



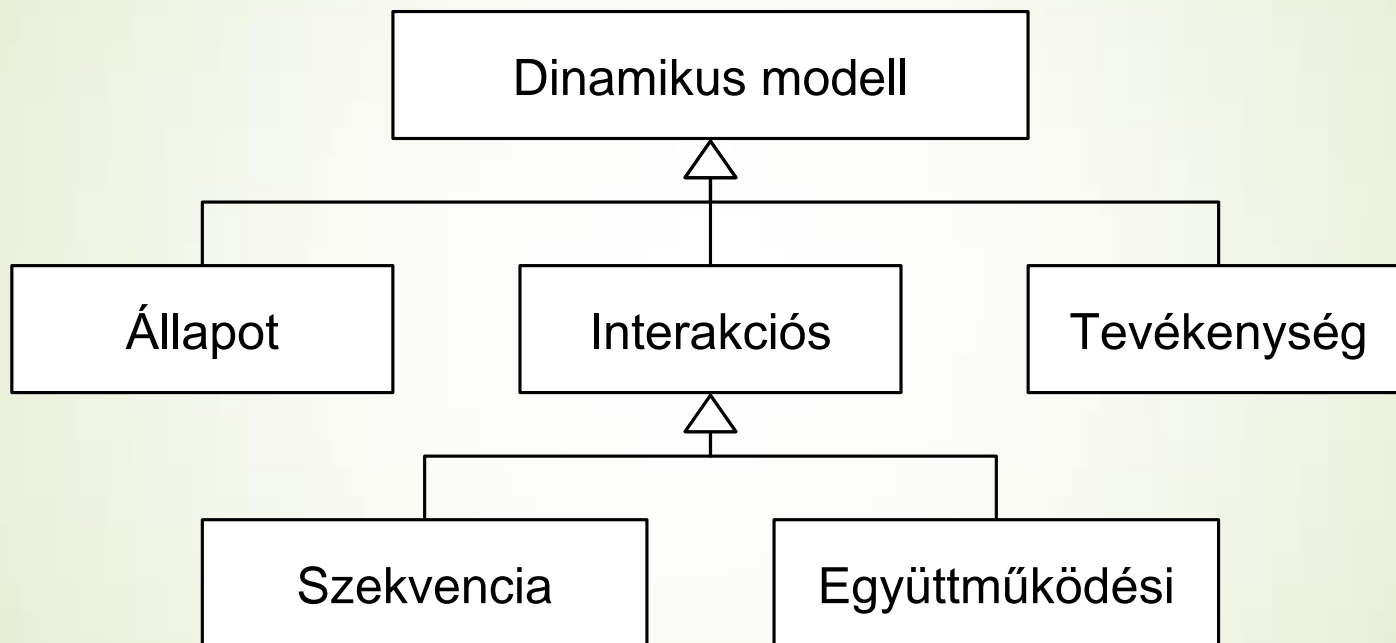
# Programozási technológia

Dinamikus modell

Állapotdiagram, szekvencia diagram

Dr. Szendrei Rudolf  
ELTE Informatikai Kar  
2018.

# Bevezető



# Alapfogalmak

## ➤ **Állapot:**

- Az objektum állapotát az attribútumok konkrét értékeinek n-esével jellemezzük.  
Például pont esetén annak koordinátái

## ➤ **Esemény:**

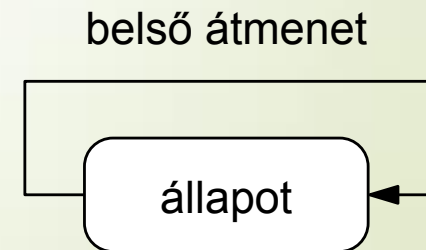
- Eseménynek nevezzük azt a tevékenységet, történést, amely valamely objektum állapotát megváltoztatja.  
Egy pont esetén például annak eltolása



