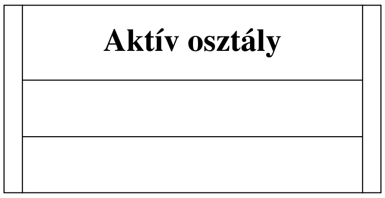
# Osztálydiagram

Egy osztály lehet *aktív* osztály. Ennek objektumai birtokolnak egy vagy több folyamatot, illetve szálat, így képesek vezérlési tevékenység kezdeményezésére. Jelölése:




Ha egy osztálydiagramban nem szerepel aktív osztály, akkor az aktivitásról nem tehetünk fel semmit (nem kötelező megadni). Ellenkező esetben csak a jelölt osztályok lehetnek aktívak.

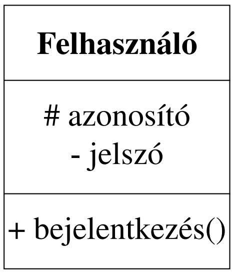
## Láthatóság

Az osztálydiagramban megadhatjuk az attribútumok és a műveletek láthatóságát, elérhetőségét a következő jelekkel:

**+** public,

**#** protected,

**-** private.




## Interfészek

Az UML-ben *interfésznek* (interface) nevezzük egy elem viselkedését jellemző műveletek névvel ellátott csoportját.

Egy interfész egy osztály (komponens) látható műveleteinek leírása, a belső szerkezet megadása nélkül. Egy interfész gyakran egy tényleges osztály viselkedésének csak egy korlátozott részét írja le. Egy osztály több (diszjunkt vagy átfedő) interfészt is támogathat. Az interfészekhez nem tartozik implementáció, nincsenek attribútumaik, állapotaik, csak műveletekkel rendelkeznek. Az interfészek között öröklődési kapcsolat fennállhat. Ekkor a származtatott interfész átveszi az ős interfész összes műveletét, és azokat újakkal egészítheti ki. Egy interfész összes művelete publikus.

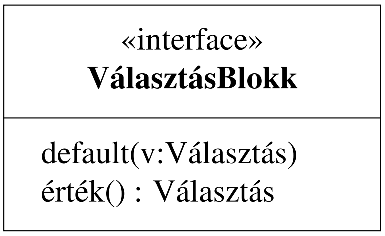
Az interfész lényegében megegyezik egy olyan absztrakt osztállyal, amelyhez nem tartoznak attribútomok és megvalósított műveletek.

Ha egy osztály megvalósít (implementál) egy interfészt, akkor az interfész összes műveletét deklarálnia kell, illetve meg kell valósítania. Ha több interészt valósít meg, akkor az összes interfész műveletére ennek kell teljesülnie. Ha ugyanaz a művelet több interfészben is szerepel, akkor a jelentésüknek meg kell egyezniük, különben konfliktust okoznak, illetve a modell helytelen.

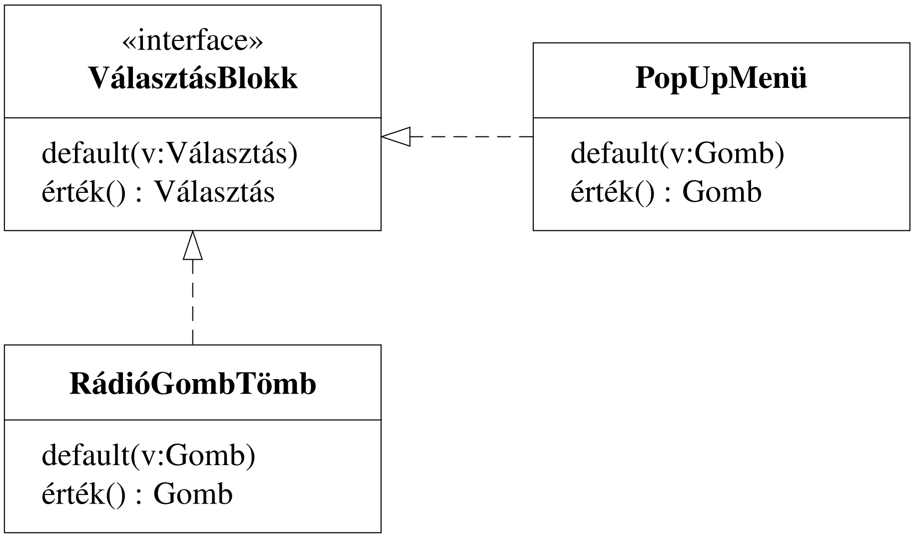
Interfészek csak úgy vehetnek részt asszociációs kapcsolatban, ha a navigálhatóság az interfész felé mutat.

