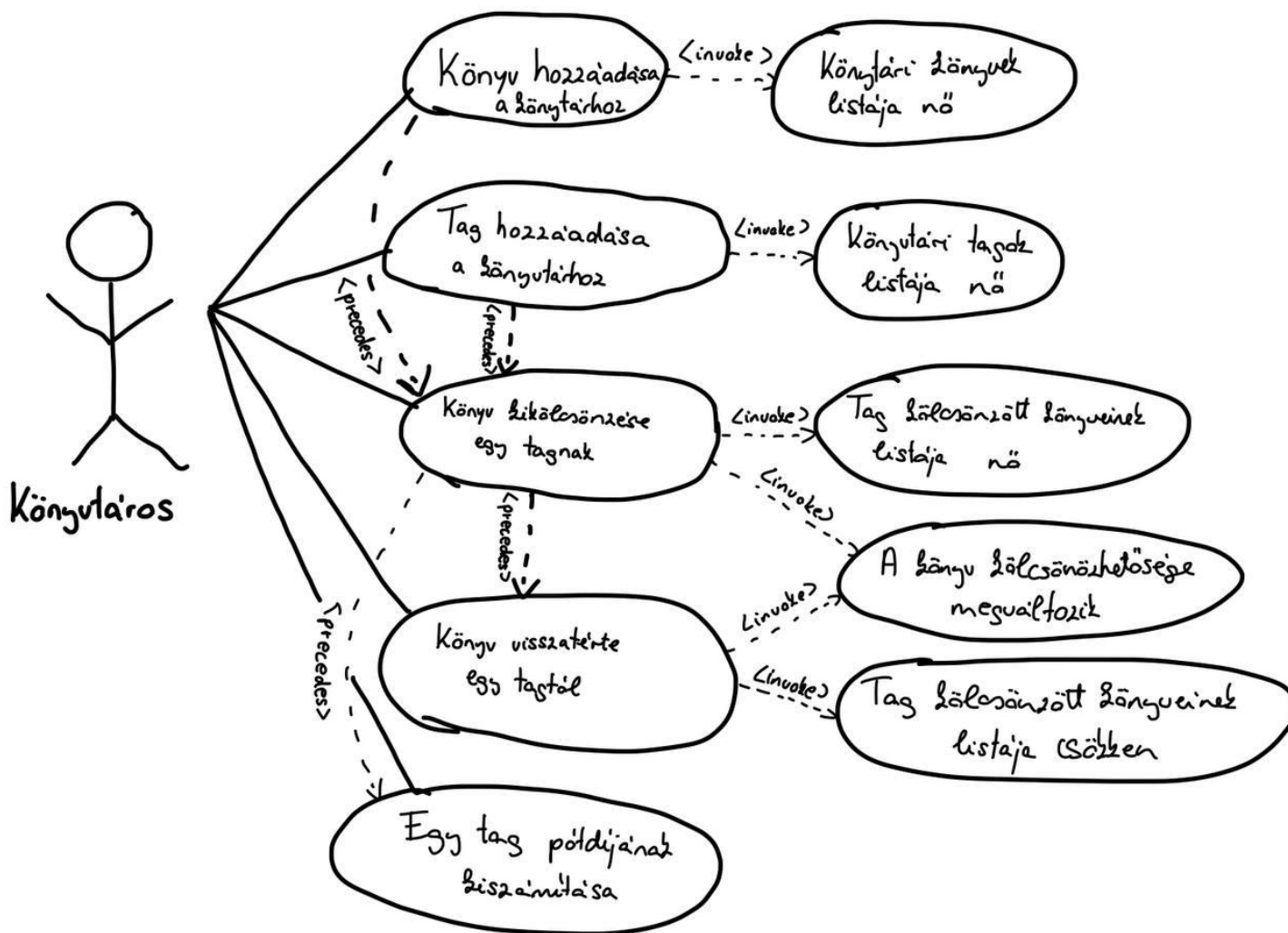


BWAF4V Márton Milán 5. Feladat : Könyvtár modellezés

Kezdeném azzal, hogy mi a volt a terv: Én elolvasva a feladatot és átgondolva, hogy való életben, hogyan is működhetne egy ilyen program, arra a következtetésre jutottam, hogy úgy oldom meg a feladatot, mintha a programomat a jövőben a könyvtárosok használnák. Ez azt jelenti, hogy a könyvtáros kezeli a programot, éppen ezért képes (mint actor) felvenni új könyveket a könyvtárba, valamint regisztrálni a tagokat, akik szimplán csak bediktálják neki a nevüket, mikor éppen be akarnak regisztrálni.