# Állapot informális definíciója

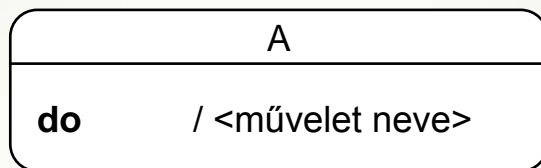
Az állapot tulajdonságai:

1. Van azonosítója, így megkülönböztethetők egymástól, például van nevük (pl.: veremnél: „normál”, „tele”, ...). Az azonosító lehet egy vagy több attribútumának értéke is, illetve ezen értékeket meghatározó állítás, állapotinvariáns.
2. Az állapot általában esemény, eseménysorozat hatására jön létre. Speciális állapot a kezdő állapot.
3. Az állapot addig marad fenn, amíg az objektumok attribútumainak értékei kielégítik az állapothoz rendelt invariánst (vagyis a belső átmenetek nem változtatják meg az objektum állapotát).

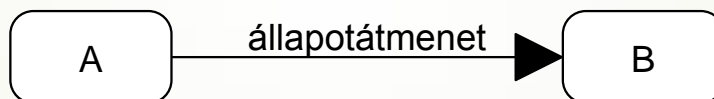


# Állapot informális definíciója – (folyt.)

- Az állapotokat gyakran a belső tevékenység, művelet nevével azonosítjuk.



- Az állapot megszűnése esemény hatására következik be, ilyenkor egy másik állapotba kerül az objektum.



- Az objektum megszűnése szintén állapotátmenet hatására következik be. Ekkor egy rendszeren kívüli befejező állapotba kerül.

# Állapotdiagram

## Állapot leírása

- ▶ **state:** <az állapot azonosítója>;
- ▶ **comment:** <az állapot rövid magyarázó leírása>;
- ▶ **pre-events:** <az állapotot előidéző események azonosítói>;
- ▶ **invariant:** <az állapot invariánsának leírása>;
- ▶ **post-events:** <az állapot megszűnéséhez kötődő események>

# Állapotdiagram

## Példa: ébresztő óra csengetés állapotának leírása

- **state:** ébresztő csengetés;
- **comment:** amikor az előzetesen beállított időpont bekövetkezik, a csengő megszólal, és 10 mp-ig csörög;
- **pre-events:** idő beállítás (ébredtési idő); minden más esemény, ami nem „idő beállítás törlés”; aktuális idő = ébredtési idő;
- **invariant:** idő beállítás bekapcsolva és  $\text{ébredtési idő} \leq \text{aktuális idő} \leq \text{ébredtési idő} + 10\text{mp}$ ;
- **post-events:**  $\text{aktuális idő} = \text{ébredtési idő} + 10 \text{ mp}$   
→ visszaállítás (ébredtési idő).

# Állapotdiagram

## Esemény informális definíciója

Az eseményeket a következő tulajdonságok jellemzik:

1. lehet paraméter nélküli  
(például az „enter” gomb megnyomása);
2. lehet paraméteres  
(például vermek esetén a  $push(s,e)$ );
3. az események között sorrendiség állhat fenn, azaz beszélhetünk
  - megelőző eseményről, illetve
  - rákövetkező eseményről;
4. az eseménynek előfeltétele is lehet.



# Állapotdiagram

## Esemény leírása

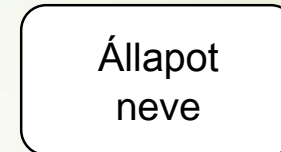
- **event:** <esemény neve, azonosítója>;
- **comment:** <az esemény jelentésének rövid leírása>;
- **parameters:** <a paraméterek listája>;
- **precondition:** <az bekövetkezését szükségszerűen megelőző állítás>;
- **preevents:** <az eseményt megelőző esemény>

# Állapotdiagram definíciója

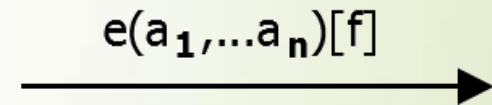
➤ Az állapotdiagram egy egyszeresen összefüggő irányított gráf, amelynek csomópontjaihoz az állapotokat, éleihez pedig az eseményeket rendeljük.

