Házi feladat

Programozás alapjai 2.

Feladatspecifikáció

Szlovák Anna (OPOFGK)

Feladat: Filmtár

„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. Tervezzen könnyen bővíthető objektummodellt a feladathoz!

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

Specifikáció:

Az objektumorientált megközelítés érdekében a rendszer a következő osztályokat tartalmazza:

- Film: ez az alapvető osztály, amely a filmek általános tulajdonságait tartalmazza: a film címét, lejátszási idejét és kiadási évét. Az osztályban ezek az adatok protected típusúak, hiszen az öröklő osztályoknak el kell érniük. A Film nem absztrakt osztály, lehet olyan film, ami nem családi, nem dokumentumfilm, csak simán film típusú.
- CsaládiFilm: a Film osztályból származik, és kiegészítő tulajdonságot tartalmaz a családi filmekhez: a megtekintési korhatárt. (A korhatár értéke csak egész szám lehet.)
- DokumentumFilm: szintén a Film osztályból származik, és kiegészítő tulajdonsága a film leírása. Ebben az osztályban a leírás privát adattagként van tárolva, getter és setter függvényekkel lehet elérni.

Az elkészülő programmal a felhasználó az alábbi műveleteket tudja elvégezni:

- Új film hozzáadása: a felhasználó megadja a film címét, lejátszási idejét és kiadási évét. Ha a film családi film, akkor a korhatár értékét is, ha dokumentumfilm, akkor a leírást is meg kell adni.
- Film törlése: a felhasználó kiválasztja a törlendő filmet a listából, majd a program eltávolítja azt az adatbázisból.
- Listázás: a rendszer listázza az összes filmet.
- Kiválasztott film adatainak lekérése: a felhasználó kiválasztja a filmet a listából, majd a rendszer kiírja a film összes tulajdonságát.

A program használata egyszerű, standard inputról olvas be. Menüpontokkal vezérelhető, hogy éppen melyik műveletet szeretnénk elvégezni.

Az adatokat dinamikus memóriakezeléssel tárolja.

Ha nem létező adatot próbálnak lekérdezni, vagy nem megfelelő adatot próbálnak eltárolni (pl.: korhatár nem egész szám), vagy nem sikerült a memóriafoglalás, arra figyelmeztet.

Terv

1. **Film** osztály:

- A konstruktorok inicializálják a `cim`, `hossz` és `kiadasEv` adattagokat.
- A másoló konstruktor lemásolja az adattagokat a másik Film objektumból.
- A `kiir()` függvény kiírja a film adatait a konzolra.
- A `beolvas()` függvény bekéri a felhasználótól a film adatait és létrehoz egy új Film objektumot a megadott adatokkal.

2. CsaladiFilm osztály:

- Az Film osztályból származik, tehát örökli annak adattagjait és függvényeit.
- A konstruktorok inicializálják a `korhatar` adattagot.
- A `kiir()` függvény kiírja a családi film adatait, beleértve az ősosztály adatait is.
- A `beolvasCsaladi()` függvény bekéri a felhasználótól a családi film adatait és létrehoz egy új `CsaladiFilm` objektumot a megadott adatokkal. Először létrehoz egy sima Film-et, majd ezt kiegészíti CsaládiFilm-mé a korhatár hozzáadásával. Ezért van szükségem a Film beolvas()-ra.

3. DokumentumFilm osztály:

- Ugyanaz, mint a CsaladiFilm és az összes többi filmtípus, ennek leírása van.

