



Programozási technológia

Dinamikus modell

Tevékenységdiagram,
Együttműködési diagram,
Felhasználói esetek diagramja

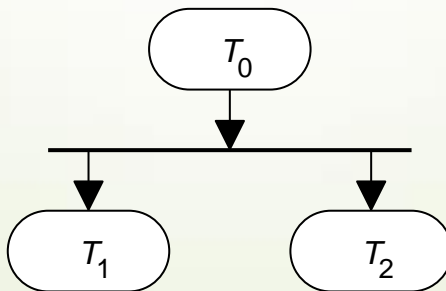
Dr. Szendrei Rudolf
ELTE Informatikai Kar
2018.

Tevékenység diagram

- ▶ A tevékenység (vagy más néven aktivációs) diagram a probléma megoldásának lépéseit szemlélteti, a párhuzamosan zajló vezérlési folyamatokkal együtt.
- ▶ Elsősorban az üzleti életben előforduló szervezetek munkafolyamatainak modellezésére használatos.
- ▶ Két fajtáját különböztetjük meg:
 - ▶ életfolyam alapú
 - ▶ sávós alapú

Sávos alapú tevékenység diagram

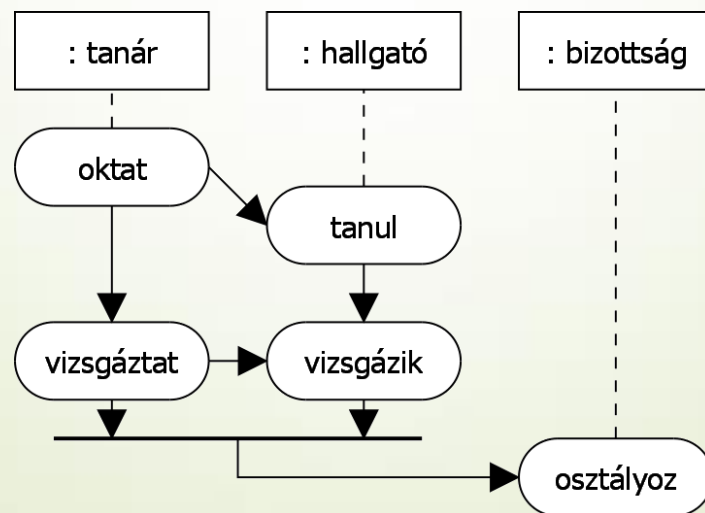
- A sávos alapú tevékenység diagram lehetőséget biztosít párhuzamos folyamatokkal leírására.
- A folyamatok közötti szinkronizációt egy vastagított vízszintes vonallal jelöljük.
- A szinkronizáció egyaránt vonatkozhat a kezdeti szinkronizációra vagy a befejeződésre, amikor is egy újabb tevékenység megkezdése előtt az adott folyamatoknak be kell fejeződniük.



Példa – életfolyam alapú tevékenység diagram 1.

Elemezzük az egyetemi oktatók, hallgatók és vizsgabizottságok tevékenységét egy adott félévben:

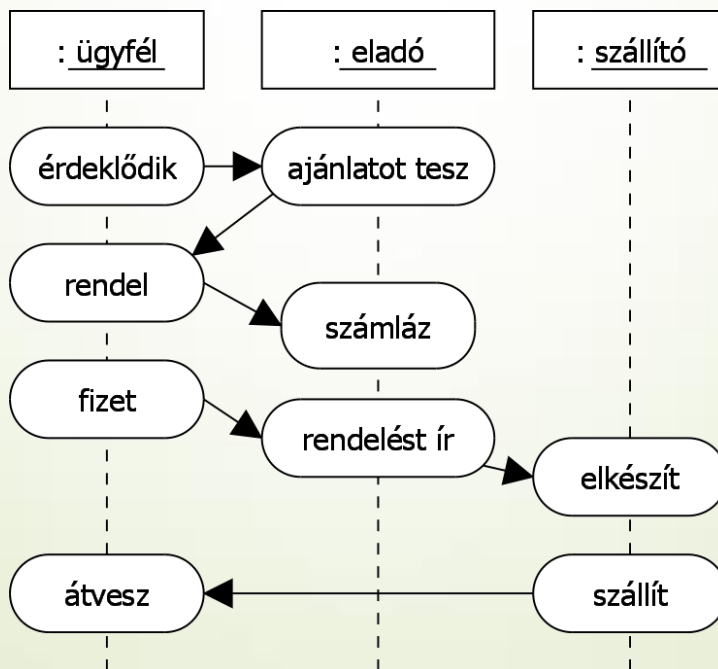
- ▶ a tanár a szorgalmi időszakban oktat
- ▶ a hallgató ekkor és a vizsgát megelőzően tanul
- ▶ a vizsgán egyszerre vizsgáztat a tanár és vizsgázik a hallgató
- ▶ a vizsga végén, amikor a tanár is befejezte a vizsgáztatást és a hallgató is a vizsgázást, a vizsgáztatókból álló bizottság osztályoz