## ➤ Jelölések

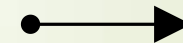
➤ Állapot



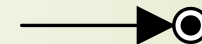
➤ Állapotátmenetet okozó esemény



➤ Kezdő állapot



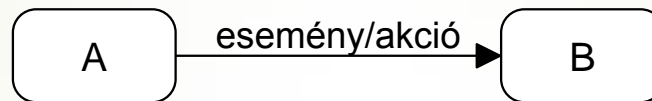
➤ Befejező állapot



# Állapotdiagram

## Esemény és akció

- Sokszor az esemény végrehajtása időbe telik, ezért célszerű megkülönböztetni az akciótól.
- Ekkor az *akció* az, ami egy időpillanathoz kötődik.



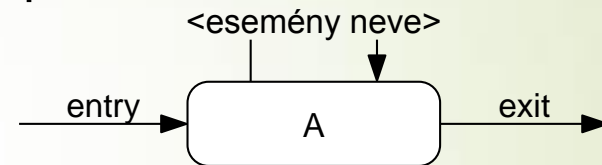
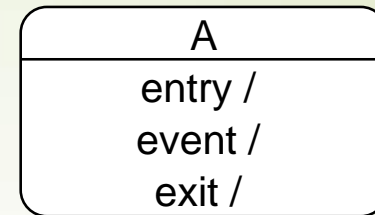
## Az események fázisai

- *entry* – a belépés akciója, amely elindítja az állapotot létrehozó eseményt / eseménysorozatot
- *event* – az állapothoz kötődő belső események sorozata, amely az adott állapothoz kötődő belső állapotokat jelenti
- *exit* – az az exit akció, amely az esemény befejezését, az állapotból való kilépést eredményezi.

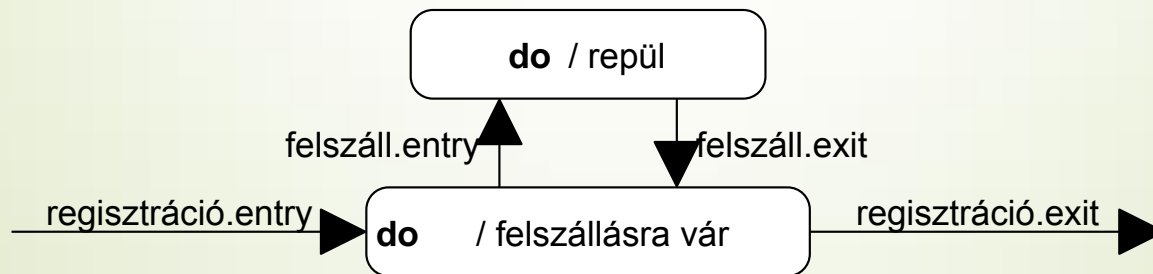
# Állapotdiagram

Az események fázisai (folyt.)

- Az esemény fázisainak jelölése
- Az esemény folyamatosságának szemléltetése



Példa – repülőjárat-nyilvántartás leegyszerűsített modellje



# Állapotdiagram

## Az állapotdiagram bonyolultsága

- Az állapotdiagram egyszerű esetekben is áttekinthetetlenné válhat (pl. adott 5 állapot, ahol mindegyikből van átmenet mindegyikbe)
- A bonyolultság csökkentésére két általános módszer létezik:
  - az állapotok általánosítása és
  - az állapotok aggregációja.

