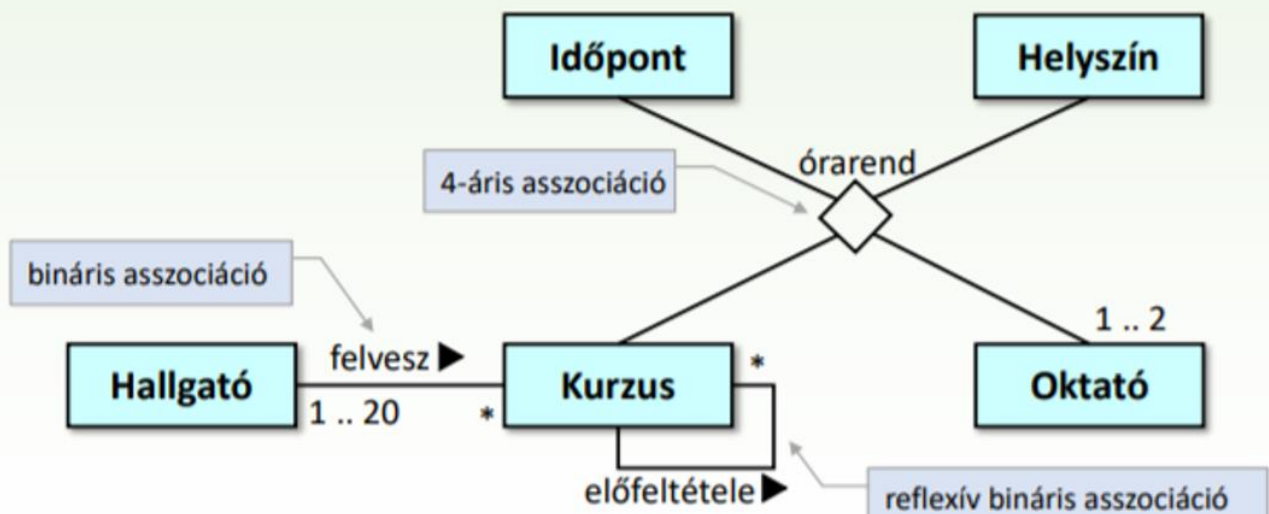
Multiplicitás

- Az asszociáció multiplicitása azt mutatja, hogy az asszociáció szerint egy objektum **hány másik objektummal létesíthet egyszerre kapcsolatot**. Ez lehet egy természetes szám (alapértelmezés szerint 1), vagy természetes számok min..max intervalluma. (Ha a szám, illetve a max helyén * áll, akkor az az adott érték tetszőleges voltára utal.)
- Előírhatjuk hogy egy objektumhoz kapcsolt objektumok
 - mind **különbözzenek** egymástól {unique},
 - megadott **sorrendben** legyenek felsorolhatók {ordered}.



Aritás

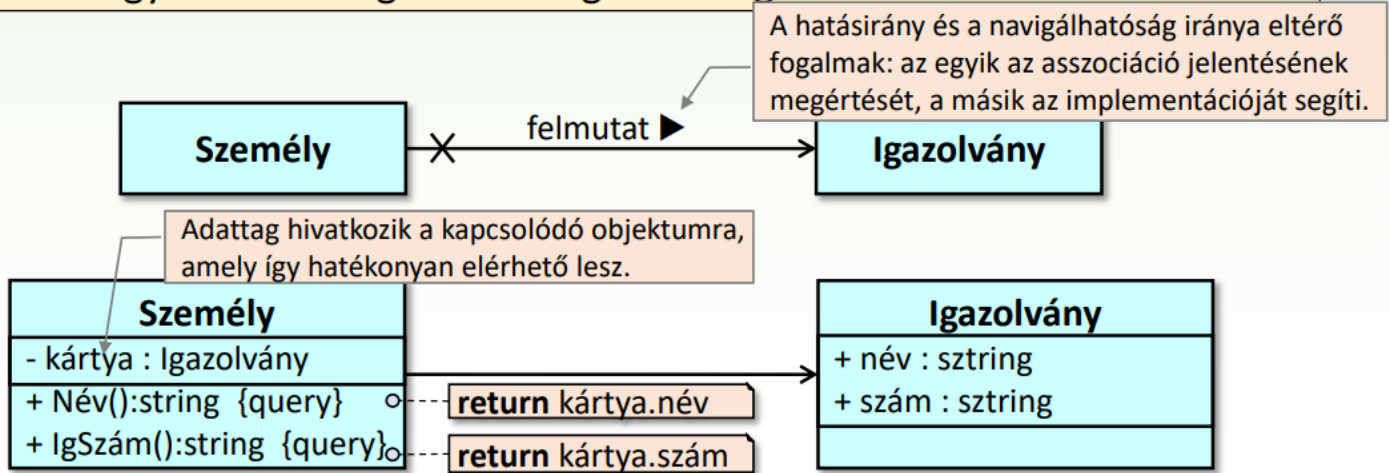
- Az asszociáció **aritása** arra utal, hogy az asszociáció egyetlen kapcsolata hány objektumot köthet össze. (Eddig csak **bináris asszociációkra** láttunk példákat, ahol egy kapcsolat mindig két objektum között jött létre.)
- Ne keverjük össze az aritás és a multiplicitás fogalmait!