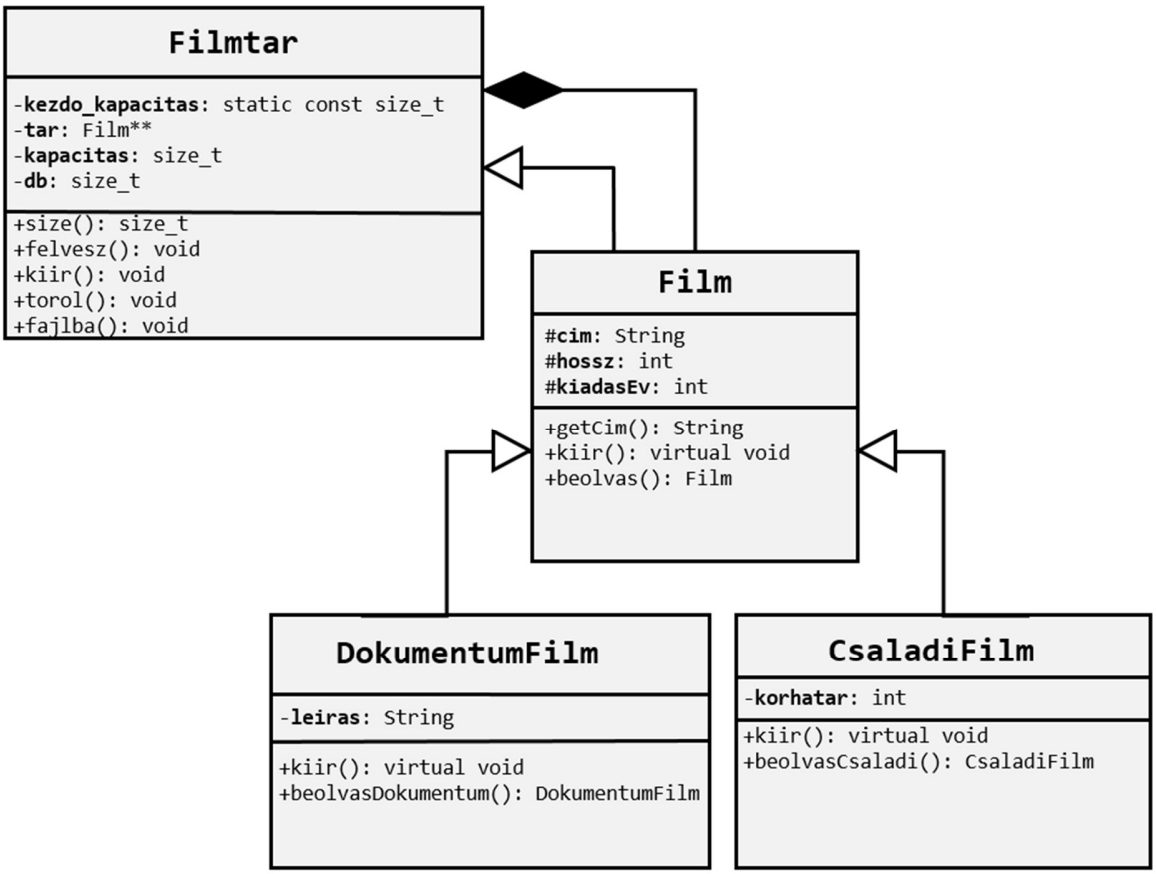
4. **Filmtar** osztály:

- A feladat legfontosabb része, a feladat lényegi megvalósítása itt történik.
- Ez az osztály többféle filmet tárolhat, ezért sablonosítva van. `Heterogén kollekcióval`, a tár alapvetően Film-eket tárol, és ha szükséges, akkor az öröklés megoldja, hogy Dokumentum- vagy CsaládiFilmet.
- A konstruktor létrehozza a `tar` pointer tömböt, inicializálja a `kapacitas` és `db` adattagokat.
- A `size()` függvény visszaadja a tárolt filmek számát.
- A `felvesz()` függvény hozzáad egy filmet a tárolóhoz. Ha a tároló tele van, akkor újat foglal 10-zel nagyobb kapacitással.
- Ennek a tárolónak is megvan a saját `kiir()` függvénye, ami egymás után kilistázza a filmeket és azok összes tulajdonságát.
- A `torol()` függvény gondoskodik a lefoglalt memóriaterület felszabadításáért.
- A fajlba() függvény a projekt mappájában lévő film.txt-be listázza a Filmtárban lévő filmeket.

5. **Tesztelés**:

A tesztelést a `gtest_lite.h`-val fogom megvalósítani, minden lehetséges függvényre kitérve. A helyes foglalásokat, kiírásokat, törlést és az indexelési hibákat is ellenőrizni fogom. A gtest mellett egy `menüvezérelt program` is készül, amivel tesztelni lehet a működést saját inputtal is.

6. Az osztályokat és kapcsolatukat ábrázoló **UML diagram:**



Std::string helyett az 5. laboron elkészített String osztályt használom, a saját megoldásom nem volt tökéletes, ezért a Jportáról használt kóddal javítottam.

Programozói Dokumentáció

Film

A Film osztály lehetővé teszi a filmek adatainak tárolását és kezelését. Ez az osztály az alapja a Filmtar osztálynak, és tartalmazza a film címét, hosszát és kiadás évét.

Funkciók

Konstruktorok `Film() Film(String c, int h = 0, int k = 0)`

A konstruktorok létrehozzák a Film objektumot. Az első konstruktor alapértelmezettként üres értékeket ad a címnek, a hossznak és a kiadás évének. A második konstruktor paraméterként megkapja a film címét, hosszát és kiadás évét.

Destruktor `~Film()`

A destruktor felelős a Film objektum felszabadításáért. Ez törli a film címét, nullára állítja a hosszt és a kiadás évet.

Getter függvények `String getCim() const int getHossz() const int getKiadasEv() const`

A getter függvények visszaadják a film címét, hosszát és kiadás évét.