# Állapotdiagram

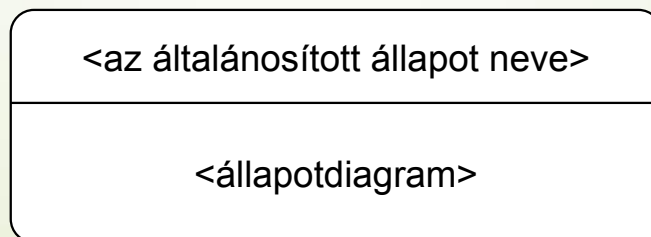
## Állapotok általánosítása

1. Az általánosított állapot véges számú részállapot összessége
2. A részállapotok örökölhetik az általánosított állapot tulajdonságait:
  - attribútumait (állapotjellemzőit);
  - eseményeit, akcióit.
3. Az általánosított állapot állapotinvaránsa a részállapotok állapotinvariánsainak diszjunkciója (azaz az objektum az általánosított állapotban mindig valamelyik részállapotban van).
4. A részállapotokhoz az az állapotdiagram tartozik, amelyben szereplő állapotátmenetek az általánosított állapotot nem változtatják meg.
5. A részállapotok lehetnek általánosított állapotok is.

# Állapotdiagram

## Állapotok általánosítása (folyt.)

6. Az állapotdiagramban kell lennie olyan részállapotnak, amely az általánosított állapot „*entry*” akcióját örökli.
7. Az állapotdiagramban kell lennie olyan részállapotnak, amely az általánosított állapot „*exit*” akcióját örökli.
8. Az általánosított állapot megfelelő objektuma az „*entry*” akció hatására létrejön, a megfelelő objektum pedig az állapotdiagram „*exit*” akciójának hatására semmisül meg.



# Állapotdiagram

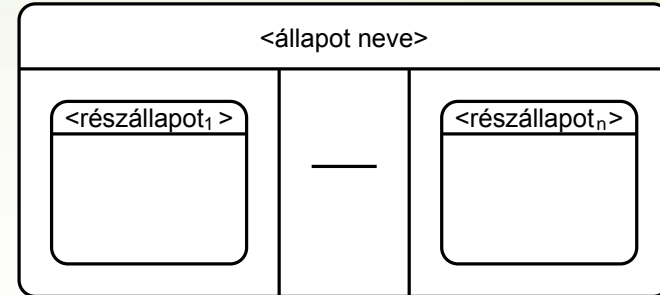
## Állapotok aggregációja

1. Az aggregációval létrejövő állapot egymástól független részállapotok egy véges halmaza.
2. A részállapotok lehetnek általánosított állapotok is, amelyekhez tehát állapotdiagramok tartoznak.
3. Minden állapotdiagram kell lennie olyan részállapotnak, amelybe az aggregátum „*entry*” esemény hatására kerül, és olyanak, amelyből kilépéskor az „*exit*” esemény hatására szűnik meg.
4. Az aggregációval létrejött állapot invaránsa a részállapotok invariánsainak konjunkciója, azaz az aggregációs állapot objektuma az aggregációt alkotó részállapotokban egyidejűleg létezik.
5. Az állapotok aggregációja az állapoton belüli állapotdiagramok közötti párhuzamosság egy megjelenési formája.



