

# *OEP BEADANDÓ*

Dokumentáció

*Soós Csaba*

*ELTE IK*

## Feladat

### 5. Modellezzük egy **könyvtár** működését!

Egy könyvtár nyilvántartja a kikölcsönözhető könyveit (ismert címe, szerzője, kiadója, ISBN száma), és a hozzá beiratkozott tagokat (ismert a neve, címe, igazolványszáma), és. A tagok kiiratkozhatnak a könyvtárból, ha már nincs náluk kikölcsönzött könyv. A könyvtár beszerezhet, illetve leselejtezhetsz könyveket.

Egy tag egy alkalommal több könyvet is kikölcsönözhet, de egyszerre ötnél több könyv nem lehet nála. A kikölcsönzött könyveket több részletben is visszahozhatja, de ügyelni kell a kölcsönzési idő betartására. Egy könyv kölcsönzési pótdíja a kölcsönzés lejáratát idejéről számított napok számának, valamint a könyv példányszámától és műfajától függő együtthatónak szorzata.

napi pótdíj	sok példány	kevés példány	ritkaság
természettudományi	20	60	100
szépirodalmi	10	30	50
ifjúsági	5	10	30

- A könyvtár tudjon beszerezni, és leselejtezni könyveket; új személy be tudjon iratkozni a könyvtárba, illetve ki tudjon lépni, ha nincs hátraléka; egy tag kikölcsönözhesse az általa megadott című könyvek közül azokat, amelyek elérhetők (feltéve, hogy nincs tartozása); visszahozhasson kikölcsönzött könyveket; befizethesse tagdíját vagy az esetleges pótdíját.
- Van-e hátraléka egy adott tagnak: kell-e tagdíjat és/vagy pótdíjat fizetnie, és mennyit?
- Megtalálható-e a könyvtárban egy adott című könyv, és kikölcsönözhető-e?
- Tagja-e egy adott nevű személy a könyvtárnak?

Készítsen használati eset diagramot! Ebben jelenjenek meg használati esetként a könyvtár fontosabb metódusainak nevei. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely egy könyvtárnak öt könyvét, és két könyvtári tagját mutatja, valamint három kölcsönzési eseményt, amelyekről leolvasható, hogy melyik tag milyen könyveket tart éppen magánál: az egyik tag kétszer kölcsönzött, először egy, majd két könyvet, a másik tag egyszer egy könyvet. Egy könyv legyen a könyvtárban. Egészítse ezt ki kommunikációs diagrammá!

