

Programozás I.

2. gyakorlat



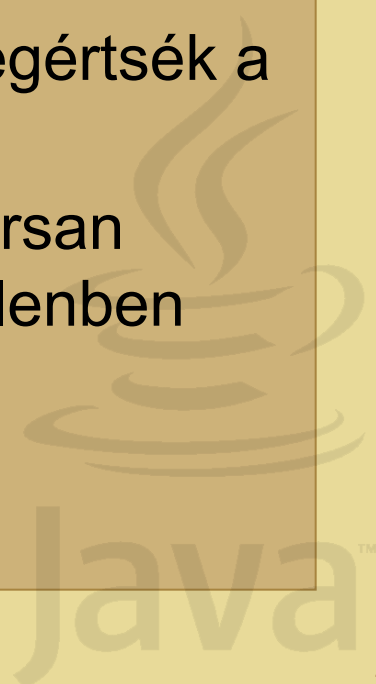
Vizuális modellezés

- Programozás:
 - Modellezés és tervezés
 - Implementálás (kódolás)
 - Dokumentálás és tesztelés
- A megtervezetlen programozás sosem vezet jóra, főleg nagy, bonyolultabb rendszereknél !!
- A tervezés hiánya garantált kudarc.
- A tervezésnek mindig meg kell előznie az implementálást!



Vizuális modellezés

- Szabványos grafikai eszközökkel írjuk le a problémát
- Célja: Jobban átlátható legyen a program felépítése
 - Programozásban nem jártas emberek is megértsék a rendszer működését
 - Hibák jelentős része itt felfedezhető, és gyorsan javítható, komolyabb mellékhatás nélkül (Ellenben egy félkész/kész programmal)



Modell és diagram

- **Modell:** A teljes rendszer leírása, amely magába foglalja a rendszer teljes életciklusát (tervezés, implementálás, dokumentáció, tesztelés)
- **Diagram:** A rendszer egy részének (vagy egészének) vizuális reprezentációja
- **Osztálydiagram:** Osztályok vizuális reprezentációja
 - UML nyelv



Objektumorientáltság

- **Objektum:** Egy valós vagy elképzelt entitás ábrázolása számítógépen, mely minden esetben rendelkezik:
 - állapottal
 - viselkedéssel
 - identitással
- Tehát az objektum lehet bármi, ami rendelkezik ezzel a három tulajdonsággal:
 - számítógép
 - személy



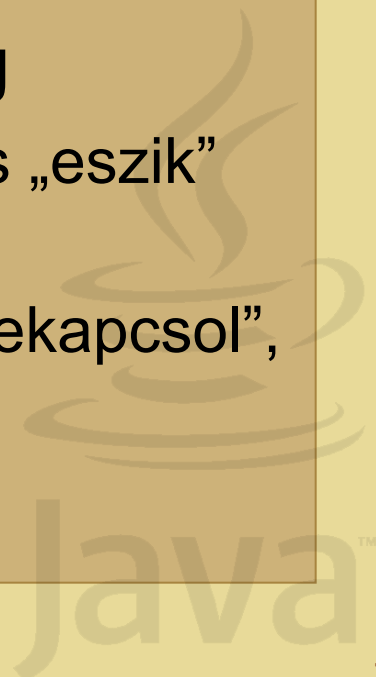
Objektum állapota

- **Objektum állapota:** egy a lehetséges létezési lehetőségek közül (aktuális tulajdonságok által meghatározva)
- Időben változó lehet
- Attribútumok határozzák meg
 - van egy lámpa, ami ki vagy be van kapcsolva
 - adott egy személy, akinek van neve
 - adott egy bor, ami fehér vagy vörös
- Ezek az attribútumok a programozásban lehetnek különböző típusú változók valamilyen értékkel.
 - pl.: adott egy Bor objektum → String típus;



Objektum viselkedése

- Annak módja, hogyan reagál más objektumok kéréseire
 - Az objektum kérésre csinál „valamit”
- Mindent definiál, amit az objektum csinálhat
- Operációk (metódusok) határozzák meg
 - Nyúl típusú objektumnak lehetnek „ugrik” és „eszik” operációi
 - egy Lámpa típusú objektumnak lehetnek „bekapcsol”, „kikapcsol”, „elromlik” operációi



