

## Dokumentáció:

A programom egy C++-ban írt program, ami egy filmeket tároló rendszert valósít meg. Konzolos megjelenésű, és egy egyszerű menürendszer van hozzá készítve. Képes a filmeket tároló listát kiírni, rá lehet keresni egy konkrét filmre, ha beírjuk a pontos címét, ki lehet listázni szűrve a filmeket cím és évjárat szerint, itt már megengedett a \*-os jelölés, ha csak egy részletet tudunk egy filmből. 4 fajta filmet különböztet meg: családi, dokumentum, fantasy és vígjáték. Minden filmnek vagy egy jellemzője. A családi filmekhez tartozik egy korhatár, a dokumentumfilmekhez, egy kis leírás, a fantasy-hez be lehet jelölni, hogy az adott film egy sci-fi film e illetve a vígjátékoknál egy típus jelző van megadva, ami elég kötetlen, olyanokra kell gondolni, hogy mondjuk játszik e a filmben valami híres színész, vagy egy híres rendező rendezte, vagy csak van valami jellemző motívuma. Természetesen a program képes a listát fájlba írni, és onnan beolvasni.

A program 7 osztályra van bontva. Mind egy-egy fontos különálló részét vezérlik. A következőkben leírom az egyes osztályok szerepét illetve, hogy milyen függvények találhatók benne.

Az ősosztály a Film osztály. Ez egy absztrakt osztály hiszen, van benne tisztán virtuális függvény. Ez felelős a filmek létrehozásáért és kiírásért. Főbb függvényei:

Getterek, amikkel hozzá lehet férni a címhez, kiadási évhez és a hosszhoz, ez a kiírás miatt fontos. Ezeken kívül van egy **virtual ostream& print(ostream& os) = 0;** függvény, ami egy paraméterként megadott ostreamre ír ki. Természetesen referenciaként veszi át, hogy lehessen változtatni. Ezen kívül van egy hasonló **virtual void fajlba\_ir(ofstream& os) = 0;** függvény, ami egy paraméterként megadott ofstreamre ír ki, azért kellett külön függvény, hiszen teljesen más formátumba tárolom a fájlban a filmeket, mint, ahogy ki szeretném írni.

Ennek az ősosztálynak leszármazottjai a Családi, a Fantasy, a Vígjáték és a Dokumentum. Ezeknek a hozzájuk rendelt plusz infó az egyetlen adattagjuk. A függvényeik a virtuális függvények megvalósítása illetve csupán egy konstruktor, aminek meg kell adni az összes adatot, és az meghívja először az ősosztály konstruktorát és így felveszi az adatokat.

Fontos osztály a Filmtarolo, ez valósítja meg a heterogén kollekciót, aminek a segítségével tárolom a filmeket. Adattagjai a db és a filmek. Az előbbi egy size\_t típusú a másik egy Film\*\* típusú adat. Konstruktor létrehoz egy üres filmek tömböt, aminek a db-ja 0 és a filmek tömbben egy NULL pointer van. Ez a memóriaszivárgások elkerülése végett kell, hogy így legyen. Függvényei a következők:

**void felvesz (Film\* ezt);**

Ennek a segítségével a megadott film pointert fel tudom venni a tárolóba.

**void listaz (ostream& os = cout);**

Ennek a segítségével a megadott ostreamre kiírja az összes filmet, ami a tárolóban van.

**void torol (Film\* ezt);**

Törli a megadott Film\* típusú filmet a tárolóból.

**Film\* keres (Film\* ezt); illetve Film\* keres (string cim);**

Ezekkel a függvényekkel lehet keresni a tárolóban, csak pontos egyezés esetén ad vissza bármit, ha nincs egyezés akkor hibát dob.

**void egesztorles ();**

Törli az egész tárolót.

**size\_t dbszam ();**

Getter a dbszámra.

**void szurveListaz (string szuro, bool (\*valamilyen)(string, string)); illetve void szurveListaz (int szuro, bool (\*valamilyen)(int, int));**

Ezekkel a függvényekkel képes a program kiírni csak az olyan típusú filmeket, amiket szeretnénk. Overloadolt függvény, működik címre és kiadási évre is. Szükséges egy függvény még, ami elvégzi az összehasonlítást. Ezek a Filmtar.cpp-ben vannak leírva, egy egyszerű ev\_eq, ami kettő intet hasonlít össze és egy benne\_van függvény, ami megkeresi a \*-ot a szóban, ha van, és ha van akkor .find függvénnyel megkeresi az egyezést, ha nincs akkor csak == operátorral hasonlítja össze a kettő megadott stringet.

**bool bennevan (Film\* ez);**

Visszatér igazgal, ha a pointer benne van a tárolóban.

Illetve megtalálható kettő operator[], egy konstans és egy sima. Természetesen a tároló i. elemével tér vissza.

A Fajlkezelő osztály felelős a programban szereplő összes fájlkezelésért. Adattagjai egy ifstream, egy ofstream és egy string, ami az elérési utat tárolja. Így egy Fajlkezelő objektum egy fájlért felel, ami fontos szempont. Két fő függvénye van:

**void fajlbair (Filmtar& tarolo);**

Ez valósítja meg a fájlba írást. Minden alkalommal először kitörli a fájlt, majd létrehozza. Ez azért fontos, hogy teljesen üres legyen a fájl, amibe írni fog. Az íráshoz segítenek a Film osztály fajlba\_ir függvényei. Meg kell adni egy tarolót természetesen, hogy az abban lévő adatokat írja a fájlba.

**void Fajlkezelő::fajlbolOlvas (Filmtar& tarolo);**

Ennek a segítségével olvassa be a program a fájlból az adatokat. A tárolás módja a következő: az első adat mindig egy int, ami a db-számot jelzi az algoritmusnak. Ez után jönnek adott sorrendben az adatok, ;-vel elválasztva, elsőnek a típus.

A menü rendszer egy egyszerű do while ciklus, ami a megadott számok alapján végzi el a feladatot.

A tesztelés esetén minden fontos függvényt teszteltem a gtest\_lite segítségével. A memóriahibákat a memtrace segítségével ellenőriztem, ami nem mutatott semmi rendelleneset.