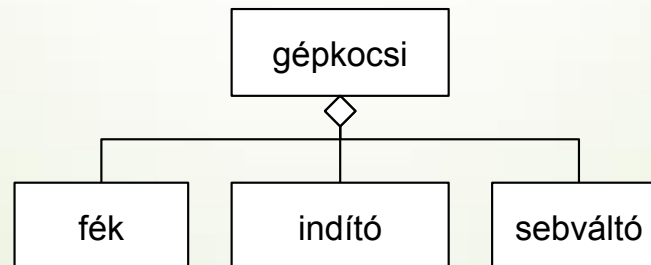
# Állapotdiagram

## Állapotok aggregációja

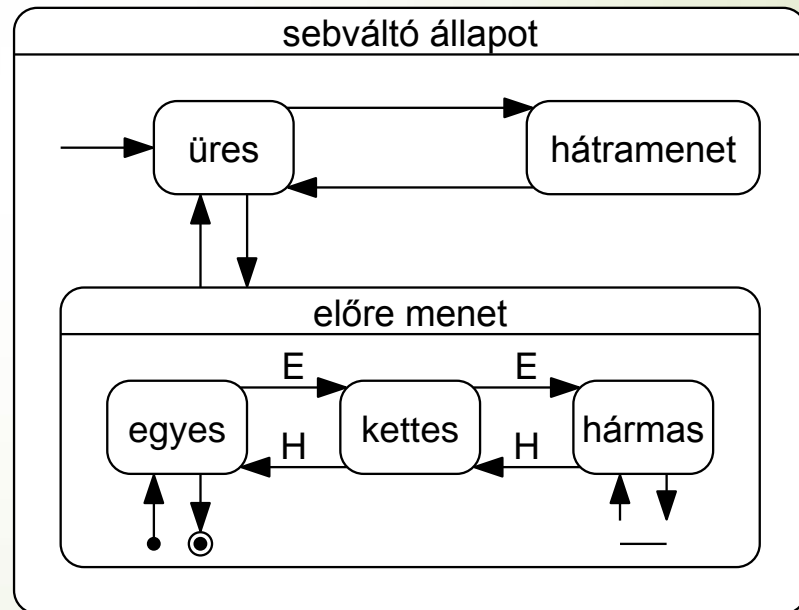
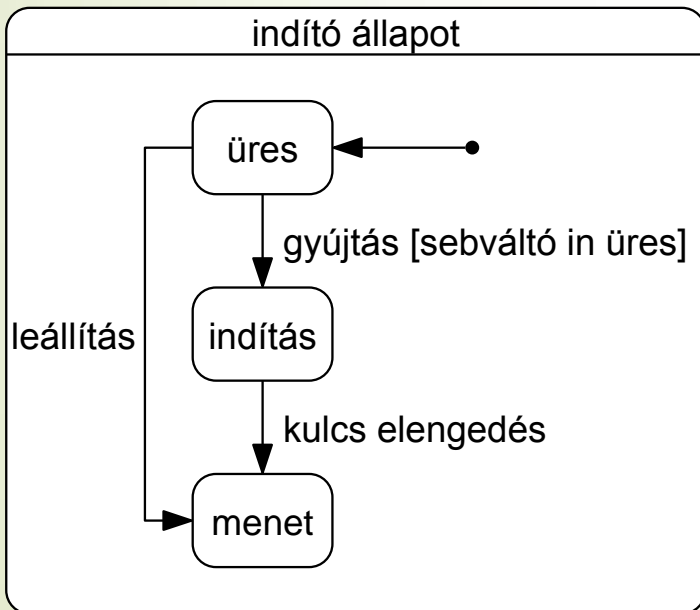
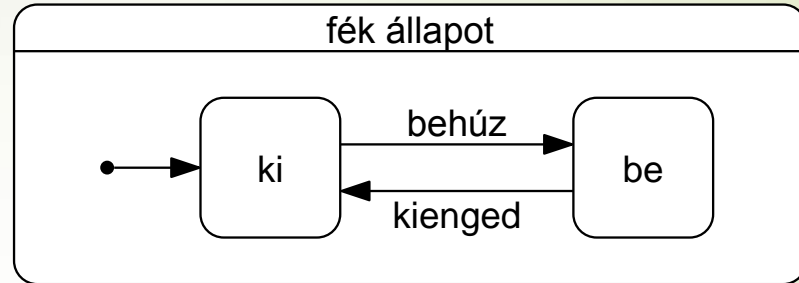


## Példa – Gépkocsi

Modellezzünk egy leegyszerűsített gépkocsit, amia következő egységekből áll: fék, indító, sebességváltó



# Állapotdiagram – Gépkocsi példa



# Állapotdiagram

## Az állapot általános fogalma

1. Az állapotnak van azonosítója.
2. Esemény hatására következik be.
3. Mindaddig fennmarad, amíg az objektumok attribútumai az állapot invariánsát kielégítik.
4. Az állapot megszűnéséhez eseménysorozat kötődik.
5. Az állapot lehet részállapotok általánosítása. Ilyenkor az állapothoz állapotdiagram is tartozik.
6. Az állapot lehet más állapotok aggregációja.
7. Az állapot lehet pseudoállapot:
  - Kezdetkor a külső állapot.
  - Befejezéskor a külső állapot.
  - Hisztorizációs állapot, melyhez hisztorizációs indikátor társul.



