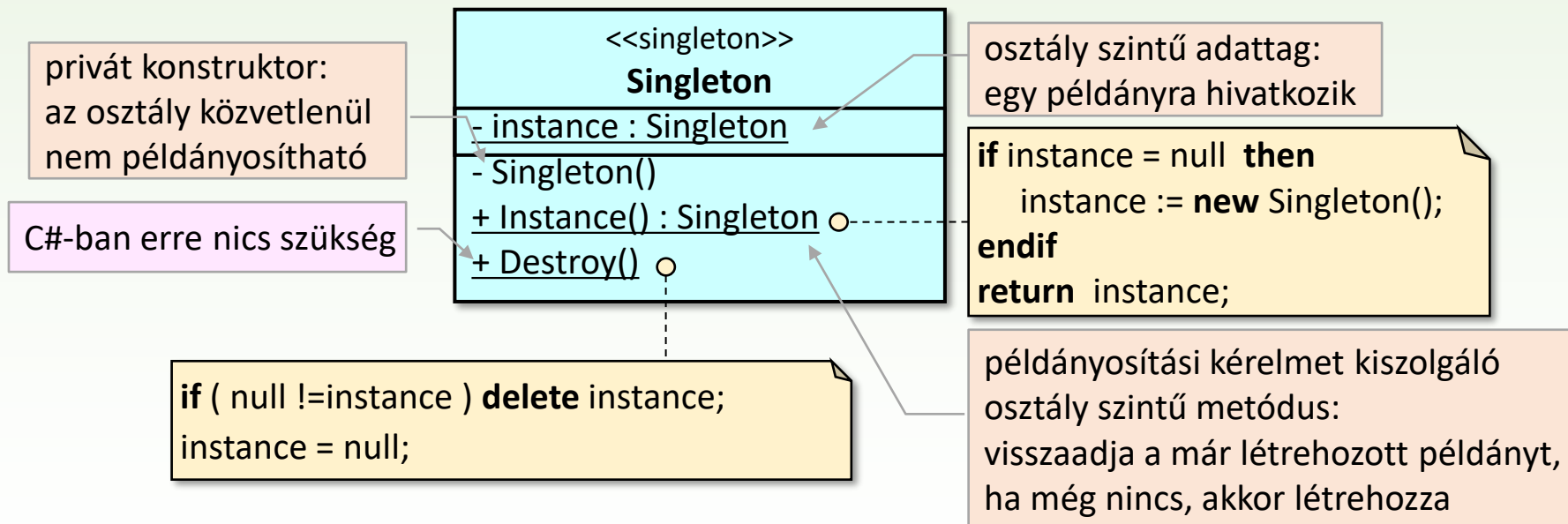


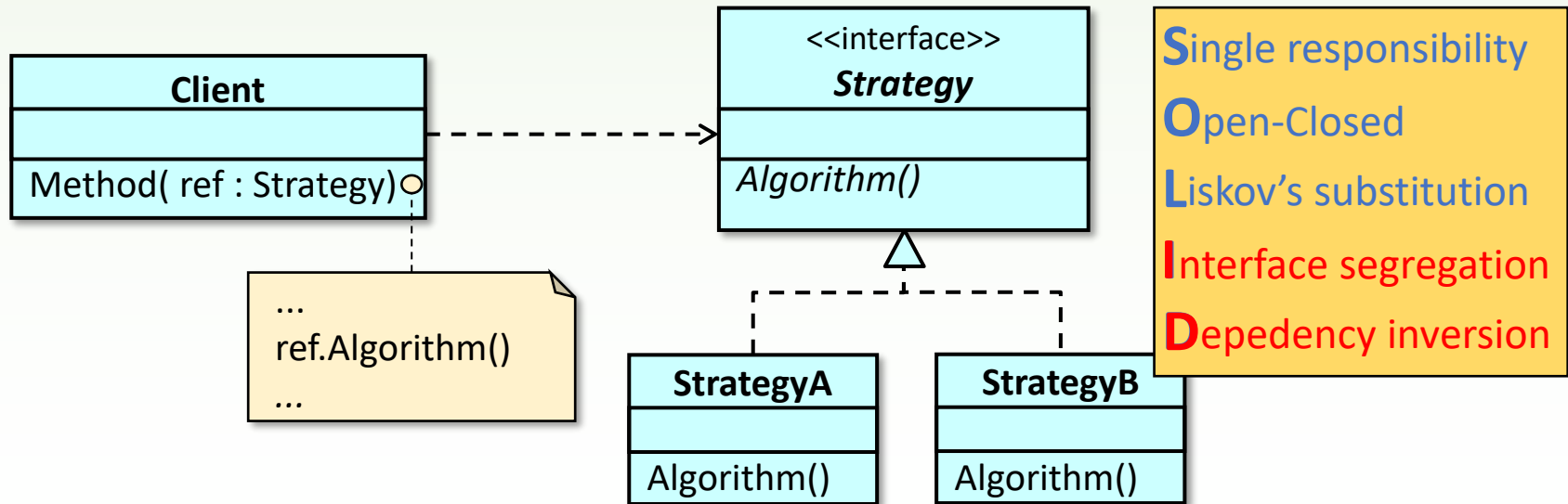
# Egyke (singleton) tervezési minta

- Amikor egy osztályhoz legfeljebb egy objektumot kell példányosítani függetlenül a példányosítási kérések számától.



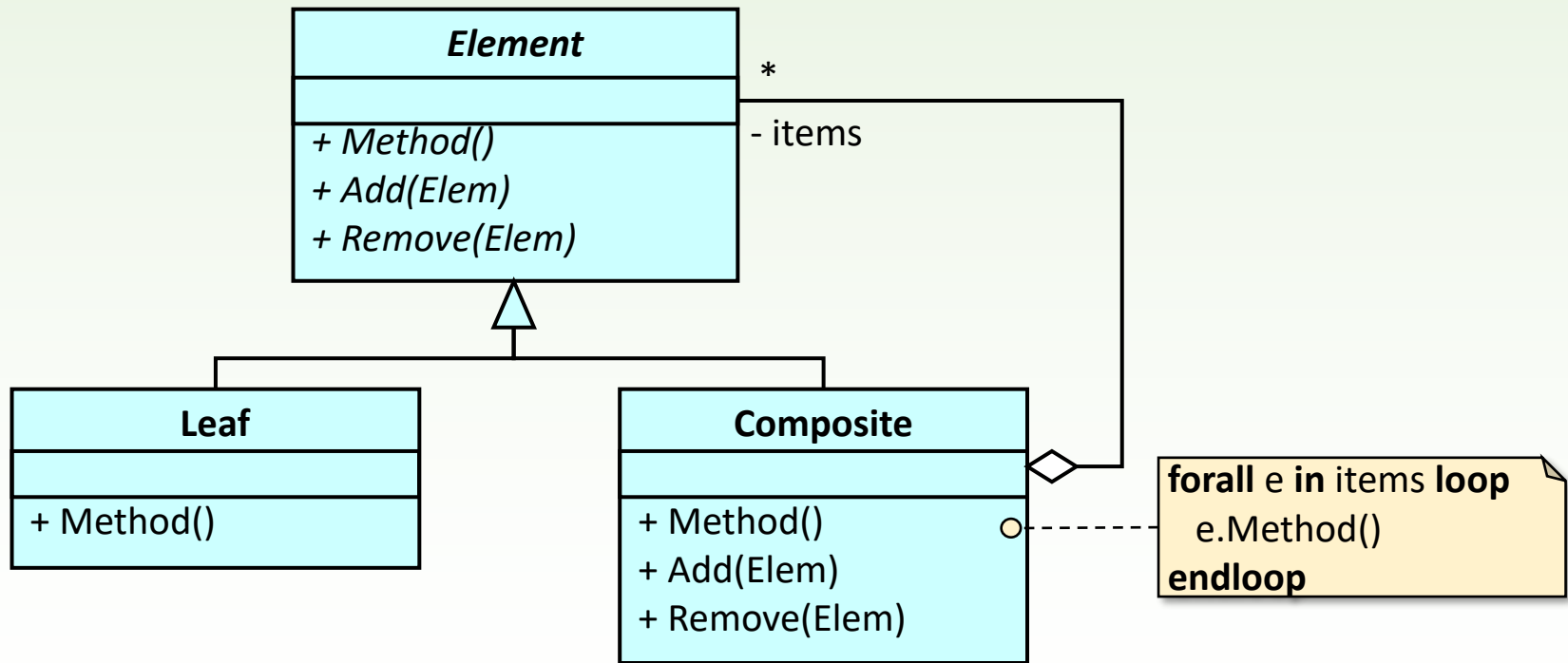
# Stratégia (strategy) tervezési minta

- Tevékenységek (algoritmusok, stratégiák) családját olyan osztályok metódusaként vezetünk be, amelyek egy közös interfészt valósítanak meg azért, hogy majd futási időben dőljön el, hogy mikor melyiket használja fel egy másik (kliens) osztály metódusa. Ezeket a tevékenységeket egy-egy (Strategy típusú) objektum hordozza a saját metódusaként, és ezt az objektumot kapja meg a kliens osztály metódusa paraméterként (vagy akár be is aggregálható a kliens objektumba).