Objektum identitása

- Minden objektum egyedi ! (még akkor is, ha az állapotuk azonos)
 - pl.: a *programozas* és *algoritmusok* különböző objektumok, de mindkettő Kurzus (ugyanabba az osztályba tartoznak)
 - a *torpeNyul* és az *angolNyul* különböző objektumok, de mindkettő Nyúl



Osztály

- Leírás objektumok csoportjához, melyeknek közösek az
 - attribútumai, operációi
 - más objektumokkal való kapcsolataik
 - viselkedésük
- az osztály az objektum típusa
- csomagok (package)
 - Az osztályok csomagokba rendezhetők, hierarchikusan



Osztálydiagram

- class diagram
- az osztályokból osztálydiagramokat állíthatunk össze, mely leírja a programunk szerkezetét



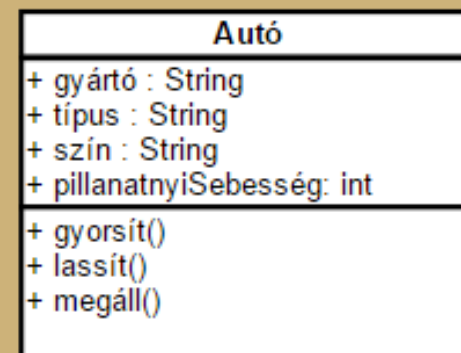
UML

- Unified Modeling Language (Egységesített Modellező Nyelv)
- Egy nyelv: szintaktikai és szemantikai szabályok összessége
- Szoftverrendszer elemeinek:
 - vizualizálására (→ Grafikus)
 - specifikálására
 - létrehozására
 - dokumentálására
- Hivatalos oldal: <http://www.uml.org>



UML osztálydiagram

- Grafikus szemléltetésre különböző diagramok készíthetők, ezek egyike az osztálydiagram
- Elemei: osztályok, interfészek, csomagok, megjegyzések
- Kapcsolatok:
 - asszociáció
 - aggregáció
 - kompozíció
 - öröklődés

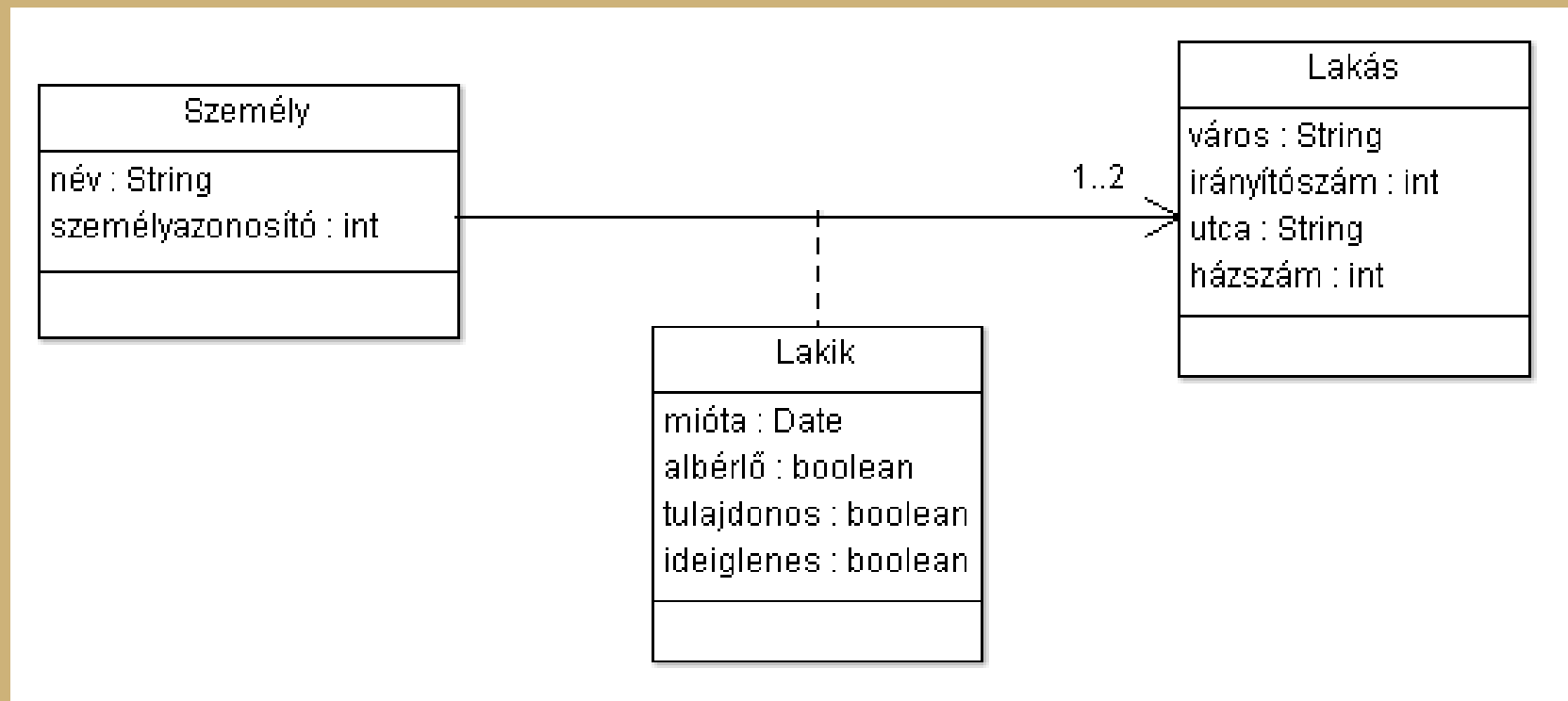


Asszociáció

- Osztályok közötti kétirányú összeköttetés
- „Használati kapcsolat” - létük egymástól általában független, de legalább az egyik ismeri és/vagy használja a másikat
- Gyakorlatilag az egyik osztály használja a másikat (vagy egy metódusát, stb)
- Asszociáció lehet egy- vagy kétirányú
- Adhatunk nevet az asszociációnak
- Asszociációs osztály:
 - Asszociáció, kapcsolat részletei



Asszociáció



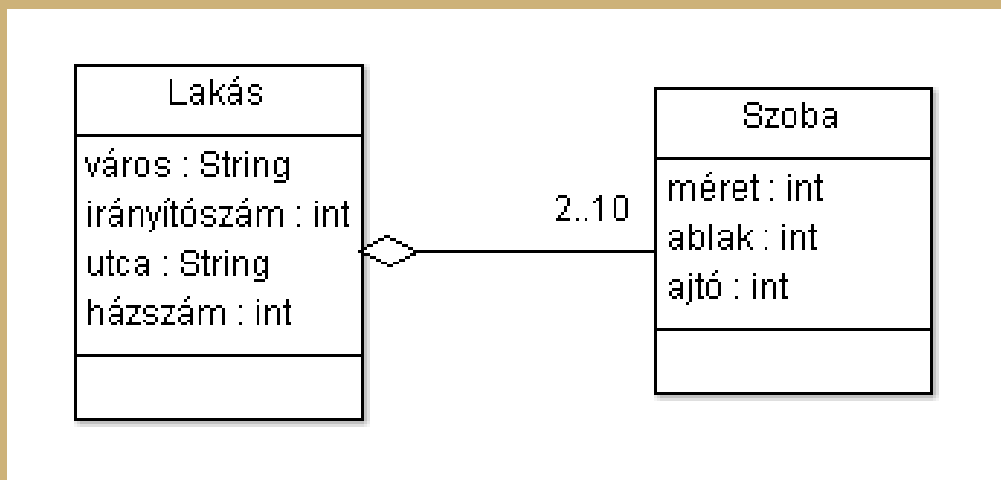
Aggregáció

- Rész-egész kapcsolat
- Az egyik objektum fizikailag tartalmazza a másikat
- A rész-objektum(ok) léte az egész-objektumtól függ
- Kétféle lehet:
 - gyenge aggregáció
 - erős aggregáció (= kompozíció)



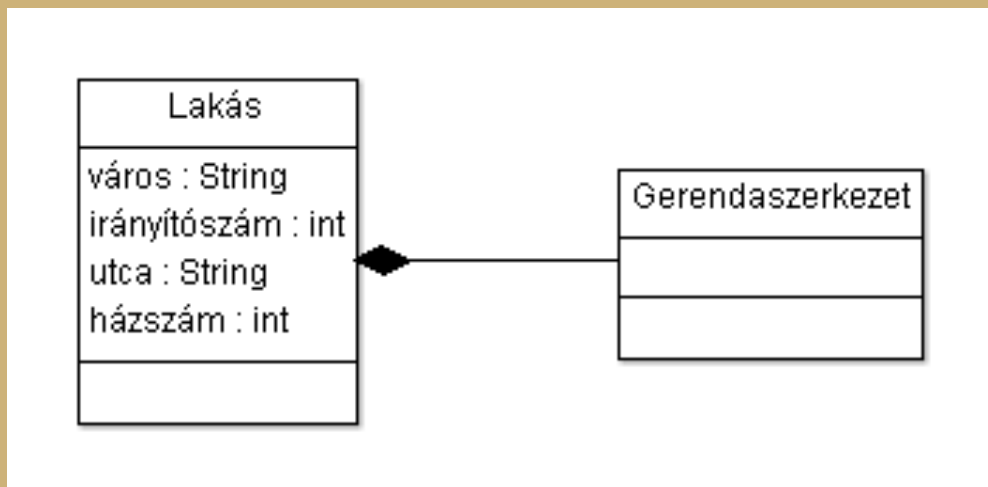
Gyenge aggregáció

- Speciális asszociáció
- A tartalmazó létezhet a tartalmazott nélkül is (pl.: Alma, Kukac)
- Egyik objektum része a másiknak
- Nem erős tartalmazás



Kompozíció

- Fizikai tartalmazás (erős aggregáció)
- Speciális aggregáció
- A tartalmazott nélkül nem létezhet a tartalmazó osztály (pl. Ember, Fej)



Öröklődés

- Osztályok közötti kapcsolat (reláció), ahol egy osztály megosztja a struktúráját és/vagy a viselkedését egy vagy több másik osztállyal
- Öröklődési hierarchia
- A származtatott (gyerek) osztály mindent örököl az őstől és kiegészítheti ezeket saját adatokkal/viselkedéssel.
 - Ősosztály: Fa
 - Gyerekosztályok: Almafa, Körtefa, Szilvafa



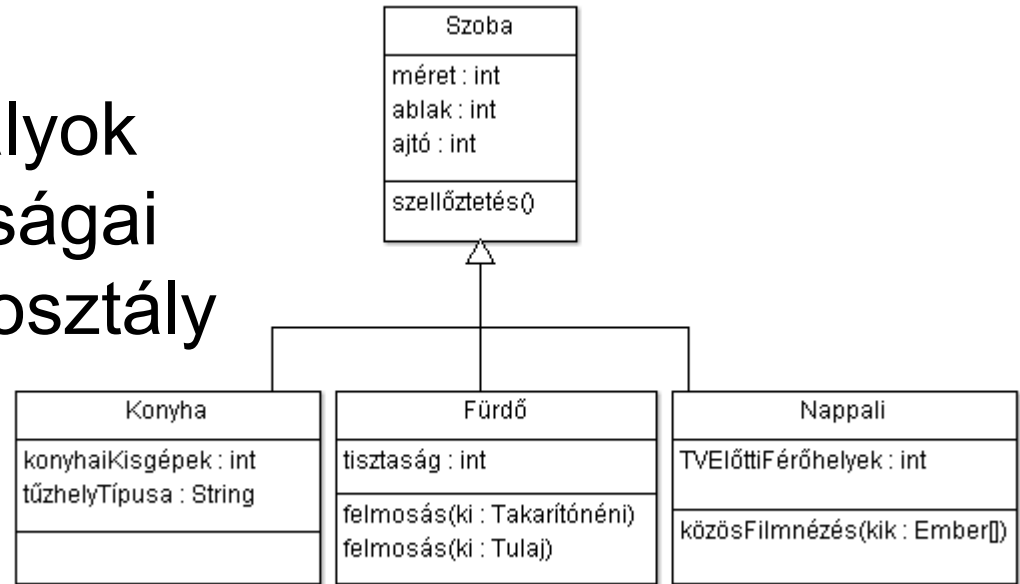
Öröklődés

- A gyerek mindent örököl az őstől, de ezen felül tartalmazhat mást is
- overriding (felüldefiniálás): az örökölt operáció saját implementációja
 - vagyis „még egyszer implementáljuk” az adott operációt a gyerekosztályon belül
 - polimorfizmus
- többszörös öröklődés Javaban nincs!

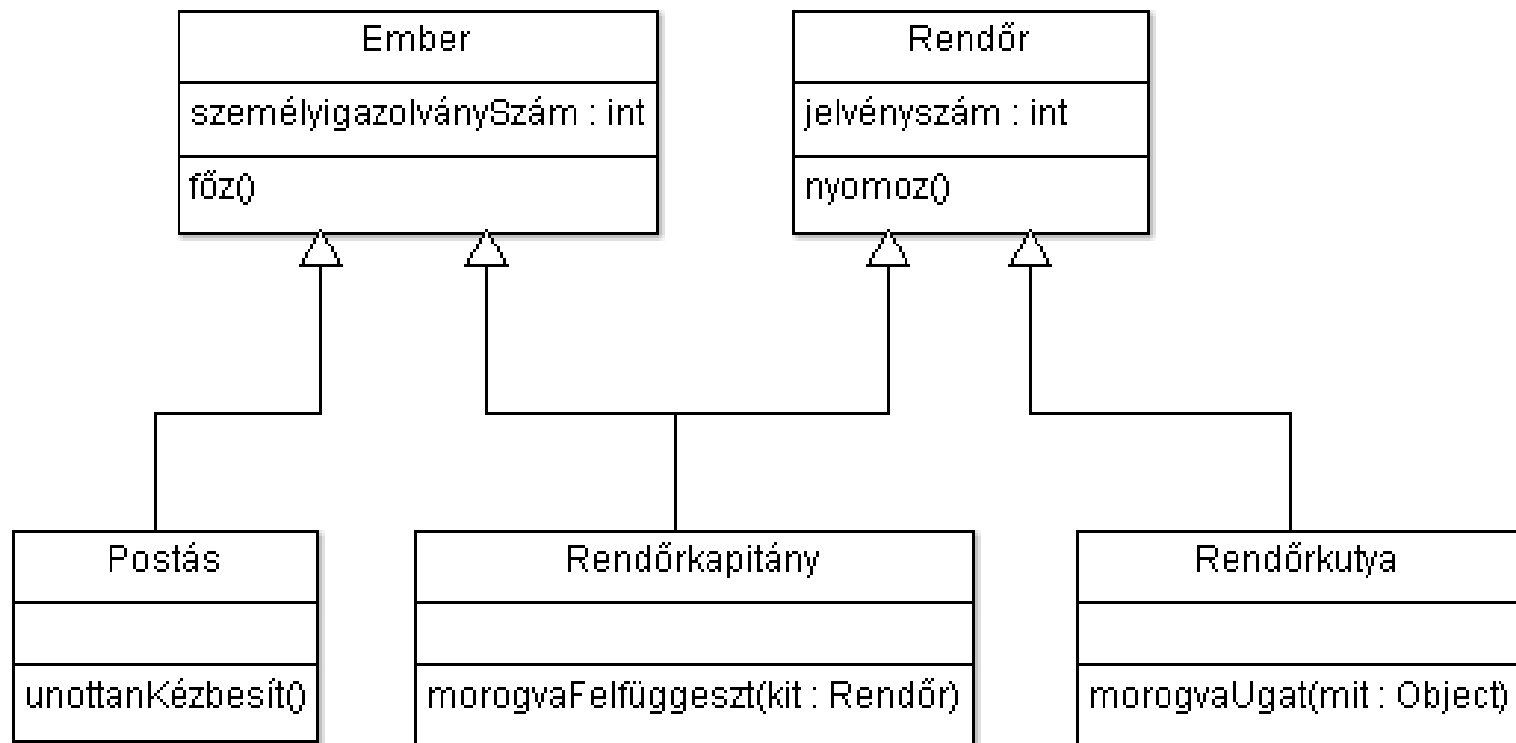


Öröklődés

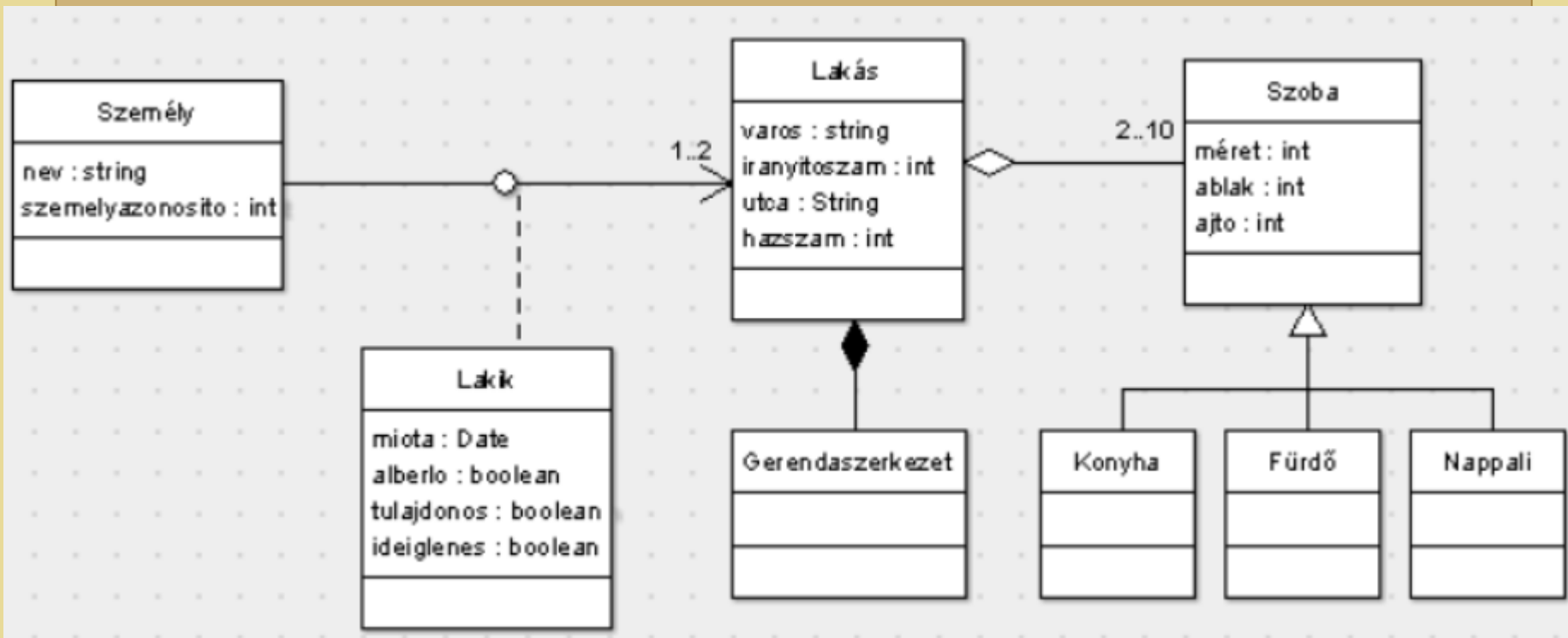
- Általánosítás
 - Bizonyos osztályok közös tulajdonságai alapján egy új osztály létrehozása
- Specializáció
 - Egy osztály speciális esete(i)