Navigálhatóság

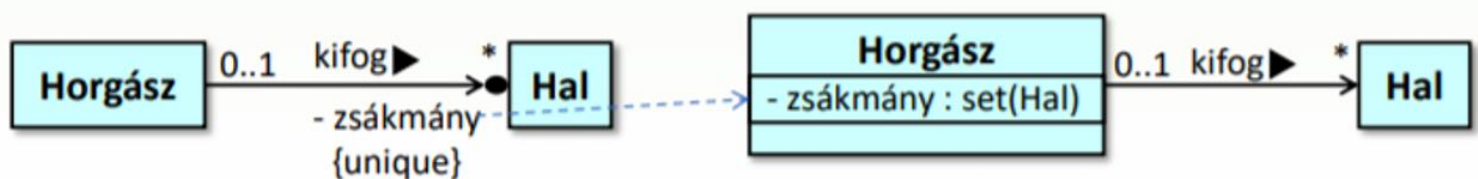
□ A navigálhatóság azt jelzi, hogy egy asszociáció kapcsolataiban melyik osztály objektumát kell a kapcsolat másik (többi) objektumának hatékonyan elérni.

- A nyíl az asszociáció végén **hatékonyan elérendő** objektum osztályára mutat.
- Az „x” jel az asszociáció végén azon osztály mellett van, amely objektumát a kapcsolatokban **nem kell (hatékonyan) elérni**.
- A jelöletlen asszociáció vég arra utal, hogy ott még **nincs eldöntve**, hogy kell-e támogatni a navigálhatóságot.



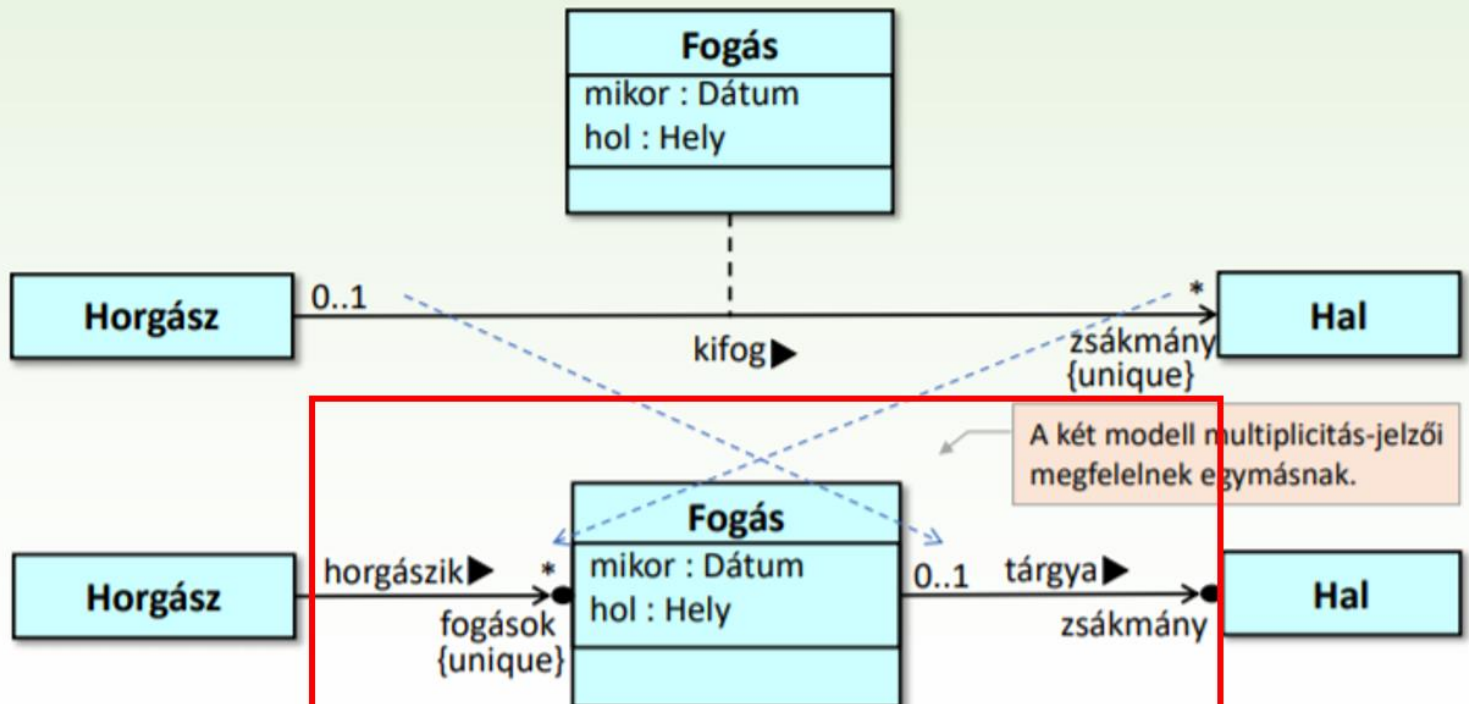
Szerepnév tulajdonosa

- Egy kapcsolat mentén zajló navigációban el kell érni a kapcsolat egyik objektumának a másik objektum hivatkozását. Hol tároljuk ezt el?
- Ki legyen egy kapcsolatban egy szerepnév által azonosított objektum hivatkozásának (röviden a szerepnévnek) a tulajdonosa?
 - Maga a **kapcsolat**, amennyiben a kapcsolat rendelkezik önálló tárhellyel, amelyet a kapcsolat objektumai elérnek.
 - A kapcsolat **másik (többi) objektuma** a saját tárhelyén. Erre utal a *fekete pötty* az asszociációs vonal végén, az adott szerepnévnél.
- A szerepnév **láthatósága** (private, protected, public) mutatja, hogy ez a név publikus, vagy sem; **multiplicitása** pedig azt, hogy gyűjtemény-e.



