

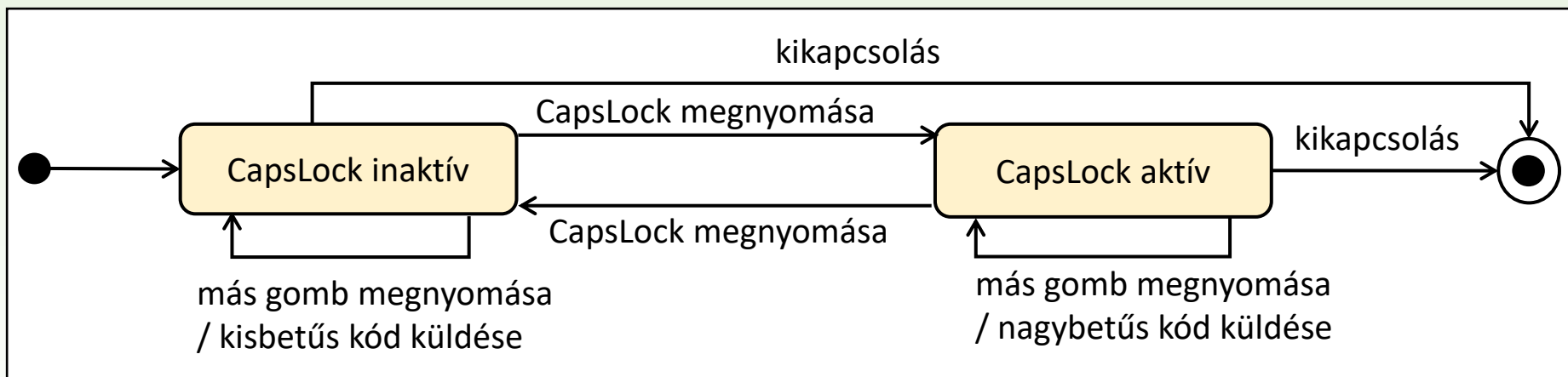
# Billentyűzet

Egyszerűsített billentyűzet modellezése:

ha a CapsLock aktivált, akkor minden más billentyű lenyomásra nagybetűs karaktereket, különben kisbetűs karaktereket kapunk.

Legyen két állapot: CapsLock aktív, illetve inaktív.

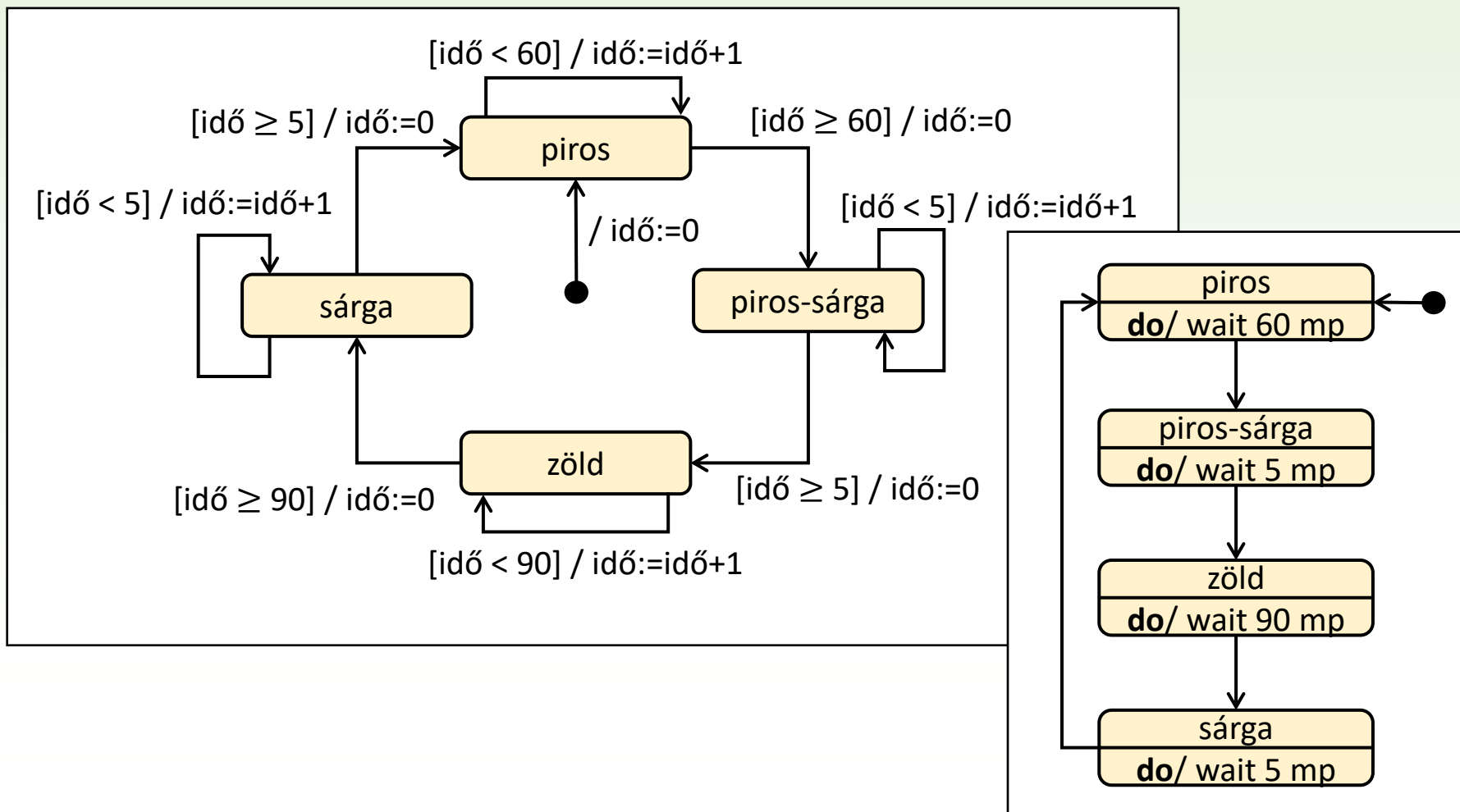
Legyen háromféle művelet: CapsLock lenyomása, más gomb lenyomása, kikapcsolás.



	inaktív	aktív
CapsLock	aktív	inaktív
más nyomógomb	inaktív / kisbetűs kód küldése	aktív / nagybetűs kód küldése
kikapcsolás	exit	exit

# Közlekedési lámpa

Egy közlekedési lámpán piros, piros-sárga, zöld, sárga fények vannak. A lámpa 60 másodpercig piros és 90 másodpercig zöld színű. Az átmeneti állapotok 5 másodpercig tartanak: pirosról a zöldre a piros-sárgán keresztül, zöldről a pirosra a sárgán keresztül. Kezdetben a lámpa piros.

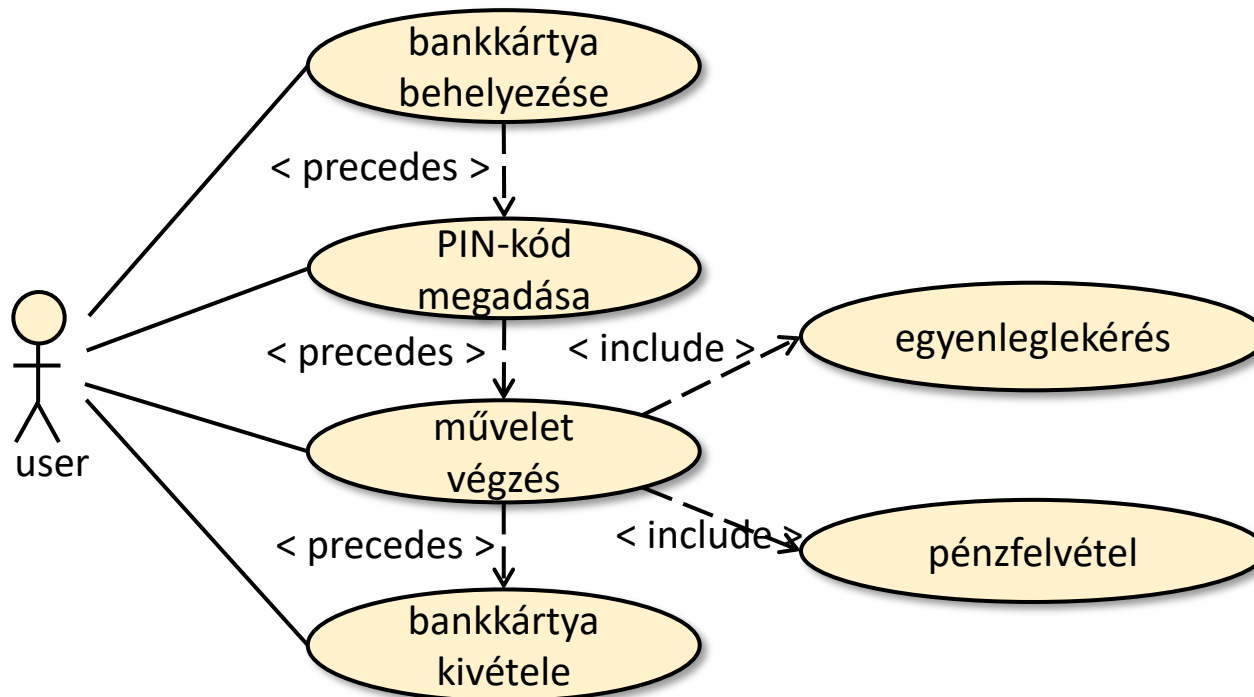


# ATM

Egy bankautomata a következő módon működik. Azzal indul, hogy az ügyfél behelyezi a kártyáját, majd beviszi a pin kódot, amivel háromszor próbálkozhat (harmadik sikertelen kísérlet után a tranzakció elutasítva).

Ha sikeres a pin kód megadása, akkor le lehet kérdezni az egyenleget, vagy ki lehet venni pénzt az automatából.