Jelölés:

Az osztályhoz hasonlóan téglalappal az «`interface`» kulcsszó feltüntetésével.

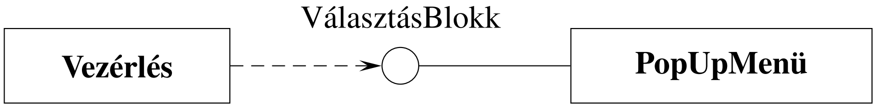



A megvalósítás jelölése:



A megvalósítás egyszerűsített jelölése:


Interfész használata:




# Állapotdiagram

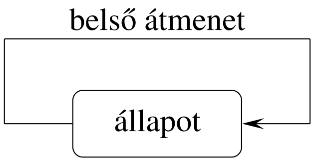
## Az állapot informális definíciója

Az állapot a következő tulajdonságokkal rendelkezik:

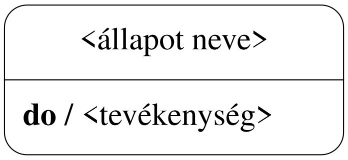
1. Az állapotnak van azonosítója. Az állapotok egymástól megkülönböztethetők, például van nevük. A verem esetében: „normál”, „tele”, ... Állapotot azonosíthat attribútumainak konkrét értéke, illetve ezen értékeket meghatározó állítás, állapotinvariáns. Az állapot jelölése egy lekerekített sarkú téglalap, amelybe az azonosító kerül.

2. Az állapot általában esemény, eseménysorozat hatására jön létre. Ezeket az eseményeket megelőző eseményeknek (pre-events) nevezzük. (Pl. a verem normál állapota a „create”, „push” események eredményeként jön létre először; az iskolás állapot pedig a beiratkozás hatására.) Egy speciális, rendszeren kívüli állapot a kezdeti állapot.

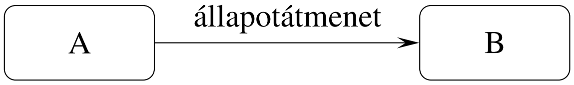
3. Az állapot időben mindaddig fennmarad, amíg az objektumok attribútumainak értékei kielégítik az állapothoz rendelt invariánst. Az állapot fennállása során belső átmenetek fordulhatnak elő. Ezek nem változtatják meg az állapotinvariáns:




4. Az állapotokban, annak fennállása során, egy objektum tevékenységet hajthat végre. Ha ez áll fenn ezt a következő módon jelölhetjük:



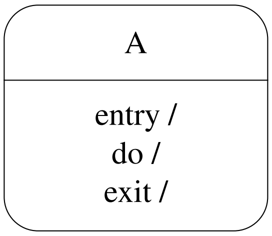

5. Az állapot megszűnése egy esemény hatására következik be. A megszűnéshez egy eseménysorozat kötődhet (rákövetkező események, post-events). Azaz az objektum egy külső állapotátmenet hatására egy másik állapotba kerül. Jele:



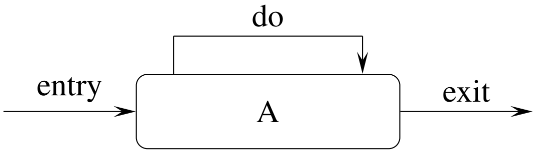

6. Az objektum megszűnése ugyancsak egy állapotátmenet hatására következik be. Ekkor az objektum egy rendszeren kívüli, úgynevezett befejező állapotba kerül.

