

Papp Bálint - HND2AH

NHF - Filmtár - Pontosított Feladat-specifikáció és Osztályterv

1. Pontosított Feladat-specifikáció

A feladat egy filmnyilvántartó rendszer elkészítése, amely képes különböző típusú filmek adatainak kezelésére. A rendszer egy parancssoros alkalmazás, amely lehetővé teszi filmek hozzáadását, törlését, keresését és listázását, valamint az adatok mentését és betöltését. A megoldáshoz heterogén kollekciót kell használni.

Bemenet és kimenet

Felhasználói parancsok

A program a standard bemenetről fogadja a felhasználói parancsokat:

- add: Új film hozzáadása a megadott típussal (normal, family, documentary)
- delete: Film törlése azonosító alapján
- list: Az összes film listázása
- search: Filmek keresése a megadott feltétel alapján
- save: A filmadatok mentése megadott fájlba
- load: Filmadatok betöltése megadott fájlból
- exit: Kilépés a programból

Filmadatok bevitele

Az add parancs után a program bekéri a film adatait:

- Minden filmtípus esetén: cím, lejátszási idő (percben), kiadási év
- Családi film esetén: korhatár (év)
- Dokumentumfilm esetén: leírás (szöveg)

Keresési feltételek

A search parancs után a következő keresési feltételek adhatók meg:

- title(szöveg): Keresés cím alapján
- year(szám): Keresés kiadási év alapján
- type(osztály): Keresés filmtípus alapján
- A feltételek kombinálhatók & jellel: title=Star&year=1977

Kimenet formátuma

- A list és search parancsok esetén minden film adatai egy-egy sorban jelennek meg, tabulátorral elválasztva:
 - [azonosító] [típus] [cím] [lejátszási idő] [kiadási év] [egyéb specifikus adatok]
- Hibaüzenetek a standard hibakimenetre kerülnek egyértelmű üzenettel
- A parancsok végrehajtása után a program visszajelzést ad a művelet sikerességéről

Adatmodell

A program a következő adatokat tárolja:

1. Film:

- Azonosító (egész szám)
- Cím (szöveg)
- Lejátszási idő (egész szám, percben)
- Kiadási év (egész szám)

2. Családi film (Film leszármazottja):

- Film adatai
- Korhatár (egész szám)

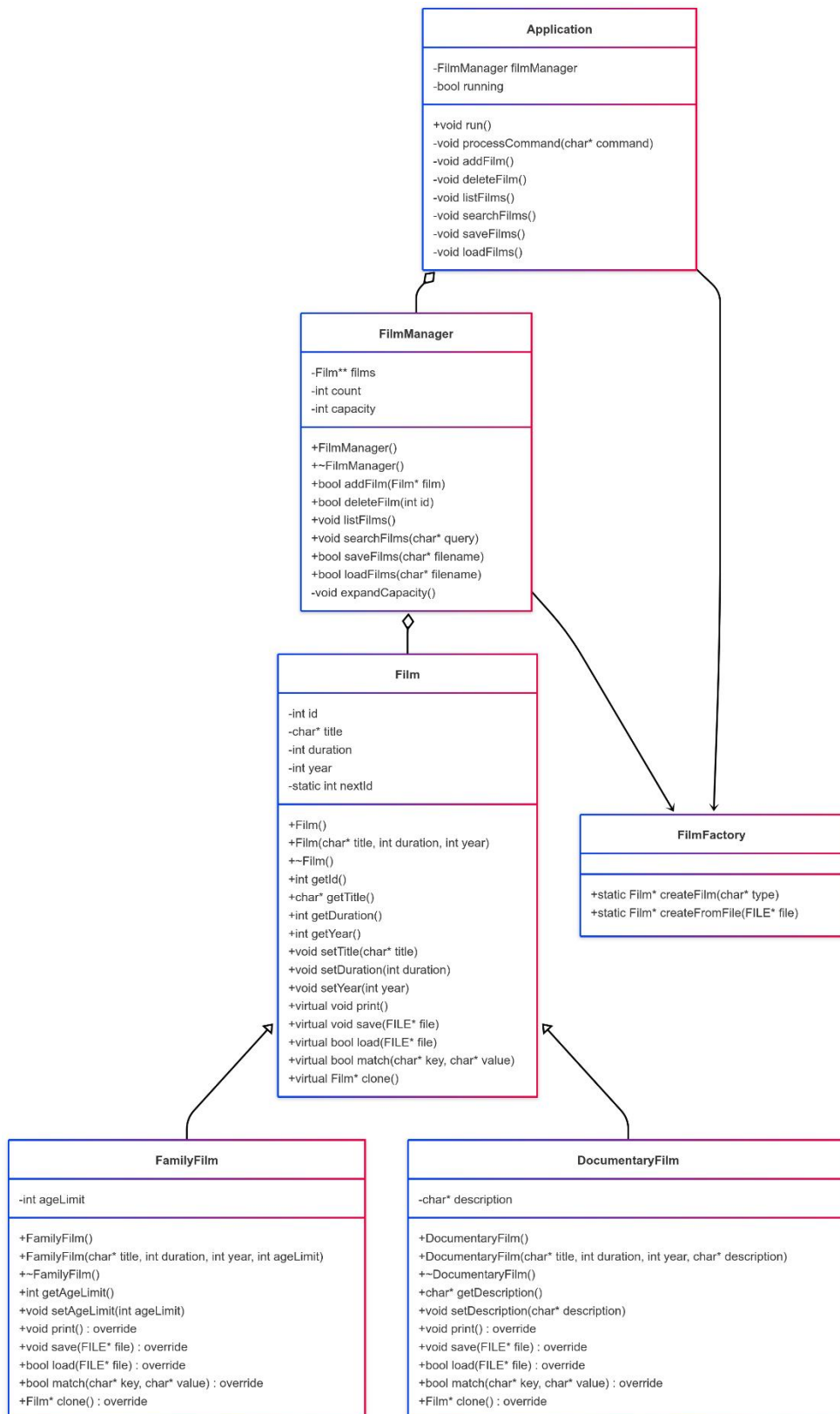
3. Dokumentumfilm (Film leszármazottja):