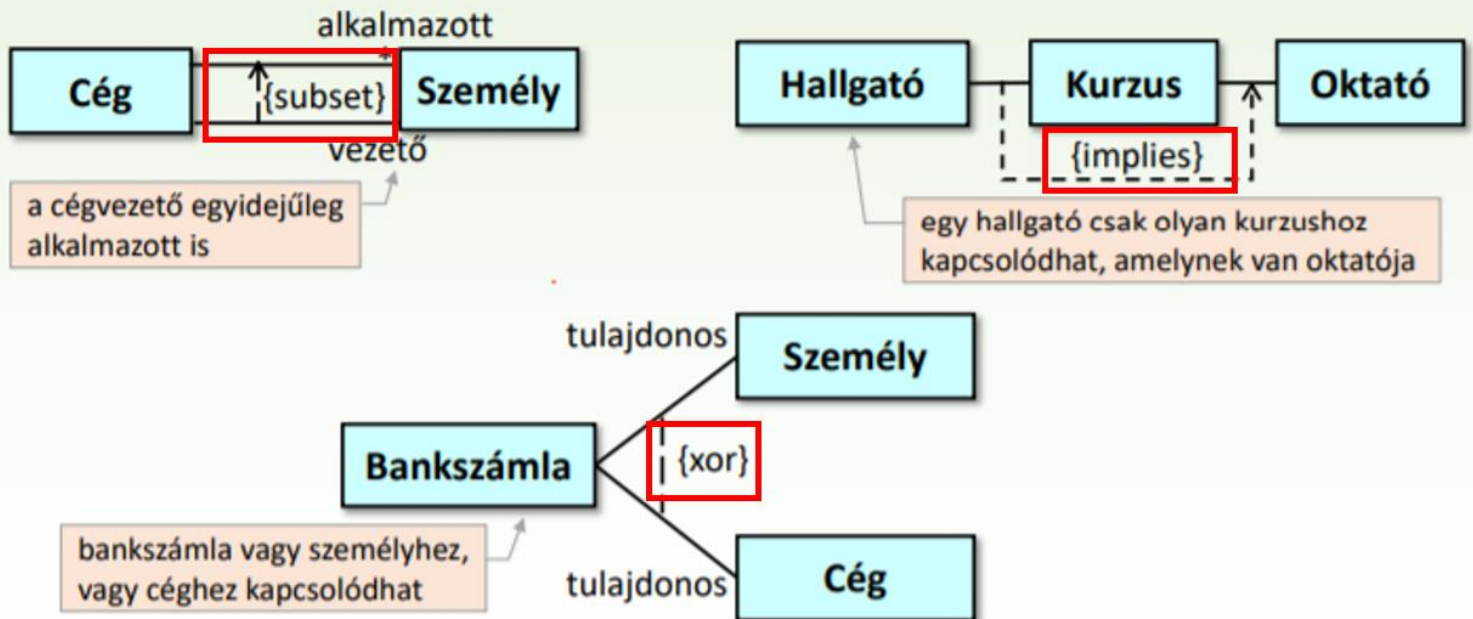
Asszociációs osztály kiküszöbölése

- A programozási nyelvek többsége az asszociációs osztály fogalmát nem támogatja: a kapcsolatokhoz nem rendelnek önálló tárhelyet. Ha erre szükség van, akkor közönséges osztály segítségével helyettesíthetjük.



Asszociációk közötti feltételek

- ❑ Megadhatunk az asszociációk között logikai feltételeket (subset, and, or, xor, implies, ...), amelyek **különböző asszociációk kapcsolatai közötti korlátozásokat** fogalmazzák meg.