Egy állapotnak általában három fázisa van:

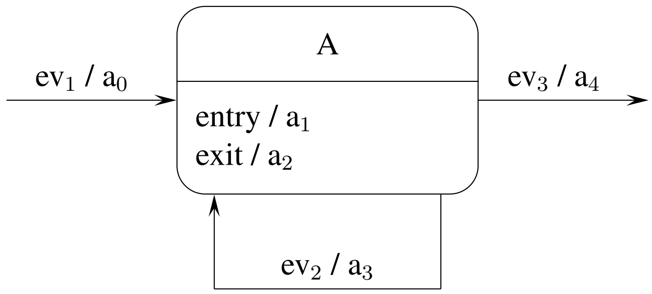
- Egy *entry fázis*, a belépés akciója, amely az állapotba lépéskor hajtódik végre.
- Egy *event fázis*, amely az adott állapothoz kötődik, az állapotban végrehajtott tevékenység(ek)et adja meg.
- Egy *exit fázis*, a kilépés akciója, amely az állapot megszűnésekor, az állapotból való kilépéskor kerül végrehajtásra.




Szemléletesen:

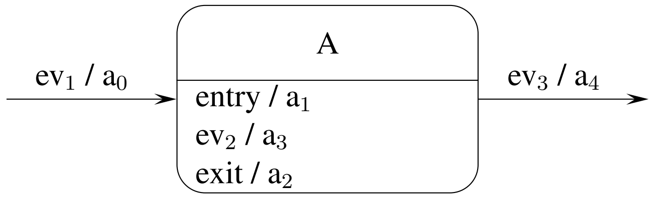


Példa:




$ev_1 : a_0, a_1$
$ev_2 : a_2, a_3, a_1$
$ev_3 : a_2, a_4$

Ha $ev_2$ belső átmenet, akkor ez helytelen, ezért a jelölés:




## Állapotok általánosítása

Az általánosított állapot a következő tulajdonságokkal rendelkezik:

1. Az általánosított állapot véges számú részállapot összessége.
2. A részállapotok örökölhetik az általánosított állapot tulajdonságait:
    a. attribútumait (állapotjellemzőit);
    b. eseményeit, akcióit.
3. Az általánosított állapot állapotinvariánsa a részállapotok állapotinvariánsainak diszjunkciója, azaz az objektum az általánosított állapotban mindig valamelyik részállapotban van.
4. A részállapotok lehetnek általánosított állapotok is.
5. Az általánosított állapothoz állapotdiagram tartozik, amelyben a részállapotok közötti átmeneteket tüntetjük fel.

6. Az általánosított állapot állapotdiagramjában a részállapotokhoz azok az állapotátmenetek tartoznak, amelyek az általánosított állapotot nem változtatják meg.

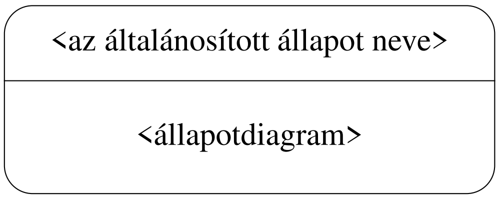
7. Az állapotdiagramban kell lennie legalább egy olyan részállapotnak, amely az általánosított állapot belépési eseményét örökli. Az állapotdiagramban a kezdő állapotból a megfelelő részállapotba vezet él.

