

Filmtár dokumentáció

Jeges Gábor

1. feladat

A feladat, amit választottam, a filmtár. A honlapról:

„Készítsen filmeket nyilvántartó rendszert. Minden filmnek tároljuk a címét, lejátszási idejét és kiadási évét. A családi filmek esetében korhatár is van, a dokumentumfilmek esetében egy szöveges leírást is tárolunk. Tervezzon könnyen bővíthető objektummodellt a feladathoz!

Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz ne használjon STL tárolót!”

2. pontosított feladatspecifikáció

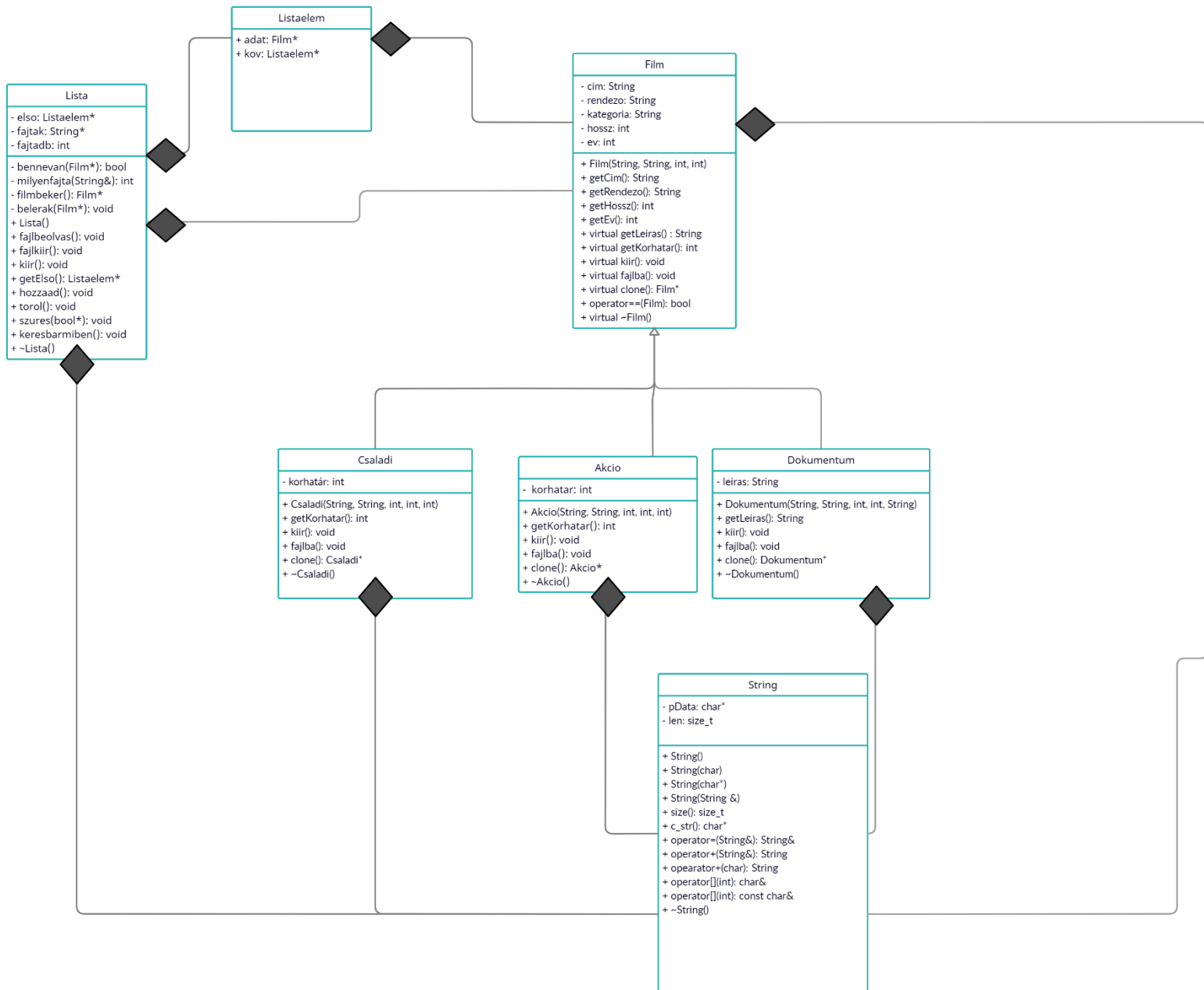
A programom egy adatbázist fog megvalósítani, melyből lehet törölni, új elemet hozzáadni, bármilyen szempont alapján szűrni (kiadás éve, korhatár, rendező, kategória stb.). Bárhol jellegű keresés is (tehát a "lad" keresésre kihozza az összes csaLADi filmet és "A hihetetlen csaLAD" c. animációs filmet is). A felhasználó egy parancssoros felületet fog maga előtt látni, ahol a megfelelő karakter beütésével választhat a menüpontok között. A program nem fog kezelni ékezetes betűket