# Szekvenciadiagram

## A szekvenciadiagram alapfogalmai, komponensei

- Osztályszerep
- Osztályszerep életrajza
- Aktivációs életrajz
- Üzenet

# Szekvenciadiagram

## Osztályszerp

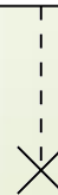
- Az osztály szerepét az osztályok közötti üzenetben megtestesítheti az osztály egy vagy több objektuma, amelyek az üzenetküldés szempontjából konform módon járnak el.
- Az osztályszerp megnyilvánulhat az osztályok egy halmazának megtestesítőjeként is (pl. generic form).
- Jelölése:

< szerep neve > : < osztály neve >

## Osztályszerp életvonala

- Az életvonal az osztályszerp időben való létezését jelenti.

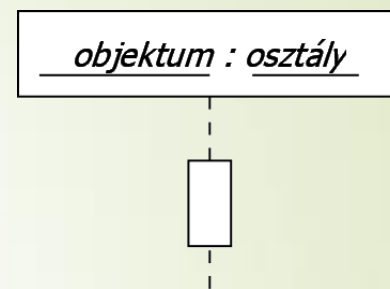
< szerep neve > : < osztály neve >



# Szekvenciadiagram

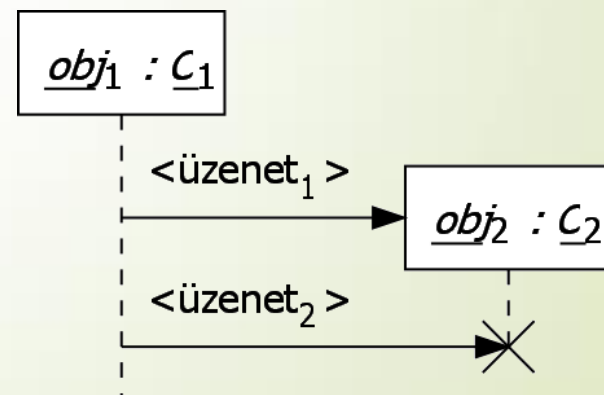
## Az osztályszerep aktivációs életrajza

- Az aktivációs életrajz az osztályszerepnek azt az állapotát jelöli, amelyben az osztályszerep megtestesítői műveletet hajtanak végre, és más objektumok vezérlése alatt állnak.



## Objektum létrehozása és megsemmisítése

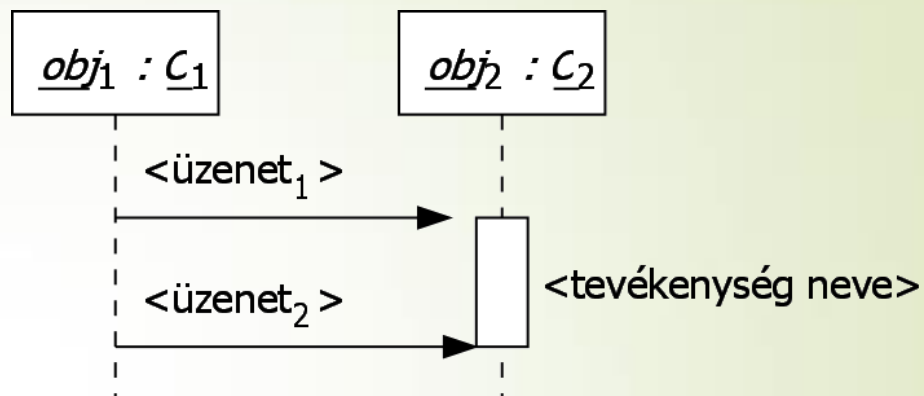
- Egy objektum létrejöhet egy másik objektum létrehozó üzenetének a hatására és megsemmisülhet, ha a másik objektum egy törlést jelentő üzenetet ad ki.