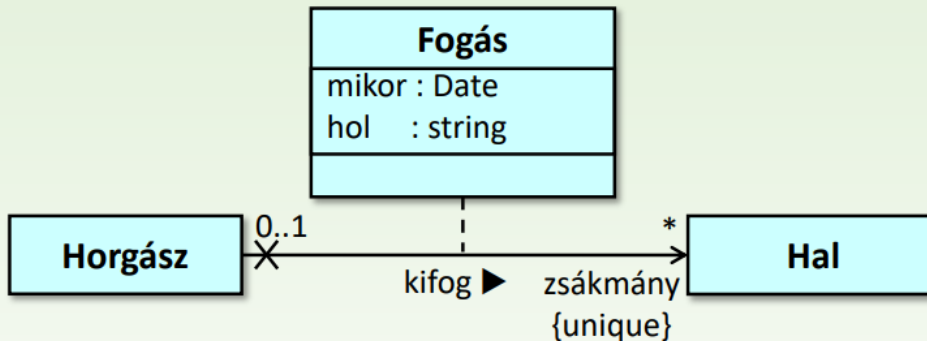


Modellezési szintek

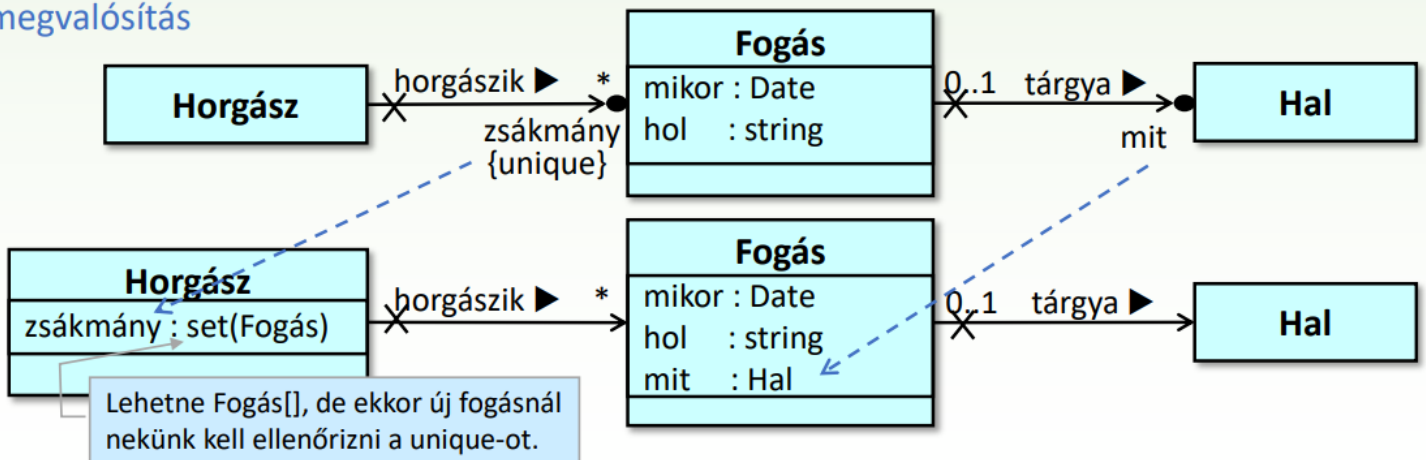
elemzés



tervezés



megvalósítás



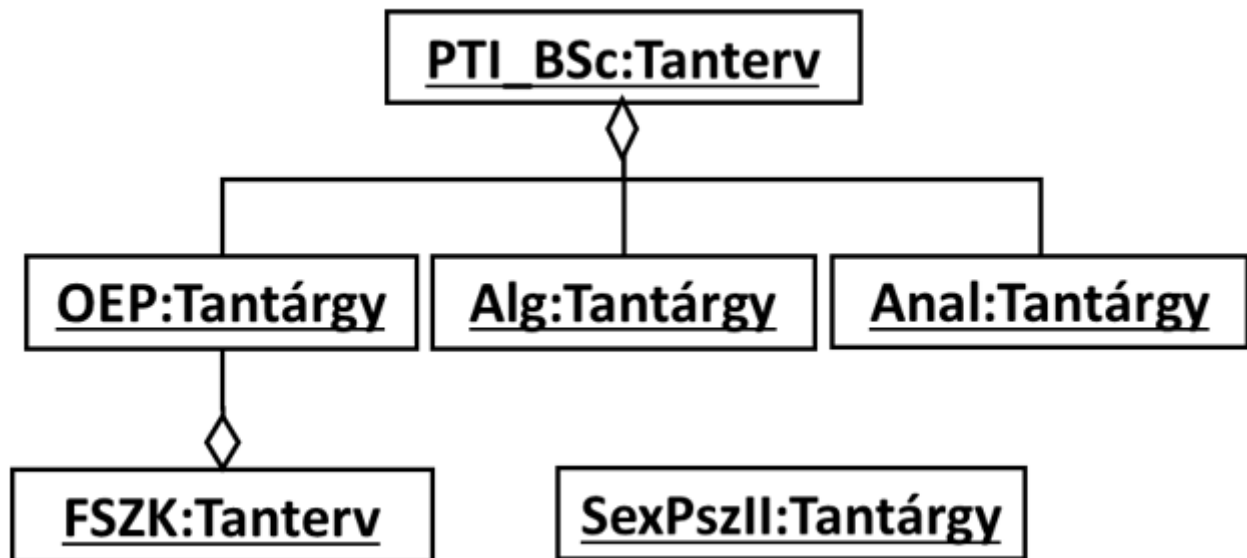
Aggregáció

- ❑ **Egész-rész** kapcsolatot kifejező bináris asszociáció, amely azt írja le, hogy egy objektumnak része, tulajdona egy másik:
 - Ez egy **aszimmetrikus, nem reflexív, tranzitív** binér reláció a tartalmazó és a beágyazott objektumok között, tehát **nem alkothat irányított kört** (azaz egy objektum még közvetett módon sem lehet önmaga része, tulajdona).
 - **Tartalmazó objektumnak nem kell mindig rendelkeznie beágyazott objektummal; a beágyazott objektum pedig létezhet önmagában is, vagy egyidejűleg több tartalmazó objektumnak is része lehet.**
- ❑ Megállapodás szerint a beágyazott objektum
 - irányába hatékonyan navigálhatunk a tartalmazó objektumból,
 - szerepneve a tartalmazó objektum tulajdona.

Kompozíció

- ❑ Speciális aggregáció (tehát aszimmetrikus, nem reflexív, tranzitív binér reláció), de
 - a tartalmazó objektum **nem létezhet beágyazott objektum nélkül,**
 - a beágyazott objektum egyszerre csak **egy objektum része lehet.**
- ❑ A kompozíció fogalmán sokan az alábbi, a beágyazott objektumra tett, egyre szigorodó megszorításokat is hozzáértik:
 - **van tartalmazó objektuma:** beágyazandó önmagában nem létezhet
 - **a tartalmazó objektuma nem változik:** a beágyazott létrehozása és megszüntetése a tartalmazó feladata
 - **élettartama azonos a tartalmazó objektumével:** a tartalmazó konstruktora példányosítja, destruktora törli a beágyazottat.

1/a. Egy tanterv tantárgyakból áll. Egy tantárgy egyszerre több tanterv része is lehet, sőt tantervhez sem kell tartoznia.