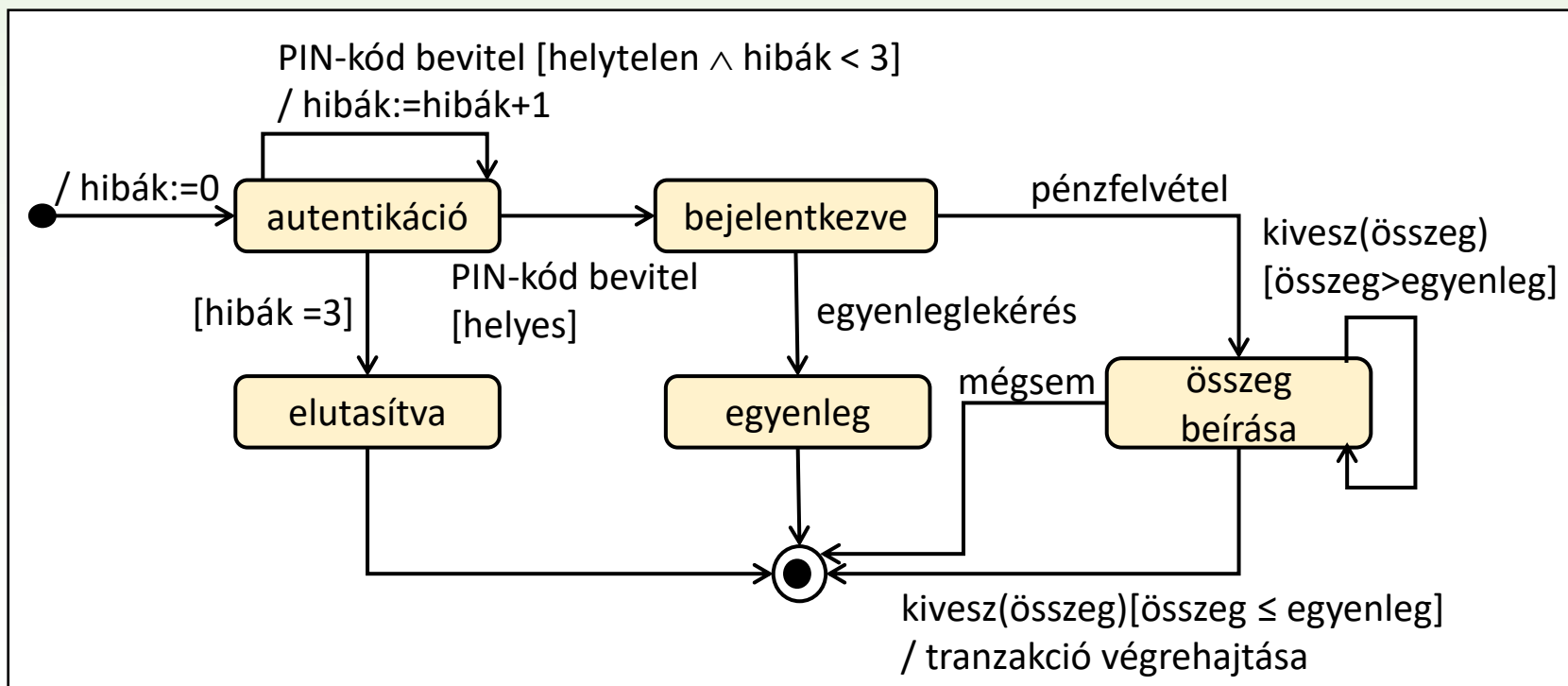
Ha a megadott összeg kisebb vagy egyenlő, mint az egyenleg, akkor sikeres a pénz kivét, különben nem.



# ATM

Egy bankautomata működése során

- az ügyfél behelyezi a kártyáját, majd
- beviszi a pin kódot, amivel háromszor próbálkozhat (harmadik sikertelen kísérlet után a tranzakció elutasítva);
- sikeres pin kód megadás esetén le lehet kérdezni az egyenleget, vagy
- ki lehet venni pénzt az automatából. Ha a megadott összeg kisebb vagy egyenlő, mint az egyenleg, akkor sikeres a pénz kivét, különben nem.



# Videomagnó

Készítsük el egy videomagnó osztálydiagramját és állapotgépét!

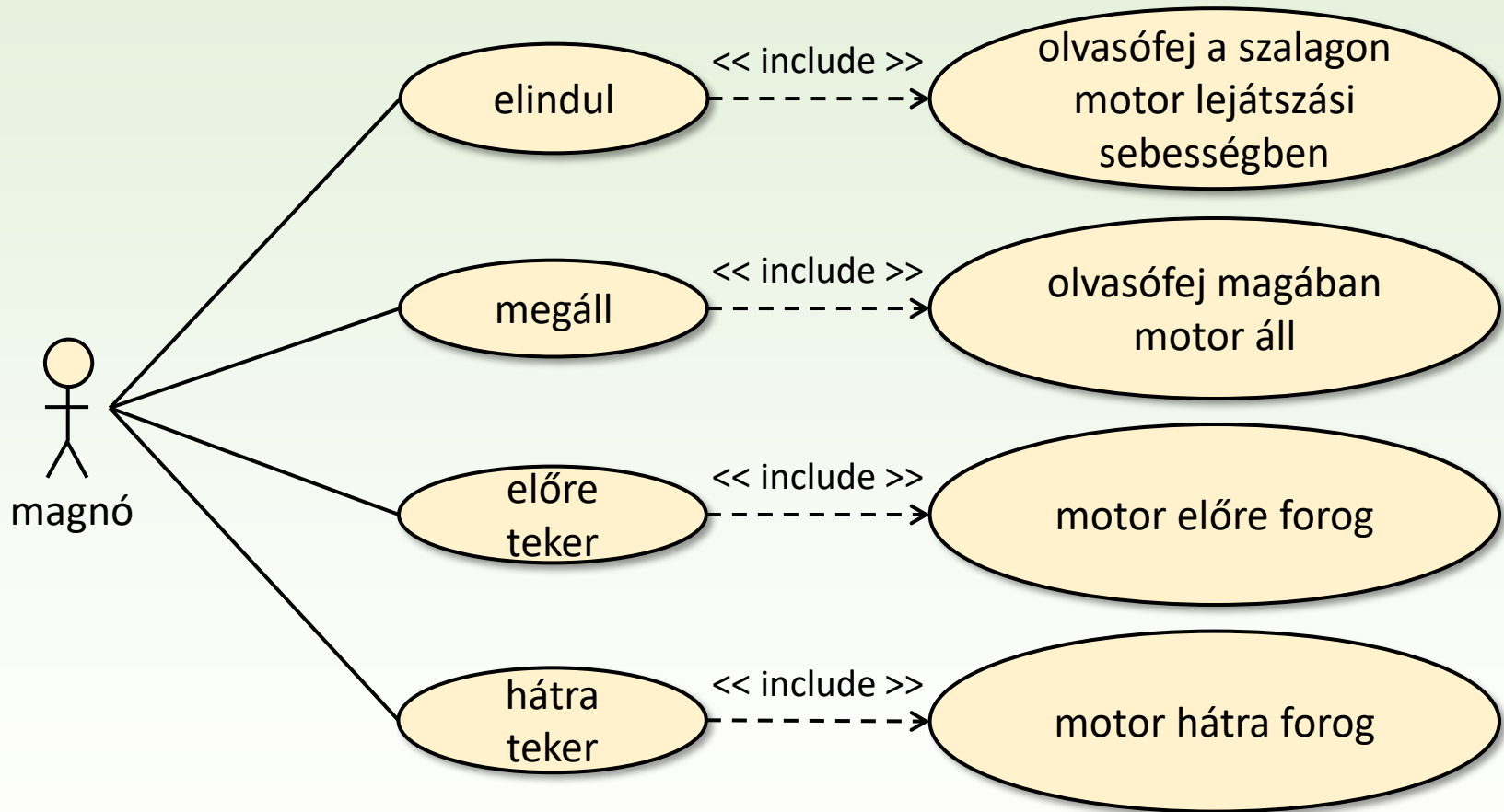
A magnóban található egy olvasó fej és egy motor, amelyeket négy gomb segítségével vezérelhetünk. A gombokat elegendő megérinteni a vezérlés során.

A négy gomb és vezérlési szerepük a következő:

- ▶ (lejátszás) : lejátszó sebességbe helyezi a motort, és a fejet a szalagra helyezi, ha még nincs ott
- ■ (állj) : leállítja a motort, és a fejet felemeli a szalagról, ha azon volt
- ▶▶ (előre) : a motor előre csévéli a szalagot
- ◀◀ (hátra) : a motor hátra csévéli a szalagot