8. Ha az általánosított állapothoz kilépési esemény tartozik, akkor az összes részállapota átveszi ezt a kilépési eseményt, azaz az esemény bekövetkezésekor elhagyjuk előbb az aktuális részállapotot, majd az általánosított állapotot. Ennek megfelelően, ha az általánosított állapothoz kilépési akció tartozik, akkor ezt az összes részállapot örökli.

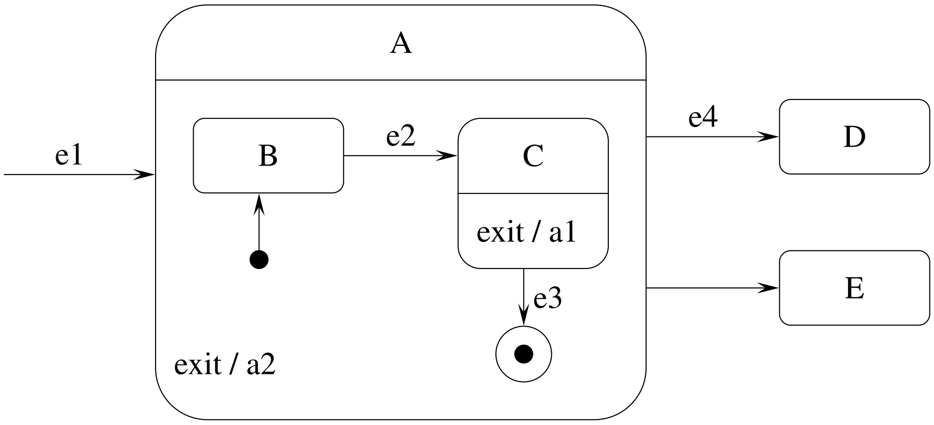
9. Ha az általánosított állapot rendelkezik befejező állapottal és egy esemény hatására ebbe kerülne, akkor az állapotot elhagyjuk.

10. Ha az általánosított állapothoz objektum tartozik, akkor az a belépés hatására létrejön, kilépés hatására pedig megsemmisül.

Az általánosított állapot jelölése:



Példa:




| események | állapotok | akciók |
|---|---|---|
| e1, e2, e3 | B, C, E | a1, a2 |
| e1, e2, e4 | B, C, D | a1, a2 |
| e1, e4 | B, D | a2 |

## Állapotok aggregációja

Az aggregációval előállított állapot a következő tulajdonságokkal rendelkezik:

1. Az aggregációs állapot egymástól független részállapotok egy véges halmaza.

2. Minden részállapothoz állapotdiagram tartozik.

3. A részállapotok lehetnek általánosított állapotok is.

4. Az aggregációval létrejött állapot invariánsa a részállapotok invariánsainak konjunkciója, azaz az aggregációs állapot objektuma az aggregációt alkotó részállapotokban egyidejűleg létezik.

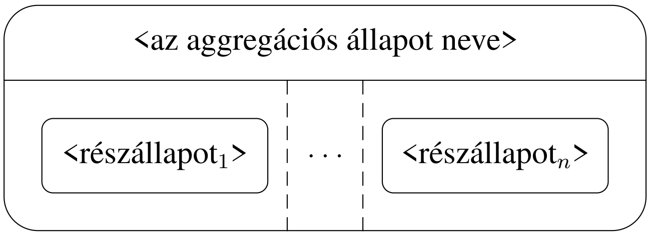
5. Minden részállapothoz tartozó állapotdiagramban kell lennie legalább egy olyan állapotnak, amelybe a részállapot az aggregátum belépési eseményének hatására kerül. A megfelelő állapotdiagramban a kezdő állapotból ilyen állapotba vezet él.

6. Ha az aggregátum rendelkezik kilépési eseménnyel, akkor minden részállapot örökli ezt a kilépési eseményt, azaz az esemény bekövetkezésekor minden részállapotot elhagyunk, majd az aggregációs állapotot is elhagyjuk. Ennek megfelelően, ha az aggregációs állapothoz kilépési akció tartozik, akkor ezt az összes részállapot örökli.