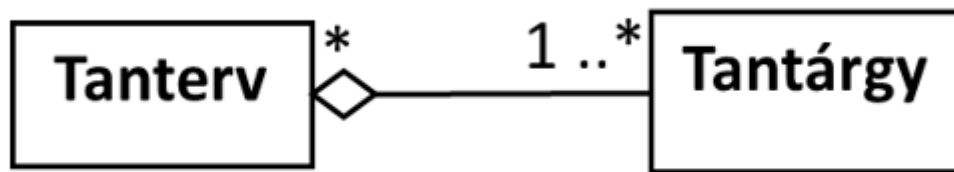


Az objektum diagramokban csak a teli (fekete) rombuszokat szokták írni, de ebben az esetben már érdemes felhívni arra a figyelmet, hogy :

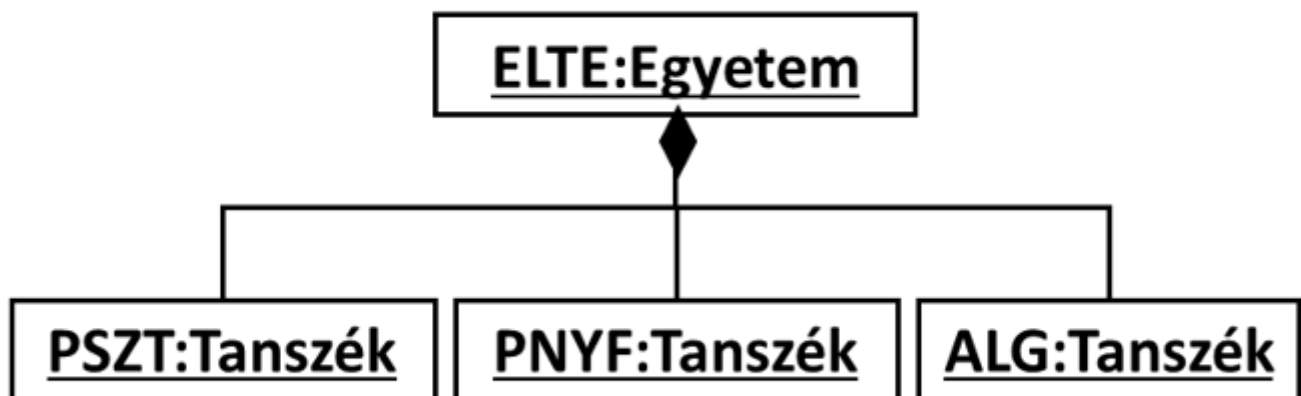
1. ugyanazt a tantárgyat egyszerre több tanterv is birtokolja,
2. sőt, lehet olyan tantárgy is, amelyik egyik tantervhez sem tartozik (de választható a szabad kreditek terhére).

Ezért a kapcsolat az osztálydiagram szintjén aggregációval jellemezhető.

Az aggregáció azt fejezi ki, hogy egy objektum birtokol egy másik objektumot, de megengedi, hogy ne legyen tulajdona, és a tulajdon objektum több objektumnak van is a birtoka lehet.

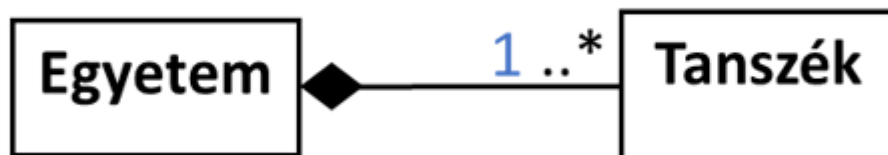


1/b. Egy egyetemen tanszékek vannak: tanszékek nélkül nincs egyetem, de tanszék sincs egyetem nélkül. Egy tanszék egy egyetemhez tartozik.