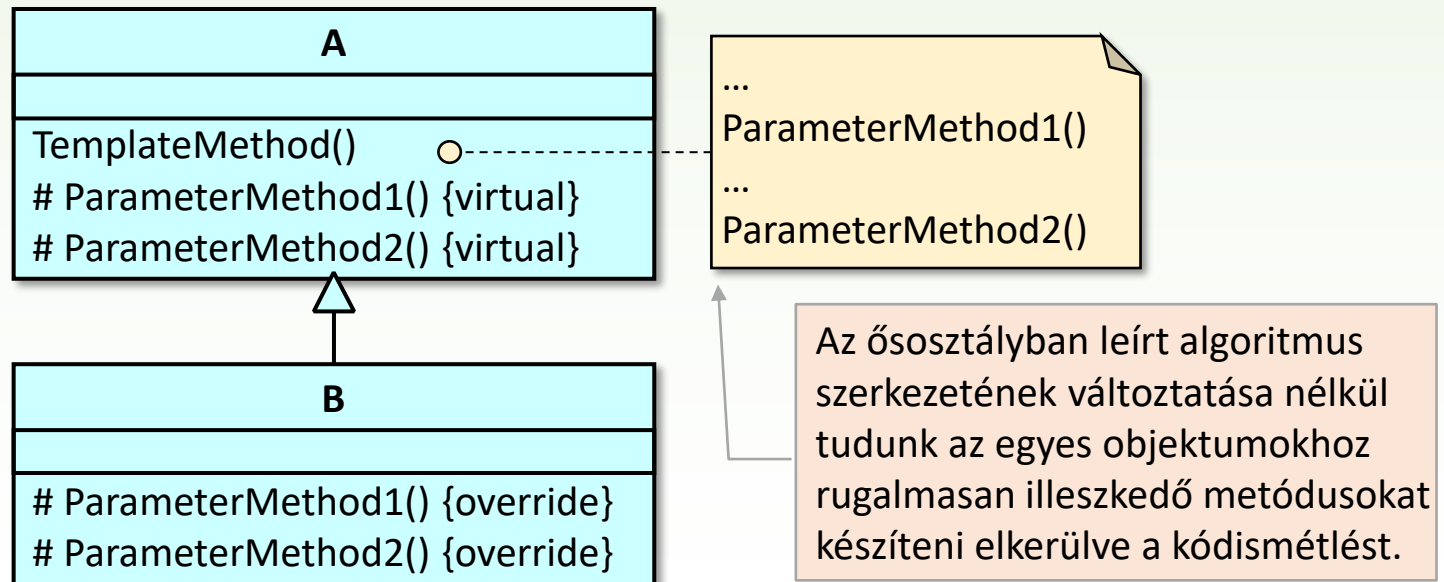
# Összetétel (composite) tervezési minta

- Objektumok fastruktúrába rendezéséhez ajánlják különösen akkor, ha a rész-egész (gyerek-szülő) viszonyokat kell kezelni. A módszer révén a levélcsúcsokat és a részfákat egységesen kezelhetjük.



