

Házi feladat

Programozás alapjai 2.

Feladatválasztás/feladatspecifikáció

Spertli Krisztián

IYX0EQ

Feladatválasztás

Készítsen filmeket nyilvántartó rendszert. A rendszer lehetővé teszi filmek adatainak tárolását, beleértve a címet, lejátszási időt és kiadási évet. A családi filmeknél korhatárt tárolunk, míg a dokumentumfilmekhez egy szöveges leírást is hozzárendelünk. Az objektummodellnek könnyen bővíthetőnek kell lennie.

Bemenetek és elvárt kimenetek

Bemenetek

- Felhasználói interakciók a filmekkel, például hozzáadás, törlés, listázás évjárat szerint stb.
- Filmek adatai, mint cím, lejátszási idő, kiadási év, korhatár és dokumentumfilmek esetében a leírás

Elvárt kimenetek

- A felhasználó által kért műveletek elvégzése (pl. film hozzáadása, listázása)
- Filmek adatainak megjelenítése a felhasználó számára
- Hibaüzenetek vagy visszajelzések, ha a felhasználó által végzett műveletek nem sikerülnek

Program működésének feltételei

- A programnak képesnek kell lennie kezelni filmeket, családi filmeket és dokumentumfilmeket.
- A filmek adatait szöveges formátumban kell tárolni, és a programnak lehetőséget kell biztosítania azok kezelésére (pl. hozzáadás, módosítás, törlés stb.).
- A rendszernek képesnek kell lennie listázni a filmeket, és külön kezelni kell a családi és dokumentumfilmeket.
- A rendszer elvárja bizonyos helyeken a megfelelő formátumokat, probléma esetén hibaüzenettel térünk vissza
- A programnak lehetőséget kell biztosítania további filmtípusok felvételére a jövőben

Egyéb körülmények

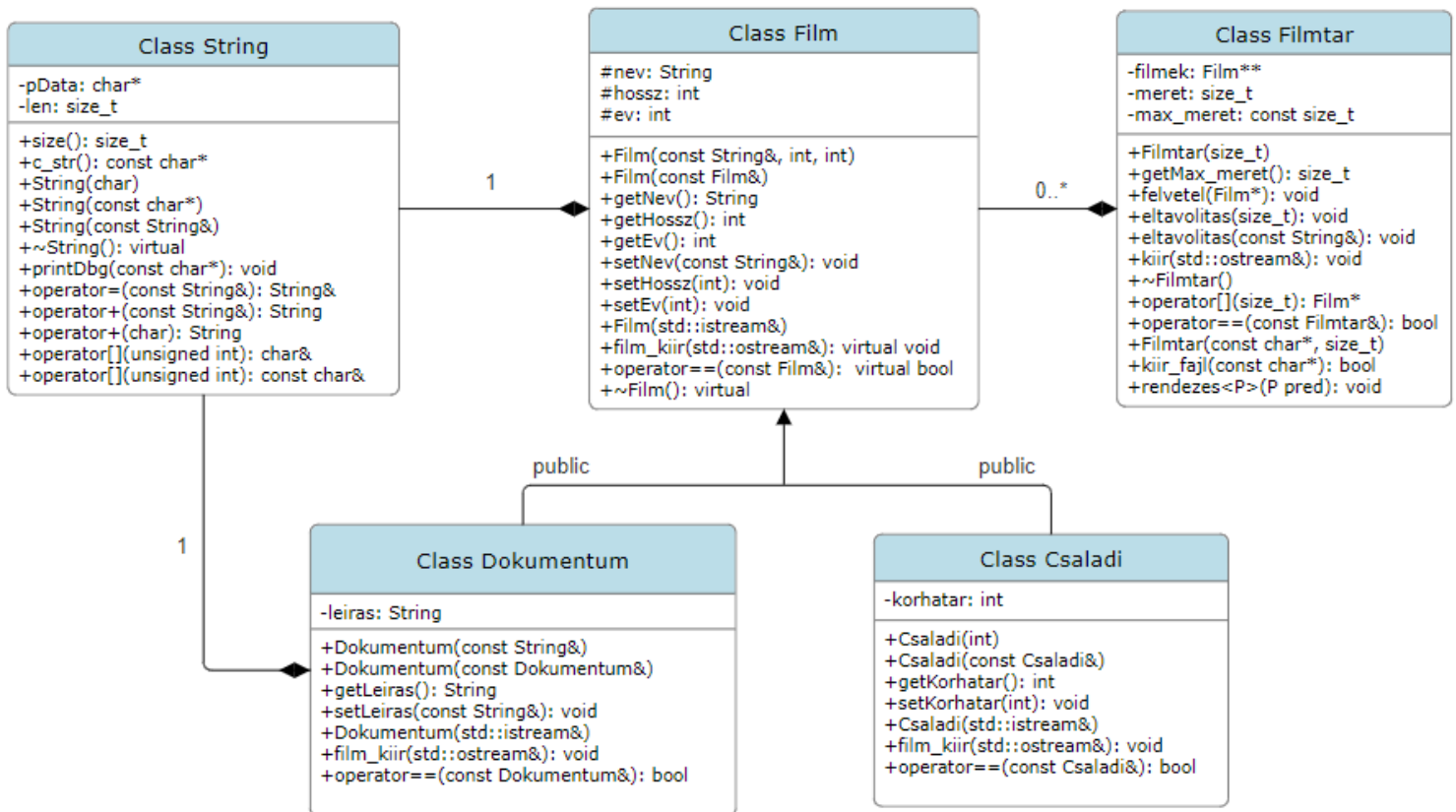
- A program ne használjon STL tárolókat, saját adatstruktúrákat kell implementálni.
- A felhasználói felület lehetővé teszi a filmek kezelését és megjelenítését.
- A rendszernek biztosítania kell a felhasználók számára egyértelmű és könnyen kezelhető felületet a filmek kezeléséhez.
- A program bizonyos százalékát lefedő tesztprogramokat kell implementálni