3. terv

Az adatok egy láncolt listában lesznek eltárolva, indulásakor ezt fájlból olvassa be, bezáráskor abba menti el. A feladathoz használni fogom a laboron elkészített String osztályt kisebb módosításokkal (pl. sort olvasson be, ne szóközöig). A lista osztálya tartalmazni fog egy segéd tömböt a kategóriák nevéből, ez majd a beolvasásnál lesz hasznos, hogy a program tudja, hogy milyen osztályból kell létrehozni egy példányt. A fájlban a kategória lesz mindig az első adat egy filmről, majd ezt követi a többi adat, 1-1 sorban. A benne van(Film&) függvényt használni fogja a hozzáadó és a törölő függvény is.

A szűrésnél a felhasználó megadja a szempontot mi alapján akar, majd ezután a program felépít egy új listát, ahol ismét lehet további adatok

alapján szűrni, ha kilép a felhasználó belőle akkor a program a végső listát írja ki. A mindenben keresés egy kapott szöveget néz meg egy film összes adatában és ha benne van, akkor kiírja azt.



4. megvalósítás

String osztály:

Karaktorsorozatot és annak hosszát tároló osztály. Fájlok: string5.h, string5.cpp. A laborfeladatokban megcsinált osztály apró módosításokkal.

4 konstruktora van: paraméter nélküli, charból C-stringből (const char*)

Stringet létrehozó, valamint másoló konstruktor.

size(): A String hosszát adja vissza.

c_str(): A String által tárolt karaktereket adja vissza C-stringként.

operator =, +, [], == : értékadó, összeadó, indexelő és egyenlőség operátorok

operator char*(): C-string konverzó, hogy a <cstring> függvényeihez lehessen használni a Stringet.

operator << és >>: ostreamre kiíró és istreamről egy sort beolvasó oprátorok.

Film osztály:

Absztrakt ősosztály, melynek a leszármazottjai tárolják a filmek adatait.

Fájlok: film.h, film.cpp.

getCim(), getRendezo(), getKategoria(), getHossz(), getEv(): Az adattagok elérését szolgáló függvények, melyek az adat típusának és értékének megfelelő visszatéréssel rendelkeznek.

getLeiras(), getKorhatar(): virtuális függvények, melyek a leszármazottak megegyező nevű függvényeik eléréséhez (Film* esetén a listából), ha nincs felülírva üres stringet, illetve 0-t adnak vissza.

kiir(), fajlba(): A film adatait képernyőre, illetve fájlba kiíró tisztán virtuális függvények.

clone(): Az objektumról másolatot készítő tisztán virtuális függvény.

operator==: Egyenlőség operátor, mivel egy filmnek lehet egyforma címe, rendezője, több változat esetén a hossz és az év változhat, így akkor lesz egyenlő két film, ha ezek az adataik mind megegyeznek.

Akcio osztály:

A Film osztály leszármazottja. Fájlok: akcio.h, akcio.cpp.

getKorhatar(): A tárolt akciófilm korhatárát adja vissza.

kiir(), fajlba(): Az akciófilm adatait képernyőre, illetve fájlba kiíró függvények.

clone(): Az akciófilmről másolatot készítő függvény, mely a másolatra mutató pointerrel tér vissza.

Csaladi osztály:

A Film osztály leszármazottja. Fájlok: családi.h, családi.cpp.

getKorhatar(): A tárolt családi film korhatárát adja vissza.

kiir(), fajlba(): A családi film adatait képernyőre, illetve fájlba kiíró függvények.

clone(): A családi filmről másolatot készítő függvény, mely a másolatra mutató pointerrel tér vissza.

