



---

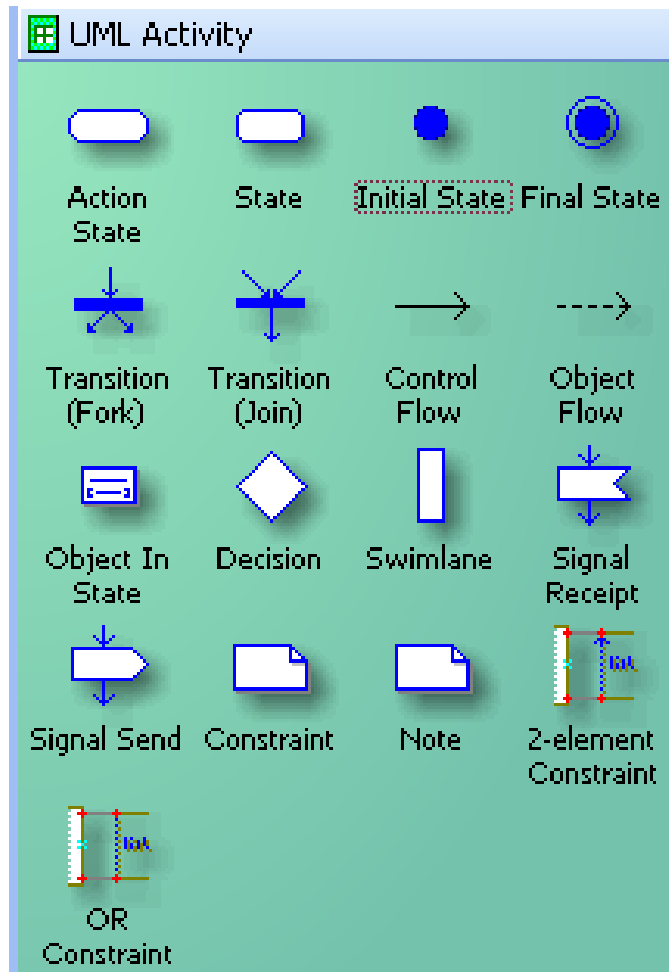
# UML Tevékenység diagram (Activity Diagrams)

# Szervezeti folyamatok modellezése (Folyamatmenedzsment)



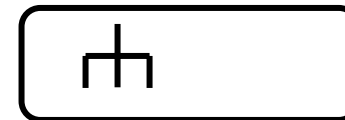
Az 1970-es években Gane és Sarson fejlesztette ki az adatfolyam modell diagramot, a szervezeti („üzleti”) folyamatok modellezésére, a legfontosabb tevékenységek végrehajtásának lépéseinek és a hozzájuk kapcsolódó adatoknak a leírására. UML-ben nincs adatfolyam diagram, de ugyanazt a célt szolgálja az UML tevékenység diagram.

# Tevékenység diagram jelölés technikája

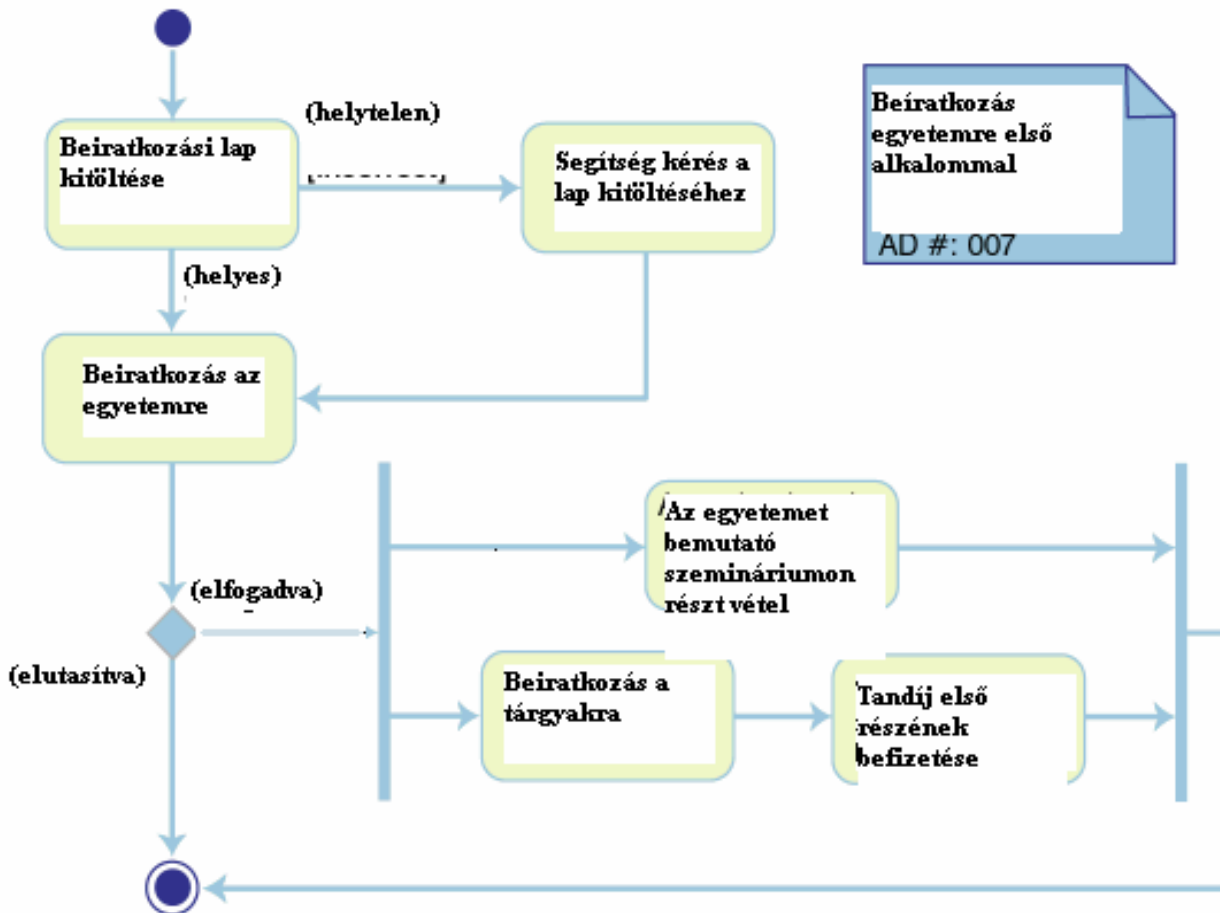
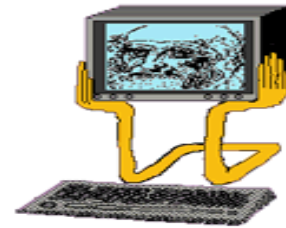