- Film adatai
- Leírás (szöveg)

Fájlformátum

- Minden sor egy filmet reprezentál
- A sorok felépítése: [típus] [cím] [lejátszási idő] [kiadási év] [típus-specifikus adatok]
- Példa normál filmre: normal Star Wars 121 1977
- Példa családi filmre: family Frozen 102 2013 6
- Példa dokumentumfilmre: documentary Planet Earth 550 2006 Természeti dokumentumfilm

2. Osztálydiagram



3. Osztályok részletes leírása

Film (Absztrakt ősosztály)

Ez az osztály az összes filmtípus közös jellemzőit és viselkedését definiálja.

Attribútumok:

- id: Egyedi azonosító (egész szám)
- title: A film címe (karaktertömb)
- duration: Lejátszási idő percben (egész szám)
- year: Kiadási év (egész szám)
- static nextId: Következő kiosztandó azonosító

Metódusok:

- Konstruktorok és destruktork
- Getter és setter metódusok
- print(): Film adatainak kiírása a standard kimenetre
- save(FILE*): Film adatainak mentése fájlba
- load(FILE*): Film adatainak betöltése fájlból
- match(char* key, char* value): Ellenőrzi, hogy a film megfelel-e a keresési feltételnek
- clone(): Másolat készítése a filmről (virtuális metódus)

FamilyFilm (Film leszármazottja)

Családi filmek specifikus adatait és viselkedését definiálja.

Attribútumok:

- ageLimit: Korhatár (egész szám)

Metódusok:

- Az ősosztály metódusainak felüldefiniálása
- Az ageLimit attribútum getter és setter metódusai

DocumentaryFilm (Film leszármazottja)

Dokumentumfilmek specifikus adatait és viselkedését definiálja.

Attribútumok:

- description: Leírás (karaktertömb)

Metódusok:

- Az űsosztály metódusainak felüldefiniálása
- A description attribútum getter és setter metódusai

FilmManager

Ez az osztály felelős a filmek tárolásáért és a rajtuk végezhető műveletekért. Heterogén kollekciót használ a különböző típusú filmek tárolására.

Attribútumok:

- films: Film pointerek tömbje (Film**)
- count: A tárolt filmek száma
- capacity: A tömb aktuális kapacitása

Metódusok:

- addFilm(Film*): Új film hozzáadása
- deleteFilm(int id): Film törlése azonosító alapján
- listFilms(): Összes film listázása
- searchFilms(char* query): Filmek keresése feltétel alapján
- saveFilms(char* filename): Filmek mentése fájlba
- loadFilms(char* filename): Filmek betöltése fájlból
- expandCapacity(): A tömb kapacitásának növelése (privát metódus)

FilmFactory

Ez az osztály felelős a különböző típusú filmek létrehozásáért.

Metódusok:

- createFilm(type): Új film létrehozása típus alapján
- createFromFile(FILE*): Film létrehozása fájlból olvasott adatok alapján

Application

Ez az osztály felelős a felhasználói interakcióért és a program futásáért.

Attribútumok:

- filmManager: A filmek kezelését végző objektum

- running: Futás jelző (logikai érték)

Metódusok:

- run(): A program fő ciklusa
- processCommand(char*): Parancs feldolgozása
- Különböző parancsokat végrehajtó metódusok

4. Fontosabb algoritmusok

4.1. Dinamikus memóriakezelés (FilmManager::expandCapacity)

```
expandCapacity() {  
    // 1. Új, nagyobb kapacitású tömb létrehozása  
    újKapacitás = régiKapacitás * 2  
    újTömb = új Film*[újKapacitás]  
  
    // 2. Adatok átmásolása  
    for i = 0 to count-1 do  
        újTömb[i] = régiTömb[i]  
  
    // 3. Régi tömb felszabadítása  
    delete[] régiTömb  
  
    // 4. Változók frissítése  
    films = újTömb  
    capacity = újKapacitás  
}
```

4.2. Filmek keresése (FilmManager::searchFilms)

```
searchFilms(query) {  
    // 1. Keresési feltételek feldolgozása  
    feltételek[] = query.split('&')  
  
    // 2. Összes film ellenőrzése a feltételek alapján  
    for i = 0 to count-1 do  
        találat = true  
  
        // 3. Minden feltétel ellenőrzése  
        for minden feltétel in feltételek do  
            kulcs, érték = feltétel.split('=')  
  
            // 4. Ha nem felel meg a feltételnek, nem találat  
            if !films[i]->match(kulcs, érték) then  
                találat = false  
                break  
  
        // 5. Ha minden feltételnek megfelel, kiírjuk  
        if találat then  
            films[i]->print()
```

```
}
```

4.3. Filmek mentése fájlba (FilmManager::saveFilms)

```
saveFilms(filename) {  
    // 1. Fájl megnyitása írásra  
    file = fopen(filename, "w")  
    if file == NULL then  
        return false  
  
    // 2. Filmek mentése egyenként  
    for i = 0 to count-1 do  
        films[i]->save(file)  
  
    // 3. Fájl bezárása  
    fclose(file)  
    return true  
}
```

4.4. Filmek betöltése fájlból (FilmManager::loadFilms)

```
loadFilms(filename) {  
    // 1. Fájl megnyitása olvasásra  
    file = fopen(filename, "r")  
    if file == NULL then  
        return false  
  
    // 2. Jelenlegi filmek törlése  
    for i = 0 to count-1 do  
        delete films[i]  
    count = 0  
  
    // 3. Új filmek betöltése  
    while !feof(file) do  
        film = FilmFactory::createFromFile(file)  
        if film != NULL then  
            addFilm(film)  
  
    // 4. Fájl bezárása  
    fclose(file)  
    return true  
}
```

4.5. Parancs feldolgozása (Application::processCommand)


```
processCommand(command) {  
    // 1. Parancs részekre bontása  
    parts[] = command.split(' ')  
  
    // 2. Megfelelő művelet végrehajtása  
    if parts[0] == "add" then  
        addFilm()  
    else if parts[0] == "delete" then  
        deleteFilm(parseInt(parts[1]))  
    else if parts[0] == "list" then  
        listFilms()  
    else if parts[0] == "search" then  
        searchFilms(parts[1])  
    else if parts[0] == "save" then  
        if parts.length > 1 then  
            saveFilms(parts[1])  
        else  
            saveFilms("filmek.dat")  
    else if parts[0] == "load" then  
        if parts.length > 1 then  
            loadFilms(parts[1])  
        else  
            loadFilms("filmek.dat")  
    else if parts[0] == "exit" then  
        running = false  
    else  
        print("Ismeretlen parancs")  
}
```