Dokumentum osztály:

A Film osztály leszármazottja. Fájlok: dokumentum.h, dokumentum.cpp.

getLeiras(): A tárolt dokumentumfilm rövid leírását adja vissza.

kiir(), fajlba(): A dokumentumfilm adatait képernyőre, illetve fájlba kiíró függvények.

clone(): A dokumentumfilmről másolatot készítő függvény, mely a másolatra mutató pointerrel tér vissza.

Listaelem struktúra:

Egy következő Listaelemre mutató pointerből és a Film osztály egyik leszármazottjára mutató pointerből (Film*) áll. Fájl: lista.h.

Lista osztály:

A programom vázát adó osztály, ez tartalmazza a filmes adatbázist.

Fájlok: lista.h, lista.cpp.

Az osztály tartalmaz egy első Listaelemre mutató pointert, egy statikus fajtdb változót és egy String tömböt a lehetséges filmkategóriákat tartalmazva.

privát függvények

bennevan(Film*): A függvény kap egy filmet és visszaadja, hogy megtalálható-e a lista elemei közt.

milyenfajta(String& kat): A kapott Stringről megmondja a fajtak tömb segítségével, hogy melyik kategóriába tartozik a film. Ezt használja a program beolvasásnál (fájl és standard input is). Visszatérési érték az indexe, a String helyének, ha nincs benne kivételt dob.

filmbeker(): A standard inputról kéri be egy film adatait a felhasználótól, majd az arra mutató pointerrel tér vissza.

belerak(Film*&): Kap egy Film pointert, majd azt elhelyezi a listában. Ha már benne, kivételt dob.

publikus függvények

fajlbeolvas(): A filmek.txt fájlból beolvassa a már a rendszerben lévő filmeket és a listába helyezi azokat.

fajlkiir(): Végigmegy a listán és minden elemre meghívja a fájlba író függvényét.

kiir(): Végigmegy a listán és minden elemre meghívja a képernyőre író függvényét.

getElso(): A Lista első elemre mutató pointerét adja vissza.

hozzaad(): Bekér a felhasználótól egy filmet, majd a listába rakja. Ha benne lenne már, azt jelzi a felhasználónak.

torol(): Bekér a felhasználótól egy filmet, majd törli a listából. Ha nem lenne benne, azt jelzi a felhasználónak.

szures(bool*): A felhasználó a különböző szempontok alapján szűkítheti a Lista elemeit. Ha megadta a felhasználó kívánságát, az adott tulajdonság alapján a függvény felépít egy új Listát az előző elemeiből, majd meghívja arra is ezt. Visszalépés esetén az utolsó listát írja ki a képernyőre. A paraméterként kapott tömb azt mondja meg mi alapján lehet még a listát szűrni (például, ha már év szerint volt, akkor a megfelelő helyen hamis érték lesz és ez alapján nem engedi a felhasználónak az újbóli szűrést).

keresbarmiben(): A felhasználtól bekér egy karaktersorozatot, majd az összes film összes szöveges adatában megnézi, hogy szerepel-e ez. Ha valamelyikben igen, akkor kiírja a film adatait.

destruktor: Végigmegy a Listán és felszabadítja a lefoglalt memóriát.

Egyebek

teszt-menu.h, teszt-menu.cpp:

teszteles(): A Jportán lefutó tesztek.

felhasznalo(): Normál esetben a felhasználónak megjelenő főmenü.

main.cpp:

A CPORTA definiáltságától függően meghívja a teszteles() vagy a felhasznalo() függvényt.

5. Tesztek

teszteles() függvény a teszt-menu.cpp-ben

Az első három teszt a Film osztály leszármazottainak helyességét ellenőrzi. A 4. teszt a Film osztály == operátorát teszteli. Ezek után a különböző Lista függvényeket teszteli a program: konstruktor, fájlbeolvasás, a két kiírás, hozzáadás-törlés, szűrés és végül keresés. A kód fájljaiba „include-olva” van a memtrace.h így ellenőriztem a memóriaszivárgást, én nem tapasztaltam hibát a futtatás során.

6. Felhasználóknak

A programban a menüpontok között számok beírásával lehet navigálni. Ahol a program számot kér be, ott jelzi, ha tévesen nem azt (nem csak azt) kapott, szöveges beolvasással nincs probléma. A menükből általában a 0 a visszalépés kódja.