Az UML tevékenység (Activity Diagram) jelölésrendszere; MS Visio-ban.

A gereblye (Rake symbol) nincs a Visio-ban, amely egy folyamat folytatódását jelzi egy másik diagramon.



# Példa: UML tevékenység diagram (UML tevékenység diagram első alkalommal egyetemre beiratkozásra)



Forrás: Scott Ambler,  
<http://www-106.ibm.com/developerworks/webservices/library/ws-tip-drawuml/>

# Lépések



## 1. A tevékenység diagram terjedelmének meghatározása

A modellezés tárgyának meghatározása. Csak egyetlen egy használati esetet? Használati eset egy részét? Olyan szervezeti folyamatot, amely több használati esetet érint? Egy osztály egyetlen egy metódusát? Ha a diagram kiterjedését meghatározták, akkor a diagram felső részét egy címkével kell ellátni, amely a diagram címét tartalmazza, valamint egy egyértelmű azonosítót. A készítő neve és a készítés dátuma is megadható.

## 2. A kezdeti és végpont hozzáadása

Az összes tevékenység diagramnak van kezdő és végpontja, ezért ezeket a kezdet kezdetén azonnal lehet ábrázolni. Gyakran egy tevékenység egyszerűen egy zsákutca, ebben az esetben sem okoz igazán gondot, ha a tevékenységet összekötjük a végponttal, jelölve azt, hogy ennek a tevékenységnek egyetlen állapot átmenete a végpontba vezet. Ezen a módon, bárki is olvassa a diagramot, tudni fogja, hogy az elemző mérlegelte azt, hogy ez a tevékenység, hogyan fog végződni.

## 3. Bővítés tevékenységekkel

Ha egy használati esetet modellezünk, akkor egy szereplő által kezdeményezett mindegyik lényeges lépésre hozzunk létre egy **tevékenységet** (egy ilyen *tevékenység* tartalmazhatja az indító lépést, valamint a rendszer összes további reakcióit erre az indító lépésre.) Ha egy magas szintű szervezeti folyamatot modelleznek, akkor minden jelentősebb folyamatra létre kell hozni egy *tevékenységet*, ami legtöbbször egy használati eset vagy használati esetek egy halmaza. Ha egy objektum metódust modellezünk, akkor általában a megírt vagy megírandó program kód minden önálló lépésére létre kell hozni egy *tevékenységet*.

# Lépések



## 4. A tevékenységeket egészítsük ki átmenetekkel

Egy tevékenységből mindig célszerű a kilépést jelölni, még abban az esetben is, ha az a végponthoz vezet. Ha több kilépési pont van, mindegyiket el kell látni címkével.

## 5. Kiegészítés döntési pontokkal

A modellezés tárgyának logikája gyakran döntési pontok leírásához vezet. Fontos megjegyezni, hogy a döntési pontok használata opcionális. Az [ábrán](#) az „elfogadott” és az „elutasított” tevékenységek közötti állapotátmenetet közvetlenül is lehetett volna ábrázolni, a „Beiratkozás az egyetemre” tevékenységből.

## 6. Azonosítsd a párhuzamos tevékenységekre a lehetőséget

Két tevékenység akkor folyhat párhuzamosan, ha nincs közöttük közvetlen kapcsolat, de mindkét tevékenységnek be kell fejeződnie, mielőtt egy harmadik tevékenység elkezdődne. Az [ábrán](#) az egyetemet bemutató szemináriumon a részt vétel, vagy a tárgyakra a beiratkozás/felvétel bármilyen sorrendben történhet, de mindkét tevékenységnek le kell folynia mielőtt a teljes folyamat lezárulna.