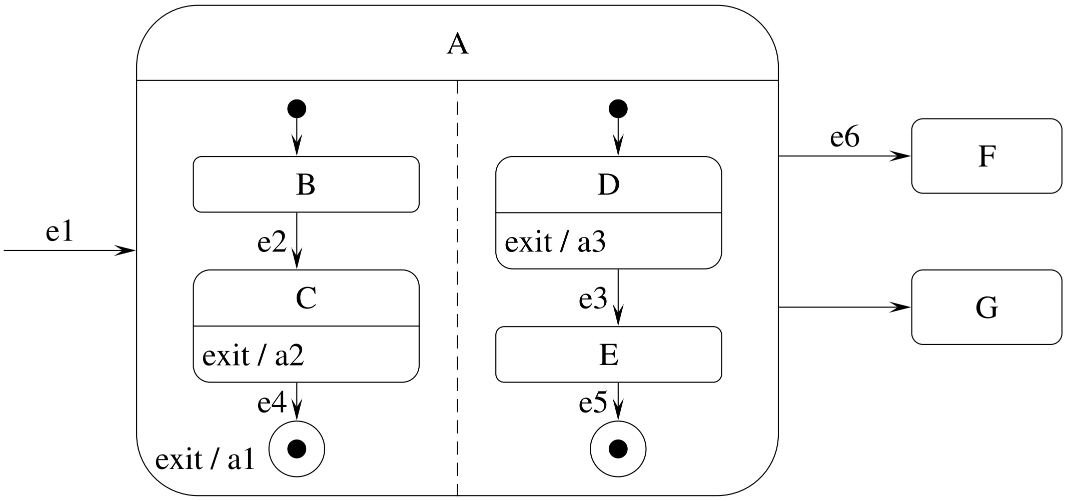
7. Ha az összes részállapot rendelkezik befejező állapottal és mindegyik ebbe kerül, akkor az aggregációs állapotot elhagyjuk. (A befejező állapotba került részállapotok abban várakoznak, amíg az aggregációs állapotot el nem hagyjuk.)

8. Az állapotok aggregációja az állapoton belüli állapotdiagramok közötti párhuzamosság egy megjelenési formája.

Az állapotaggregáció jelölése:



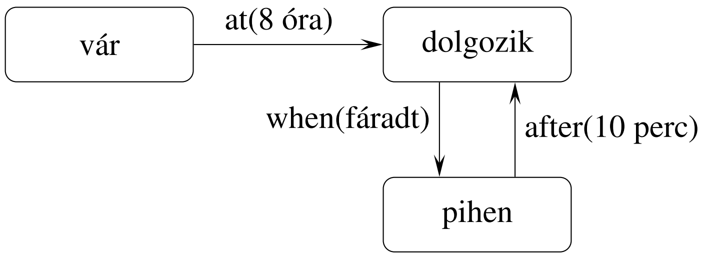

Példa:




| események | állapotok | akciók |
|---|---|---|
| e1, e2, e3, e6 | (B,D), (C,D), (C,E), F | a3, a2, a1 |
| e1, e6 | (B,D), F | a3, a1 |
| e1, e3, e6 | (B,D), (B,E), F | a3, a1 |
| e1, e2, e3, e4, e6 | (B,D), (C,D), (C,E), (.,E), F | a3, a2, a1 |
| e1, e2, e3, e5, e4 | (B,D), (C,D), (C,E), (C,.), G | a3, a2, a1 |

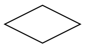
## Speciális állapot-átmenetek

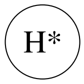
- Adott $t$ idő letelte utáni állapot elhagyás: *after(t)*.
- Állapot elhagyása adott időpillanatban: *at(t)*.
- Változás bekövetkezése (feltétel igazzá válása): *when(felt)*.




## További jelölések

 szétválasztás, összefuttatás

 választás



mély hisztorizációs indikátor