Példa – életfolyam alapú tevékenység diagram 2.

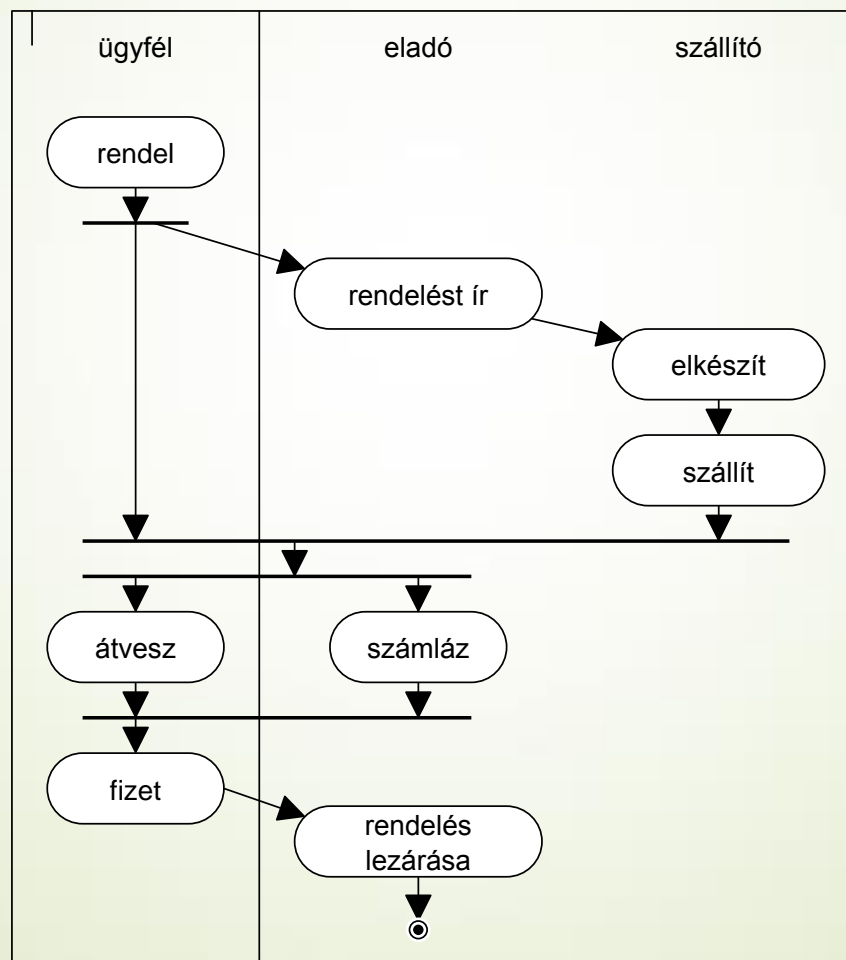
Adjuk meg az árurendelés és a házhozszállítással összekötött áruvásárlás tevékenység diagramját.

- Az ügyfél először érdeklődik, és az eladó ajánlatot tesz.
- Ha az ügyfél elfogadja az ajánlatot, akkor megrendeli az árut, az eladó elkészíti a számlát, az ügyfél pedig kifizeti.
- Fizetés után a szállító házhoz viszi az árut.



Példa – sávos alapú tevékenység diagram

Adjuk meg az előbbi árurendelés feladat sávos alapú tevékenység diagramját.