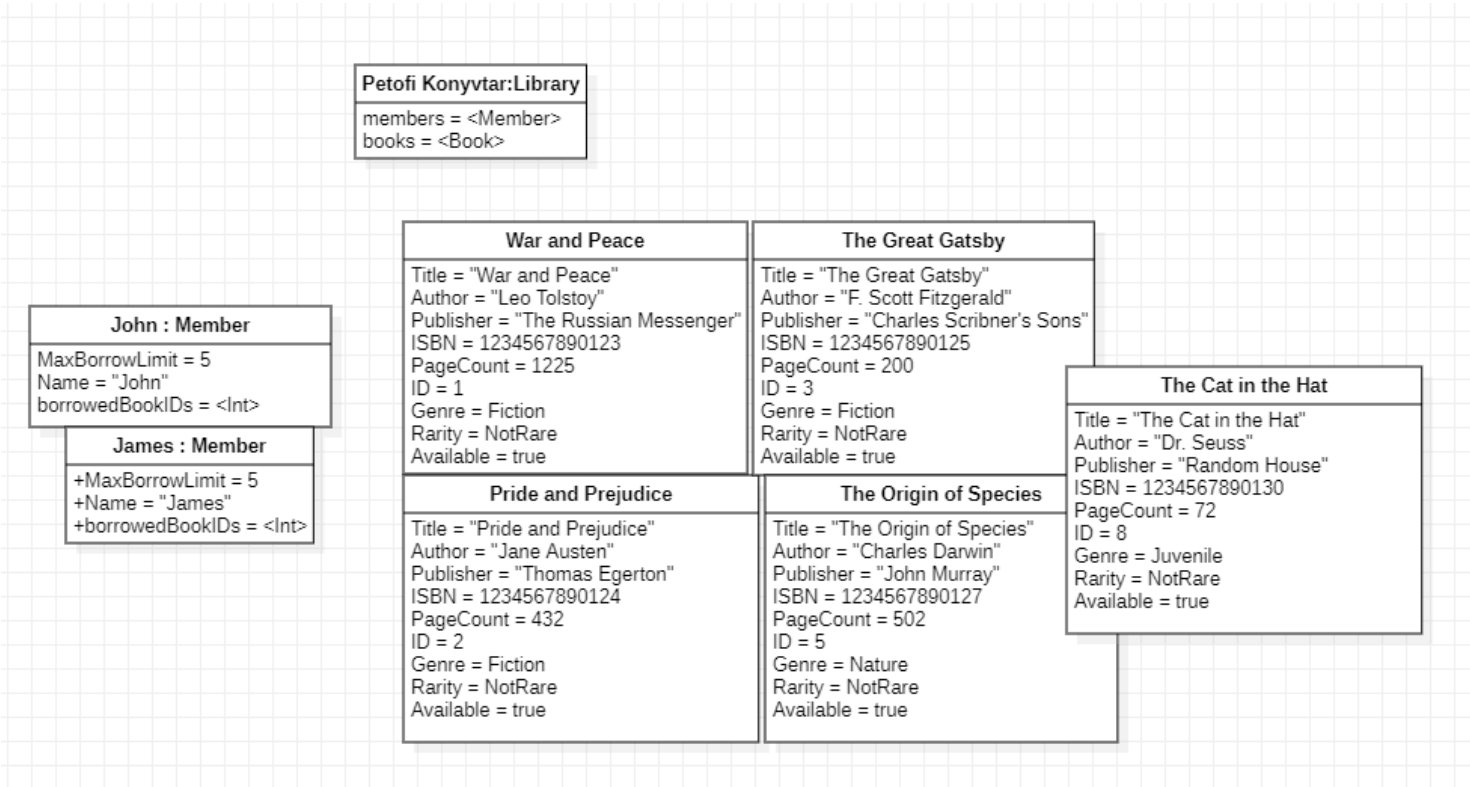
A tárgyakat is ő kölcsönözteti ki a könyvtár tagjainak (könyvtárban a tag levesz egy könyvet a polcról, majd mikor megy ki a könyvtárból kell odamenni egy pulthoz, hogy ezt te ki szeretnéd kölcsönözni) ezt a kikölcsönző neve és a könyv könyvtári azonosítója(ID) alapján teszi meg. Azért választottam azt hogy a könyveket ID alapján lehessen kikölcsönözni(nem cím szerint, vagy más módon), mivel ez az egyetlen biztosan egyedi adattagja a könyvnek. (az ISBN majdnem teljesen egyedi de lehet a könyvtárban egy könyvnek több példánya is és nem mindegy, hogy melyiket vitték el). Metódust nem írtam arra, hogy címből meg lehessen találni a könyveket, mivel mikor a tag odaviszi fizikailag a könyvet a pulthoz, akkor a könyvtáros vagy le fogja olvasni róla az ID-t szemmel, vagy bescanneli azt, így a programban nem kell ezzel foglalkozni. A könyvek lejáratát idejét is beállíthatóan adtam meg, mivel lehetséges, hogy nem minden könyvet lehet ugyanannyi ideig kikölcsönözni (nem írt a feladat erről semmit). Az 'egyszerre kikölcsönzött könyveket' én úgy értelmeztem, hogy időben egyszerre kölcsönözni ki őket a tag, de mivel a könyvtáros egyszerre csak 1 könyvet tud bescannelni (+ megadnia lejáratát idejét), ezért nem csináltam külön metódust arra, hogy tényleges egyszerre lehessen kikölcsönözni könyveket, hanem csak rögtön egymás után. Mindennek tudatában jönnek a diagramok.

