

Programozói dokumentáció

Könyvtárkezelő program

Osztályok és adatszerkezetek bemutatása

Az osztályok és a metódusok dokumentációja [ide](#) kattintva elérhető. Az osztálydiagram a dokumentum végén található (teljes méretben megtekinthető [itt](#)).

- **ApplicationFrame:** JFrame-ből öröklődik, a program főablaka. Beállítja a rajta lévő komponenseket és megjeleníti őket. Egyszerűbb listenereket tartalmaz, események fellépésekor (pl. gombra kattintás) meghívja a program szükséges függvényét.
- **Book:** Egy könyvet megvalósító osztály a programban, tárolja egy könyv adatait.
- **BookCategory:** Egy könyv lehetséges kategóriáit tartalmazó enum. Lehetőség van az enumok magyar megfelelőjét String-ként lekérdezni tőle, illetve a magyar megfelelőt visszaalakítani enum értéké.
- **BookData:** A főképernyőn megjelenő könyveket tartalmazó táblázat modellje.
- **BookPanel:** Egy könyv adatainak megadását lehetővé tevő panelt hoz létre, lekérdezhetők tőle a felhasználó által megadott értékek.
- **BookSearchBar:** Megvalósítja a kereséshez használt szövegmezőt, a JTextField-ből öröklődik. A mező fókuszbba kerülése vagy a fókuszt elvesztése esetén automatikusan frissíti a szövegmező tartalmát.
- **BorrowData:** Az aktuális kölcsönzéseket megjelenítő fa struktúra modellje.
- **BorrowManager:** Egy könyv kölcsönzéssel kapcsolatos tulajdonságainak (kölcsönözhető-e, ki kölcsönözte ki) megváltozását kezeli. Ha a kölcsönző visszahozza a könyvet a könyvtárba, az általa elvitt könyvek listájából törlődik a könyv, és a könyv borrowedBy adataja ismét null lesz.
- **Library:** A program adatait összefogó osztály, kezeli a szerializációt és a deszerializációt.
- **Member:** Egy könyvtári tagot reprezentáló osztály.
- **MemberData:** A tagokat tartalmazó táblázat modellje.
- **MemberPanel:** Egy tag adatainak megadását lehetővé tevő panelt hoz létre, lekérdezhetők tőle a felhasználó által megadott értékek.
- **ValidityChecker:** Megvizsgálja, hogy a felhasználó által megadott adatok megfelelnek-e az elvárt formátumnak egy könyv vagy tag adatainak megadása során (pl. a név mező nem üres; a telefonszám mező vagy csak számokat tartalmaz, vagy üres; kiadási év mezője csak számokat tartalmaz).

A programban használt adatszerkezetek (pl. a könyvek- és tagok adatait tárolók) ArrayList-et használnak. Azért ezt a típust választottam, mert egyszerű lineáris adatstruktúrára volt szükségem, és erre a szabványos ArrayList bizonyult a legmegfelelőbbnek. Az esetleges későbbi módosításokat szem előtt tartva az ArrayList referenciát tároló tagváltozók List interfész típusúak.

Use case-ek

- Tagok kezelése
 - A felhasználó megadja egy személy adatait, majd eltárolja a programban