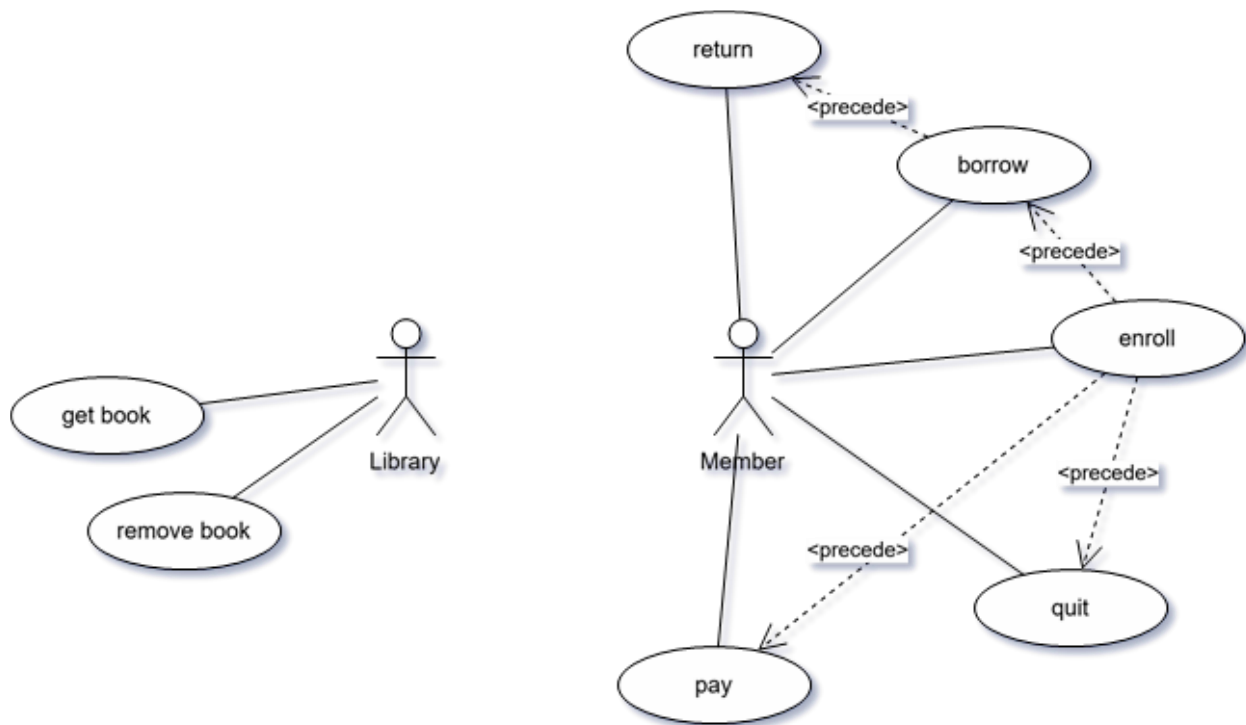
Készítse el egy kölcsönzés objektum állapotgépét! Különböztesse meg az „üres”, a „köztes”, és a „teli” állapotokat aszerint, hogy 0, 1-4, vagy 5 könyv van a kölcsönzés eseményhez rendelve. Az állapot-átmeneteket megvalósító tevékenységeket majd a kölcsönzés osztály metódusaiként definiálhatja. Rajzolja fel a feladat osztály diagramját! Felteheti, hogy a rejtett adattagokhoz mindig tartozik egy publikus getter: ha mégsem, akkor azt a „secret” megjegyzéssel jelölje. Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírása legyen minél tömörebb (például ciklusok helyett a megfelelő algoritmus minta specifikációs jelölését használja). Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta. Egy könyv pótdíját a késedelmi idő és a könyv fajtájától (természettudományos, szépirodalmi, ifjúsági), valamint a könyvtárban található példányszámától (sok, kevés, ritka) függő szorzótényező határozza meg.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni egy könyvtár könyveit, könyvtári tagjait, néhány kölcsönzést és könyv visszahozást. Válaszoljuk meg a b. c. d. kérdéseket. Készítsen teszteseteket, néhányak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

## Terv

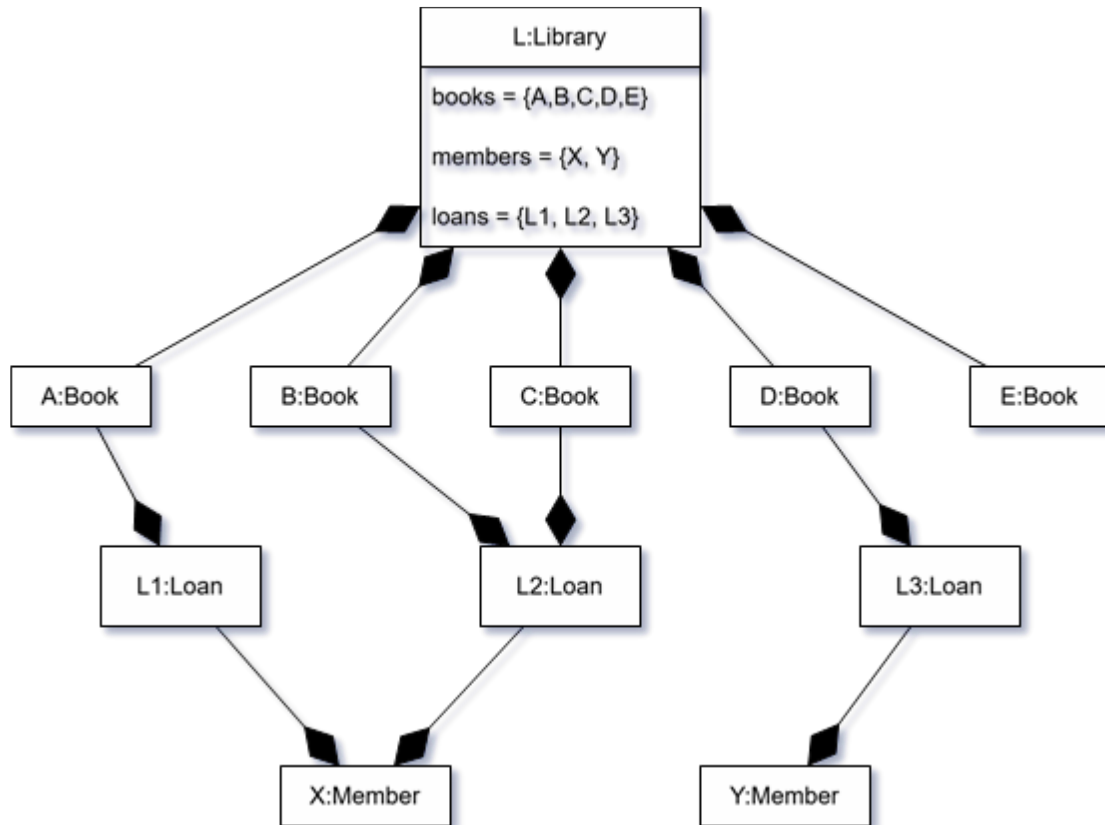
### Használati eset diagram:

A diagram egy könyvtári rendszer fő szereplőit és azok alapvető műveleteit mutatja be. A Library aktor könyvek hozzáadását és eltávolítását végezheti. A Member aktor regisztrálhat, könyvet kölcsönözhet, visszavihet, fizethet, illetve megszüntetheti tagságát. Bizonyos műveletek sorrendisége is jelölve van: például a kölcsönzés csak regisztráció után, a kilépés pedig fizetés után lehetséges.



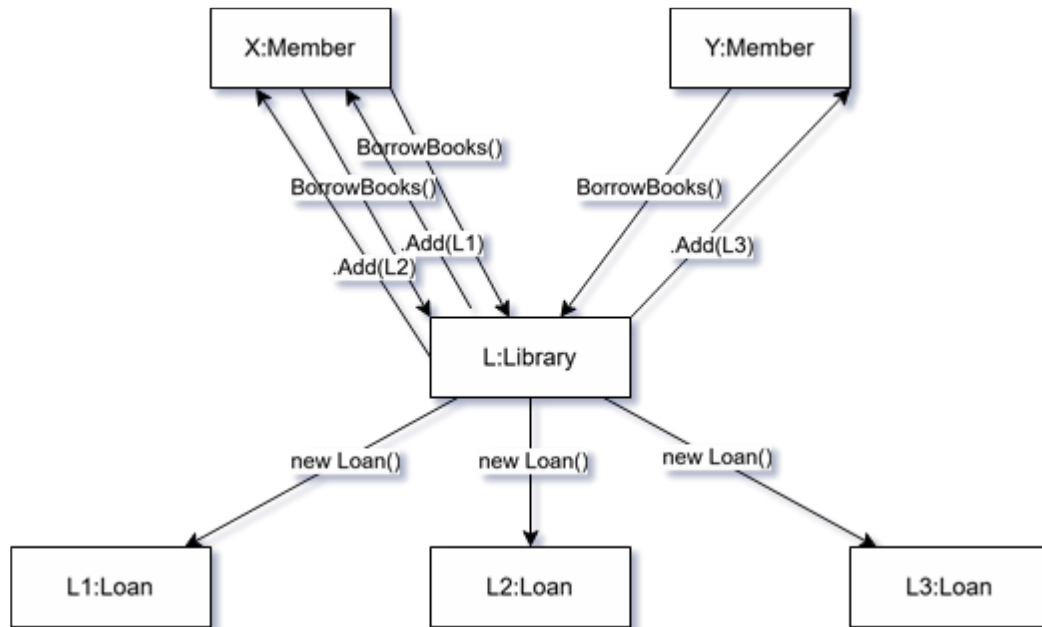
### Objektum diagram:

A diagram egy könyvtári rendszer objektumdiagramja, amely bemutatja, hogy az L könyvtár öt könyvet (A, B, C, D, E), két tagot (X, Y) és három kölcsönt (L1, L2, L3) tart nyilván. Az egyes kölcsönök konkrét könyvekhez és tagokhoz kapcsolódnak, szemléltetve a könyvtár, a könyvek, a tagok és a kölcsönzések közötti kapcsolatokat.