Használati eset diagram



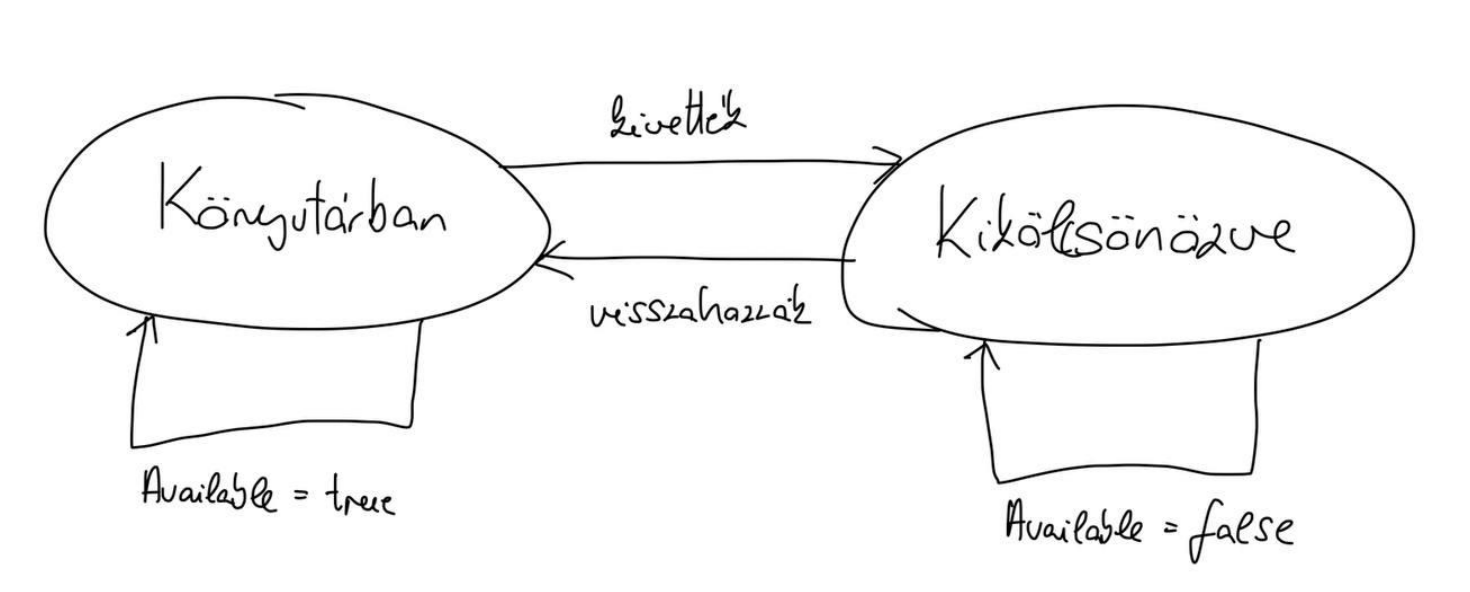
Mint jól látszik az Actor szerepét a könyvtáros tölti be.