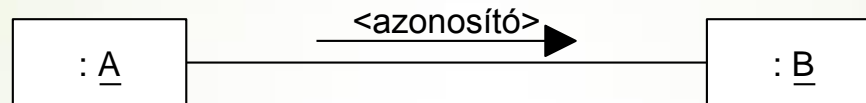


Együttműködési diagram

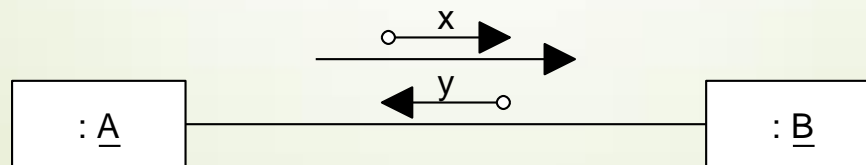
- Az együttműködési (kollaborációs) diagram azt mutatja be, hogy miként működnek együtt az osztályok objektumai a probléma megoldásában, milyen üzenetek cseréje révén valósul meg ez az együttműködés.
- Az együttműködés az asszociációs kapcsolattal összekötött objektumok között valósul meg.
- A diagram mutatja az összekapcsolást és az ehhez tartozó üzenetváltásokat, ezért az együttműködési diagram az objektumdiagram egyfajta kiterjesztének is tekinthető.

Együttműködési diagram (folyt.)

- Az üzenet küldését egy nyíl mutatja, amely a társítás mellett kap helyet, és a címzett irányába mutat. Az üzenet azonosítója a nyíl mentén helyezkedik el.

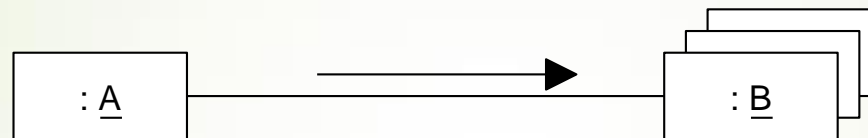


- Az üzenetnek lehet argumentuma és eredménye. Ezeket egy kis körből induló nyíl mellett helyezzük el, ahol a nyíl az információ áramlásának irányát jelzi.

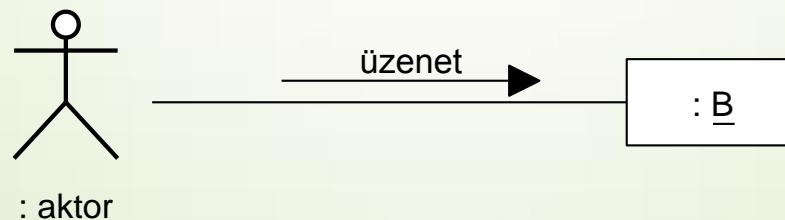


Együtműködési diagram (folyt.)

- Az üzenetnek lehet egy osztályon belül több címzettje is. Ekkor ez három, egymáshoz képest elcsúsztatott téglalappal jelöljük.

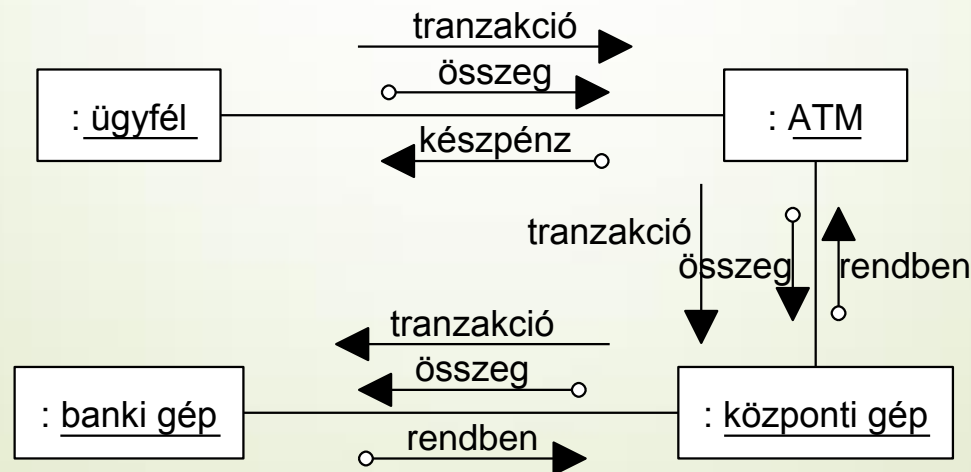


- Az együttműködésben részt vehet egy rendszeren kívüli objektum is, amely aktor szerepet játszik. Az első üzenet, amely az együttműködést elindítja, ettől az objektumtól származik.