A fenti specifikáció a feladat nagyvonalú leírását adja meg. A specifikáció pontosítása és tervezése a további megoldás részeként történik.

Terv

Osztályok kapcsolata

A jelölésrendszer az UML diagrammnak megfelelő.



Osztályok

String

Az osztály karaktersorozatok tárolását végzi el dinamikus területen, amelynek pontos specifikációját és algoritmikus működését az 5. laborfeladaton végeztem el.

Film

A program ősosztálya, amely filmeket tartalmaz. Az osztály adatai tartalmazzák a film nevét, hosszát(percben), illetve a kiadási évét. A tagfüggvények között szerepel a megfelelő konstruktorok, getterek, setterek és operátorok megvalósítása. Emellett a heterogén kollekció szempontjából elengedhetetlen a film_kiir virtuális tagfüggvény. A további virtuális függvények a leszármazás miatt fontosok.

Dokumentum

Az osztály a Film osztály egyik leszármazottja, amely specializál a dokumentumfilmekre. Plusz adataja az adott film leírását tartalmazza. A

tagfüggvények között szerepel továbbá az űosztály tovább definiált virtuális függvényei.

Csaladi

Hasonlóan az előzőhöz, a Film osztály másik leszármazottja a Családi osztály, amely plusz adattagként a film korhatárát kapta meg. Az alapértelmezett tagfüggvényein kívül szintén tartalmaz overloadolt függvényeket.

Filmtar

Ez az osztály kiemelt funkciót lát el, ugyanis heterogén kollekcióként hasznosítom a filmek eltárolásához. Tárolja az adatbázis maximális méretét, hogy éppen hány elem van benne, illetve a filmtár filmeire mutató pointerek tömbjét. Az adatbázis megkönnyebbítése érdekében a program a filmtárban való bővítést, eltávolítást, rendezést és keresést is biztosítja.

Kiemelt algoritmusok

Filmtár::Fájlba kiíratás

Amikor a program fájlba ír, átiterál a filmeket tartalmazó tömbön, és mindegyik elemre meghívja a film_kiir tagfüggvényt. Az objektum típusától függően a megfelelő függvény kerül meghívásra a virtualitás miatt, így a leszármazott osztályok adatai is kiíródnak.