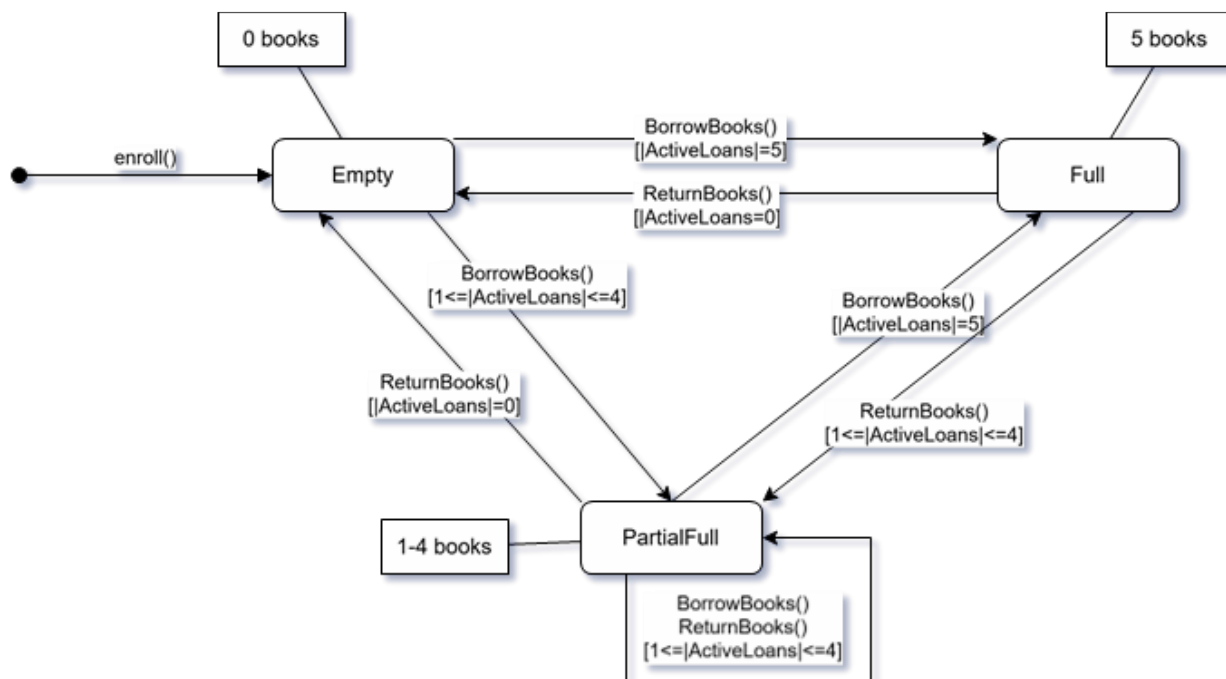
### Kommunikációs diagram:

A kommunikációs diagram bemutatja, hogyan működik együtt a könyvtári rendszerben két tag (X és Y), a könyvtár (L:Library) és a kölcsönzések (L1, L2, L3). A tagok könyvet kölcsönöznek a könyvtártól (BorrowBooks()), majd a könyvtár új kölcsön objektumokat hoz létre (new Loan()), amelyeket hozzáad a rendszerhez (Add(L1), Add(L2), Add(L3)). A diagram szemlélteti az üzenetküldéseket és az objektumok közötti kapcsolatokat a kölcsönzési folyamat során.

