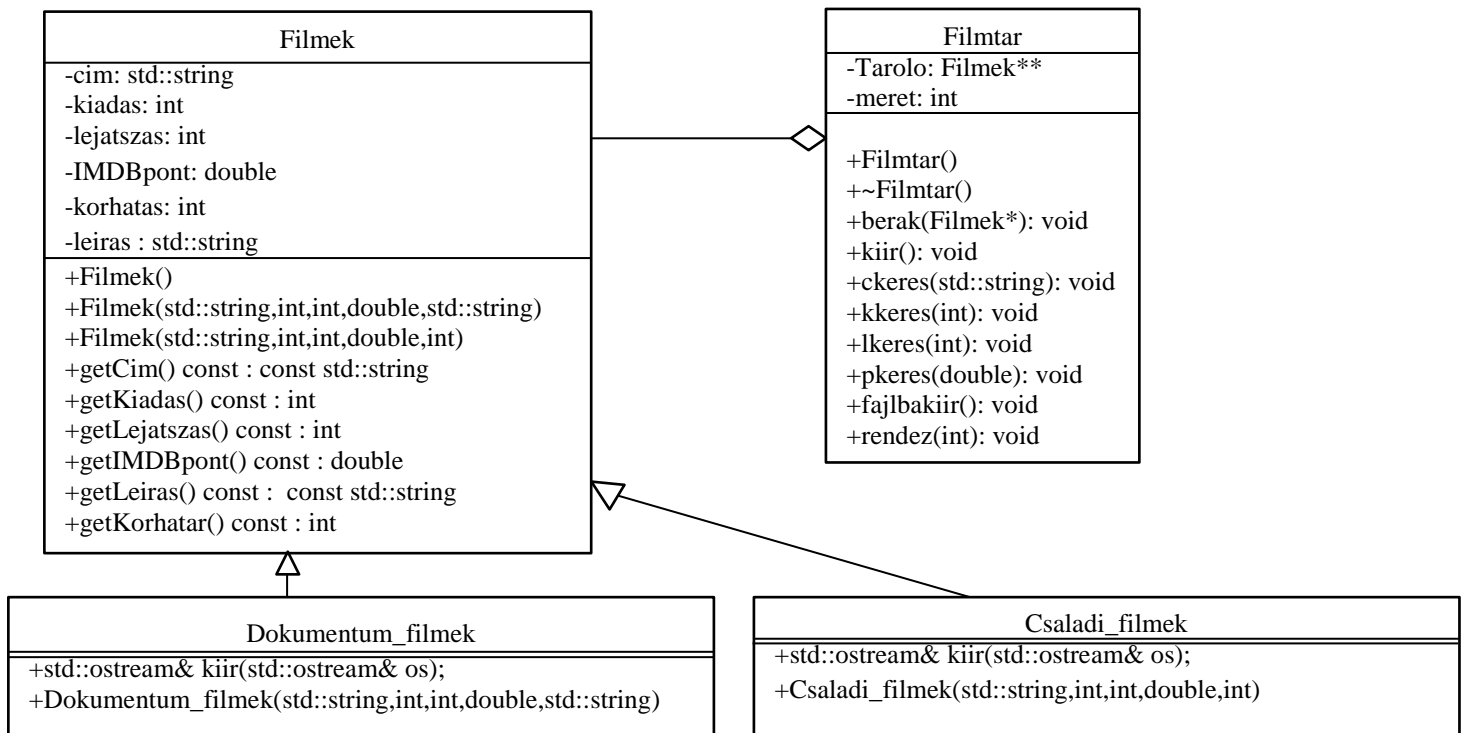


# Filmtár

Egy filmeket nyilvántartó rendszert készítek el. Tárolom a filmek címét, lejátszási idejét, kiadási évét és az elért pontszámát az IMDb-n. A családi filmeknél korhatár is van, a dokumentumfilmeknél pedig egy szöveges leírást is tárolok.

A filmeket egy szöveges fájlból olvasom be a programba. A filmek között lehet keresni (cím, kiadási év, lejátszási idő, pont szerint) és kiírni őket egy másik szöveges fájlba (lemasolt). Lehet a filmeket rendezni is (cím, kiadási év, lejátszási idő, pont) szerint. Magában a programban is fel lehet venni tagokat (most: dokumentumfilm, családi film). A szülő osztály a **Filmek**, amelynek jelenleg van kettő gyermeke, a **Családi filmek** és a **Dokumentum filmek**. A **Filmtár** egy tarolót hoz létre, amely különböző fajtájú filmek adatait tárolja.

## UML jelölésű osztálydiagramot



## **Filmtar:**

A Filmtar osztályban definiált Tarolo Filmek pointereket tarol. Ezáltal tárolja az összes fajta film adatait, amit betesznek belé. Ilyen formában tovább is bővíthető más fajta filmekkel. A meret intiger pedig a jelenlegi méretét adja meg (pl.: egy filmfajta Taroloba helyezése után +1 lesz az értéke). Dinamikusan tárolja a Filmeket így nincsen egy fix kapacitása a tárolónak.

### **Filmtar():**

Az osztály paraméter nélküli konstruktora, amely a Filmtar osztály adattagjainak kezdőértéket ad.

### **void berak(Filmek\*):**

A Taroloba betesz egy, valamilyen fajta filmet (adataival együtt).