Filmtár::Fájlból beolvasás

Az adatok beolvasása a fájlból egy konstruktoron keresztül történik, amely két paramétert kap: a fájl nevét, illetve a felveendő filmek maximális számát. Ezután a függvény beolvassa a megadott számú sort, és az első szó alapján meghívja a megfelelő konstruktort, majd hozzáadja az így létrehozott objektumot a heterogén kollekcióhoz.

Filmtár::Rendezés

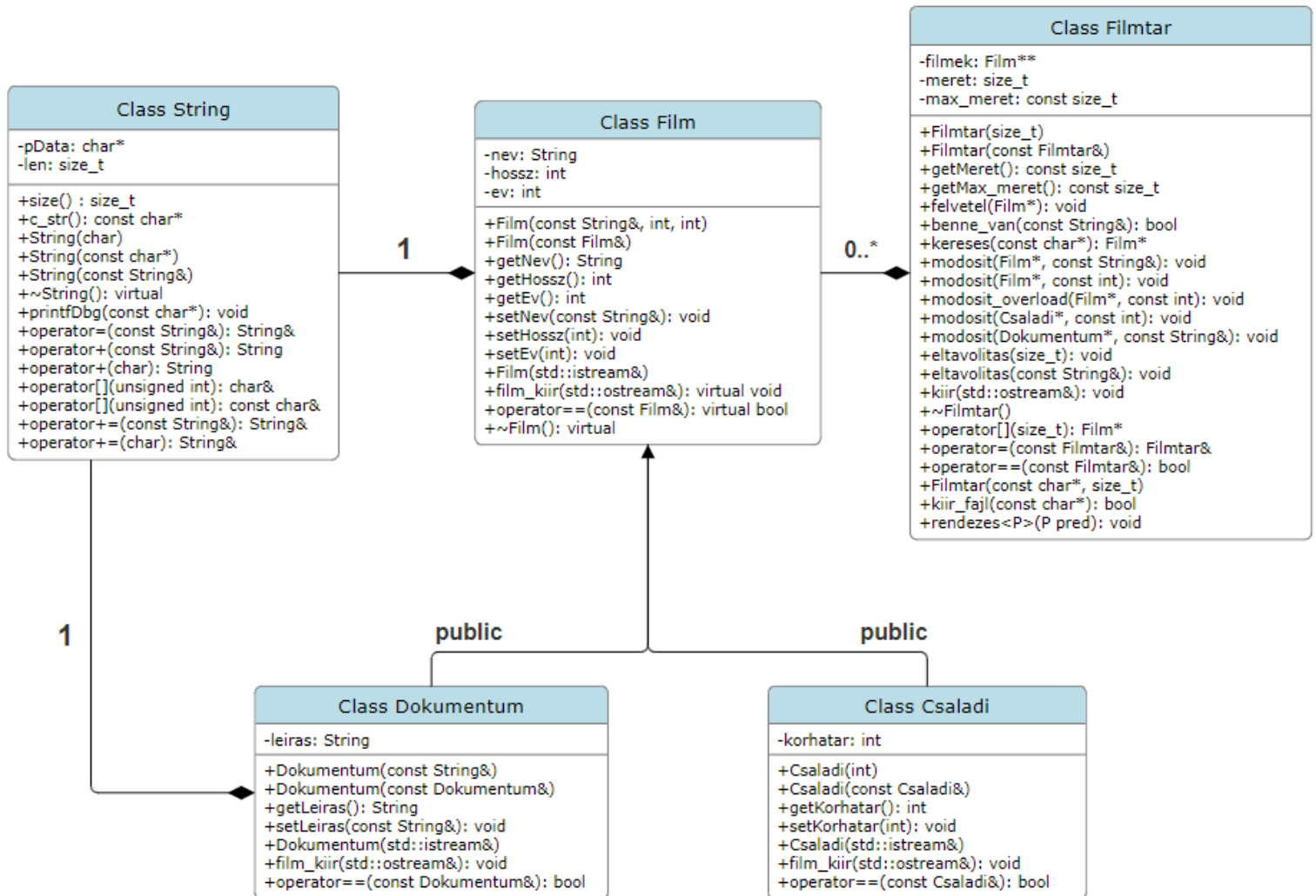
Az adatbázis rendezhető, amely alkalmas lehet a felhasználói igények kielégítésére. A rendezési algoritmus még tervben van, ami buborékredezést vagy kertitörpe-rendezeit fog használni, ami az előző évi programozási ismeretekre alapoz.