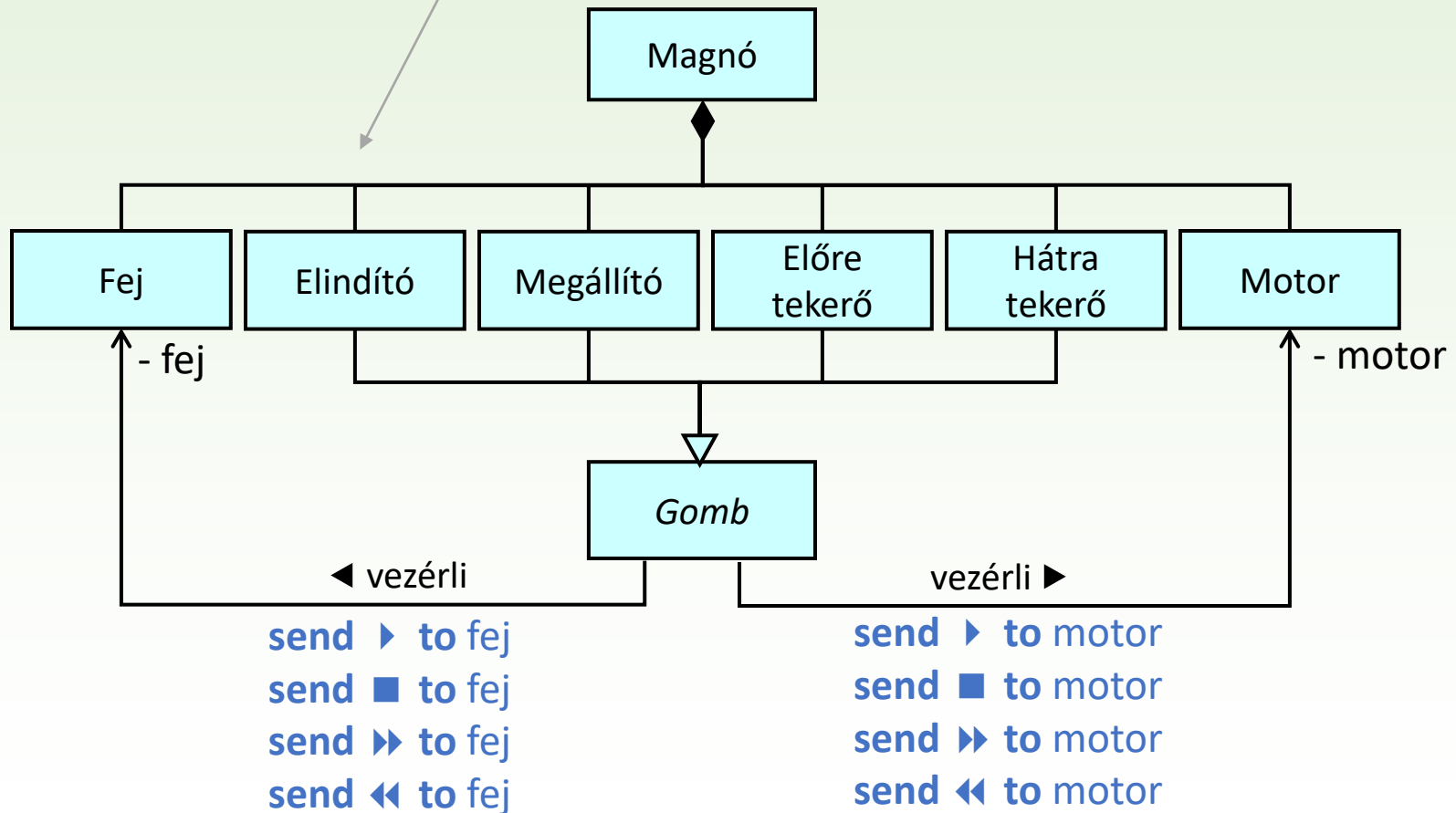
Előre, illetve hátracsévélés alatt a fej rajta maradhat a szalagon (gyorskeresés)

# Videomagnó

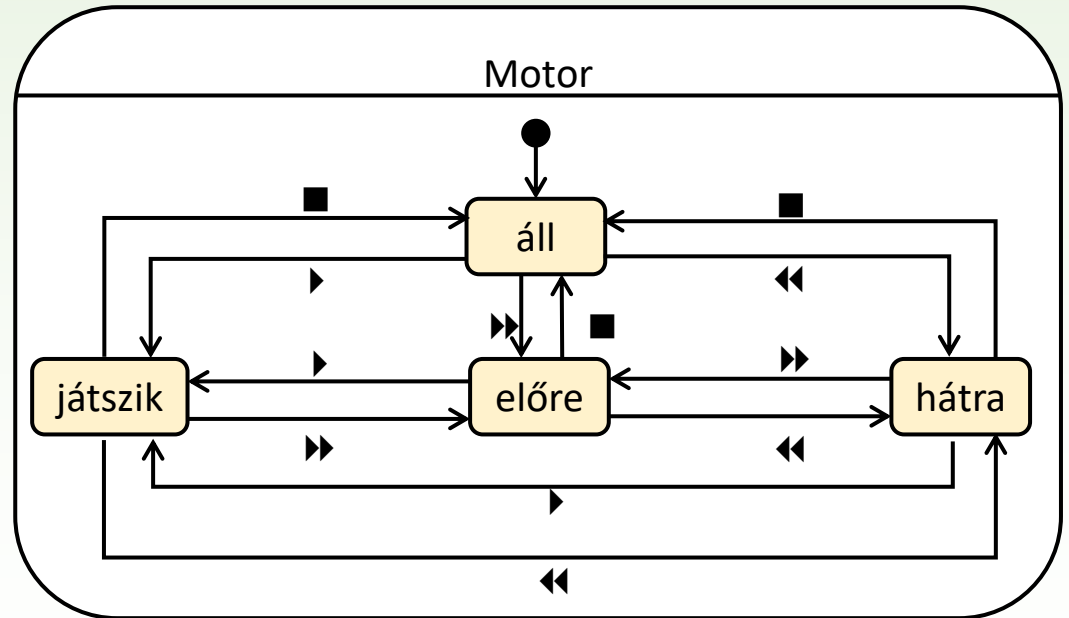
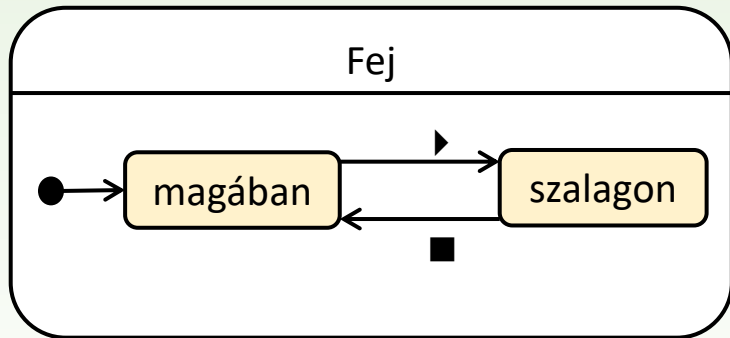
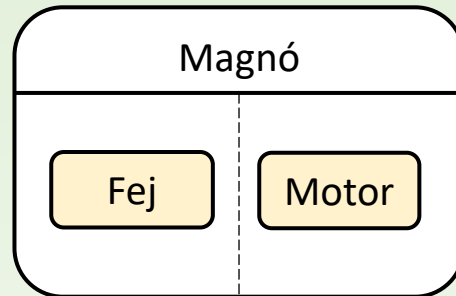


# Videomagnó

Egy kompozíció komponensei közötti navigálás biztosítható:  
1. közvetlen kapcsolatok mentén (fej, motor szerepnevek)  
2. a kompozíció szerepneveinek bevezetésével, amelyek láthatóságát biztosítani kell (getter).



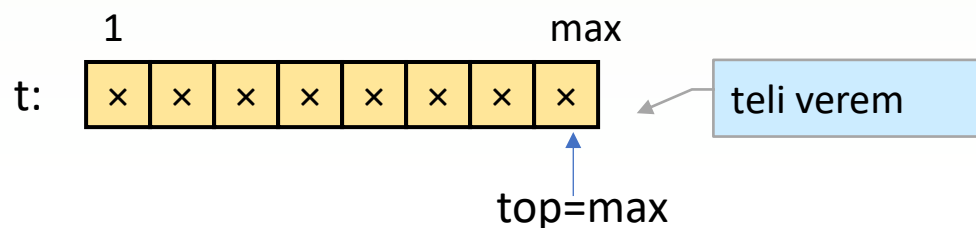
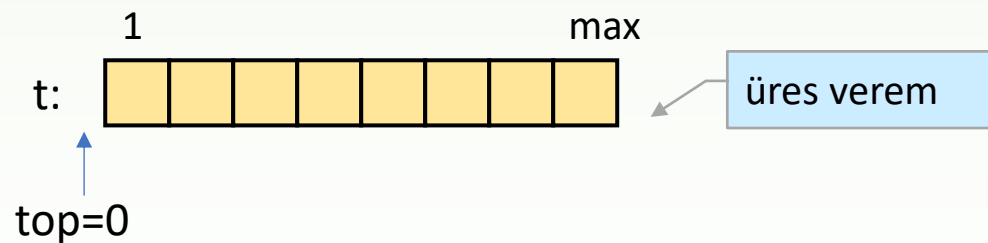
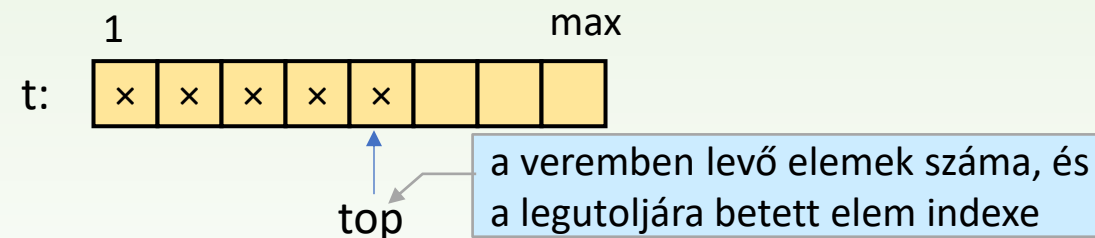
# Videomagnó