Objektum diagram

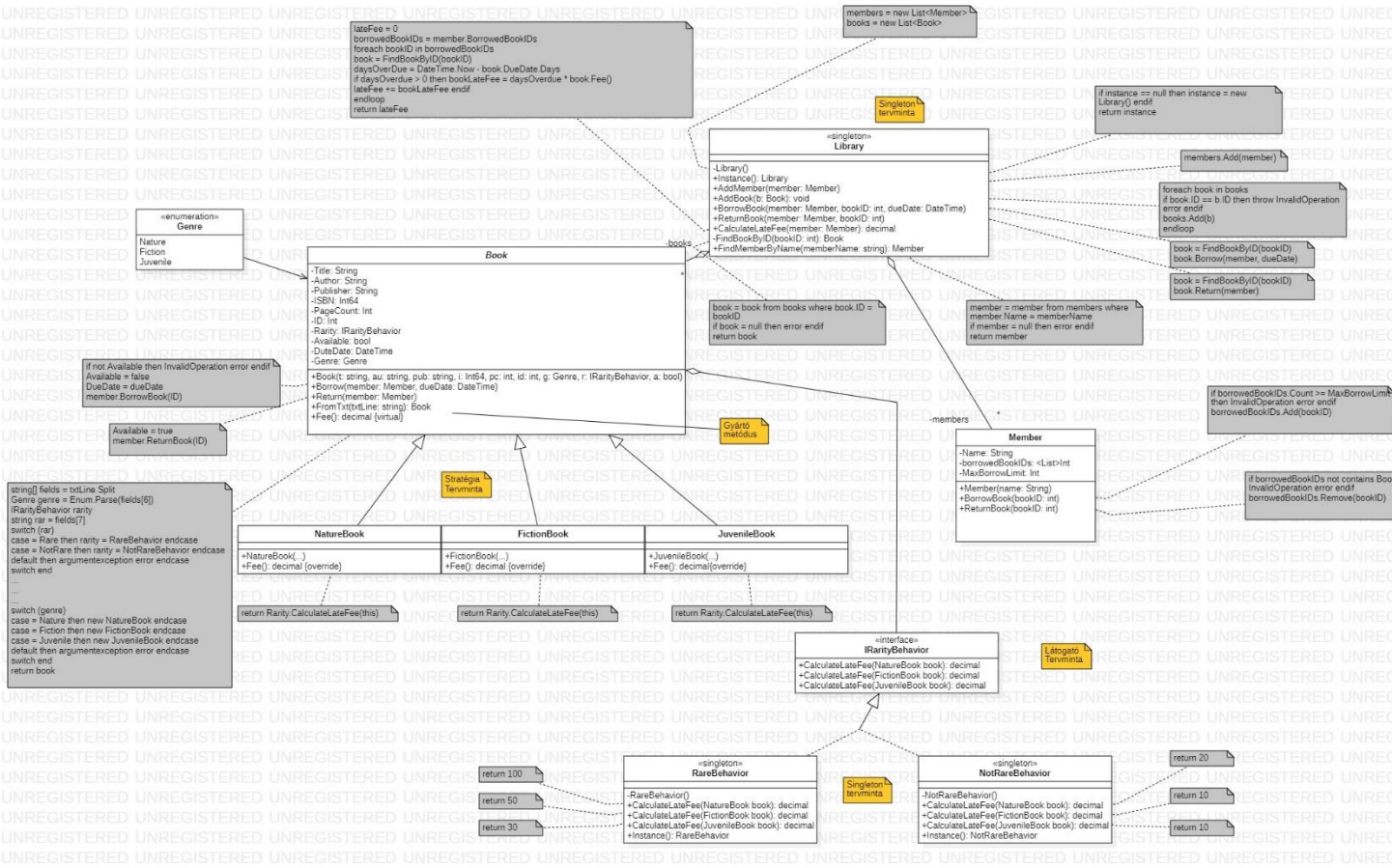


Az én megoldásomban a kölcsönzési tevékenységek nem külön objektumok, így azokat nem tudom itt ábrázolni.