Főprogram

A tesztprogram létrehoz egy filmtárat, amely tartalmaz mindenféle filmet, és kipróbálja mindegyik filmfajta minden konstruktorát. Ezt a filmtárat elmenti egy fájlba, majd beolvassa az adatbázist a fájlból, és minden rendelkezésre álló rendezési módot alkalmaz rá. Minden lépés után ellenőrzi a helyességet, és figyeli a memóriaszivárgást.

Dokumentáció

A végleges osztálydiagramm a következőképpen alakult:



Az osztályok működése és a tagfüggvények feladata

Az osztályok célja a különböző típusú filmek kezelése, valamint ezek tárolása és módosítása. A diagrammon szereplő osztályok közötti kapcsolatok és az osztályok metódusai is részletes bemutatásra kerülnek.

String osztály

Áttekintés

A String osztály egy egyedi karakterlánc típus, amely a C++ standard könyvtárainak std::string típusához hasonló funkcionalitást kínál. Az osztály a karakterlánc adatait és hosszát tárolja, valamint számos konstruktort és operátort biztosít a karakterlánc kezelésére.

Attribútumok

- pData: char* típusú mutató, amely a karakterlánc adatokat tárolja.
- len: size_t típusú változó, amely a karakterlánc hosszát tárolja.

Metódusok

- size(): Visszaadja a karakterlánc hosszát (size_t).
- c_str(): Visszaad egy konstans karakter mutatót (const char*) a karakterlánc adatokhoz.
- String(const char*): Konstruktor, amely egy const char* típusú karakterláncból hoz létre egy String objektumot.
- String(char): Konstruktor, amely egy char típusú karakterből hoz létre egy String objektumot.
- String(const String&): Konstruktor, amely egy másik String objektumból hoz létre egy új String objektumot.
- ~String(): Destruktor, amely az String típusú objektumot szünteti meg.
- operator=(const String&): Értékadás operátor, amely egy String objektumot másol egy másikba.
- operator+(const String&): Két String objektumot összefűz és egy új String objektumot ad vissza.
- operator+(char): String objektumhoz karaktert hozzáfűz és egy új String objektumot ad vissza.
- operator[](unsigned int): Visszaad egy hivatkozást (char&) a karakterlánc adott indexű karakterére.
- operator+=(const String&): Egy másik String objektumot ad hozzá az aktuális String objektumhoz.
- operator+=(char): Egy karaktert ad hozzá az aktuális String objektumhoz.
- operator==(const String&): Összehasonlít két String objektumot és visszaad egy logikai értéket (egyenlőség).

Film osztály

Áttekintés

A Film osztály egy filmet reprezentál, amelynek neve, hossza és év azonosítója van. Az osztály biztosítja a film adatok kezeléséhez szükséges metódusokat, valamint támogatja a film adatainak kiírását és módosítását.

Attribútumok

- nev: String típusú változó, amely a film nevét tárolja.
- hossz: int típusú változó, amely a film hosszát tárolja.
- ev: int típusú változó, amely a film évét tárolja.

Metódusok

- `Film(const String&, int, int)`: Konstrktor, amely inicializálja a film nevét, hosszát és évét.
- `Film(const Film&)`: Másoló konstruktor, amely egy másik Film objektumból hoz létre egy új Film objektumot.
- `getNev()`: Visszaadja a film nevét (String).
- `getHossz()`: Visszaadja a film hosszát (int).
- `getEv()`: Visszaadja a film évét (int).
- `setNev(const String&)`: Beállítja a film nevét.
- `setHossz(int)`: Beállítja a film hosszát.
- `setEv(int)`: Beállítja a film évét.
- `Film(std::istream&)`: Konstrktor, amely egy bemeneti folyamat (std::istream) használ a film adatok beolvasásához.
- `film_kiir(std::ostream&)`: Kiírja a film adatait egy kimeneti folyamra (std::ostream).
- `operator==(const Film&)`: Összehasonlítja két Film objektumot és visszaad egy logikai értéket (egyenlőség).
- `~Film()`: Destruktor, amely Film típusú objektumot szüntet meg (virtual).