# Korlátos verem

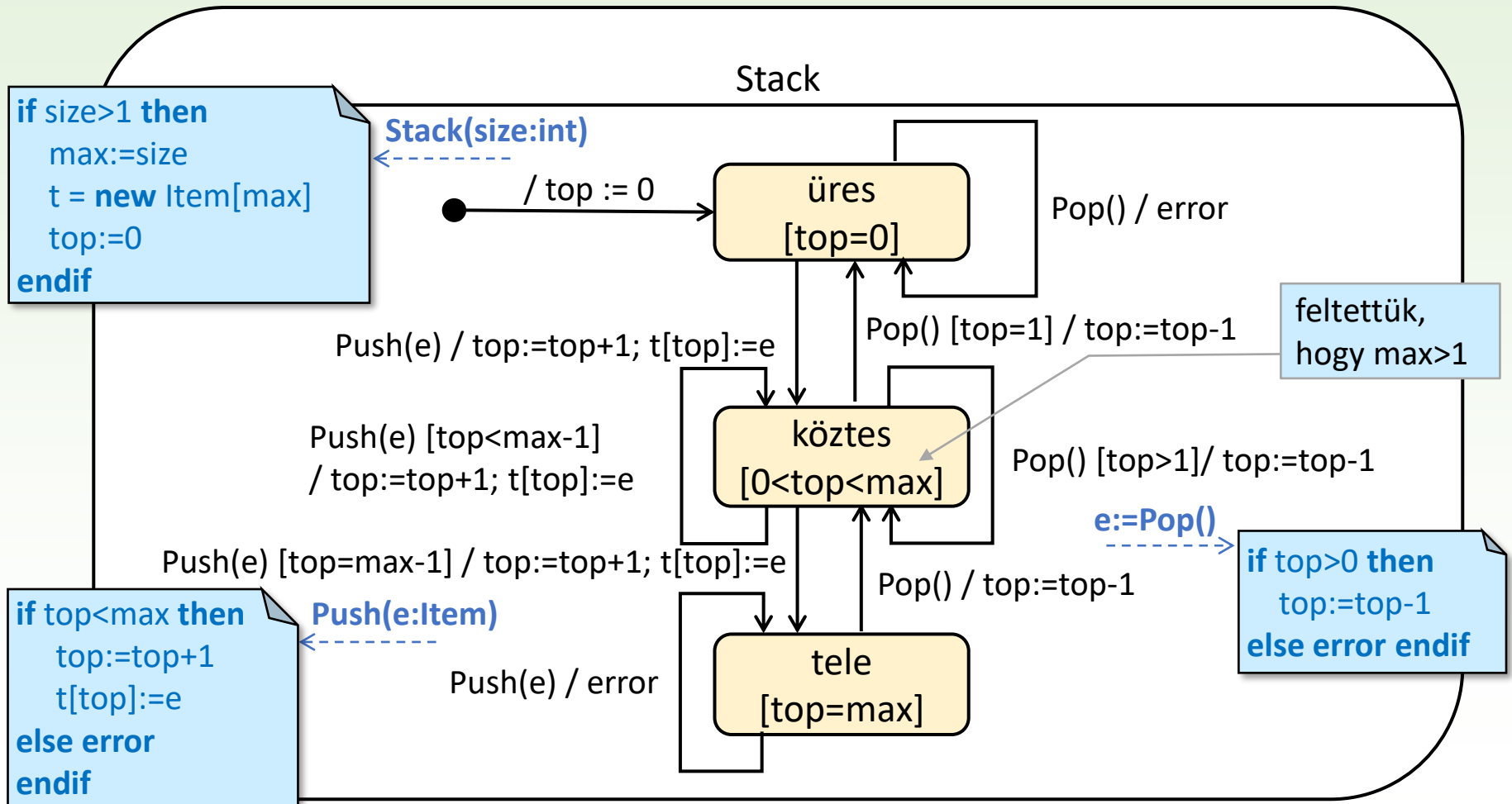
- ❑ Tervezzünk egy tárolót, amely veremként (LIFO) működik.
- ❑ Az előre nem meghatározott típusú elemeket egy rögzített méretű ( $\text{max}$ ) tömb ( $t : \text{array}[1..\text{max}] \text{ of Item}$ ) tárolja, amelynek az 1. és a  $\text{top}$ -adik közötti része felel meg a veremben tárolt elemeknek: az 1. indexű a verem alja, a  $\text{top}$ -adik a teteje.



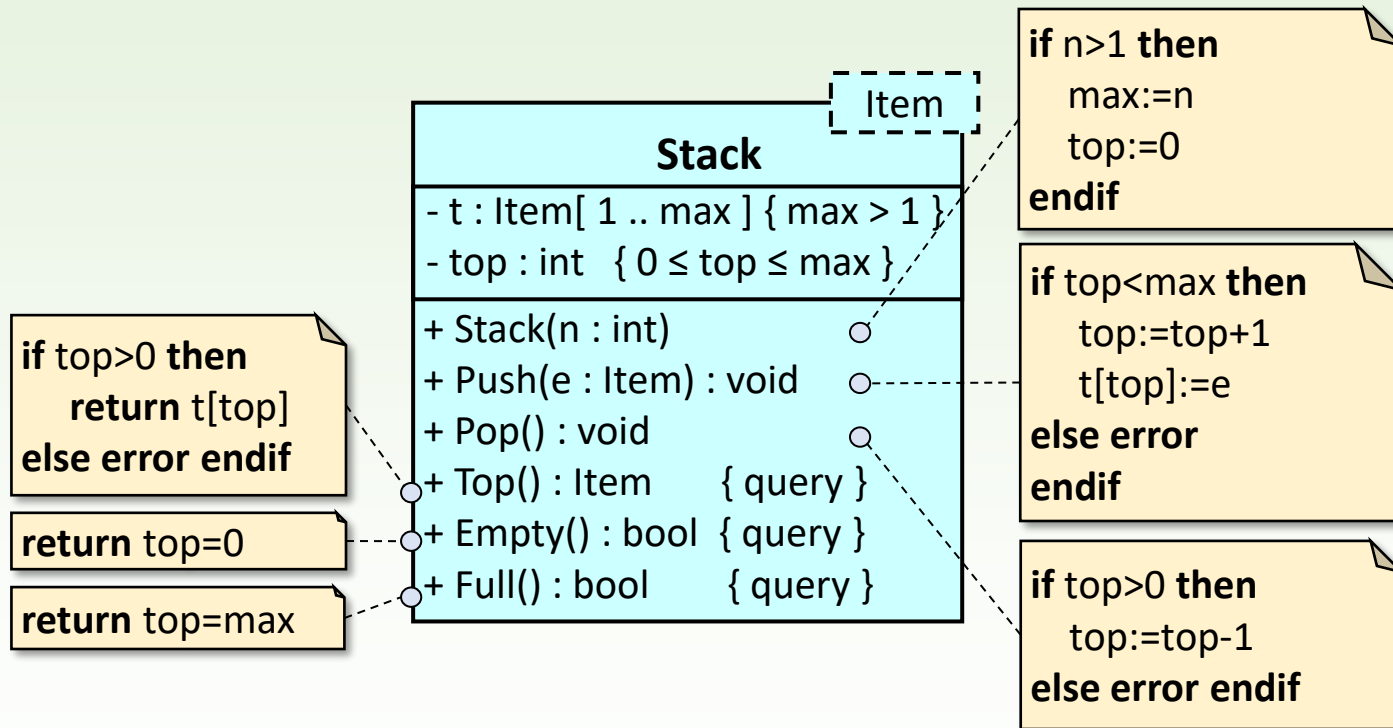
Stack	
- $t : \text{Item}[1 .. \text{max}] \{ \text{max} > 1 \}$	
- $\text{top} : \text{int} \{ 0 \leq \text{top} \leq \text{max} \}$	
+ $\text{Stack}(n : \text{int})$	
+ $\text{Push}(e : \text{Item}) : \text{void}$	
+ $\text{Pop}() : \text{void}$	
+ $\text{Top}() : \text{Item} \quad \{ \text{query} \}$	
+ $\text{Empty}() : \text{bool} \quad \{ \text{query} \}$	
+ $\text{Full}() : \text{bool} \quad \{ \text{query} \}$	

# Korlátos verem

- Három állapotot vezetünk be, amelyek között a verem Push() és Pop() műveletei hatására következhet be átmenet.

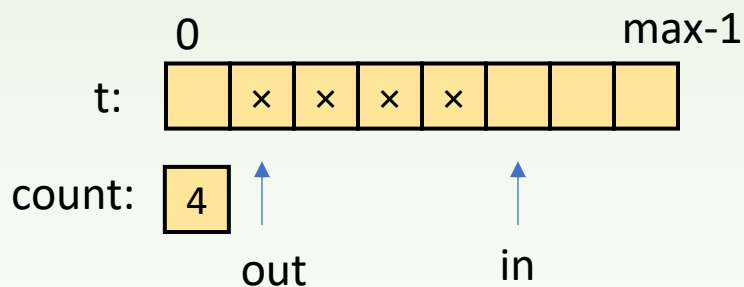


# Korlátos verem



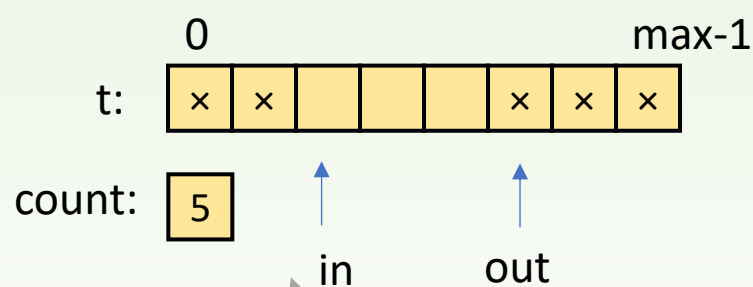
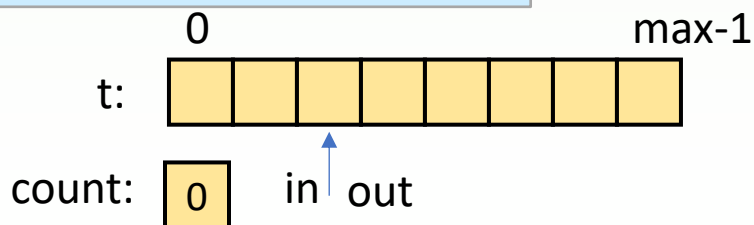
# Példa: Korlátos sor

- ❑ Készítsünk egy sorként (FIFO) működő tárolót előre nem meghatározott típusú (Item) elemek számára az Enqueue() és Dequeue() műveletekkel.
- ❑ Az elemeket egy tömbben ( $t : \text{Item}^{0..max-1}$ ) tároljuk úgy, hogy a sor elejét (a legrégebben betett elem indexét) egy out adattag, a sor végét (a legkésőbb betett elem utáni hely indexét) egy in adattag tárolja. A sor elemeinek számát (count) külön is nyilvántartjuk.