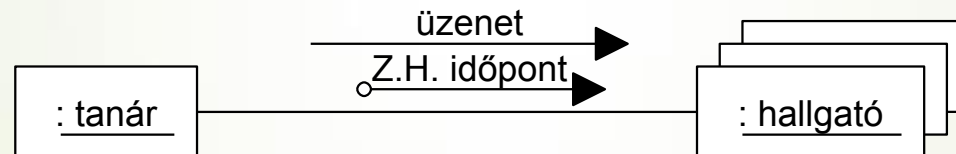
Együttműködési diagram – Példa

- Adjuk meg egy ATM készüléken folytatott pénzügyi tranzakció együttműködési diagramját.
- Közreműködő felek:
 ügyfél, ATM, központi számítógép, banki számítógép
- Az egyszerűség kedvéért csak a pénzfelvétel és az ahhoz kötődő adatok jogosságának ellenőrzésével, a kért összeg megnevezésének útjával és a készpénz kiadásával foglalkozunk.



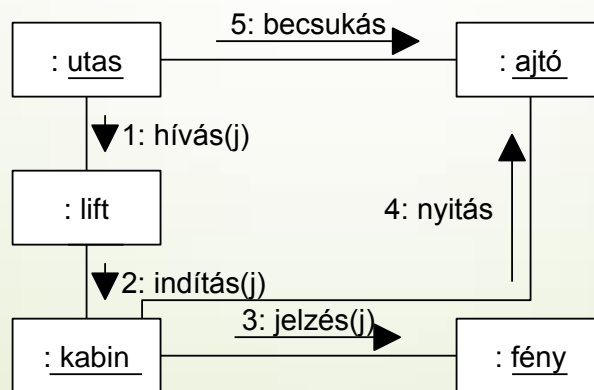
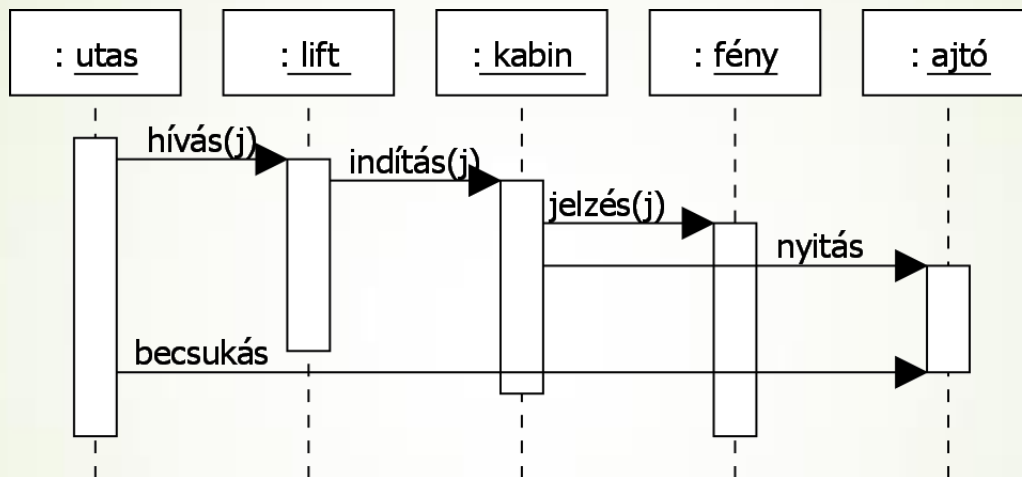
Együttműködési diagram – Példa

- Adjuk meg az együttműködési diagramját annak, amikor a tanár közli a hallgatókkal a zárthelyi időpontját.
- Az üzenetet jelen esetben több hallgatónak is el kell juttatni.