Copy konstruktor `Film(const(Film& rhs)`

A copy konstruktor lemásolja a Film objektumot. Másolatot készít a film címéről, hosszáról és kiadás évéről.

Értékadó operátor `Film& operator=(const Film& rhs)`

Az értékadó operátor másolja a Film objektumot. Másolatot készít a film címéről, hosszáról és kiadás évéről.

kiir() `void kiir() const`

A kiir() függvény kiírja a film adatait a standard kimenetre. Ha a film címe üres, kivételt dob.

beolvas() `Film* beolvas()`

A beolvas() függvény lehetővé teszi a film adatainak beolvasását a standard bemenetről. A felhasználótól bekéri a címet, a hosszt és a kiadás évet, majd létrehoz egy új Film objektumot ezekkel az adatokkal. Ha a cím üres vagy a hossz negatív vagy nulla, kivételt dob.

operator<< `std::ostream& operator<<(std::ostream& os, const(Film& film)`

Az operator<< függvény lehetővé teszi a Film objektumok kiírását egy ostream objektumba. A film címét, hosszát és kiadás évét írja ki.

Filmtar

egy osztály, amely lehetővé teszi a különböző típusú filmek tárolását és kezelését. Az osztály egy heterogén kollekció, amelyben különböző film objektumokat tárolhatunk, például dokumentumfilmeket és családi filmeket.

Funkciók

Konstruktor `Filmtar()`

A konstruktor létrehozza a Filmtar objektumot. Ez inicializálja a tárat, a kezdeti kapacitást és a darabszámot.

Destruktor`Filmtar()`

A destruktor felelős a Filmtar objektum felszabadításáért. Ez törli a tárolt filmeket és felszabadítja a memóriát.

size() `size_t size() const`

A size() függvény visszaadja a tárolt filmek számát.

felvesz() `void felvesz(Film* ap)`

A felvesz() függvény felelős egy új film hozzáadásáért a tárhoz. Paraméterként megkapja a hozzáadandó film pointerét. Ha a tároló kapacitása megtelt, dinamikusan növeli a tárat a filmek befogadásához.

kiir() `void kiir() const`

A kiir() függvény kiírja az összes tárolt filmet a standard kimenetre. Ha a tároló üres, kivételt dob.

fajlba() `void fajlba(const String& filename)`

A fajlba() függvény fájlba írja a tárolt filmeket. Paraméterként megkapja a fájl nevét. Ha nem sikerül megnyitni a fájlt, kivételt dob.

torol() `void torol size_t(idx)`

A torol() függvény törli a megadott indexű filmet a tárból. Paraméterként megkapja a törlendő film indexét. Ha az index hibás, kivételt dob.

operator[] `Film& operator[](size_t idx) const`

Az operator[] függvény lehetővé teszi a filmek indexelését a tárolóban. Paraméterként megkapja a kívánt film indexét. Ha az index hibás, kivételt dob.

MAIN()

a **main** függvény egy egyszerű menüvezérelt programot valósít meg egy filmeket tároló rendszerhez. A program indulásakor egy üdvözlő üzenet jelenik meg a konzolon, majd létrehoz egy **Filmtar** objektumot, amely a filmeket tárolja.

A **Filmtar** objektumhoz néhány **Film**, **CsaladiFilm** és **DokumentumFilm** objektumot ad hozzá a **felvesz** függvény segítségével. Ezután egy menü jelenik meg, amelyben a felhasználó választhat különböző műveletek között. A választott menüpontnak megfelelő műveletet hajtja végre a program, majd újra megjelenik a menü, amíg a felhasználó ki nem lép.

Az egyes menüpontok végrehajtásához a program az alábbiakat teszi:

1. **Filmlista kiírása**: A **Filmtar** objektum **kiir** függvényét hívja meg, amely kiírja a tárolt filmek adatait.
2. **Keresés cím alapján**: A felhasználótól bekéri egy film címét, majd végigmegy a tárolt filmeken és megkeresi azt, amelyiknek a címe megegyezik a bekért címmel. Ha talál ilyen filmet, kiírja annak adatait, különben pedig jelzi, hogy nincs ilyen film.
3. **Új film hozzáadása**: A felhasználótól bekéri a film típusát, majd ennek megfelelően létrehoz egy új **Film**, **CsaladiFilm** vagy **DokumentumFilm** objektumot a megadott adatokkal. A létrehozott objektumot hozzáadja a **Filmtar** objektumhoz, majd kiírja a frissített filmlistát.
4. **Film törlése**: A felhasználótól bekéri a törölni kívánt film címét, majd végigmegy a tárolt filmeken és törli azt, amelyiknek a címe megegyezik a bekért címmel. Ezután kiírja a frissített filmlistát.
5. **Filmlista fájlba írása**: A **Filmtar** objektum **fajlba** függvényét hívja meg, amely kiírja a tárolt filmek adatait egy "filmek.txt" nevű fájlba.
6. **Kilépés**: Kilép a programból. A destruktorok felszabadítják a lefoglalt memóriát.