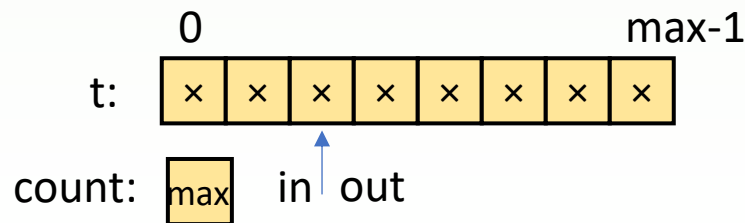


Invariáns állítás:

$$\text{count} = \begin{cases} \text{in} - \text{out} & \text{ha } \text{in} > \text{out} \\ \text{max} - (\text{out} - \text{in}) & \text{ha } \text{in} < \text{out} \\ \text{max vagy } 0 & \text{ha } \text{in} = \text{out} \end{cases}$$



a sor használata közben az in és out értéke ciklikusan (azaz modulo max) nő



# Korlátos sor állapotgépe

- Az állapotgép modellje segíti a metódusok tervezését.
- Három állapotot vezetünk be, amelyek között a sor Enqueue() és Dequeue() műveleteinek hatására következhet be átmenet.

## Queue

**Queue(n:int)**

/ in,out,count := 0,0,0

**if** n>1 **then**  
 max:=n  
 t = **new** Item[max]  
 in,out,count:=0,0,0  
**else error endif**

in := (in+1) mod max

**if** count<max **then**  
 t[in]:=e  
 Inc(in); ++count  
**else error endif**

Enqueue(e)  
 / t[in]:=e; Inc(in); ++count

Enqueue(e) [count<max-1]  
 / t[in]:=e; Inc(in); ++count

Enqueue(e) [count=max-1]  
 / t[in]:=e; Inc(in); ++count

**Enqueue(e:Item)**  
 Enqueue(e) / error

üres  
 [count=0]

e:=Dequeue() / error

fel kell tennünk,  
 hogy max>1

e:=Dequeue() [count=1]  
 / e:=t[out]; Inc(out); --count; **return** e

köztes  
 [0<count<max]

e:=Dequeue() [count>1]  
 / e:=t[out]; Inc(out); --count; **return** e

e:=Dequeue()  
 / e:=t[out]; Inc(out); --count; **return** e

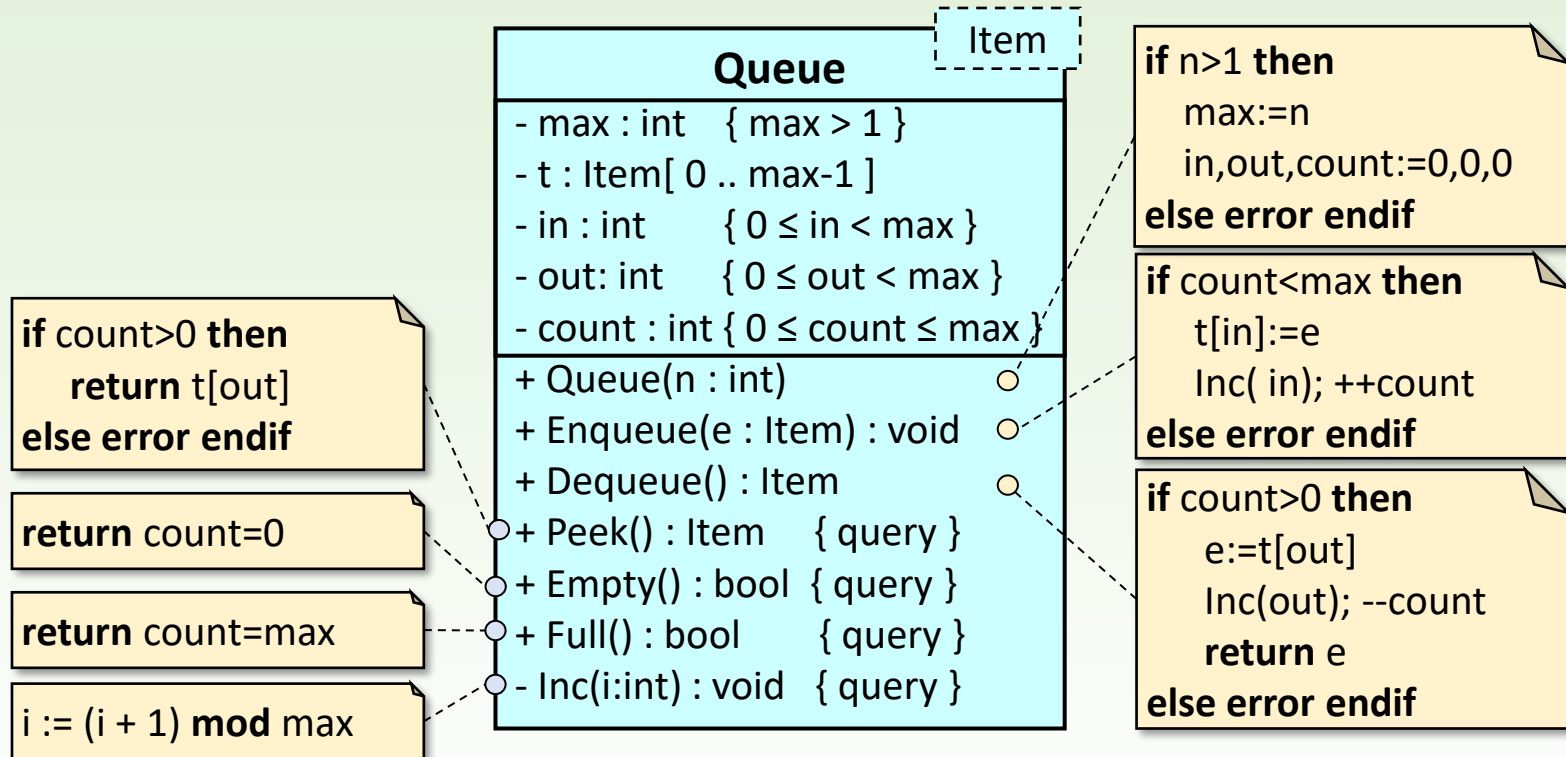
out := (out+1) mod max

tele  
 [count=max]

**e:=Dequeue()**  
 ----->

**if** count>0 **then**  
 e:=t[out]  
 Inc(out); --count  
**return** e  
**else error endif**

# Korlátos sor osztály diagramja

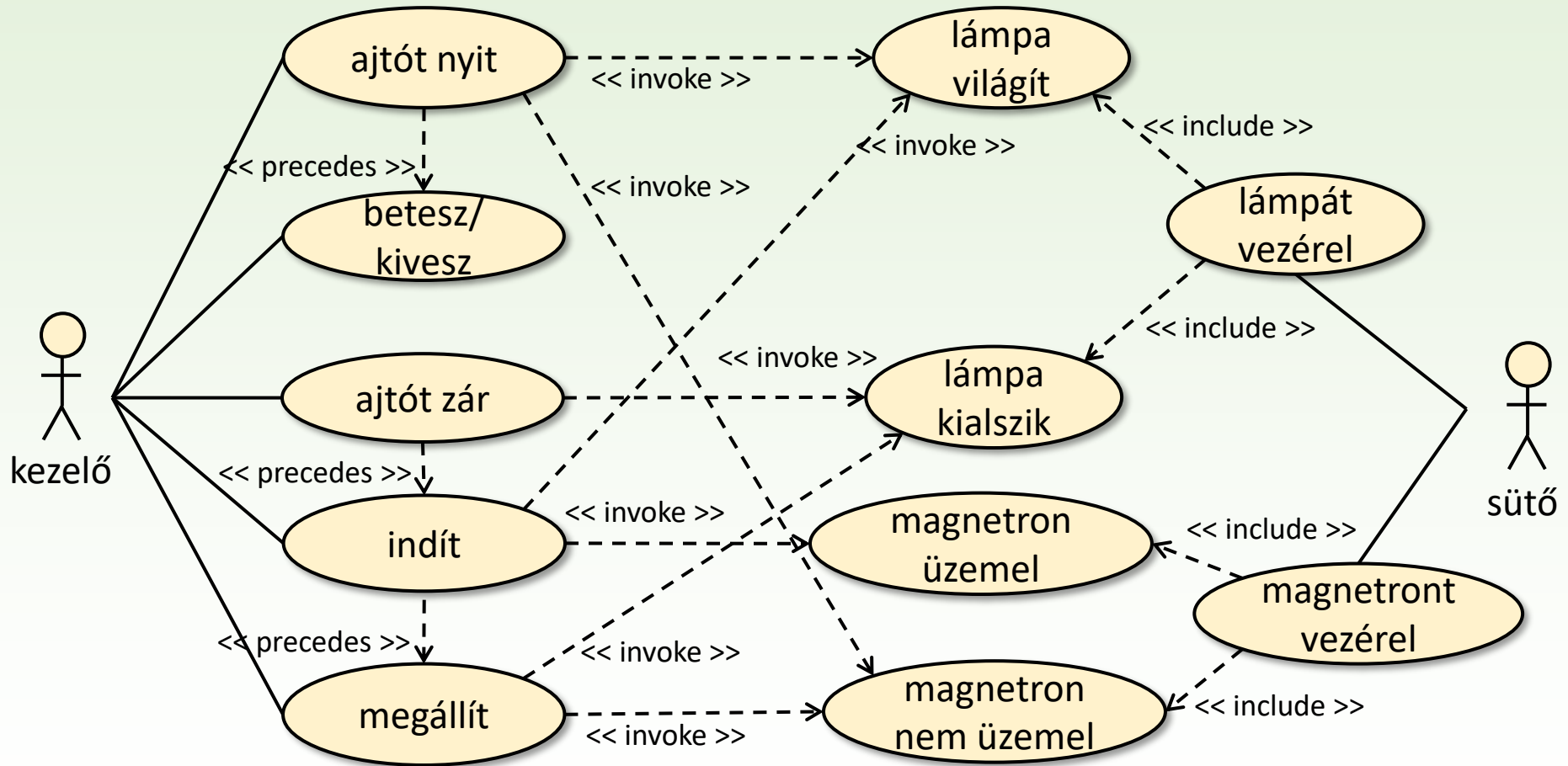


# Mikrohullámú sütő

Modellezzünk egy mikrohullámú sütőt!