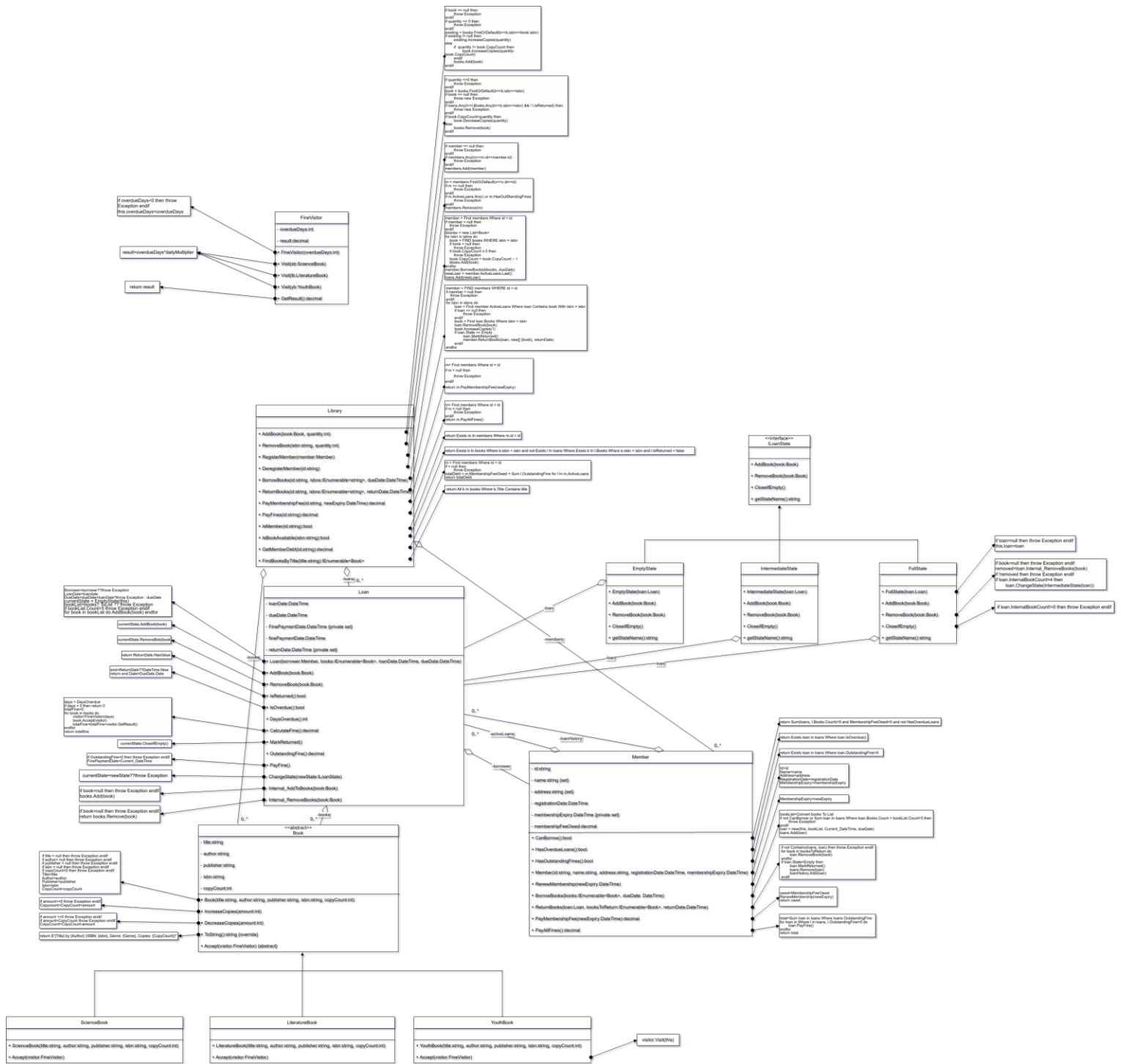


## Kölcsönzés objektum állapotgép:

A diagram a kölcsönzés objektum állapotgépét mutatja be egy könyvtári rendszerben. Az objektum három fő állapotot vehet fel: üres (Empty), részben teli (PartialFull) és teli (Full). Az üres állapot azt jelenti, hogy a tagnak nincs aktív kölcsönzése (0 könyv), a részben teli állapot 1–4 aktív kölcsönzést jelent, míg a teli állapotban a maximális, 5 könyv van kikölcsönözve. Az állapotok közötti váltásokat a könyvek kölcsönzése (BorrowBooks) és visszahozása (ReturnBooks) műveletek idézik elő, az aktuális kölcsönzésszámnak megfelelően. A diagram szemlélteti, hogy a kölcsönzési folyamat során hogyan változik az objektum állapota a kölcsönzött könyvek számának függvényében.



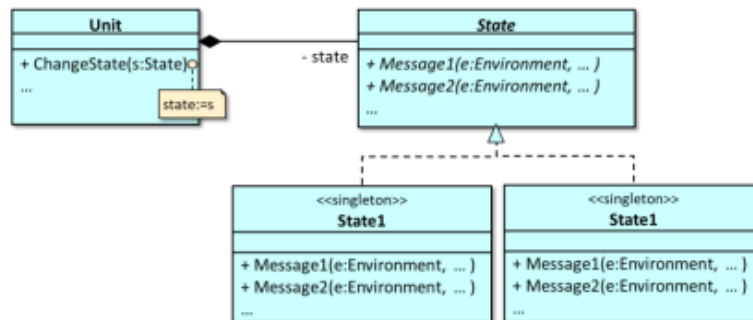
# Osztály diagram:



## Tervezési minták:

### Állapot (State – viselkedési tervezési minta)

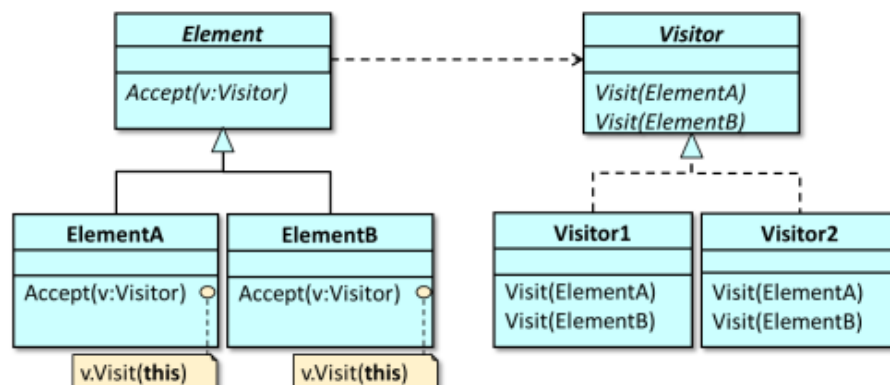
Amikor egy objektum metódusának az objektum állapotától függően kell viselkednie, akkor ahelyett, hogy ezen metódusba az objektum állapotától függő elágazást építenénk, a viselkedést kiszervezzük az objektum állapotait megtestesítő közös interfésszel rendelkező (többnyire egyke) objektumokba. A közös interfésze specifikálja az állapotfüggő metódusokat, míg az eredeti (Unit típusú) objektum rendelkezik az állapotváltoztató metódusokkal.



### Látogató (Visitor – viselkedési tervezési minta)

Amikor két objektum család egy-egy objektuma közötti interakcióra kerül sor, akkor ezt a működést a látogató tervminta alapján érdemes szervezni. Az `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.<sup>2</sup>

Megfigyelhető, hogy az `Accept()` metódusok törzse alakilag azonos, de eltér bennük a `this` típusa. Ezért amikor sor kerül a `v.Visit(this)` hívására, akkor a `Visit()` több változata közül az fut majd le, amelyet egyfelől a `v` paraméternek konkrét osztálya, másfelől a paraméteréről adott `this` konkrét osztálya határoz meg.