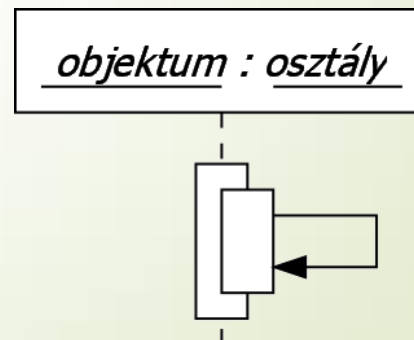
# Szekvenciadiagram

## Az objektum aktivációja

- ▶ Életvonalala során az objektum aktív módon viselkedhet, amíg valamilyen tevékenységet végez. Ezt előidézheti, de meg is szüntetheti egy másik objektum.



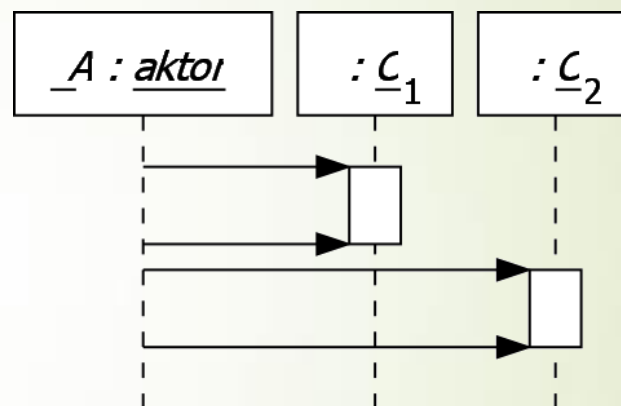
- ▶ Az objektum saját magát is aktiválhatja (*rekurzív aktiváció*).



# Szekvenciadiagram

## Az objektum aktivációja (folyt.)

- Az aktiváció *centrális*, ha minden objektumot egy *aktor* objektum aktivizál.



- Az aktor egy másik jelölése



A : aktor



# Szekvenciadiagram

## Üzenettípusok

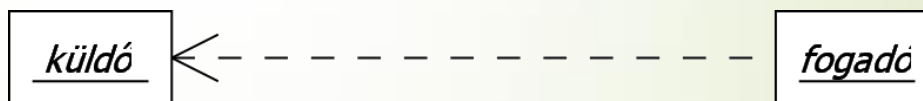
### ➤ Aszinkron üzenet

Egy aktív objektum üzenetet küld egy passzív objektumnak és folytatja a munkáját



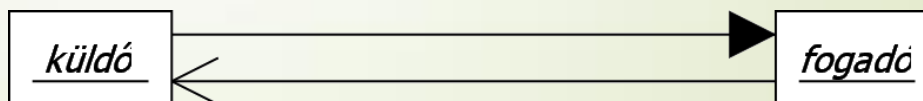
### ➤ Válasz üzenet

A küldő válasz üzenetet kap egy korábbi aszinkron üzenetére.



### ➤ Szinkronizációs üzenet

Az üzenet elküldésekor a küldő blokkolt állapotba kerül, amíg a fogadó nem válaszol az üzenetre.



# Szekvenciadiagram

## Üzenettípusok (folyt.)

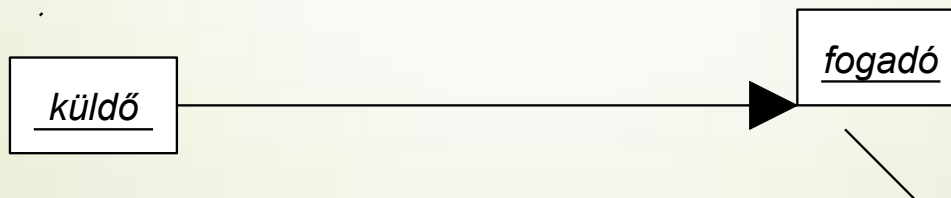
### ➤ Létrehozó üzenet

A küldő objektum létrehozza egy új objektumot.