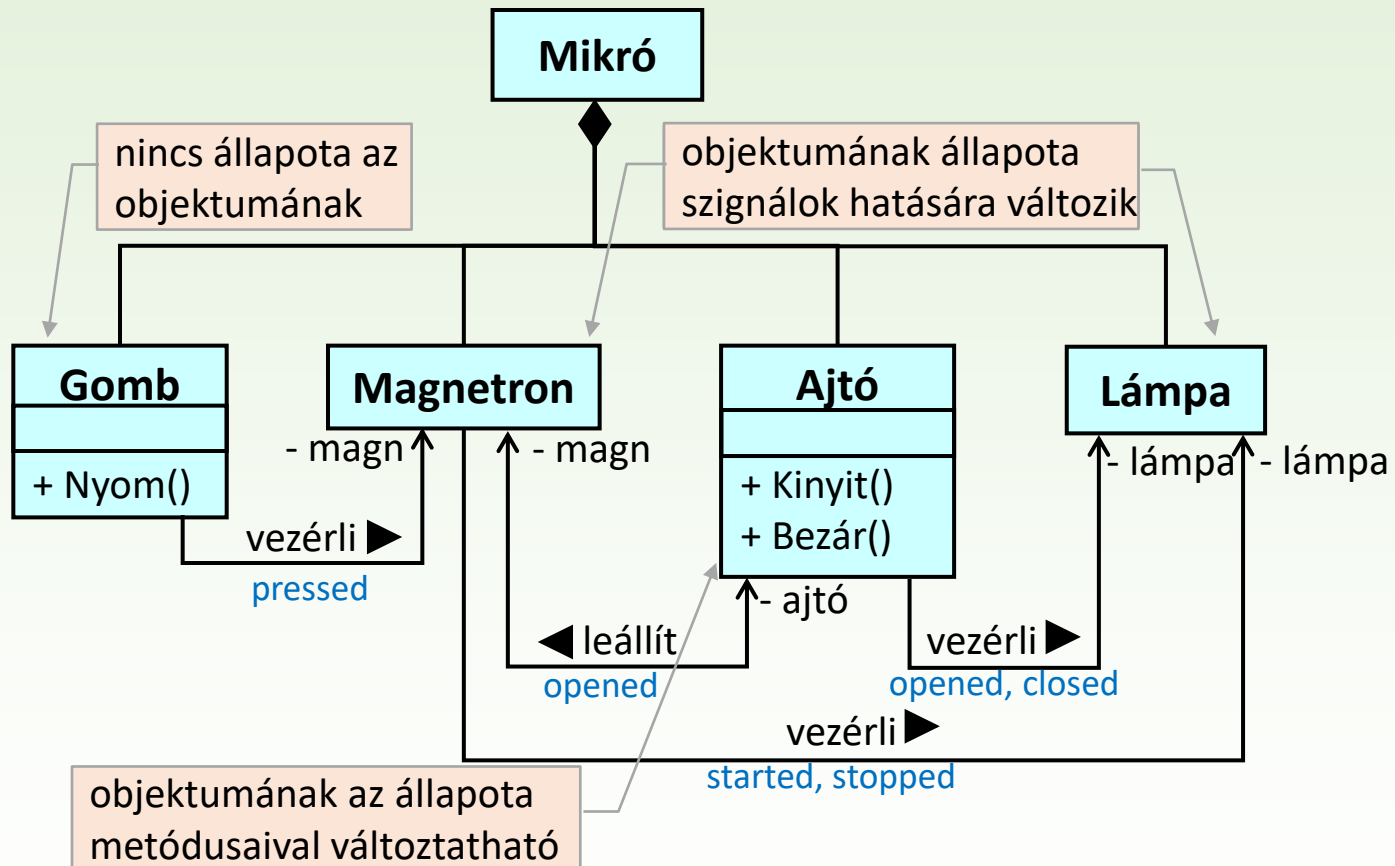
A mikrohullámú sütő meghatározó elemei: az ajtó, a lámpa, egy vezérlő gomb és a magnetron. A magnetront a vezérlő gomb megnyomása indítja el, feltéve, hogy az ajtó csukva van, és ilyenkor a lámpa is világítani kezd. A magnetron működését vagy a vezérlő gomb megnyomásával állíthatjuk le, ilyenkor a lámpa is kialszik, vagy az ajtó kinyitásával, de ilyenkor a lámpa égve marad. Az ajtó kinyitása mindig felkapcsolja a lámpát, bezárása pedig lekapcsolja.

# Mikrohullámú sütő

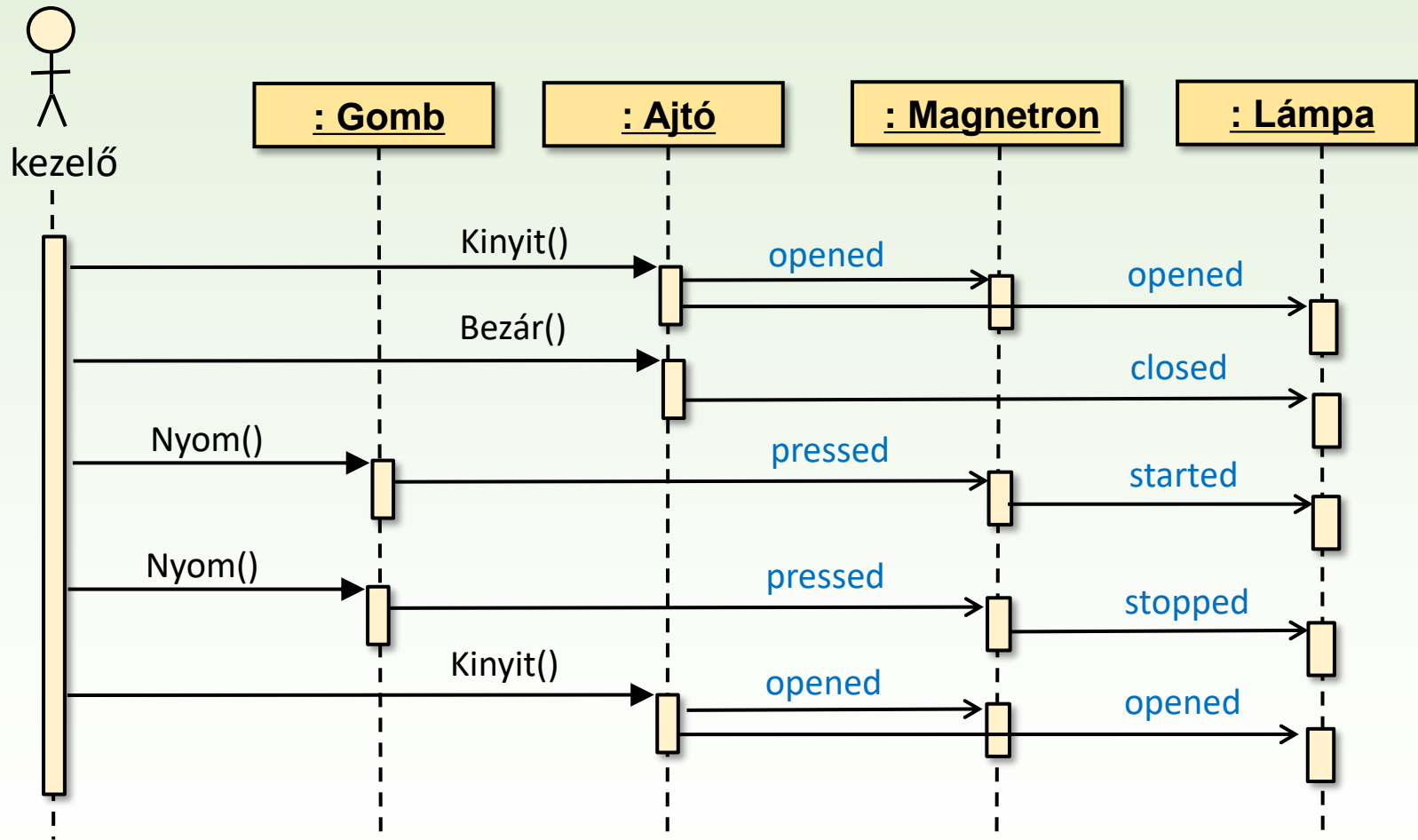




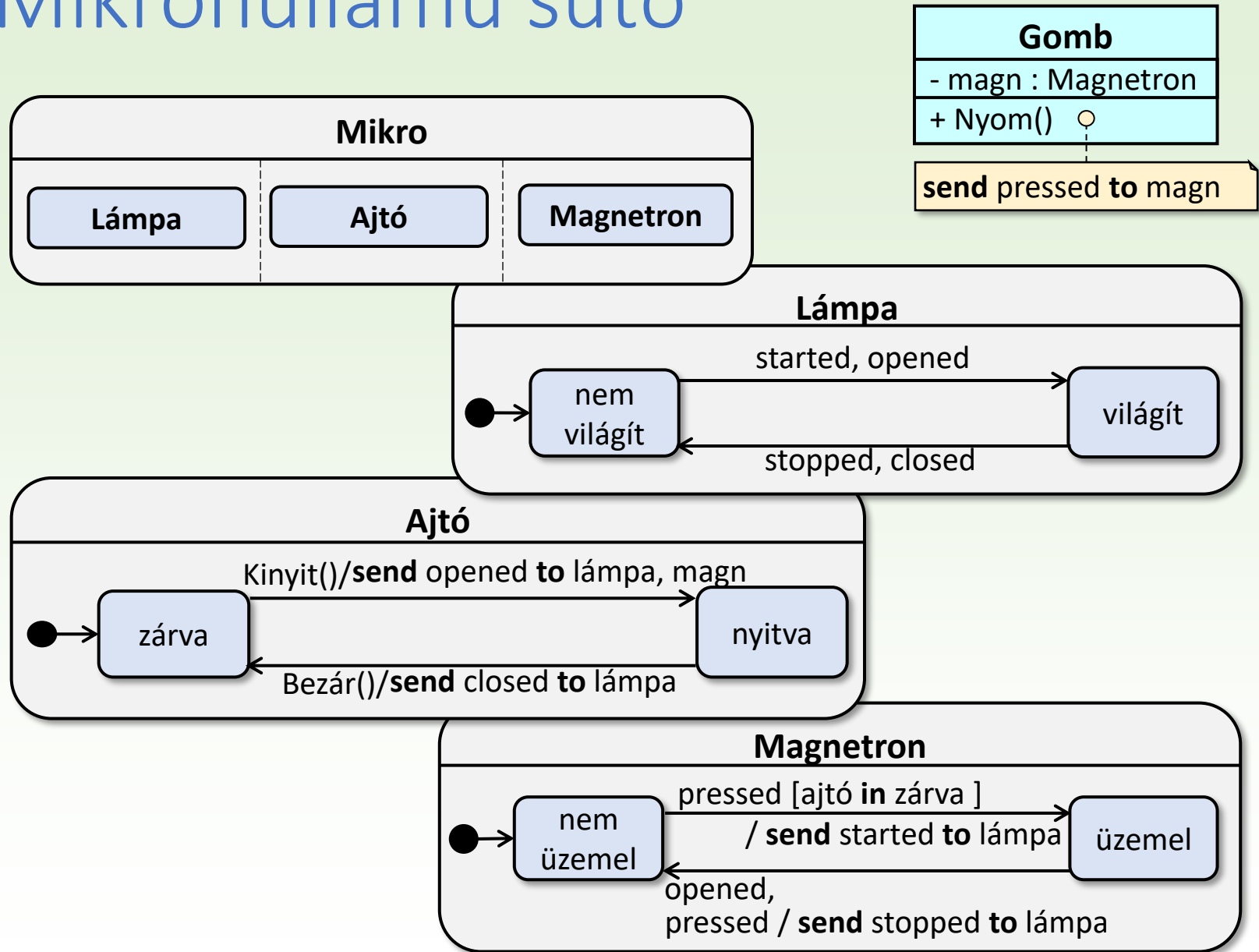
# Mikrohullámú sütő



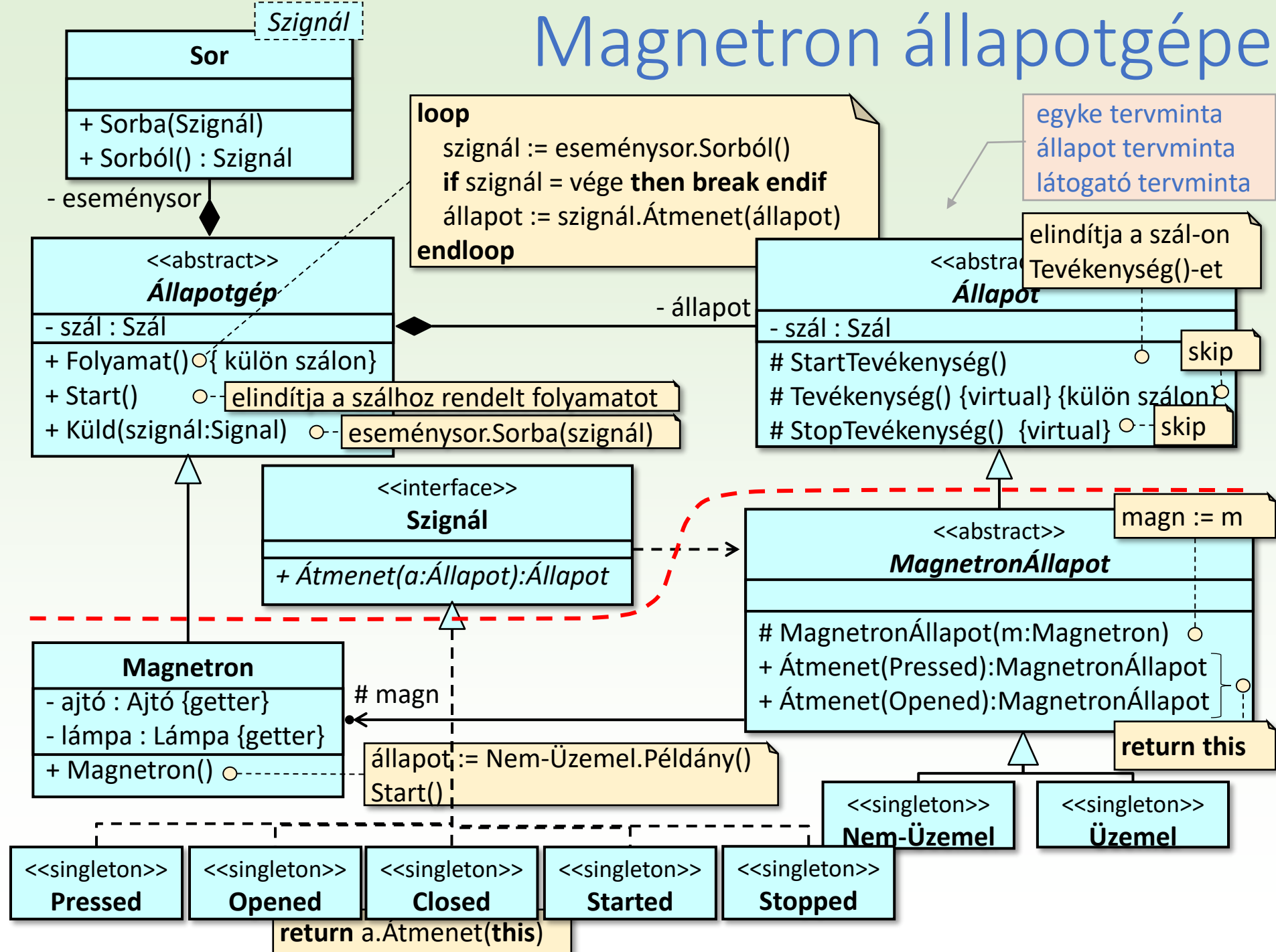
# Mikrohullámú sütő



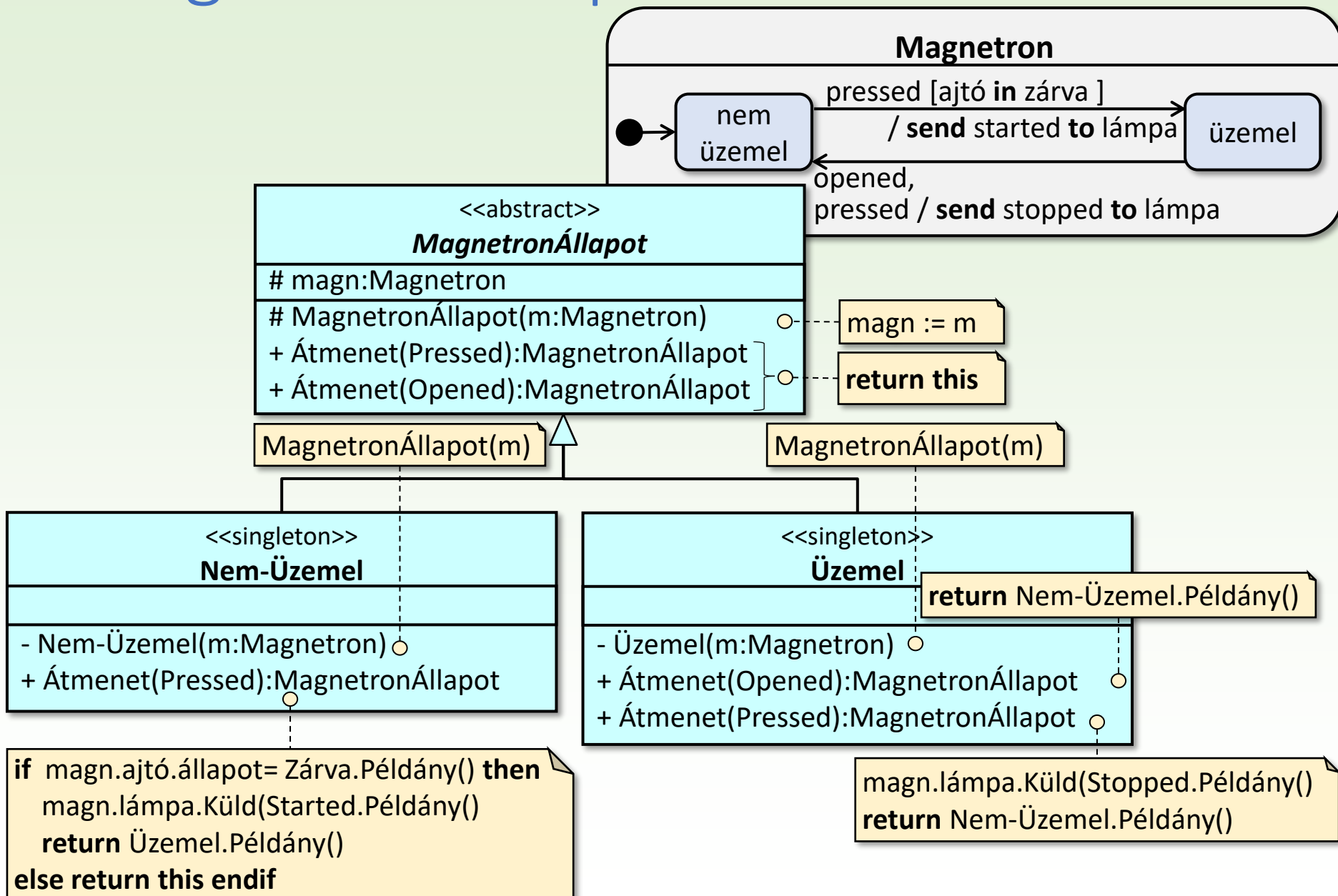
# Mikrohullámú sütő



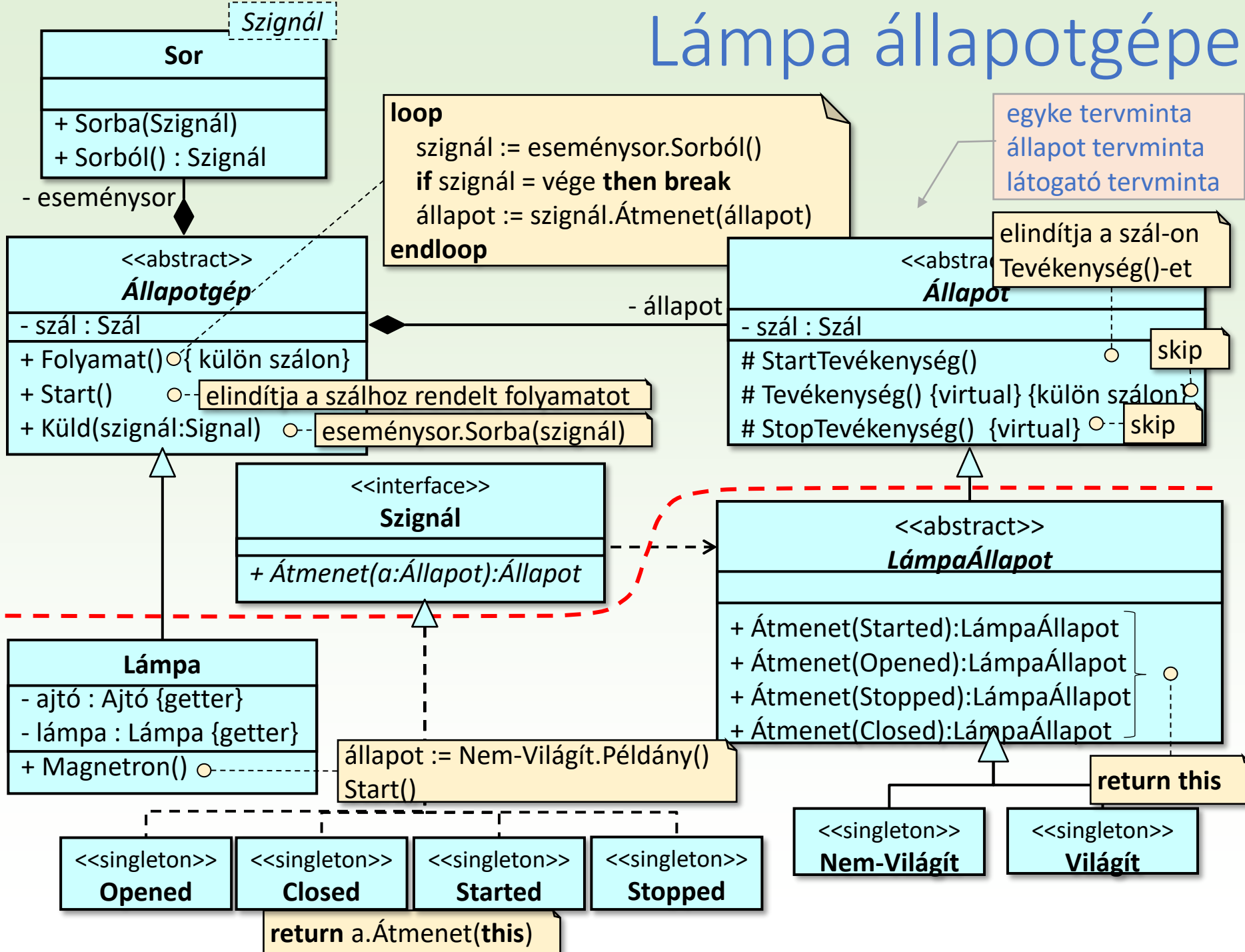
# Magnetron állapotgépe



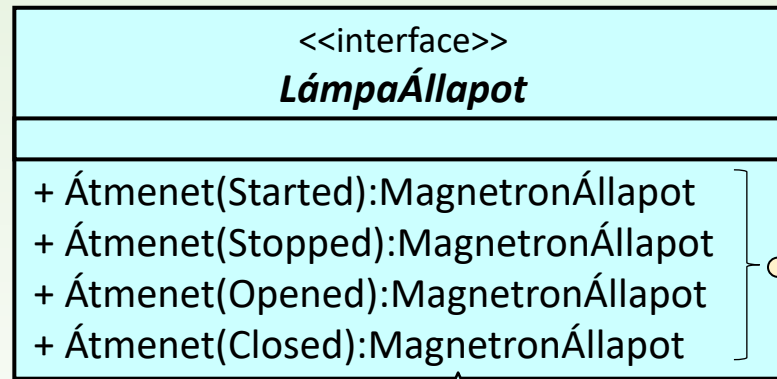
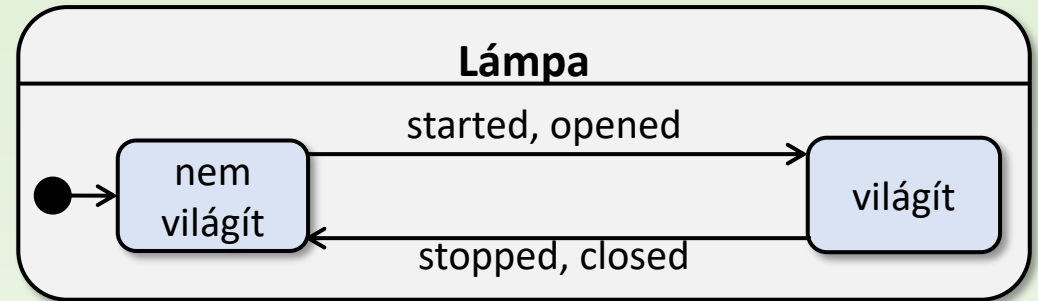
# Magnetron állapotai



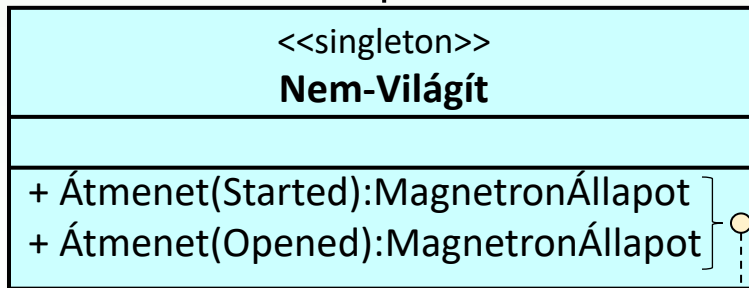
# Lámpa állapotgépe



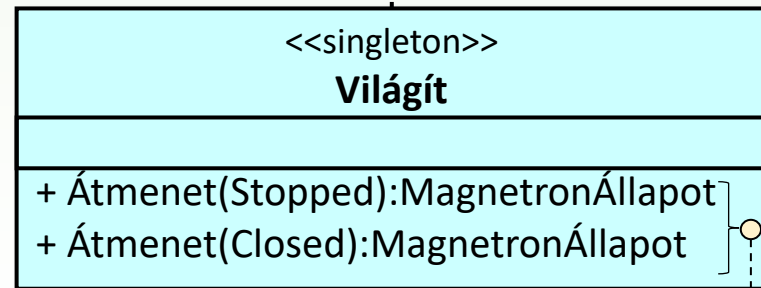
# Lámpa állapotai



return this

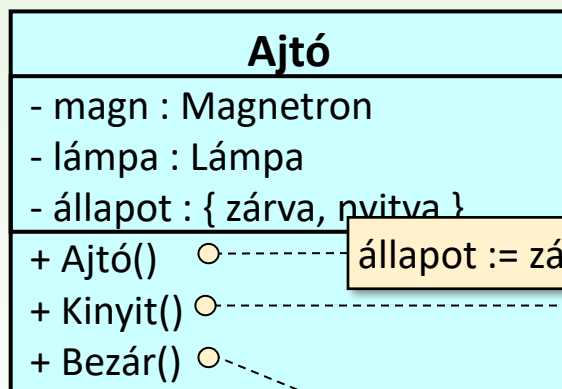
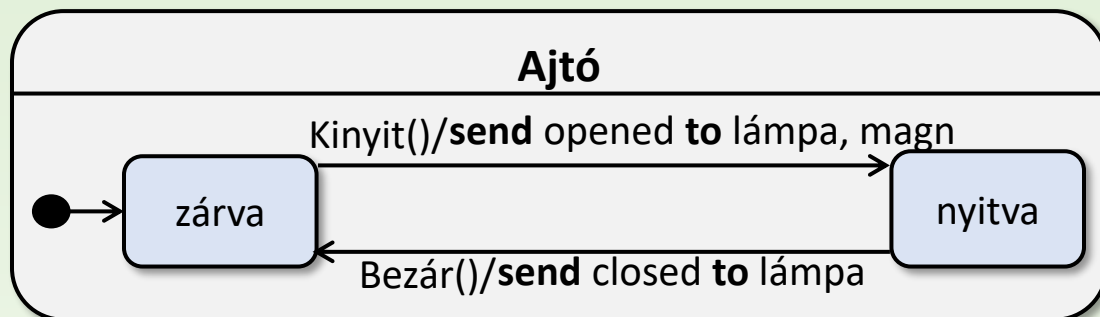


return Világít.Példány()



return Nem-Világít.Példány()

# Ajtó



állapot := zárva

```
if állapot = zárva then
    magn.Küld(Opened.Példány() )
    lámpa.Küld(Opened.Példány() )
    állapot := nyitva
endif
```

```
if állapot = nyitva then
    lámpa.Küld(Closed.Példány() )
    állapot := zárva
endif
```