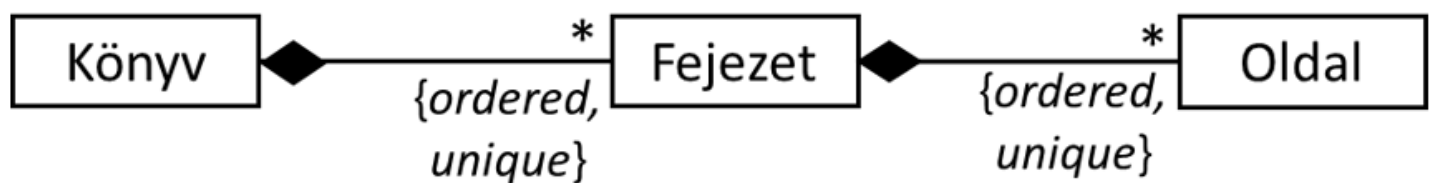
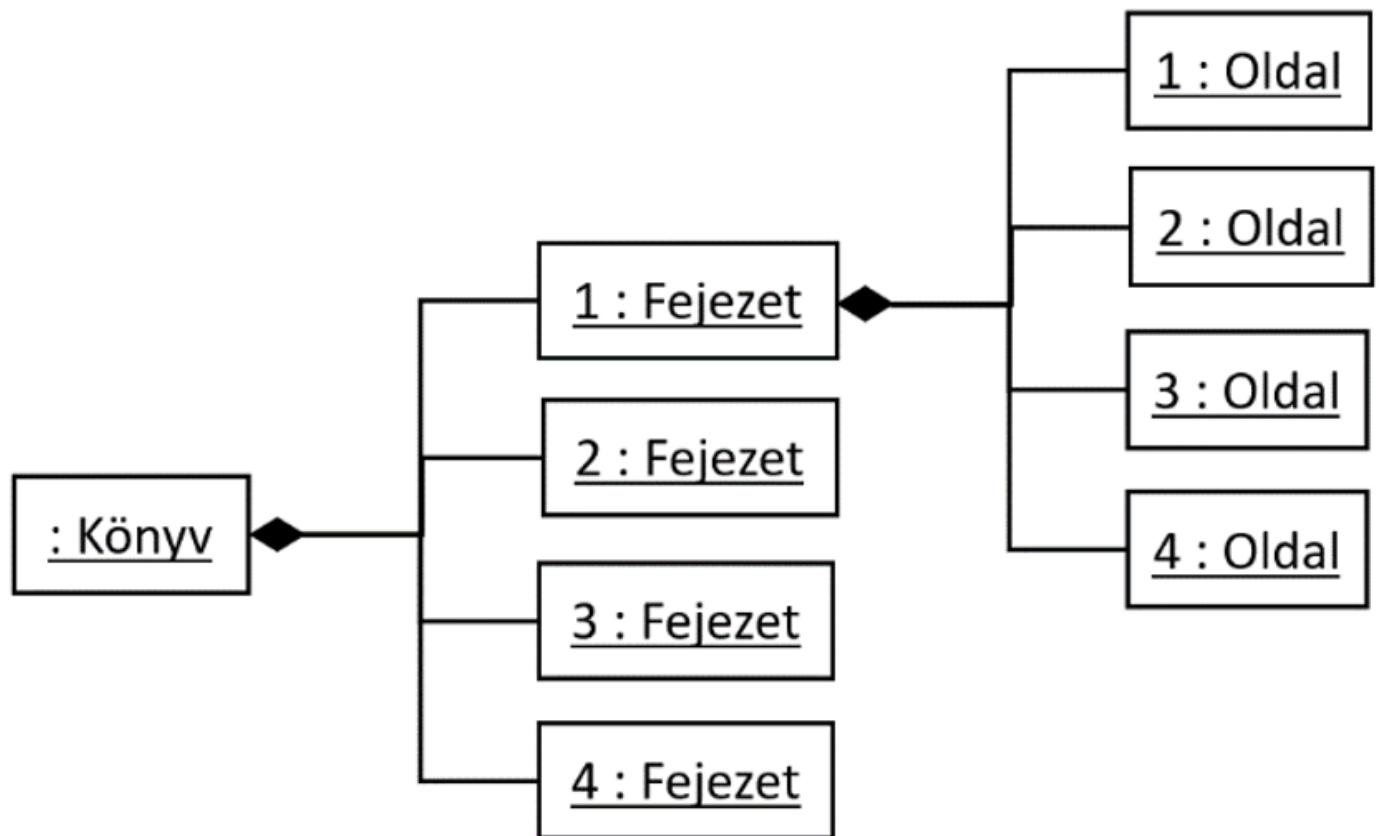


A kompozíció egy szigorú aggregáció, amely egész-rész kapcsolatot fejez ki:

a tartalmazó nem lehet meg tartalmazott nélkül, és a tartalmazott legfeljebb egy tartalmazónak a része lehet.