A **main** függvényben a menüvezérlés egy **while** ciklusban történik, amíg a felhasználó a kilépés opciót nem választja.

TESZTELÉS

A gtest-tel a különböző funkciókat és osztályokat tesztelem a Film és Filmtar nevű osztályokban. Az egyes tesztesetekben különböző helyzeteket vizsgálnak meg, és az elvárt eredményeket hasonlítják össze a tényleges eredményekkel.

Film(és örökölt osztályok) osztály tesztesetei:

- **TEST(Film, ures)**: Ebben a tesztesetben tesztelem, hogy az üres paraméterekkel létrehozott Film objektum helyesen inicializálódik-e.
- **TEST(Film, konstruktor)**: Ez a teszteset ellenőrzi, hogy a konstruktor helyesen inicializálja-e a Film objektumot a megadott paraméterek alapján.
- **TEST(Film, copy konstruktor)**: Ebben a tesztesetben tesztelem a másoló konstruktor helyes működését, hogy a másolt objektum azonos értékeket tartalmaz-e.

- DokumentumFilm osztály tesztesetei:
- **TEST(DokumentumFilm, konstruktor)**: Ebben a tesztesetben ellenőrzik, hogy a DokumentumFilm osztály konstruktora helyesen inicializálja-e a paraméterek alapján létrehozott objektumot, beleértve a leírást is.
- CsaladiFilm osztály tesztesetei:
- **TEST(CsaladiFilm, konstruktor)**: Ez a teszteset ellenőrzi, hogy a CsaladiFilm osztály konstruktora helyesen inicializálja-e a paraméterek alapján létrehozott objektumot, beleértve a korhatárt is.
- Filmtar osztály tesztesetei:
- **TEST(Filmtar, mukodik a filmtarolo 1 filmre)**: Ebben a tesztesetben tesztelik, hogy a Filmtar osztály helyesen kezeli-e a Film objektumok tárolását és visszakérdezését egyetlen film esetén.
- **TEST(Filmtar, mukodik a filmtarolo tobb filmre)**: Ez a teszteset teszteli, hogy a Filmtar osztály helyesen kezeli-e a Film objektumok tárolását és visszakérdezését több film esetén.
- **TEST(Filmtar, mukodik a filmtarolo mas tipusra is)**: Ebben a tesztesetben ellenőrzik, hogy a Filmtar osztály helyesen kezeli-e más típusú filmek (például CsaladiFilm) tárolását és visszakérdezését.
- **TEST(Filmtar, torles)**: Ellenőrzi, hogy a Filmtar osztály helyesen törli a filmet a megadott index alapján. A teszt hozzáad egy filmet a tárolóhoz, majd törli a megadott index alapján. Ellenőrzi, hogy a tároló mérete 0, és hogy a torles() metódus helyesen működik.
- **TEST(Filmtar, torles idx hiba)**: Ellenőrzi, hogy a Filmtar osztály helyesen kezeli a hibát, amikor egy érvénytelen indexet adnak meg a torles() metódusban. A teszt hozzáad egy filmet a tárolóhoz, majd megpróbálja törölni egy érvénytelen index alapján. Ellenőrzi, hogy const char* kivételt dob a helytelen indexre.
- **TEST(Filmtar, indexelo op hiba)**: Ellenőrzi, hogy a Filmtar osztály helyesen kezeli a hibát, amikor egy érvénytelen indexet használnak a [] operátorban. A teszt hozzáad egy filmet a tárolóhoz, majd megpróbálja lekérni egy érvénytelen indexű filmet. Ellenőrzi, hogy const char* kivételt dob a helytelen indexre.
- **TEST(Filmtar, torles indexelo op hiba)**: Ellenőrzi, hogy a Filmtar osztály helyesen kezeli a hibát, amikor egy érvénytelen indexet használnak a torles() metódusban és a [] operátorban. A teszt hozzáad egy filmet a tárolóhoz, majd megpróbálja törölni egy érvénytelen index alapján. Ellenőrzi, hogy const char* kivételt dob a helytelen indexre.
- **TEST(Filmtar, fajlba iras)**: Ellenőrzi, hogy a Filmtar osztály helyesen írja ki a filmeket egy fájlba. A teszt hozzáad egy CsaladiFilm objektumot a tárolóhoz, majd a fajlba() metódust használva kiírja a tárolót egy tesztfájlba. Ellenőrzi, hogy a kiírás nem dob kivételt.
- **A többi pedig hasonlóan működik.**