### ➤ Megsemmisítő üzenet

A küldő objektum megsemmisíti a fogadó objektumot.



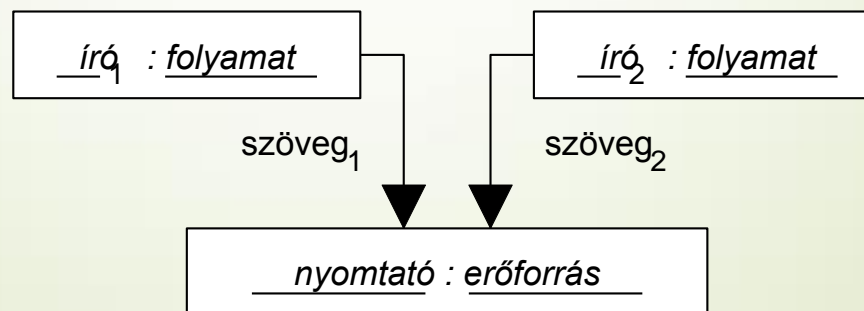
# Szekvenciadiagram

## Üzenettípusok (példák)

- Adott egy nyomtató, amit többen is szeretnének használni
- A nyomtatóhoz, mint erőforráshoz kölcsönös kizárással lehet, amit szinkronizációs üzenettel oldjuk meg
- A kölcsönös kizárás osztálydiagramja



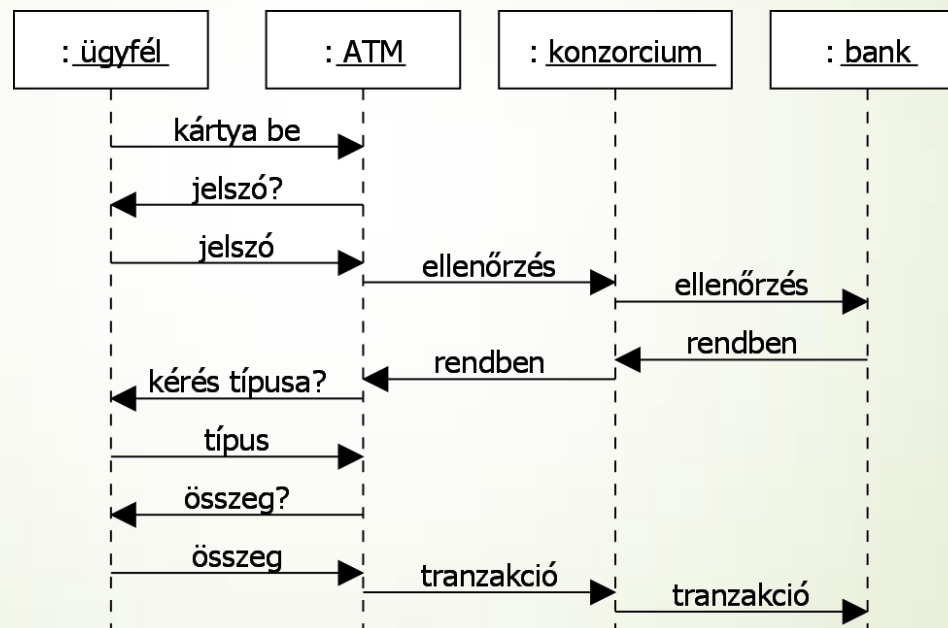
- A kölcsönös kizárás két folyamat esetén



# Szekvenciadiagram

## Üzenettípusok (példák)

- Szemléltessük az ATM készülék működését
- Most elegendő csak az üzenetek időben lejátszódó sorrendjét szemléltetni



- (A feladat egyben példa a decentralizáltan vezérelt rendszerre.)