### **void kiir():**

A Tarolo filmjeinek adatait kiírja a standard kimenetre.

### **void ckeres(std::string):**

A Tarolo filmjei között cím szerint keres.

Egy char tömbb-öt kap, amely a keresés feltétele.

Az eredményt kiírja a standard kimenetre.

### **void kkeres(int):**

A Tarolo filmjei között kiadási év szerint keres.

Egy intiger-t kap, amely a keresés feltétele.

Az eredményt kiírja a standard kimenetre.

### **void lkeres(int):**

A Tarolo filmjei között lejátszási idő szerint keres.

Egy intiger-t kap, amely a keresés feltétele.

Az eredményt kiírja a standard kimenetre.

### **void pkeres(double):**

A Tarolo filmjei között IMDB pont szerint keres.

Egy double-t kap, amely a keresés feltétele.

Az eredményt kiírja a standard kimenetre.

### **void fajlbakiir():**

Kiírja a Tarolo filmjeit egy lemasolt.txt fájlba.

### **void rendez(int):**

A Tarolo filmjeit rendezi, valamilyen feltétel szerint.

Egy intiger-t kap, amely megmondja, hogy mi szerint rendezzen sorba.

### **~Filmtar():**

Az osztály destruktora felszabadítja a Tarolo filmjeit.

## **Dokumentum filmek:**

A Dokumentum\_filmek osztálya egy filmfajta. Az osztálynak van egy külön adattagja (std::string leiras), amely csak ennek az filmfajtának van.

### **Dokumentum\_filmek(std::string,int,int,double,std::string):**

Az osztály paraméteres konstruktora, amely beállítja adott értékekre a dokumentumfilm összes adatát.

### **std::ostream& kiir(std::ostream& os):**

Kiír a konzolra egy speciális szöveget ("A természetfilm elozetes leirasa:"<<std::endl<<Filmek::getLeiras()), amelyet csak a dokumentum filmek esetén ír ki.

## **Csaladi filmek:**

A Csaladi\_filmek osztálya egy másik filmfajta. Ennek az osztálynak is van egy külön adattagja (int korhatar), amely csak ennek a filmfajtának van.

### **Csaladi\_filmek(std::string,int,int,double,int):**

Az osztály paraméteres konstruktora, amely beállítja adott értékekre a családi film összes adatát.

### **std::ostream& kiir(std::ostream& os):**

Kiír a konzolra egy speciális szöveget ("A film "<<Filmek::getKorhatar()<<" ev alatt szuloi felugyelet nelkul nem ajanlot!"), amelyet csak a családi filmek esetén ír ki.

## **Filmek:**

A Filmek osztály a heterogén kollekció megvalósítása miatt egy absztrak osztály. Kettő, virtuális absztrak tagfüggvénye van. A Filmek osztály adattagjai a filmek közös adatai (std::string cím,int kiadas,int lejatszas,double IMDBpont), amely minden filmfajtnál megvan.

### **Filmek();**

Az osztály paraméter nélküli konstruktora, amely a közös adatoknak kezdőértéket ad.

### **Filmek(std::string,int,int,double,std::string);**

Az osztály paraméteres konstruktora, amely beállítja adott értékekre az összes adatot.

### **Filmek(std::string,int,int,double,int);**

Az osztály paraméteres konstruktora, amely beállítja adott értékekre az összes adatot.

### **const std::string getCim() const;**

Visszaadja a filmfajta címét.

### **int getKiadas() const;**

Visszaadja a filmfajta kiadási évét.

### **int getLejatszas() const;**

Visszaadja a filmfajta lejátszási idejét (percben).

### **double getIMDBpont() const;**

Visszaadja a filmfajta IMDB pontját.

### **virtual const std::string getLeiras() const = 0;**

Absztrak, virtuális tagfüggvény, hogy lefusson a filmfajták azonos nevű tagfüggvénye.

### **virtual int getKorhatar() const = 0;**

Absztrak, virtuális tagfüggvény, hogy lefusson a filmfajták azonos nevű tagfüggvénye.

## **Tesztelés:**

A filmeket nyilvántartó rendszerem helyes működését különféle módszerekkel ellenőriztem le.

A memóriaszivárgás elkerülése érdekében a memtrace.h és memtrace.cpp fájlokat alkalmaztam, melyek nem jeleztek memóriaszivárgást a projektben.

A menürendszeremnél beírtam olyan számokat (pl.: 12), melyek érvénytelenek. A programomba épített hibakezelő jelezte, hogy rossz számot írtam be és próbáljam újra beírni a jó számok valamelyikét.

A projektben van egy hibakezelő függvény is (main.cpp), amely megakadályozza, hogy szám helyett betű beírásakor végtelen ciklusba jusson a programom. Ezt is ellenőriztem betűk beírásával (pl.: alma) és a projekt jelezte, hogy nem számot írtam be.

A fájl megnyitásának ellenőrzésére is van hibakezelőm, mely jelzi, ha a programnak nem sikerült megnyitni az adott fájlt.

Úgy ellenőriztem, hogy átírtam a programomban a megnyitni kívánt fájl nevét egy nem létező fájl névére (pl.: banan.txt). Ekkor a hibakezelő jelezte, hogy hiba van a fájl megnyitásakor és a program leáll.

A program helyes működését pedig az inputon megadott helyes adatokkal ellenőriztem le.