Együtműködési diagram – Példa

- Adjuk meg a lift működését leíró szekvencia és együttműködési diagramot.



Felhasználói esetek diagramja

- A felhasználói esetek diagramja a felhasználók szempontjából mutatja be, hogy a rendszer miként működik, függetlenül attól, hogy a szolgáltatásait miként valósítja meg.
- A rendszer által nyújtott szolgáltatásokat együtt mutatja be azokkal, akik ezekkel a szolgáltatásokkal kapcsolatba kerülnek.
- A diagram segítségével a felhasználók és a rendszer fejlesztői a rendszerrel szemben támasztott követelményeket azonos módon értelmezhetik.
- A diagram részei:
 - felhasználói esetek,
 - felhasználók,
 - felhasználási relációk.

Felhasználói esetek

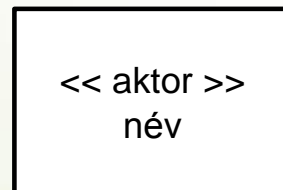
- A felhasználói esetek a rendszer funkcióinak összefoglalásai, szolgáltatási egységek.
- Ez az egység az akcióknak egy olyan sorozata, amelyekkel a rendszer a felhasználók egy csoportjával működik együtt.
- Egy információs rendszerben ilyen egység lehet a számlamozgásokról történő információ szolgáltatás az ügyfelek számára.
- A felhasználói eset jelölése



<funkció neve>

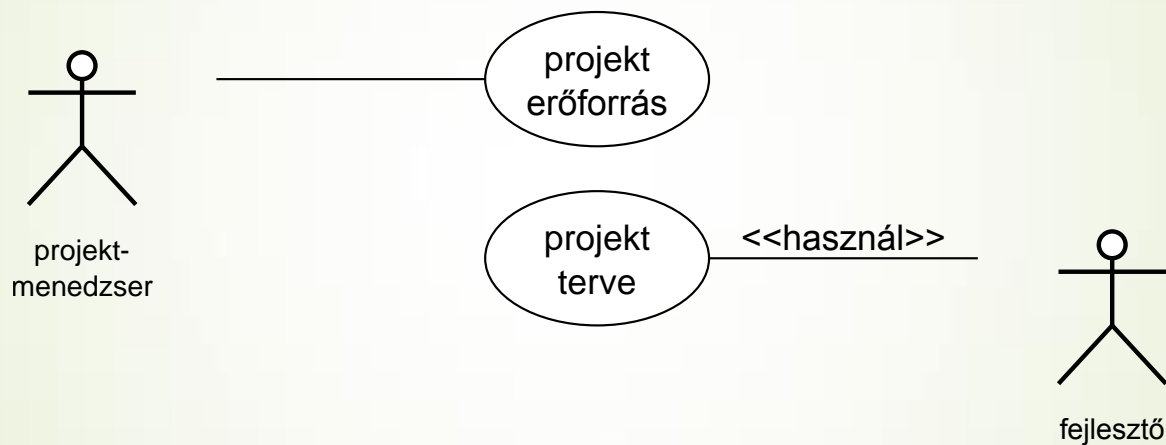
Felhasználók

- A felhasználók az adott rendszeren kívüli egységek, más programrendszerek, alrendszerek, osztályok, illetve személyek lehetnek.
- Ezek aktor szerepet töltenek be.
- A felhasználók jelölése:



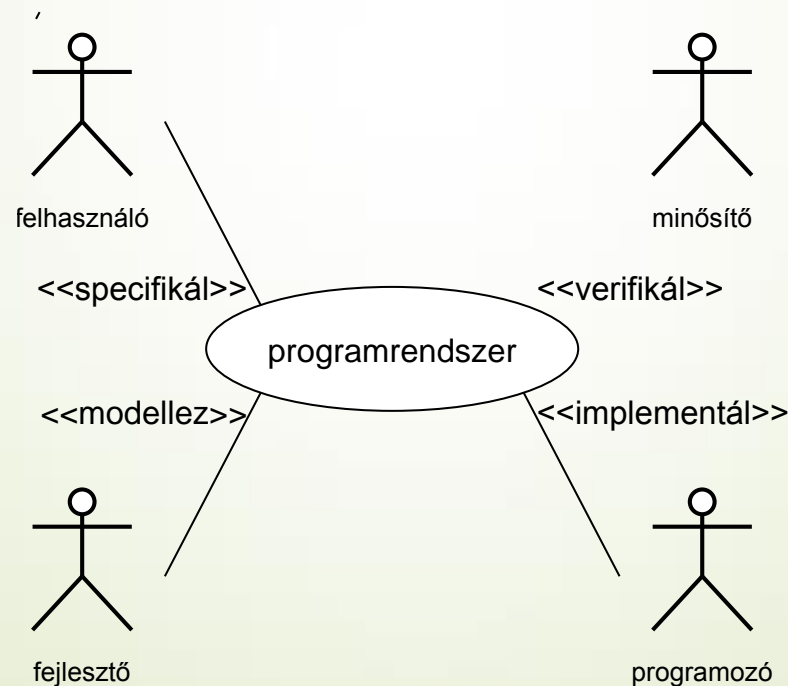
Felhasználási relációk

- A felhasználói relációkat a felhasználás módjának alapszava egészítheti ki.