Könyv állapotgépe



Osztálydiagram



Szöveges állomány

AWar and Peace,Leo Tolstoy,The Russian Messenger,1234567890123,1225,1,Fiction,NotRare,true
APride and Prejudice,Jane Austen,Thomas Egerton,1234567890124,432,2,Fiction,NotRare,true
AThe Great Gatsby,F. Scott Fitzgerald,Charles Scribner's Sons,1234567890125,200,3,Fiction,NotRare,true
M
John Doe
M
Jane Doe
B
John Doe
1
-7
B
Jane Doe
2
-14
C
John Doe
C
Jane Doe
R
John Doe

1
R
Jane Doe
2
AThe Great Gatsby,F. Scott Fitzgerald,Charles Scribner's Sons,1234567890125,200,3,Fiction,NotRare,true
B
Jane Doe
2
5
R
John Doe
2

Tesztesetek

1. **Test_AddBook:** Teszteli, hogy a rendszer kivételt dob-e, ha megpróbálunk ugyanazt a könyvet kétszer hozzáadni a könyvtárhoz.
2. **Test_BorrowBook:** Teszteli, hogy egy tag sikeresen kölcsönözhet-e egy könyvet a könyvtárból, és hogy a könyv státusza megváltozik-e nem elérhetőre.
3. **Test_ReturnBook:** Teszteli, hogy egy tag sikeresen visszaadhat-e egy könyvet a könyvtárnak, és hogy a könyv státusza megváltozik-e elérhetőre.
4. **Test_CalculateLateFee:** Teszteli, hogy a rendszer helyesen számítja-e ki a késedelmi díjat egy lejárt kölcsönzésű könyv esetén.
5. **Test_RarityBehavior:** Teszteli, hogy a ritka és nem ritka könyvek esetében helyesen számolódik-e a késedelmi díj.
6. **Test_BorrowUnavailableBook:** Teszteli, hogy kivételt dob-e a rendszer, ha valaki megpróbál kölcsönözni egy már kölcsönzött, nem elérhető könyvet.
7. **Test_ReturnNotBorrowedBook:** Teszteli, hogy kivételt dob-e a rendszer, ha valaki olyan könyvet próbál visszaadni, amit nem is kölcsönzött.
8. **Test_MemberExceedBorrowLimit:** Teszteli, hogy kivételt dob-e a rendszer, ha egy tag túllépi a maximálisan kölcsönözhető könyvek számát.
9. **Test_FindNonExistentBookByID:** Teszteli, hogy kivételt dob-e a rendszer, ha egy nem létező könyv azonosítóval próbálunk könyvet kölcsönözni a könyvtárban.

5. Modellezzük egy **könyvtár** működését!

A könyvtár nyilvántartja a könyvtárba beiratkozott személyeket, és a kölcsönözhető könyveit. Könyvtári tag az a személy lehet (ismerjük a nevét), aki beiratkozik a könyvtárba.

Egy könyvtári tag egy alkalommal legfeljebb öt, a könyvtárban meglévő könyvet kölcsönözhet ki. Egy könyvnek ismert a címe, szerzője, kiadója, ISBN száma, az oldalszáma, és van egy könyvtári azonosítója, miután a könyvtárba kerül. Az egyszerre kikölcsönzött könyveket több részletben is vissza lehet hozni, így egy kölcsönzés eseményhez tartozó könyvek listája folyamatosan csökkenhet.

Egy könyv kölcsönzési pótdíja a kölcsönzés lejáratí idejétől számított napok számától függ, de az egy napi pótdíj a könyv műfajától (természettudományi, szépirodalmi, ifjúsági) és példányszámát jellemző kategóriától (ritkaság, sok példány) függ.

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

- Tegye lehetővé, hogy a könyvtár beszerezhesen egy könyvet, egy új személy be tudjon iratkozni, egy tag kikölcsönözhesse az általa kért könyvek közül azokat, amelyek jelenleg elérhetők, és bármikor visszahozhasson egy kikölcsönzött könyvet.
- Mennyi pótdíjat kell fizetnie egy tagnak a vissza nem hozott könyvei után?

Készítsen használati eset diagramot! Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutat öt könyvet, két könyvtári tagot, hozzájuk kapcsolható három kölcsönzési tevékenységet, ahol az egyik kölcsönzés egyszerre két könyvet is tartalmaz.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy könyv objektum állapotgépét! Különböztesse meg a „könyvtárban”, és a „kikölcsönözve” állapotokat. Az állapot-átmeneteket megvalósító tevékenységeket majd a könyv osztály metódusaiként definiálhatja.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég „...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni egy könyvtár könyveit, könyvtári tagjait, néhány kölcsönzést és könyv visszahozást. Számoljuk ki egy tag pótdíját! Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!