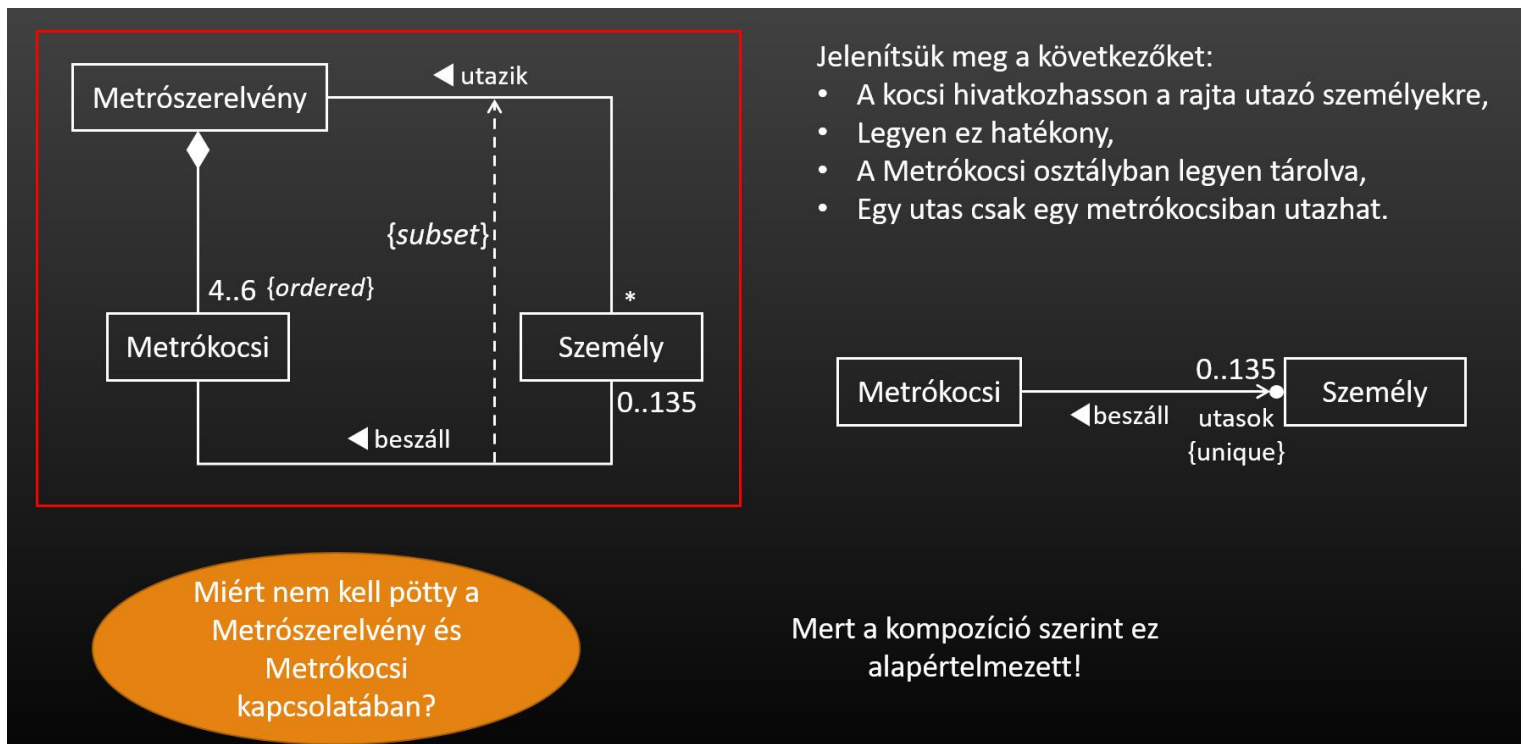
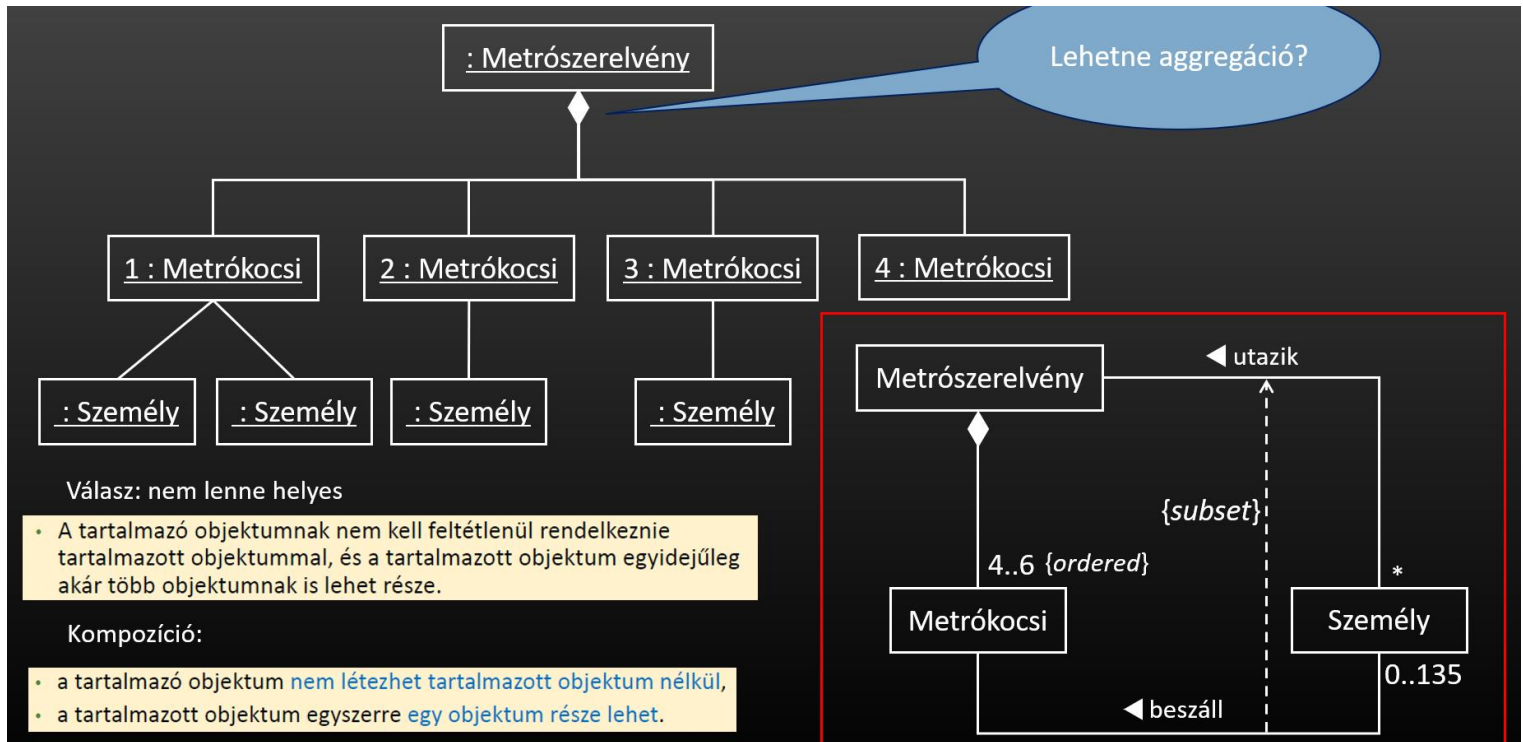
2. Egy könyv legalább egy fejezetből áll, egy fejezet pedig legalább egy oldalt tartalmaz.



Elég lenne csak '*' -ot írni?