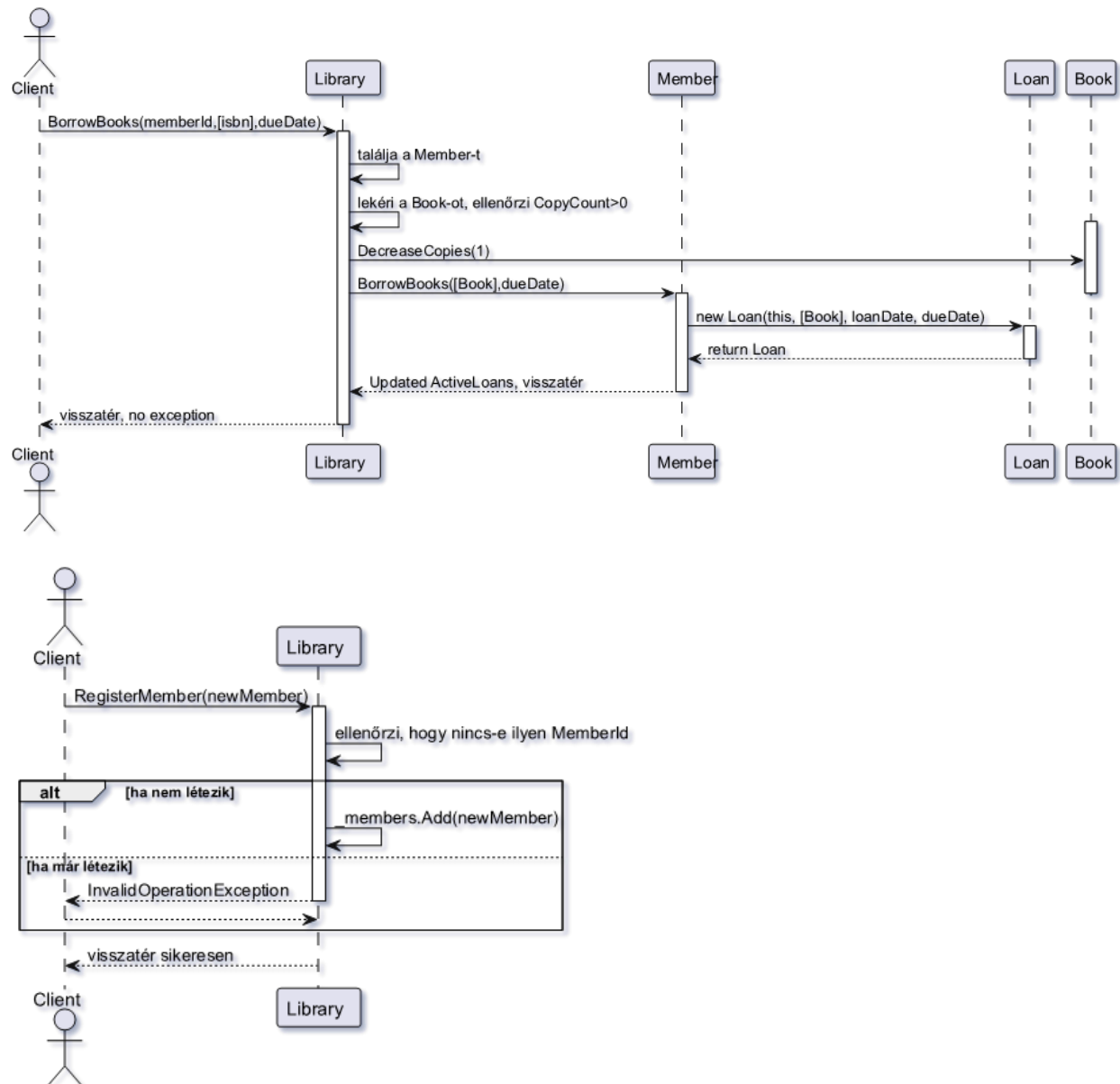




# Tesztelés

## Szekvencia diagram:

A szekvencia-diagram a könyvkölcsönzés folyamatát szemlélteti egy könyvtári rendszerben. A folyamatot a Client indítja a BorrowBooks művelettel, amely során a Library először azonosítja a tagot (Member), majd lekéri a kívánt könyvet (Book) és ellenőrzi, hogy van-e elérhető példány. Ha igen, csökkenti a példányszámot, majd a kölcsönzési folyamat továbbhalad: a Library meghívja a Member objektumon a BorrowBooks műveletet, amely létrehoz egy új Loan objektumot a kölcsönzött könyvre és a határidőre vonatkozóan. Ezután a kölcsön objektum visszatér a Memberhez, a Library frissíti az aktív kölcsönzések listáját, és visszajelzést küld a Client számára a sikeres kölcsönzésről. A diagram lépésről lépésre mutatja be az objektumok közötti üzenetküldéseket és a kölcsönzési folyamat logikai sorrendjét.



## **Tesztelési terv:**

1. Könyv hozzáadás
  - a. Új könyv bevétele egy példánnyal
  - b. Új könyv bevétele több példánnyal
  - c. Már meglévő könyv példányszám-növelése
  - d. Hibás mennyiség
2. Könyv eltávolítása
  - a. Példányszám csökkentése, maradnak példányok
  - b. Teljes törlés
  - c. Eltávolítás kölcsönzött könyvnél
  - d. Hibás mennyiség
  - e. Nem létező ISBN
3. Tagok
  - a. Új tag regisztrálása
  - b. Ismételt regisztrálás
  - c. Törlés tagság nélkül
  - d. Törlés aktív kölcsönzéssel / tartozással
4. Tagság megújítása és díjfizetés
  - a. Lejárt tagság befizetése
  - b. Még érvényes tagság befizetése
5. Kölcsönzés
  - a. Egy könyv kölcsönzése
  - b. Több könyv egy eseményben
  - c. Könyvkorlát túllépése
  - d. Példányhiány
  - e. Helytelen memberId vagy isbn
6. Visszahozás
  - a. Részleges visszahozás
  - b. Teljes visszahozás
  - c. Nem kölcsönzött könyv visszahozása
  - d. Helytelen memberId vagy isbn
7. CalculateFine() és OutstandingFine
  - a. Lejárat előtti visszahozás
  - b. Késedelmes visszahozás
  - c. Még be nem hozott, de lejárt kölcsönzés

8. PayFine()

- a. Fizetendő pótdíj befizetése
- b. Fizetés nélkül pótdíj nélkül