**Papp Bálint - HND2AH**

**NHF - Filmtár - Pontosított Feladatspecifikáció és Osztályterv**

**1. Pontosított Feladatspecifikáció**

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

**Hibakezelés**

* A program szigorú hibaellenőrzést végez minden felhasználói bemenetre
* Bármilyen érvénytelen bemenet esetén a program azonnal megszakítja az aktuális parancs végrehajtását, hibaüzenetet jelenít meg, és visszatér a főciklusba
* Hibás keresési formátum esetén a keresés nem fut le, hibaüzenet jelenik meg
* Hibás filmtípus esetén a film hozzáadása megszakad
* Érvénytelen numerikus értékek esetén az aktuális művelet megszakad

**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. OsztálydiagramA diagram of a computer code

AI-generated content may be incorrect.**

**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 destruktor
* 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)
* createFromFile(FILE\*): Statikus metódus film létrehozására fájlból

**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 ősosztály 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
* addFilm(const char\* type, const char\* title, int duration, int year, const char\* extraData): Film létrehozása és 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)
* createFilm(): Film létrehozása típus alapján (privát metódus)

**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
* addFilm(): Új film hozzáadása
* deleteFilm(): Film törlése
* listFilms(): Filmek listázása
* searchFilms(): Filmek keresése
* saveFilms(): Filmek mentése
* loadFilms(): Filmek betöltése
* readLine(): Szöveg beolvasása
* readInt(): Egész szám beolvasása

**4. Fontosabb algoritmusok**

**4.1. Dinamikus memóriakezelés (FilmManager::expandCapacity)**

A FilmManager osztály expandCapacity metódusa felelős a filmek tárolására használt tömb méretének dinamikus növeléséért. A módszer először létrehoz egy új, nagyobb méretű tömböt (általában a jelenlegi kapacitás kétszeresét), majd átmásolja az összes meglévő filmet az új tömbbe. Végül felszabadítja a régi tömb által használt memóriát, és frissíti a FilmManager kapacitás attribútumát.

**4.2. Filmek keresése (FilmManager::searchFilms)**

A keresési algoritmus először validálja a keresési lekérdezést, ellenőrizve, hogy a megfelelő formátumban van-e. Ezután felbontja a keresési lekérdezést egyedi feltételekre az '&' elválasztó mentén. Ezután végigmegy az összes filmen, és minden filmre ellenőrzi, hogy megfelel-e az összes feltételnek. Ezt a Film osztály virtuális match metódusának hívásával teszi, amely különböző módon van implementálva a különböző filmtípusokban. Ha egy film megfelel az összes feltételnek, akkor az kilistázásra kerül. A keresés szigorú ellenőrzést végez a keresési feltételekre, és érvénytelen formátum esetén hibaüzenetet jelenít meg és megszakítja a keresést.

**4.3. Filmek mentése fájlba (FilmManager::saveFilms)**

A mentési algoritmus megnyitja a megadott fájlt írásra, majd végigmegy az összes filmen, és mindegyikre meghívja a virtuális save metódust. Ez a metódus különböző módon van implementálva a különböző filmtípusokban, hogy megfelelően tárolják a típus-specifikus adatokat. A címekben és leírásokban lévő szóközöket aláhúzásjelekre (\_) cseréli, hogy a fájlba írás és beolvasás során ne okozzanak problémát. A művelet végén a fájl bezárásra kerül.

**4.4. Filmek betöltése fájlból (FilmManager::loadFilms)**

A betöltési algoritmus először megnyitja a megadott fájlt olvasásra, majd törli az összes jelenleg tárolt filmet. Ezután sorról sorra beolvassa a fájlt, és a Film::createFromFile statikus metódus segítségével új filmeket hoz létre a beolvasott adatok alapján. Az aláhúzásjeleket (\_) visszaalakítja szóközökké a címekben és leírásokban. Minden új film hozzáadódik a FilmManager által kezelt kollekcióhoz. A művelet végén a fájl bezárásra kerül.

**4.5. Parancs feldolgozása (Application::processCommand)**

A parancs feldolgozó algoritmus felbontja a beérkező parancsot a szóközök mentén, majd az első szó alapján meghatározza a végrehajtandó műveletet. Az egyes parancsokhoz tartozó metódusok felelősek az adott művelet végrehajtásáért, például új film hozzáadásáért, filmek listázásáért vagy a keresési feltétel alapján történő filmek keresésért. Ha érvénytelen parancsot észlel, hibaüzenetet jelenít meg.

**4.6. Felhasználói bemenet ellenőrzése**

A program szigorú ellenőrzést végez minden felhasználói bemenetre. A readInt metódus ellenőrzi, hogy a megadott érték csak számjegyeket tartalmaz-e, és érvényes egész szám-e. A readLine metódus megfelelően kezeli a bemenet ellenőrzését. Bármilyen érvénytelen bemenet esetén a program hibaüzenetet jelenít meg és visszatér a főciklusba.

**5. Fájlszerkezet**

A projekt a következő fájlokból áll:

1. **Osztály Definíciók** 
   * film.h és film.cpp: Film alap osztály
   * familyFilm.h és familyFilm.cpp: Családi film osztály
   * documentaryFilm.h és documentaryFilm.cpp: Dokumentumfilm osztály
   * filmManager.h és filmManager.cpp: Filmkezelő osztály
   * application.h és application.cpp: Alkalmazás osztály
2. **Fő Program** 
   * main.cpp: A program belépési pontja
3. **Tesztelés**
   * Főprogram futtatásakor bementre: test