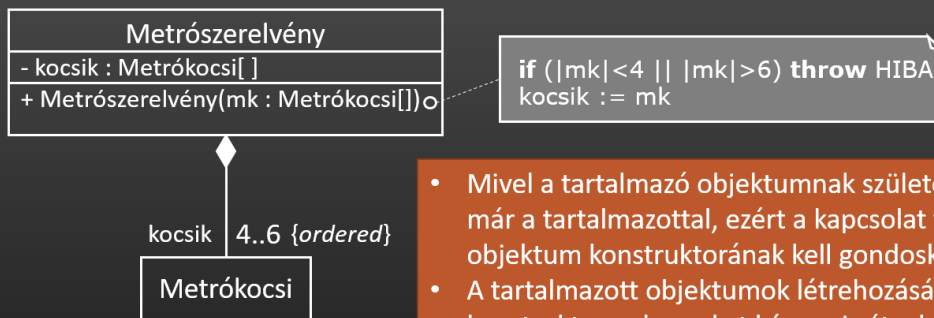
Igen, mert a kompozíció miatt az 1 kötelező!

3. Egy metrószerelvény négy, öt, vagy hat kocsiból áll. Egy személy akkor szállhat be egy metrókocsiba, ha az része egy szerelvénynek. Egy metrókocsiba legfeljebb 135 személy fér el.



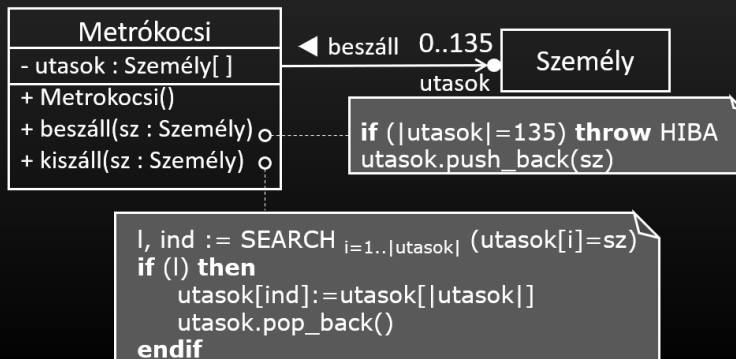
A metrókocsik a szerelvény részei, ahol a tartalmazó (a szerelvény) nem nélkülözheti a kocsikat (kocsik nélkül nincs szerelvény), továbbá egy kocsi egyszerre csak egy szerelvénynek lehet része: ezért a Metrószervény és a Metrókocsi között kompozíció áll fenn.

Milyen metódus kell a kompozíció megvalósításához?



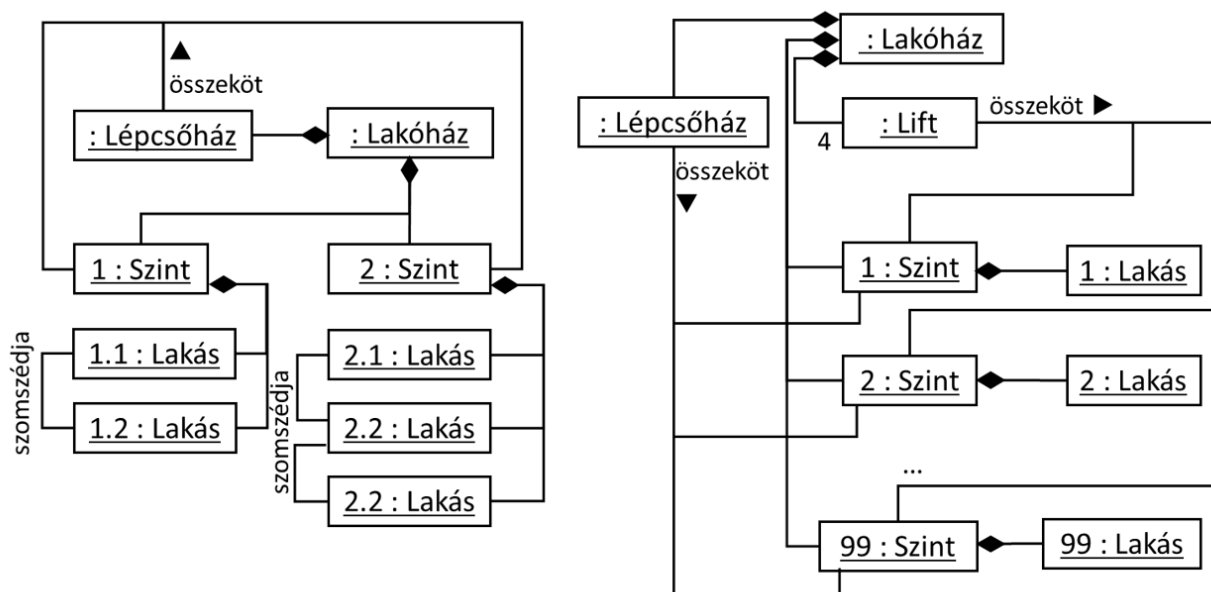
- Mivel a tartalmazó objektumnak születése pillanatában rendelkeznie kell már a tartalmazottal, ezért a kapcsolat felépítéséről a tartalmazó objektum konstruktorának kell gondoskodnia!
- A tartalmazott objektumok létrehozását elvégezheti a tartalmazó konstruktora, de azokat készen is átveheti.

Milyen metódus kell az asszociáció megvalósításához?



- Az asszociáció által leírt kapcsolatot itt nem a konstruktor hozza létre, illetve szünteti meg.
- Erre külön metódusokat vezetünk be.
- Metódusok helye: Metrókocsi osztály, mert abban van az utasokat megadó gyűjtemény.

4. Egy lakóház legalább két szintből áll, szintenként legalább egy lakás található. A szinteket lépcsőházak, illetve liftek kötik össze. Minden szinthez legalább egy lépcsőház kapcsolódik. Minden lakásnak legfeljebb két szomszédja lehet.



Lehet ezt egyszerűbben is, amely megmutatja az összes ilyen ház szerkezetét:

