



# UML

# UML

- *Unified Modeling Language*
- Mi ez?
  - Grafikus modellező nyelv
- Mire való?
  - Objektorientált modellezés, tervezés
  - Probléma specifikációja
  - Probléma megoldása
  - Dokumentáció
  - Grafikus szemléltetés több nézetben
    - Use case, class diagram,...

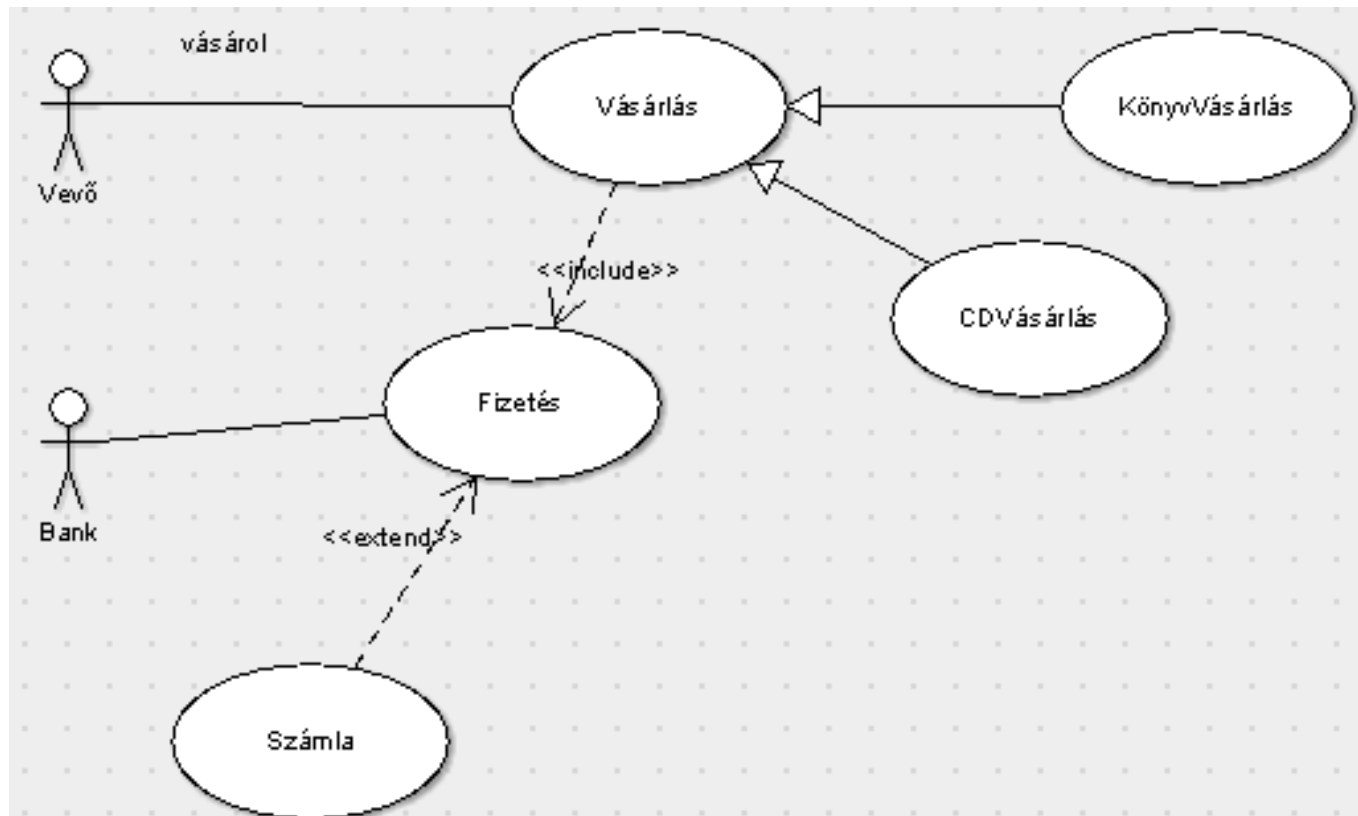
# Használati eset modellezés

- Rendszer viselkedését modellezi:
  - funkcionalitás
  - kommunikáció eszköze (felhasználó felé is)
  - legmagasabb absztrakció
- Megvilágítja:
  - rendszer tervezett funkcióit (használati eseteit)
  - rendszer környezetét (szereplők – Actor)
  - ezek közötti kapcsolatokat (Use Case Diagram)

# UML – Use Case Diagram kapcsolatok

- **Asszociáció**
  - Felhasználó és használati eset közötti kommunikáció, általában ige
- **Általánosítás**
  - Egyik használati eset vagy aktor általánosabb formája a másiknak
- **Kiterjesztés**
  - Egyik használati eset kiterjeszti újabb funkcionalitással a másikat
  - Extend kulcsszó
- **Tartalmazás**
  - Egyik használati eset tartalmazza a másikat
  - Include kulcsszó

# Use case példa

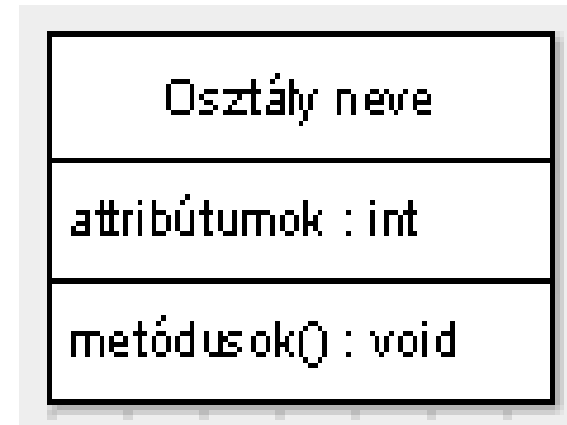


# Eseményáramlás

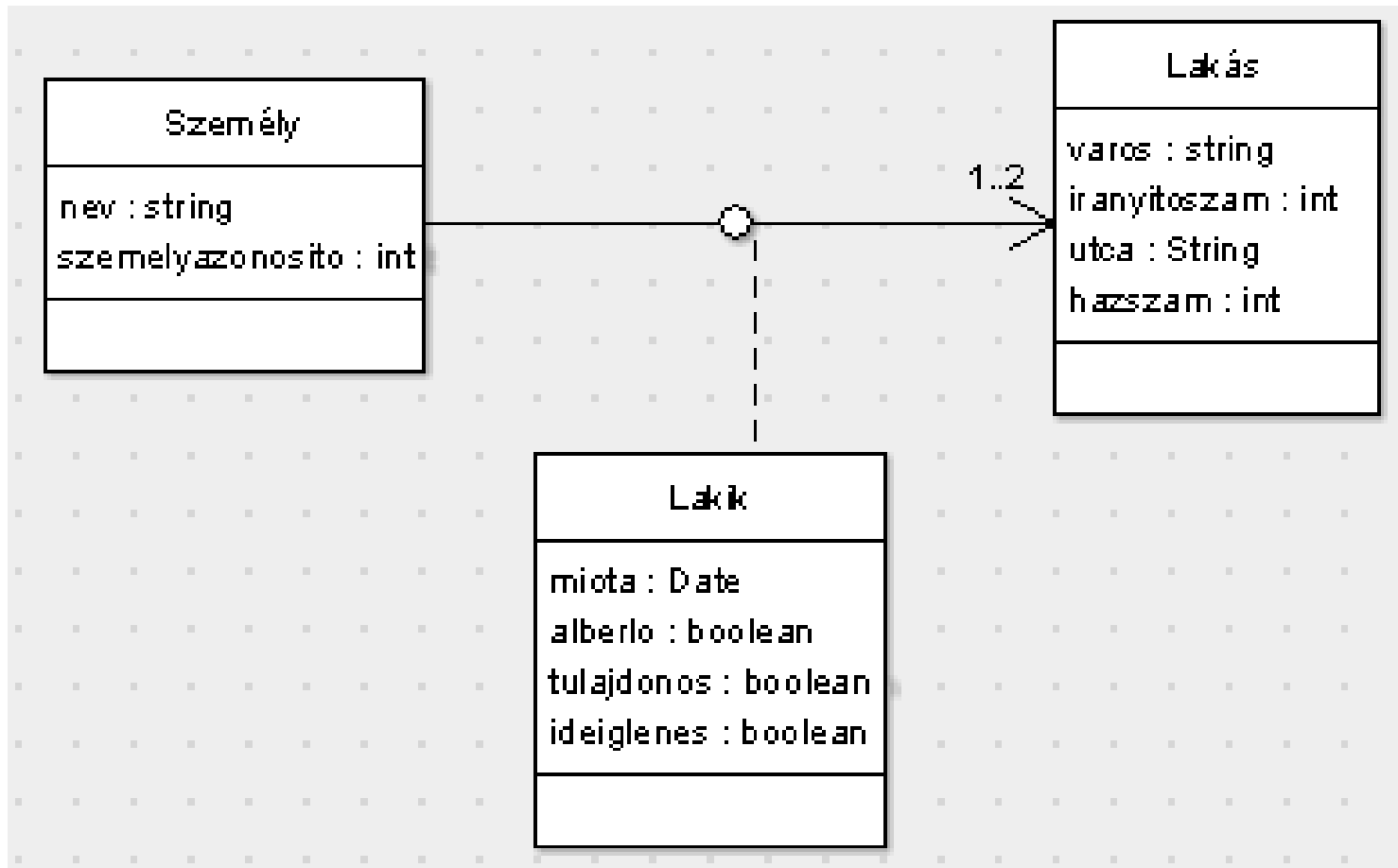
- A megrendelő által érthető legyen
  - teljes mondatokat írjunk, szóhasználatban igazodjunk a megrendelőhöz  
(pl. ügyfél – kliens, aru – cikk, ...)
- Adjuk meg pontosan a használati esetet elindító eseményt
- Az eseményáramlást külső szemszögből mutatjuk be, nem taglaljuk a rendszer belső működését
- Adjuk meg hogy mi alapján ellenőrizhető hogy a használati eset elérte célját

# UML – Class Diagram

- *Osztálydiagram*
- A probléma megoldását leíró szerkezeti diagram
- Statikus
- Elemei
  - Osztályok, csomagok, interfészek, megjegyzések
- Kapcsolatok
  - Asszociáció
  - Aggregáció
  - Kompozíció
  - Öröklődés
  - Függőség



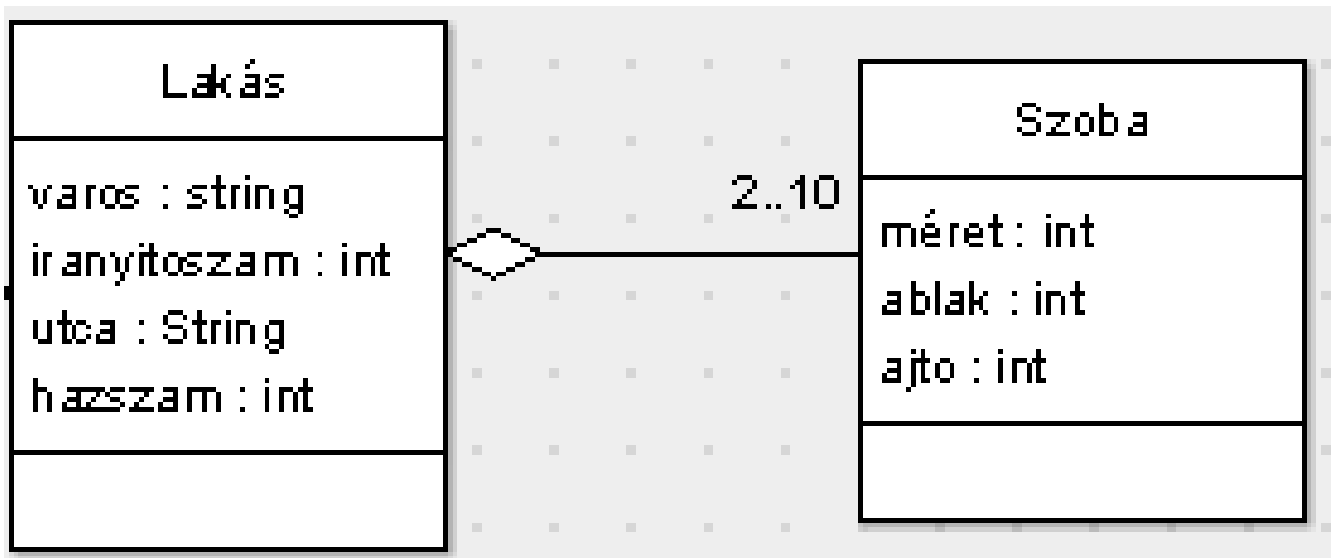
# UML - Class Diagram - Association





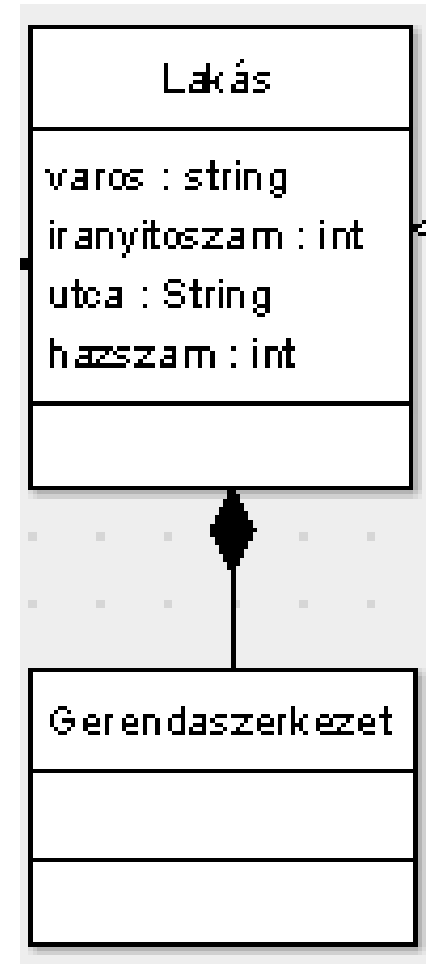
# UML – Class Diagram - Aggregation

- Speciális asszociáció
- Egyik objektum része a másiknak
- Nem erős tartalmazás



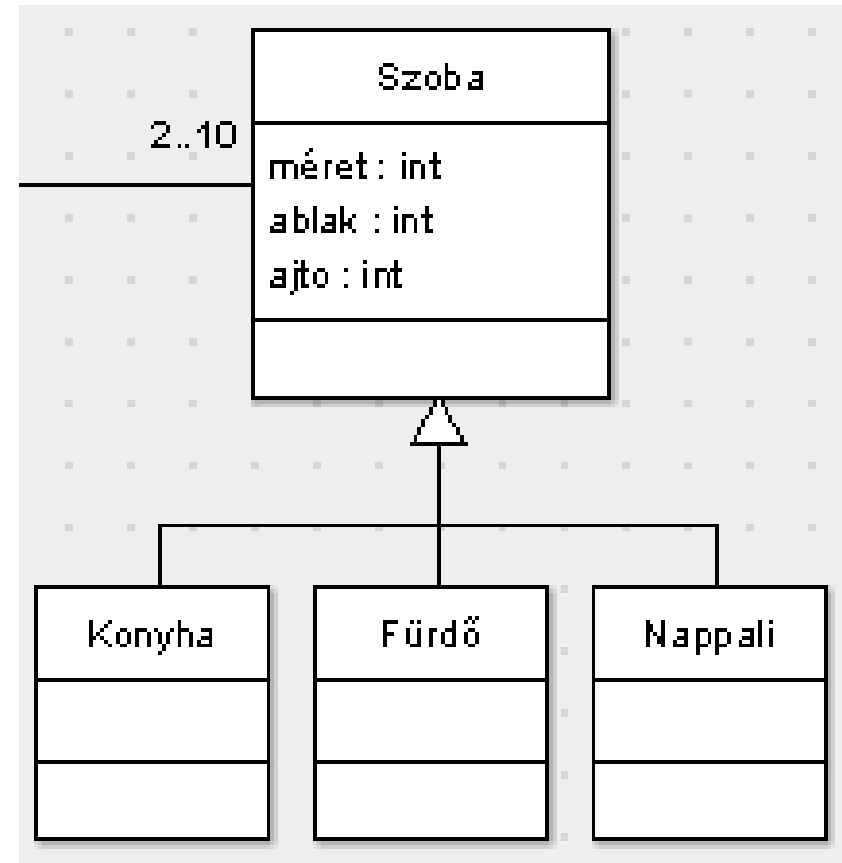
# UML Class Diagram - Composition

- Speciális aggregáció
- Fizikai, erős tartalmazás
  - A tartalmazó a tartalmazott nélkül nem létezhet



# UML Class Diagram – Általánosítás és specializáció (öröklődés)

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



# Sztereotípusok

- Osztálydiagramon sztereotípusok feltüntetése
- Osztály neve felett
- Osztályok felfedezése
  - Nehéz feladat!
  - ajánlatos sztereotípusonként így kezdeni: boundary, control, entity
- <<Boundary>> (határ) osztályok:
  - rendszer környezete és belseje közötti kommunikációt valósítják meg
  - interfészt képeznek a felhasználó vagy más rendszer (Szereplő) felé
  - felhasználói interfész is ide tartozik

# Sztereotípusok

- <<Control>> (vezérlő) osztályok:
  - használati eset(ek) szekvenciális viselkedését valósítják meg
  - „használati eset végrehajtását” végzi
  - általában egy szereplő/használati eset párhoz hozzátartozik egy Control osztály
  - entitás osztályhoz tartozzon a funkcionalitás, vagy új vezérlő osztályba kerüljön?

# Sztereotípusok

- <<Entity>> (entitás) osztályok:
  - olyan információt/viselkedést modellez, amely általában hosszú életű
  - valós világ entitásai, kevésbé érzékenyek a környezetük változásaira
  - általában alkalmazás-függetlenek
  - jó kiindulás lehet: Use Case leírásokban a feladatkiosztások főnevei

# Csomagok (Package)

- Nagy rendszereknél elkerülhetetlen az osztályok csoportosítása
- Hierarchikus szerkezetet biztosít
- Magasabb szintű absztrakciót valósít meg

## VideoKölcsönzes

## Kölcsönzes

&lt;&lt;control&gt;&gt;

**KölcsönzesControl**

```
dbControl : DBControl
finish(tagID : int, movieID : int) : void
setTagId(tag : Tag) : void
```

## Adatbáziskezeles

&lt;&lt;control&gt;&gt;

**DBControl**

```
dbConnect : DBConnect
makeSelectString(inout columns : string[], inout table : string, inout where : string[]) : string
makeUpdateString(inout column : string, inout table : string, inout where : string) : void
makeInsertIntoString(inout table : string[], inout values : string[]) : void
```

&lt;&lt;control&gt;&gt;

**VideoKölcsönzoMain**

```
main() : int
```

## Rendeles

&lt;&lt;control&gt;&gt;

**RendlesControl**

## Tagok

&lt;&lt;control&gt;&gt;

**TagControl**

&lt;&lt;boundary&gt;&gt;

**DBConnect**

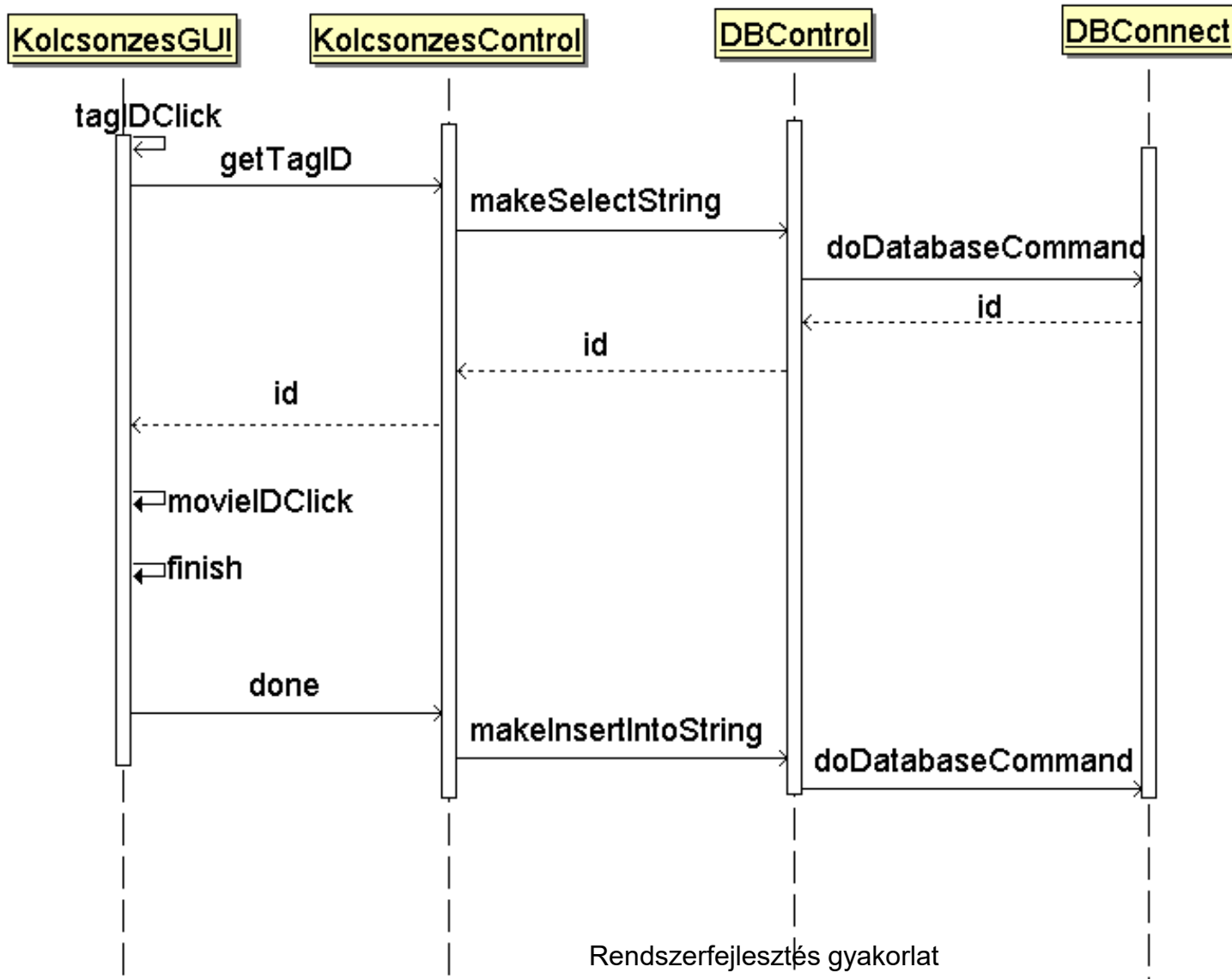
```
doDBCommand(inout command : string) : string
initDBConnect() : void
```



# Szekvencia Diagramok

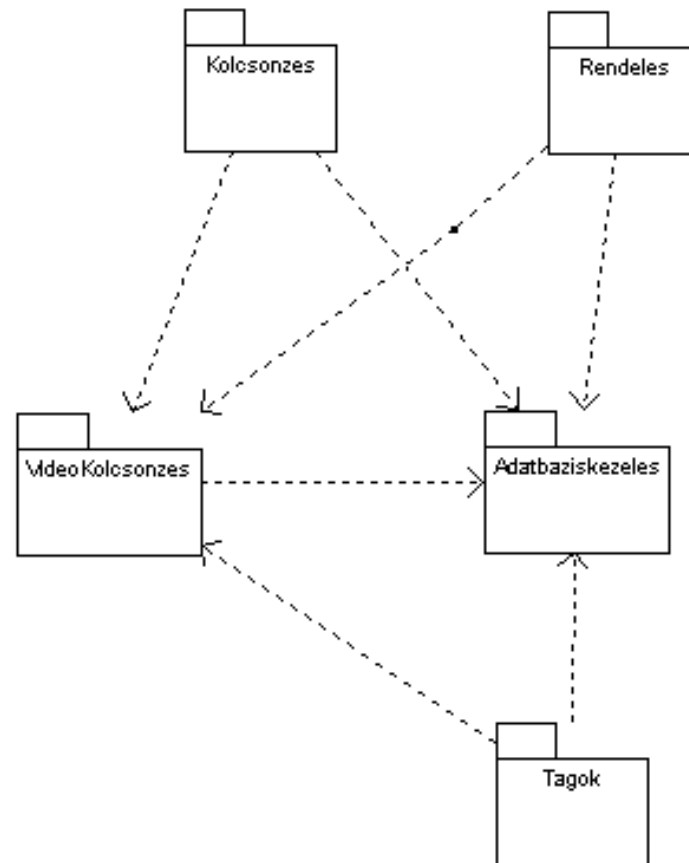
- Objektum-kölcsönhatásokat mutat be az idő függvényében
- A scenárióban szereplő objektumokat és osztályokat ábrázolja a közöttük küldött üzenetekkel
- Idő-orientált nézet
- Az üzenetek (Message) a scenárió funkcionalitását valósítják meg
- Használati esetekkel szoros kapcsolatban állnak (általában a Használati Eset nézetnek a részei)