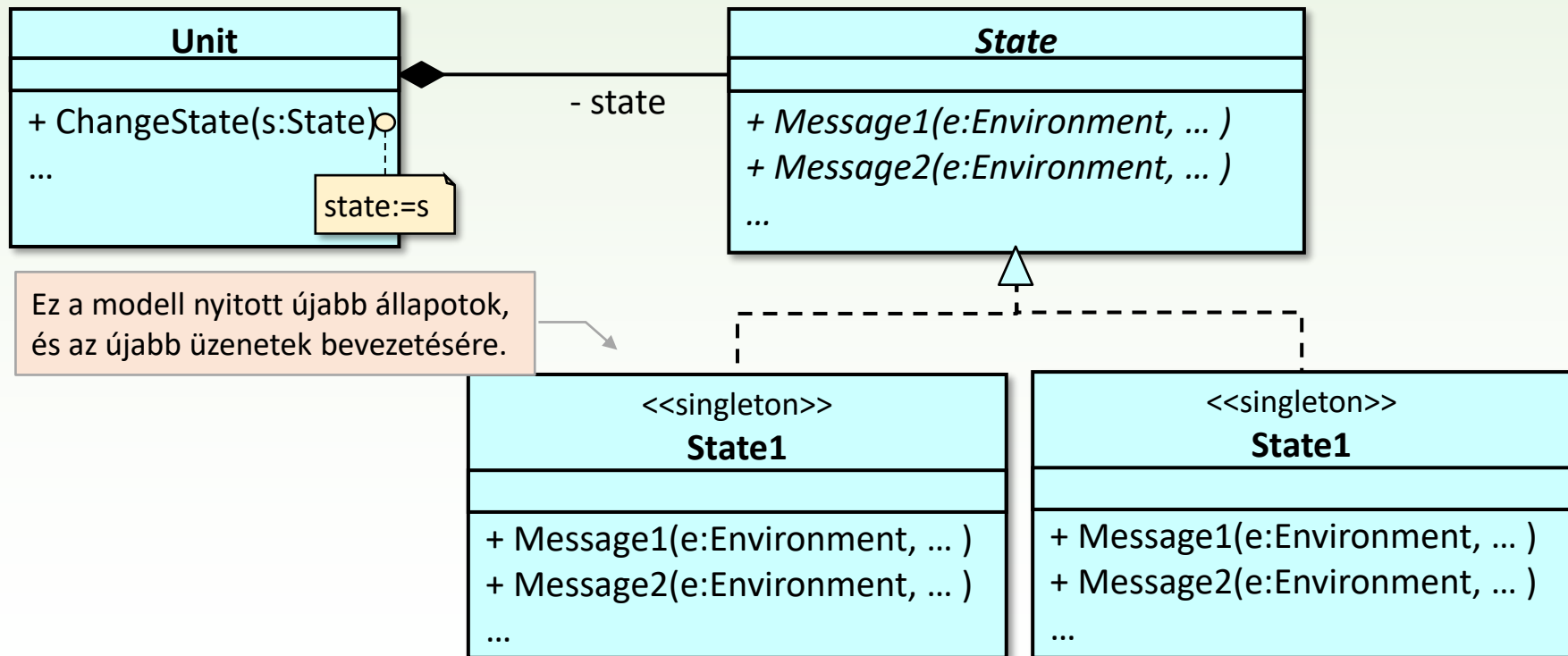
# Sablonfüggvény tervezési minta (template method)

- Egy tevékenységet (sablonfüggvényt) egy őosztály metódusaként úgy definiálunk, hogy annak speciális, altípustól függő részeit csak virtuális metódus-hívások jelzik (sablon-paraméterek), majd ezen metódusokat az alosztályokban az ott elvárt módon definiáljuk felül.



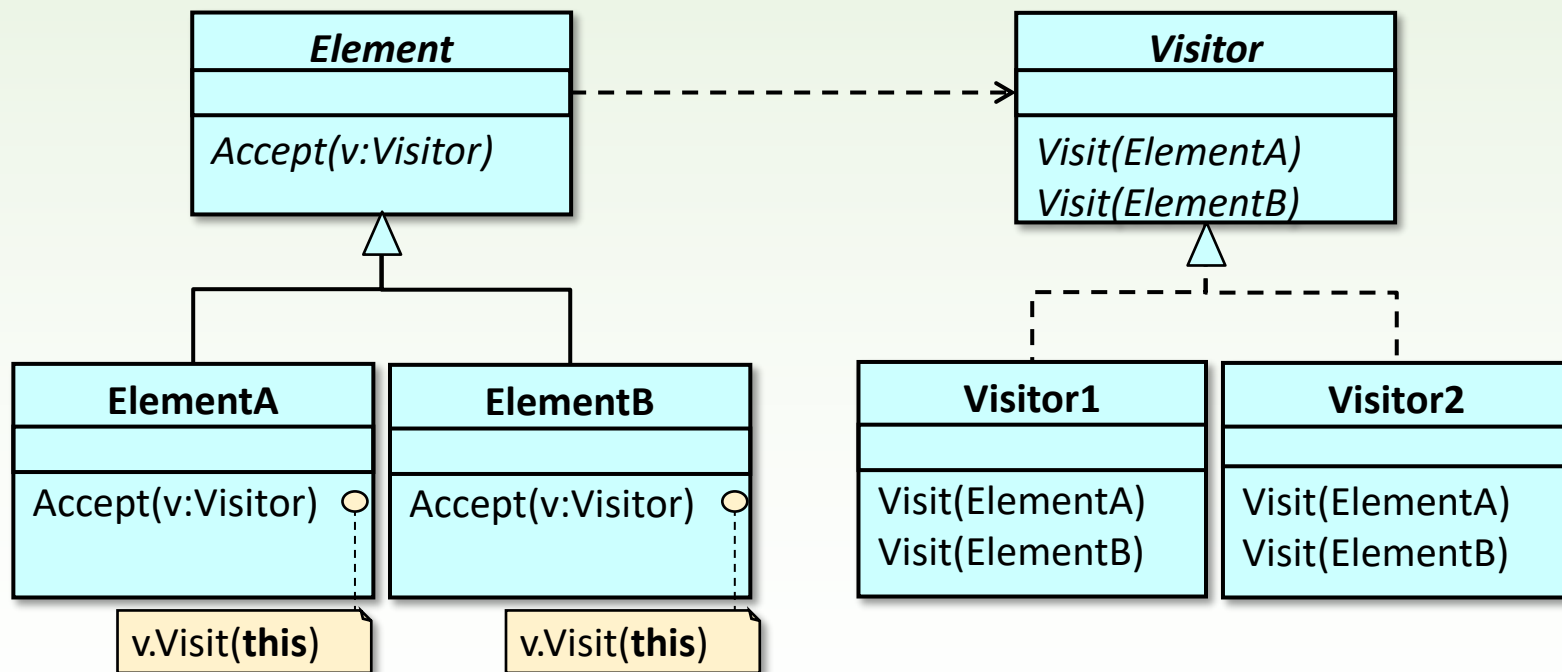
# Állapot (state) tervezési minta

- Amikor egy objektum metódusai az objektum állapotától függően viselkednek, akkor – az ugyanolyan szerkezetű elágazások ismétlése helyett – az objektum állapotait külön egyke objektumokba szervezzük. Ezek közös interfésze specifikálja az állapotfüggő metódusokat, míg az eredeti objektum rendelkezik az állapotváltóztató metódussal.



# Látogató (visitor) tervezési minta

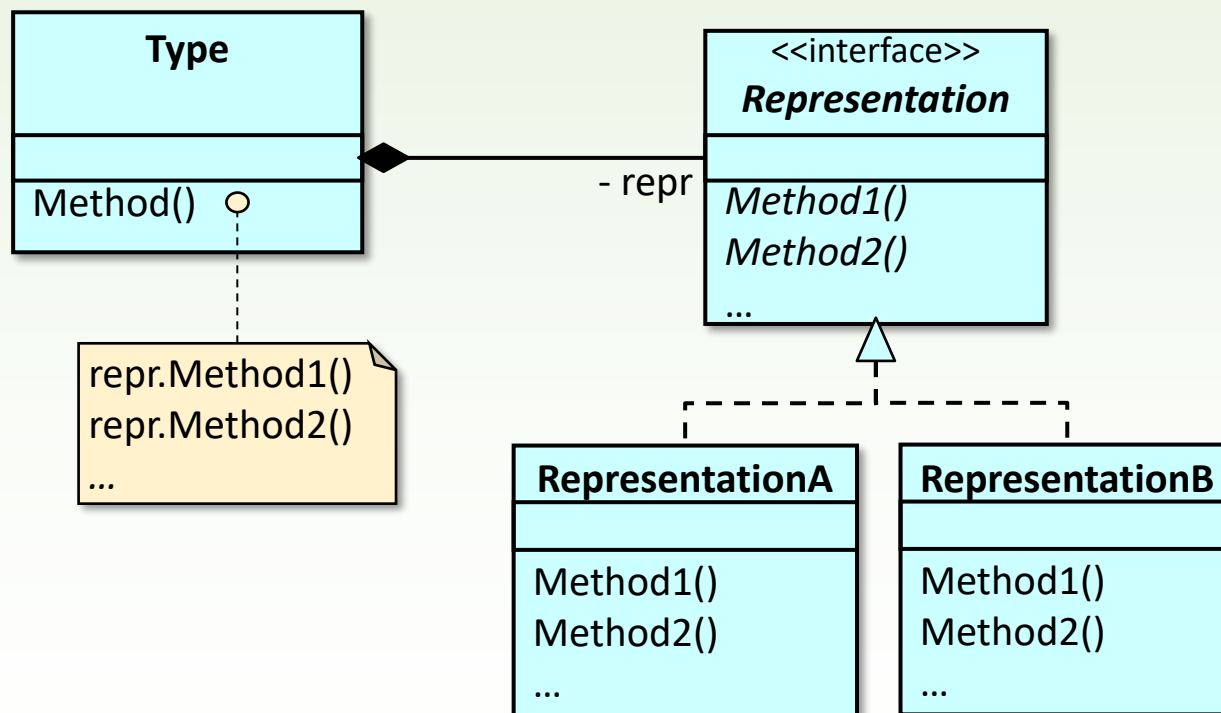
- Amikor egy metódus (Accept()) működése a saját osztályán kívül attól is függ, hogy a paramétere egy osztály melyik alosztályának objektumára hivatkozik, és nem akarjuk, hogy ez a függőség egy elágazás (vagy ugyanolyan szerkezetű elágazások) formájában jelenjen meg a kódban.