Többszörös öröklődés

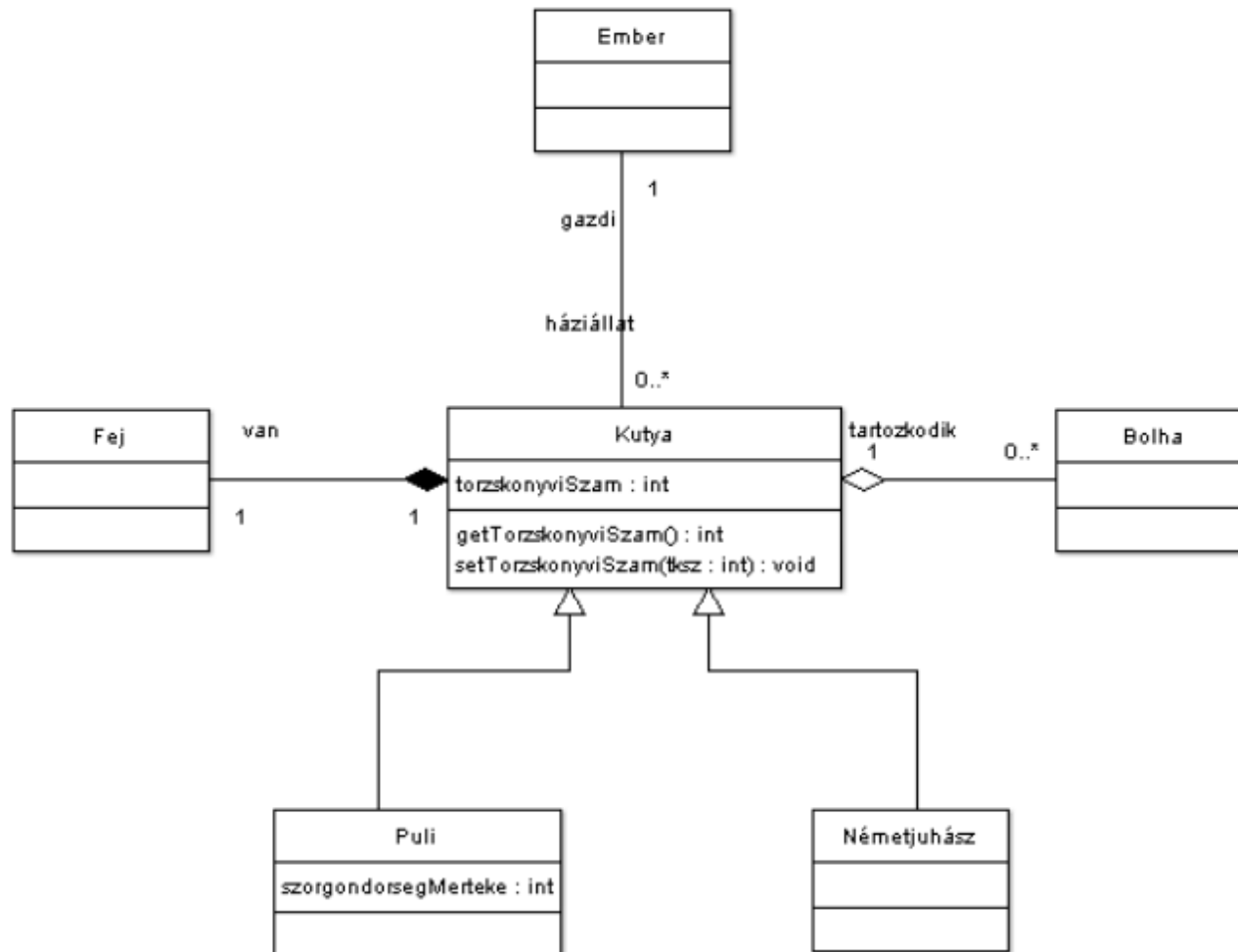


UML Osztálydiagram példa



UML Osztálydiagram példa

AggregacioPeldakiegesztve



Osztályok közötti kapcsolatok – implementációs példák

- Asszociáció
 - A osztály egy metódusának paramétere B típusú
 - A osztály egy metódusa hívja B egy metódusát
 - A osztály egy metódusa (lokálisan) példányosítja B-t
 - A osztály egy metódusának visszatérési értéke B típusú
- Aggregáció
 - A osztály tartalmaz egy B-re mutató pointert/referenciát
- Kompozíció
 - A osztály tartalmaz egy B adattagot
- Öröklődés
 - A osztály B osztályból származik



UML Eszközök

- ArgoUML:
 - <http://argouml.tigris.org/>
- BOUML:
 - <http://www.bouml.fr/>
- yEd
 - http://www.yworks.com/en/products_yed_about.html
- Gliffy:
 - <https://www.gliffy.com>
- UML Eszközök listája:
 - http://en.wikipedia.org/wiki/List_of_UML_tools



Feladatok

- Modellezd a bankkártyás fizetést!
- A modellben szerepeljenek a következő osztályok:
 - Bank
 - Számla
 - Személy
 - Kártya
 - Terminál (két fajta: automata és POS)
- valamint a közöttük lévő kapcsolatok.