# Szekvencia diagram - példa



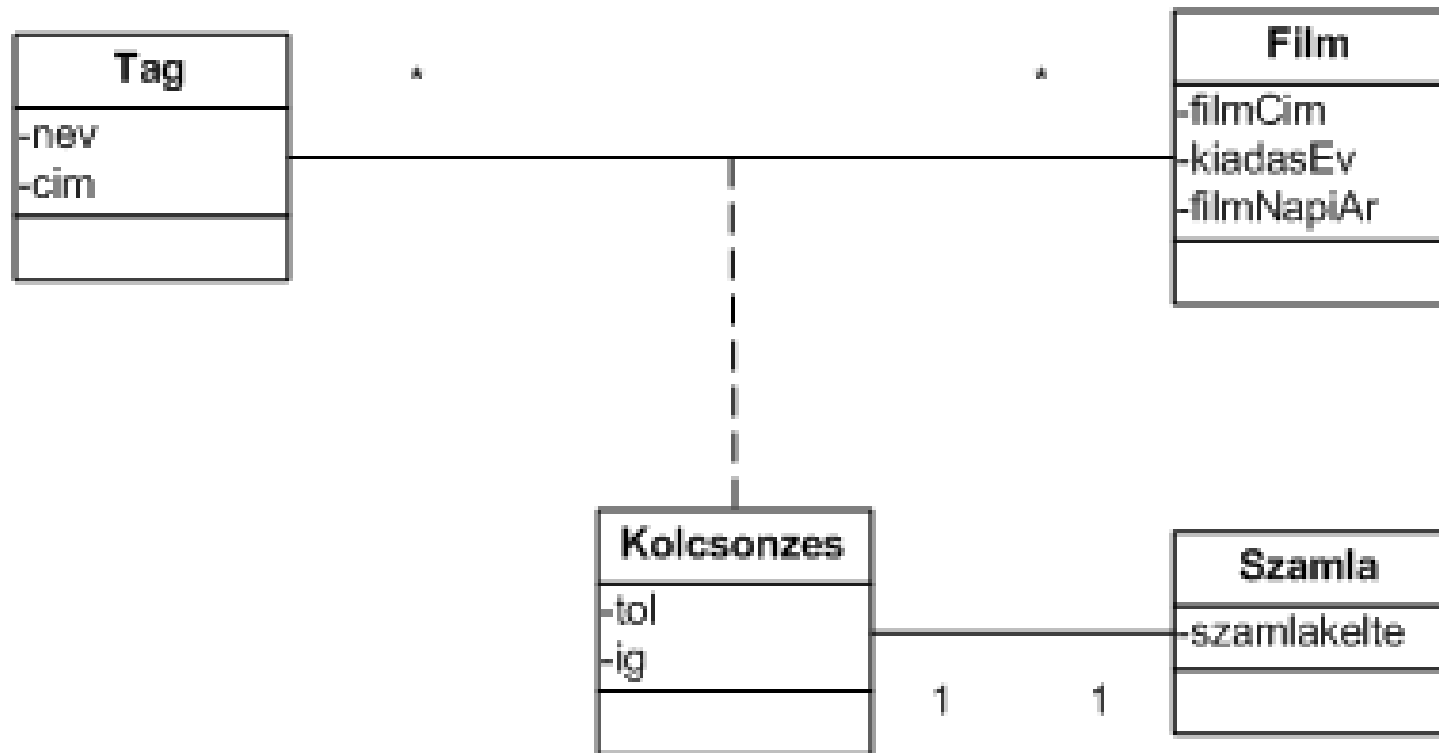
# Package diagram

- Csomagok és a közöttük fennálló függőségek



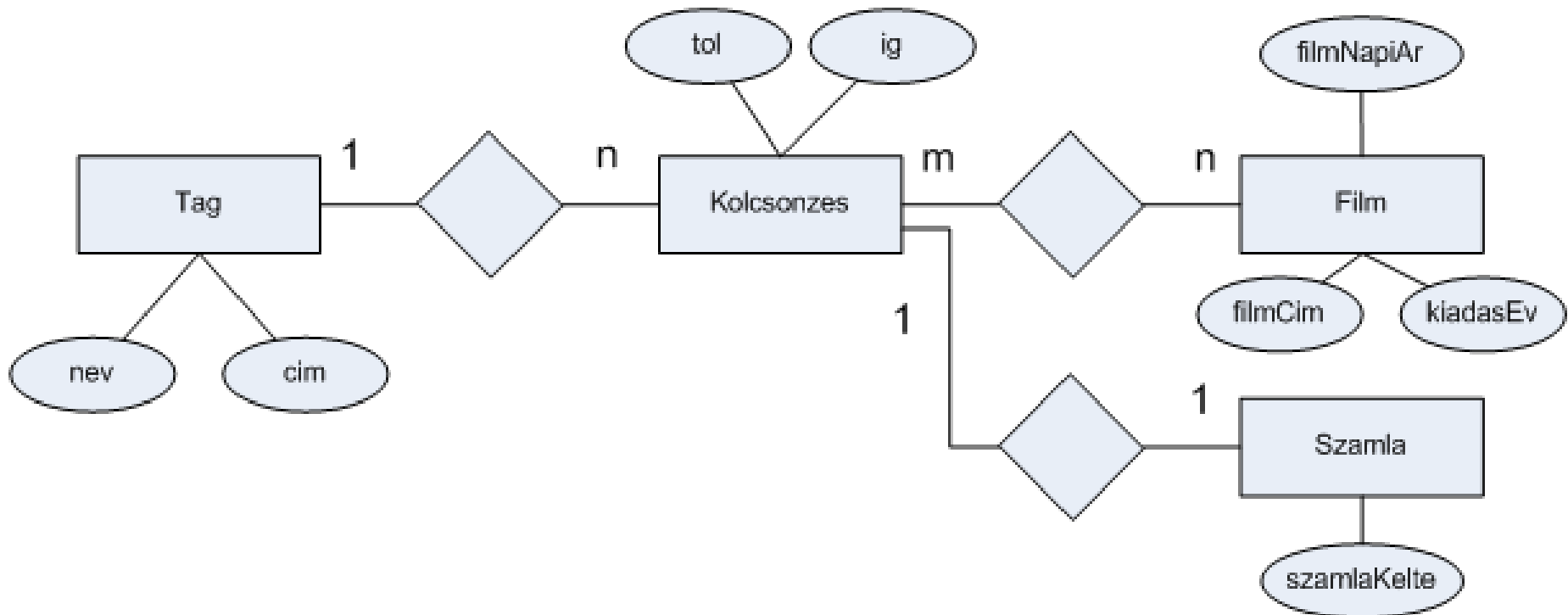
# UML-EK leképezés

## ■ Entitások és kapcsolataik (UML)



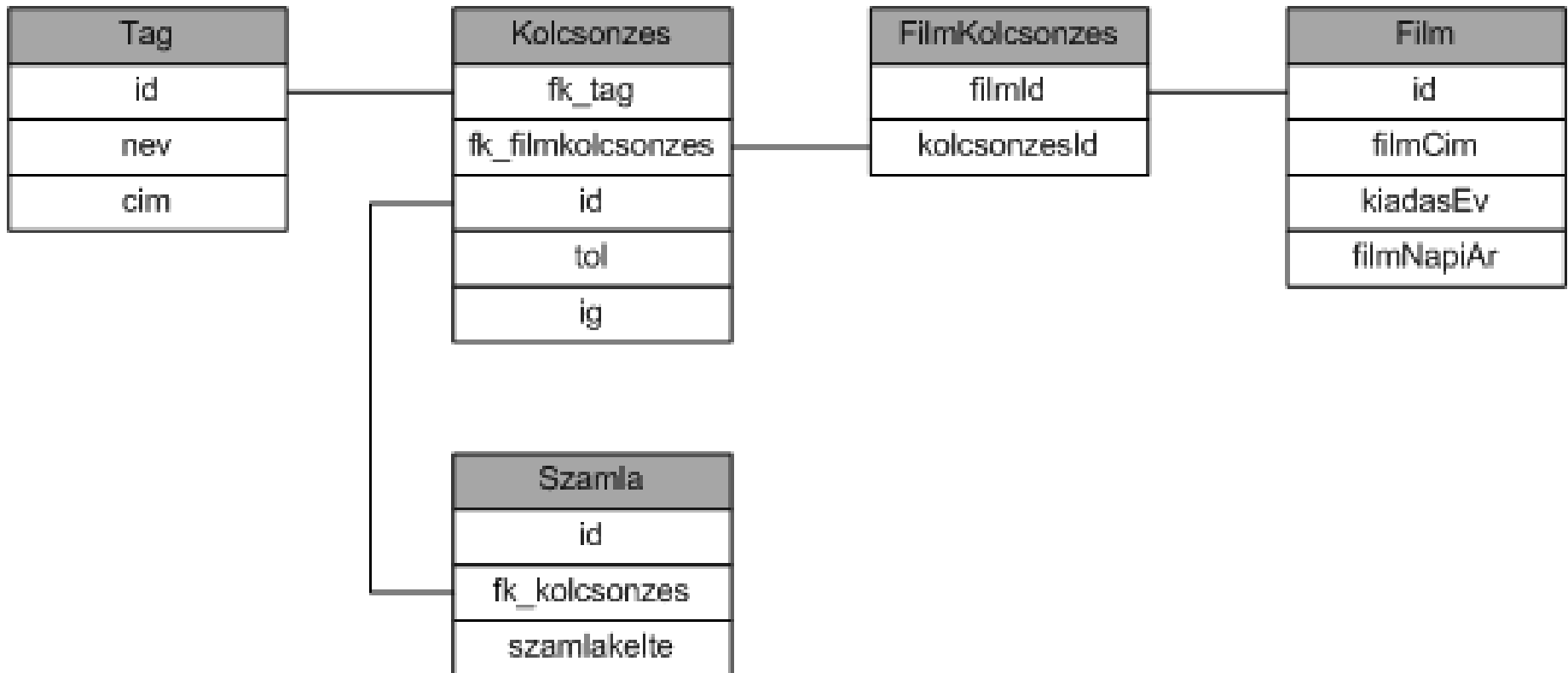
# Video kölcsönző

## ■ Entitások és kapcsolataik (EK)



# Video kölcsönző

## ■ Entitások és kapcsolataik (db séma)

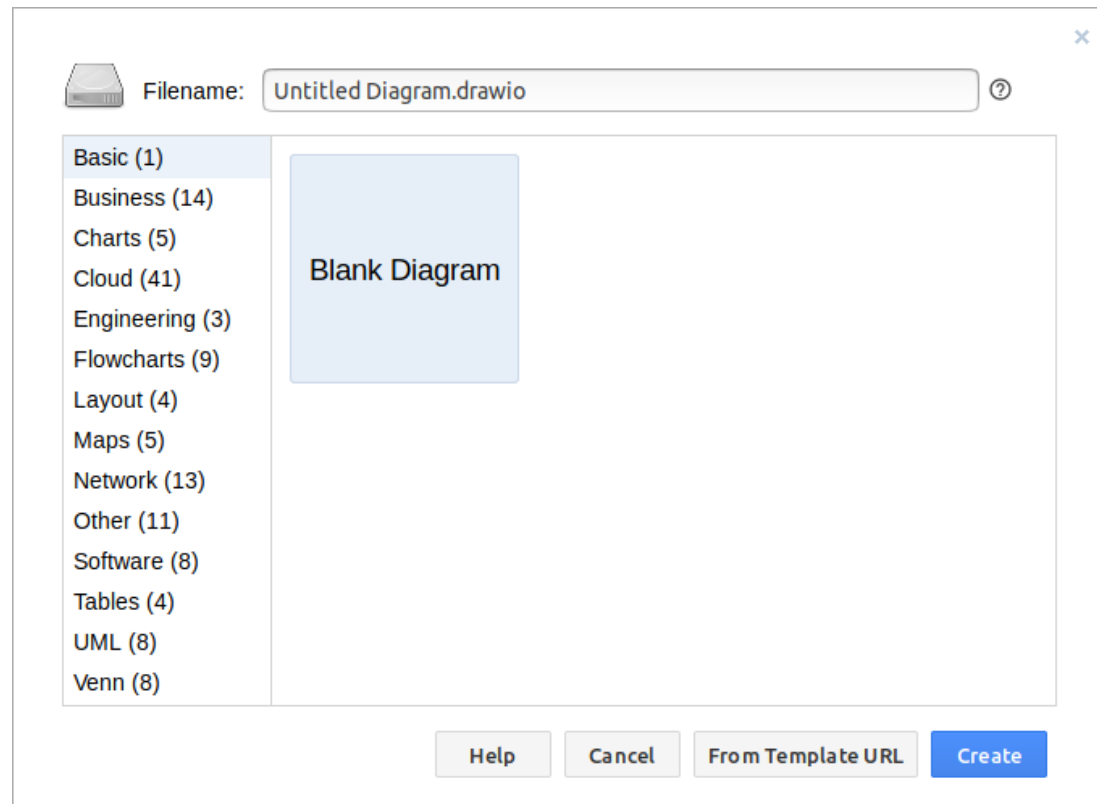


# Ajánlott UML tool: draw.io

- Nyissunk egy böngészőt és gépeljük be:
- <https://www.draw.io/>
- Nyílt forráskódú böngésző alapú diagramszerkesztő
- A diagramok XML struktúrába/ból exportálható/importálható
- Nem csak UML diagramokhoz!

# Ajánlott UML tool: draw.io

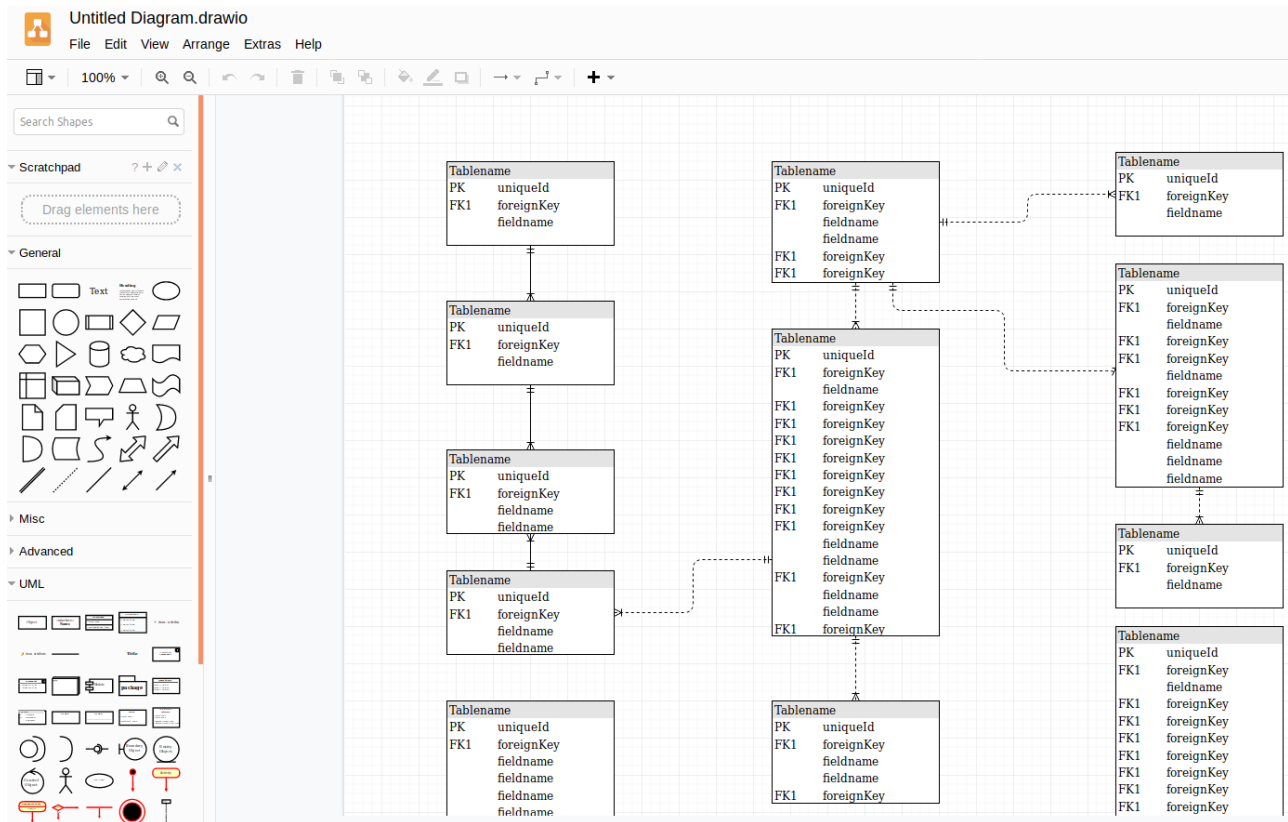
- Software vagy az UML kategóriákat érdemes választani





# Ajánlott UML tool: draw.io

- Bal oldalon nyissuk le az UML komponenst



# Ajánlott UML tool: draw.io

- Use Case: UML
- Class: Software
- EK/Database: Software, General komponensek
- Package: UML
- Sequence: UML

# Más UML toolok

## ■ Ingyenesen letölthető toolok

### □ ArgoUML

- <http://argouml.tigris.org/>

### □ BOUML

- <http://bouml.free.fr/download.html>

### □ StarUML

- <http://staruml.sourceforge.net/>

### □ További

- [http://en.wikipedia.org/wiki/List\\_of\\_UML\\_tools](http://en.wikipedia.org/wiki/List_of_UML_tools)
  - (Non-proprietary UML tools)

- Végezetül, az rf-uml\_peldak.ppt fájlban, és a mellékelt drawio fájlokban találunk pár konkrét megoldást.