altípusos és parametrikus  
polimorfizmus

# Híd (bridge) tervezési minta

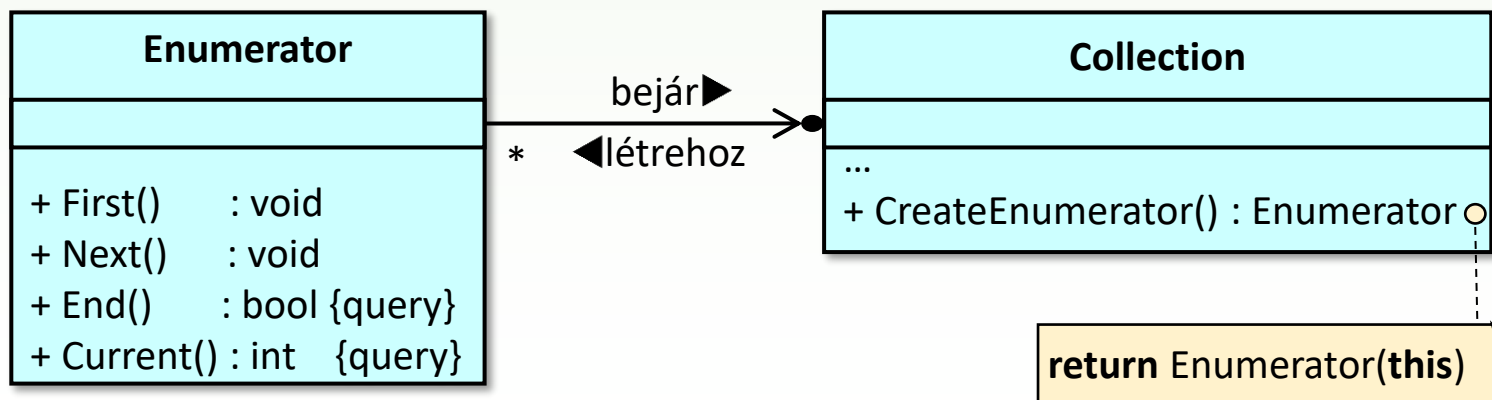
- Egy típus reprezentációját leválasztjuk a típust biztosító osztályról azért, hogy azt rugalmasan, futási időben választhassuk ki.



# Felsoroló (iterátor) tervezési minta

- ❑ Egy gyűjtemény elemeinek felsorolását (bejárását) a gyűjteménytől elkülönülő, önálló objektummal (felsorolóval) érdemes végezni.
- ❑ A felsorolónak el kell tudni érni a felsorolandó gyűjteményt: hivatkozni kell tudnia rá, de nem módosítja azt.
- ❑ A felsoroló objektumot a felsorolni kívánt gyűjtemény hozza létre, akár többet is, és nyilvántarthatja az őt bejáró felsorolókat.

Célszerű, ha a felsorolás műveletei csak a felsorolandó gyűjteménynek a lekérdező (getter) metódusaira támaszkodnak azért, hogy egyszerre több felsoroló is dolgozhasson a gyűjteményen anélkül egymást zavarnák.





# Gyártófüggvény (factory method) tervezési minta

- Egy őosztály alosztályainak objektumait egy segédosztály metódusával (Create) hozzuk létre, amely a példányosítást átruházza a segédosztály megfelelő alosztályára.