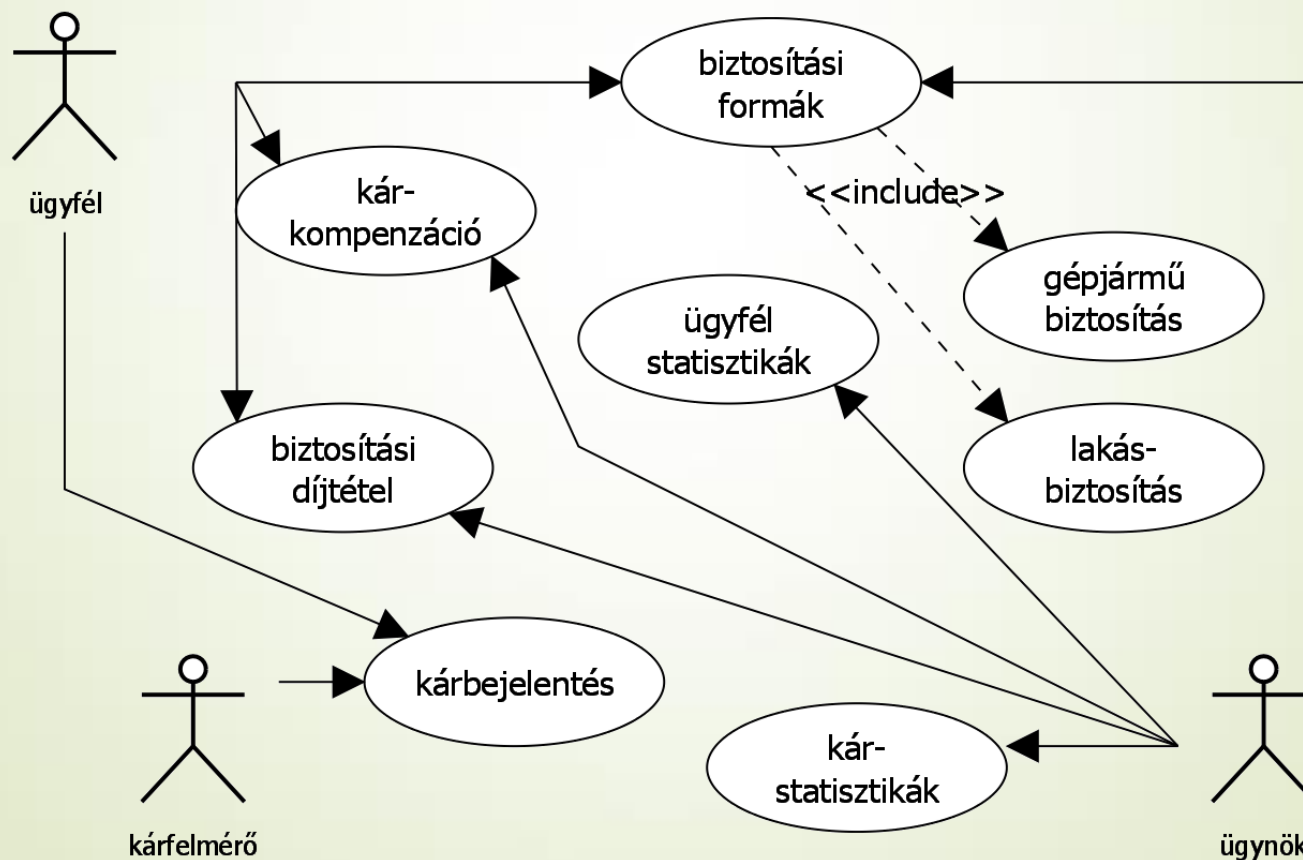
Felhasználói esetek diagramja

- A felhasználói esetek diagramja rendszerint szintekre tagolt az áttekinthetőség miatt.
- Például a programrendszer – mint felhasználói eset – és a projekt tagjainak kapcsolatát a fejlesztés során szemlélteti a legfelső szinten az alábbi diagram



Felhasználói esetek diagramja

- ▶ Adjuk meg egy biztosítási rendszer felhasználói esetek diagramját.
- ▶ Aktorok: ügyfél, ügynök, kárfelmérő.



Implementációs szempont szerinti diagramok

Komponens- és alrendszer diagram

- A rendszer elkészítésekor figyelembe kell vennünk az implementációra vonatkozó megkötéseket, korlátozásokat.
- Az UML a korlátozások leírására két diagramot ad:
 - komponens diagram,
 - alrendszer diagram (következő félévben)

Komponens diagram

Komponens diagram

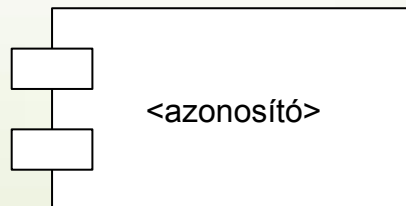
- A komponensdiagram a probléma megoldására szolgáló rendszer tulajdonságait implementációs szempont szerint fejezi ki.
- A diagram részei:
 - komponens
 - reláció

Komponens

- A *komponens* a rendszer egy fizikailag létező és kicserélhető része.
- A kicseréléshez az új komponens csatlakozási felületét a környezettel konform módon kell megvalósítunk.
- Új komponens használatához a csatlakozási felületét hozzá kell illeszteni a már meglévő rendszerünkhöz.

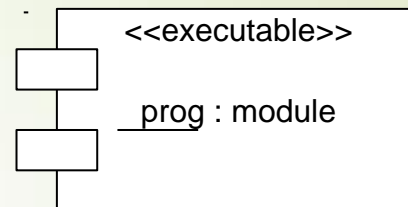
Komponens informális definíciója

1. A komponensnek van azonosítója.
2. A komponens az objektumok egy osztálya, amely a fizikai objektumokat definiálja.
3. A fizikai objektumok a szoftvermodulok fejlesztési vagy végrehajtási formáiban léteznek.
4. A fejlesztési forma lehet szöveges forma, a fordítás fázisaiban létező kódforma valamelyike.
5. A végrehajtási forma a modul végrehajtásra kész formája.
6. A komponenshez tartozhat egy sztereotípus, amelyeknek jelentése rögzített, és a létezési formát fejezi ki.
Például: <<dokumentum>>, <<forrás>> stb.



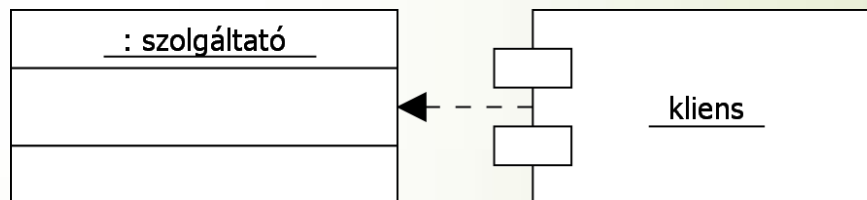
Komponens diagram – Jelölések

- Végrehajtásra kész formában megjelenő `prog` modulnak megfelelő komponens:



- A komponensek közötti relációkat két csoportba sorolhatjuk:

- fejlesztés során fennálló



(A szolgáltatást nyújtó komponens egy osztály, ami csak a fejlesztés során létezik.)

- meghívási reláció