Dokumentum osztály

Áttekintés

A Dokumentum osztály a dokumentumfilmeket kezeli, és örökli a Film osztályt. A dokumentumfilmek rendelkeznek egy leírással, amelyet az osztály tárol és kezel.

Attribútumok

- `leiras`: String típusú változó, amely a dokumentumfilm leírását tárolja.

Metódusok

- `Dokumentum(const String&)`: Konstrktor, amely inicializálja a dokumentumfilm leírását.
- `Dokumentum(const Dokumentum&)`: Másoló konstruktor, amely egy másik Dokumentum objektumból hoz létre egy új Dokumentum objektumot.
- `getLeiras()`: Visszaadja a dokumentumfilm leírását (String).
- `setLeiras(const String&)`: Beállítja a dokumentumfilm leírását.
- `Dokumentum(std::istream&)`: Konstrktor, amely egy bemeneti folyamat használ a dokumentumfilm adatok beolvasásához.
- `film_kiir(std::ostream&)`: Kiírja a dokumentumfilm adatait egy kimeneti folyamra.
- `operator==(const Dokumentum&)`: Összehasonlítja két Dokumentum objektumot és visszaad egy logikai értéket (egyenlőség).

Csaladi osztály

Áttekintés

A Csaladi osztály a családi filmeket kezeli, és öröklí a Film osztályt. A családi filmek rendelkeznek egy korhatárral, amelyet az osztály tárol és kezel.

Attribútumok

- korhatar: int típusú változó, amely a családi film korhatárát tárolja.

Metódusok

- Csaladi(int): Konstruktor, amely inicializálja a családi film korhatárát.
- Csaladi(const Csaladi&): Másoló konstruktor, amely egy másik Csaladi objektumból hoz létre egy új Csaladi objektumot.
- getKorhatar(): Visszaadja a családi film korhatárát (int).
- setKorhatar(int): Beállítja a családi film korhatárát.
- Csaladi(std::istream&): Konstruktor, amely egy bemeneti folyamatot használ a családi film adatok beolvasásához.
- film_kiir(std::ostream&): Kiírja a családi film adatait egy kimeneti folyamra.
- operator==(const Csaladi&): Összehasonlít két Csaladi objektumot és visszaad egy logikai értéket (egyenlőség).

Filmtar osztály

Áttekintés

A Filmtar osztály egy filmgyűjteményt kezel, ami heterogén kollekcióként funkcionál. Az osztály egy dinamikus tömböt használ a filmek tárolására, és különféle metódusokat biztosít a filmek hozzáadására, eltávolítására és módosítására.

Attribútumok

- filmek: Kettős mutató a Film típusú objektumokra, amely a filmeket tárolja.
- meret: size_t típusú változó, amely a jelenlegi méretet tárolja.
- max_meret: const size_t típusú változó, amely a maximális méretet tárolja.

Metódusok

- Filmtar(size_t): Konstruktor, amely inicializálja a filmgyűjteményt a maximális mérettel.
- Filmtar(const Filmtar&): Másoló konstruktor, amely egy másik Filmtar objektumból hoz létre egy új Filmtar objektumot.
- getMeret(): Visszaadja a jelenlegi méretet (size_t).
- getMax_meret(): Visszaadja a maximális méretet (const size_t).
- felvetel(Film*): Hozzáad egy új filmet a gyűjteményhez.

- `benne_van(const String&)`: Ellenőrzi, hogy egy adott nevű film benne van-e a gyűjteményben.
- `kereses(const char*)`: Keres egy filmet a megadott név alapján és visszaadja a film mutatóját.
- `modosit(Film*, const String&)`: Módosítja egy film nevét.
- `modosit_overload(Film*, const int)`: Módosítja egy film hosszát.
- `modosit_overload(Film*, const int)`: Módosítja egy film évét.
- `modosit(Dokumentum*, const String&)`: Módosítja egy dokumentum leírását.
- `modosit(Csaladi*, const int)`: Módosítja egy családi film korhatárát.
- `eltavolitas(size_t)`: Eltávolít egy filmet adott index alapján.
- `eltavolitas(const String&)`: Eltávolít egy filmet a gyűjteményből a név alapján.
- `kiir(std::ostream&)`: Kiírja a filmgyűjteményt egy kimeneti folyamra.
- `~Filmtar()`: Destruktor, amely megszünteti a filmeket a gyűjteményben, illetve felszabadítja a filmek tömbjét.
- `operator[] (size_t)`: Visszaad egy filmet a megadott index alapján.
- `operator=(const Filmtar&)`: Értékadás operátor, amely egy Filmtar objektumot másol egy másikba.
- `operator==(const Filmtar&)`: Összehasonlítja két Filmtar objektumot és visszaad egy logikai értéket (egyenlőség).
- `Filmtar(const char*, size_t)`: Konstruktor, amely egy fájl tartalmából állítja elő a gyűjteményt maximális méret megadása mellett.
- `kiir_faj(const char*)`: Kiírja a filmeket egy fájlba.
- `rendezes(P pred)`: Rendezi a filmeket egy megadott predikátum alapján.