- Ha a felhasználó hibás adatokat ad meg (pl. nem ad címet a könyvnek), akkor a program ezt hibaüzenetben jelzi és a tag nem kerül hozzáadásra
- A hozzáadott tagok megjelennek a program főképernyőjén található táblázatban és fa struktúrában
- A tagokat tartalmazó táblázat egy sorára kétszer kattintva a felhasználó szerkesztheti egy tag adatait
- A tagokat tartalmazó táblázat egy sorát kijelölve, majd az eltávolítás gombot megnyomva a felhasználó eltávolíthat egy tagot a programból
- Ha a felhasználó olyan tagot szeretne eltávolítani, akinél van kikölcsönzött könyv, és a felhasználó megerősíti a törlési szándékát, akkor a program az adott könyvet újra kölcsönözhetővé teszi
- Ha egy tag kölcsönzött könyvei megváltoznak, frissítésre kerül az aktuális kölcsönzéseket megjelenítő fa modell
- Könyvek kezelése
 - A felhasználó megadja egy könyv adatait, majd eltávolítja a programban
 - Ha a felhasználó hibás adatokat ad meg (pl. nem ad címet a könyvnek), akkor a program ezt hibaüzenetben jelzi és a könyv nem kerül hozzáadásra
 - A hozzáadott könyvek megjelennek a program főképernyőjén található, könyvek adatait tartalmazó táblázatban
 - A könyvek adatait a felhasználó közvetlenül a táblázatban szerkesztheti
 - Hibás adat megadása esetén a módosítások nem kerülnek mentésre, megmaradnak a könyv korábbi adatai
 - A könyveket tartalmazó táblázat egy sorát kijelölve és a Könyv eltávolítása gombot megnyomva a kijelölt könyv törölődik a programból
 - A könyv kölcsönzéséhez a felhasználó a könyv adatait tartalmazó sor utolsó oszlopában található legördülő menü segítségével adhatja meg, hogy melyik tag kölcsönözte ki
 - A kölcsönzés megszüntetéséhez a legördülő menü első, üres elemét kell választani vagy a Könyv eltávolítása gomb melletti Könyv visszavétele gombra kell kattintani
- Fájlkezelés
 - A program bezárásakor az adatok automatikusan mentésre kerülnek
 - A felhasználónak lehetősége van korábban mentett fájlok megnyitására a Fájl menü Megnyitás opcióját választva. Ekkor a választott fájl tartalma megjelenik a programban
 - A felhasználó megadhat egy, az alapértelmezettől eltérő célkönyvtárat a Fájl menü Mentés másként... lehetőségét választva. Ezt követően a módosítások mentése az így megadott helyre történik
 - Amennyiben a fájlműveletek során a program hibára fut (pl. sérült a fájl), azt hibaüzenettel jelzi

Fájlkezelés

A program a fájlok létrehozásához a Serializable interfészt használja. A program alapértelmezetten a futtatási könyvtárba ment egy library.libdat fájlt, amit induláskor automatikusan vissza is olvas. Ha ez nem található, hibaüzenetet ad, és egy üres könyvtárat nyit. Ha a felhasználó ezt nem írja felül, akkor a módosítások mentése ebben az esetben is a library.libdat fájlba történik. A felhasználónak lehetősége van megadni a mentés helyét a Fájl>Mentés másként... opciót választva, ekkor mentéskor a megadott helyre

íródnak az adatok. Az így mentett fájlok visszaolvasásához a Fájl menü Megnyitás lehetőségét kell választani. Amennyiben a program hibát észlel a fájlműveletek során, akkor azt hibaüzenettel jelzi.

Változtatások a specifikációhoz képest

A legnagyobb eltérés a specifikációtól, hogy a program Author objektum helyett egy könyvhöz csak a szerző nevét rendeli hozzá Stringként. A változtatást legfőképp az indokolta, hogy a program használata során egy-egy könyv adatainak megadását jelentősen lassította ez a módszer, ha az adott szerzőt még nem vettük fel a programba. Egy másik hátráltató tényező volt, hogy egy szerzőről nem minden esetben állnak rendelkezésre a program által várt személyes adatok: például egy-egy kiadvány szerzőjéről nem feltétlenül tudjuk, hogy mikor született, ez az adat leginkább híres írók, költők esetében releváns.

A fent ismertetett okok miatt végül a szerző osztály elhagyása mellett döntöttem, helyette készítettem el az aktuális kölcsönzéseket nyilvántartó fa struktúrát.

Egy másik eltérés pedig a kölcsönzés menete. Eredetileg úgy terveztem, hogy a kölcsönzés adatait egy külön párbeszédablakban lehet majd megadni, de a program írásának kezdetekor jobban utánanéztem a JTable által biztosított lehetőségeknek, és úgy döntöttem, hogy a táblázatban egy legördülő menü segítségével a kölcsönző módosítása kényelmesebb, mint egy gombra kattintást követően az adatok begépelése.

