

readme

Könyvtár

Feladat

Egy könyvtár nyilvántartja a könyvtárba beiratkozott személyeket, és a kikölcsönözhető 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

1. Tegye lehetővé, hogy a könyvtár beszerezhessen egy könyvet, egy új személy be tudjon iratkozni, egy tag kikölcsönözhesse az általa kért könyvek közül azokat, amelyek jelenleg elérhetőek, és bármikor visszahozhasson egy kikölcsönzött könyvet.

2. 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 a paraméterek alapján, 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 modelljét! 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**!

Jegyzet

A kölcsönzési tevékenységet nem láttam effektív módjának a megvalósításnak, de ehhez hasonlóan létrehoztam egy `MemberDTO` ([Data Transfer Object](#))-t amely segített a bejelentkezést és a könyv kölcsönzést biztonságossá tenni, mert ez a DTO eltárolja a member-t és egy egyedi ID-t amit csak a könyvtár és a member ismer, (kívülről nem elérhető)

Osztály diagramok

[Összes osztály megtekintése böngészőben](#)

[Követelmény szerinti osztálydiagram megtekintése](#)

Egyéb diagramok

- [Link a használati eset diagramhoz](#)
- [Link a objectum diagramhoz](#)
- [Link a szekvenciális diagramhoz](#)
- [Link a állapotgép diagramhoz](#)

Alkalmazott tervezési minták

- Abstract Factory
- Template Method
- Strategy pattern
- Dependency Inversion
- **SOLID** elvek a liskov elv nélkül

Fontos linkek

- [Diagram struktúra megtegmentése GitHub-on](#)
- [Ez a README legfrisseb változata GitHub-on](#)