Kapcsolatok az osztályok között

String és Film: Egy Film objektumnak van egy String típusú attribútuma (nev), amely a film nevét tárolja. Ez egy 1:1 kapcsolatot jelent.

Film és Filmtar: Egy Filmtar objektum több Film objektumot tartalmazhat (Film** filmek). Ez egy 1:0..* kapcsolatot jelent.

Film és Dokumentum: A Dokumentum osztály örökli a Film osztályt, és további attribútumokat és metódusokat ad hozzá a dokumentumfilmek kezeléséhez. Ez egy 1:1 kapcsolatot jelent.

Film és Csaladi: A Csaladi osztály örökli a Film osztályt, és további attribútumokat és metódusokat ad hozzá a családi filmek kezeléséhez. Ez egy 1:1 kapcsolatot jelent.

Összefoglalás

Az osztálydiagram egy átfogó rendszert mutat be, amely különféle filmek és azok gyűjteményének kezelésére szolgál. Az egyes osztályok különféle funkciókat biztosítanak a film adatok kezelésére, módosítására és kiírására. Az öröklési kapcsolatok lehetővé teszik a kód újrafelhasználhatóságát és bővíthetőségét, míg a dinamikus tömbök használata rugalmas tárolást biztosít.

Felhasználói dokumentáció

Program feladata:

A program egy nyilvántartás jellegű program, mely képes kezelni filmek gyűjteményét. A program képes új filmet hozzáadni adott típus alapján (film, családi, dokumentum). Minden hozzáadott film megjelenik az 5-ös menüpont alatt, amelyek közül, ha valamelyiket törölni szeretnénk azt a 2-es menüpont alatt tehetjük meg. A 6. menüpontban a filmeket rendezni tudjuk, időtartam alapján növekvő és megjelenési év alapján csökkenő sorrendben. Ha szeretnénk megtudni, hogy egy adott film benne van-e a filmtárban, akkor a 3. menüpont alatt rá tudunk keresni a címe szerint. Továbbá, ha módosítani szeretnénk egy adott film információit azt a 4. menüpontban tehetjük meg.

A menürendszer a következőképpen alakul:

```
*-----*
|  FILMTAR KEZELO  |
*-----*
[1] Uj film hozzaadasa
[2] Film torlese
[3] Kereses film neve alapjan
[4] Modositasok
[5] Filmtar kiirasa
[6] Rendezesek
[7] Tesztek
[0] Kilepes
Valasztas: |
```

A menüpontot kiválasztva, ha kéri, akkor megfelelő adatokat kitöltve feltölthetjük az adatbázisba a kívánt új adatokat. Ezt követően a kilépés gombbal menthetjük őket fájlba, ha később is fel szeretnénk használni az információkat.

Legfontosabb feladatokat egyike, hogy mindig kövessük a konzolon kiírt utasításokat és a megfelelő helyeken helyes formátumot adjunk meg. Például, ha menüpontot kell választanunk, akkor ne betűt adjunk, vagy ha megfelelő karakterláncot vár akkor azt helyesen tegyük. Továbbá a program ékezetes karaktereket nem tud kezelni, ezt vegyük figyelembe a feladatok végrehajtásánál. Minden almenüben a tevékenység végrehajtása egyértelmű. Ha esetleges konzol lefagyás vagy probléma tűnne fel, csak zárjuk be a programot és nyissuk meg újra.