

Programozás alapjai 2.

NHF – Dokumentáció

Makay Petra – WQNKWH

Feladat:

Egy könyveket nyilvántartó rendszer egyszerűsített objektummodeljét tervezem meg. A program minden könyvről adatokat tárol, melyeken tud műveleteket végezni.

Specifikáció:

Az egyes könyvekről az alábbi adatok lesznek eltárolva:

- könyv címe
- könyv szerzője
- könyv műfaja
- kiadó
- kiadási év

Ezenkívül szeretnénk tudni, hogy ha az adott könyv esetleg egy sorozathoz tartozik. Abban az esetben, ha így van, akkor a program tárolja a sorozat nevét, valamint azt, hogy az adott könyv a sorozat hányadik része.

A jelenlegi könyvkiadók körében az egyik legnépszerűbb a Könyvmolyképző kiadó, ahol minden kiadott könyv egy külön alcsoportba tartozik. Ezek az alcsoportok határozzák meg egy-egy könyvről, hogy körülbelül milyen korosztálynak íródtak, valamint milyen témájú a könyv. Például a „Vörös pöttyös” könyvek inkább a 16 éven aluliaknak ajánlottak, és többnyire fantasy a műfajuk.

Éppen ezért, ha valamelyik könyvet a Könyvmolyképző kiadó adta ki, pluszban meg kell adni, hogy melyik ilyen alcsoportba tartozik az adott könyv.

Természetesen, ha egy olyan könyvet szeretnénk felvenni a nyilvántartásba, amely sorozatba tartozik, és a Könyvmolyképző kiadó adta ki, azt is meg tudjuk tenni.

Dokumentáció:

Main.cpp:

A main.cpp alapvetően két részből áll: az egyik a gtest_lite tesztjeit futtatja, a másikkal a felhasználó tud műveleteket végrehajtani a programon belül. Egy makró definiálásával lehet váltani a két menü között: ha a CPORTA makró definiálva van, akkor a tesztjelek futnak, ellenkező esetben a rendes menü jelenik meg.

A menü egy try blokkban helyezkedik el, ami az esetleges kivételek keletkezésénél kapja el a kivételeket, és kezeli azokat.

A memtrace.h definiálva van, ez ellenőrzi, ha esetleg memóriaszivárgás van a programban.

Gtest lite tesztek: összesen 16 tesztet futtat a program, amelyek során a függvények helyes működését teszteli. Minden teszt egy metódusáért felel a programnak: külön megnézi a fájlbeolvasást, a konstruktorok helyes működéseit, illetve a nyilvántartásban lévő hozzáad, töröl és keres függvényeket.

Felhasználói menü: abban az esetben, ha a CPORTA makró nincs definiálva, a felhasználó egy menüvel találkozhat a program futtatása során. A menün keresztül különböző utasításokat tud végrehajtani a nyilvántartott könyvekkel, melyeknek a végrehajtásában a menü kiírásai segítenek.

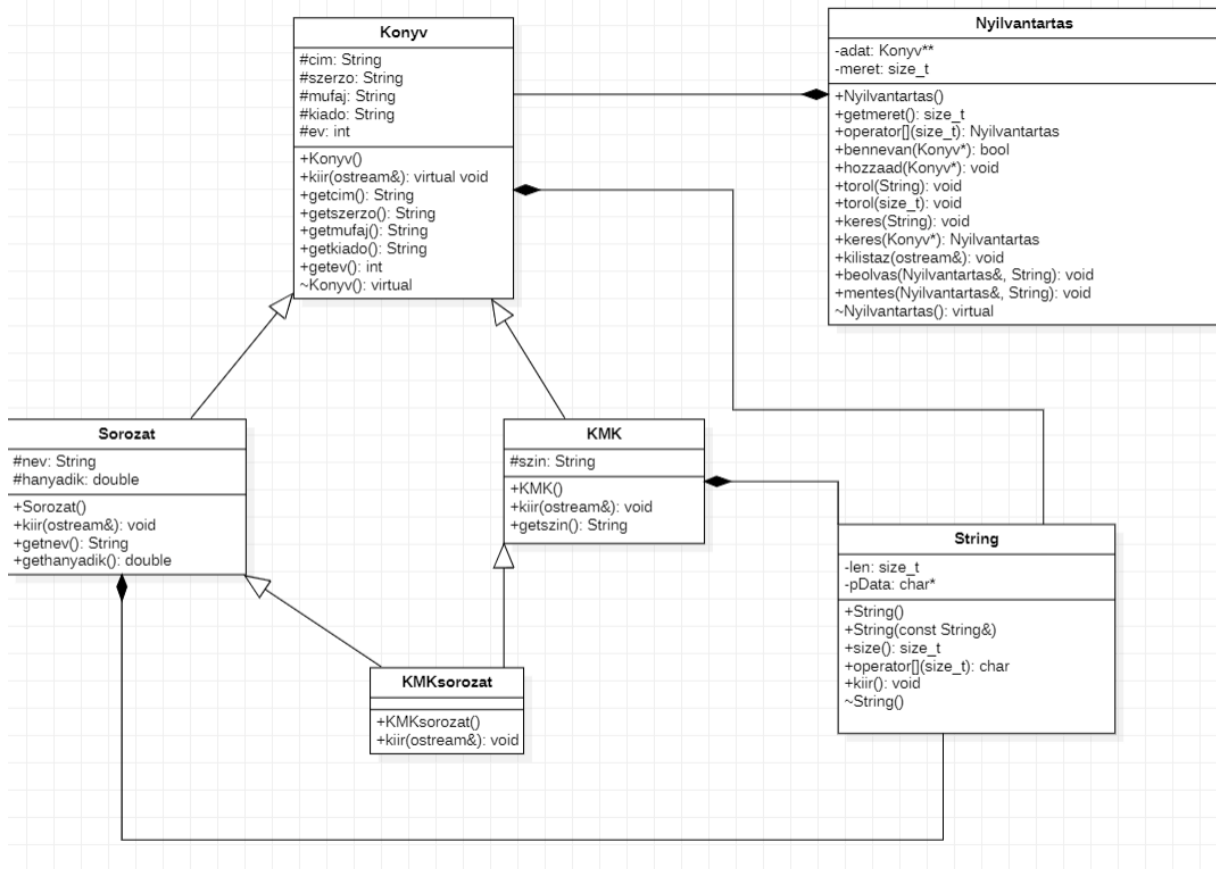
Menü:

A felhasználó egészen addig tud műveleteket végezni, amíg a 0-ás számmal ki nem lép:

- 1 - új könyv hozzáadása: a felhasználó sorban meg tudja adni a felvenni kívánt könyv adatait. A kiadási év esetében, ha hibás formátumot szeretne bevinni, a program szól, és a felhasználó újra próbálkozhat. A menü külön megkérdezi, hogy sorozatról van-e szó, mert abban az esetben kéri a sorozat címét, valamint a sorozatban lévő sorszámát.
- 2 – könyv törlése: a felhasználó kétféleképpen tud törölni: index és könyv cím alapján. A felhasználónak először választania kell a két opció között, ahol hibás formátumért szintén szól a program. Rossz index, valamint olyan könyv esetén, amely nincs benne a nyilvántartásban, kivétel keletkezik.
- 3 – keresés: a felhasználó könyv cím-, vagy szerző alapján tud keresni a nyilvántartásban. A találatokat a program kilistázza. Érvénytelen cím/szerző esetében értesíti a felhasználót, hogy olyan adat nem található a listában.
- 4 – sorozat listázása: a felhasználó ebben az esetben ki tudja listázni egy adott könyvsorozatnak a részeit. A program a találatokat kilistázza, érvénytelen név esetén - az előző ponthoz hasonlóan - a program értesít a hibáról.
- 5 – könyvek listázása: ezzel a menüponttal lehetőség van a nyilvántartásban lévő összes könyvnek a kilistázására.
- 6 – mentés: a felhasználó menteni tudja a változtatásait.

A menüből való választásnál abban az esetben, ha a felhasználó érvénytelen karaktert/számot üt be, a program jelzi, és ez esetben újra próbálkozhat.

Objektum terv:



Az objektummodell-en látszanak az egyes osztályok közötti összefüggések. A KMKsorozat esetében többszörös öröklést használtam, ezért a Sorozat és a KMK osztályok virtuálisan örökölnék a Konyv osztályból.

Nyilvantartas.h – nyilvantartas.cpp:

A heterogén kollekció megvalósításáért a Nyilvantartas osztály felel, itt valósulnak meg a program műveleteit végrehajtó függvények. Tárolja a különböző könyvekre mutató pointereket, valamint a lista méretét is, amelynek nincs maximális értéke.

```
class Nyilvantrtas{
    Konyv** adat;
    size_t meret;
```

Az index operátor esetén kivétel keletkezik, ha egy érvénytelen indexű elemet szeretnénk elérni.

A hozzáadás és a törlés függvények könnyebb működésének érdekében van két segéd bool függvény, amelyek a nyilvántartás adattagjain végighaladva ellenőrzik, hogy egy adott könyv szerepel-e már a listában. Abban az esetben, ha igen, *igaz*, ha pedig nem, akkor *hamis* értékkel térnek vissza.

A hozzáadás és töröl függvények *void* típusúak. A függvénytörzsön belül hoznak létre egy ideiglenes listát, amibe a függvénynek megfelelő adatokat átmásolják, majd ezt teszik egyenlővé az eredeti listával. Abban az esetben, ha a *bool* függvények hamis értéket adnak, *invalid_argument* kivétel keletkezik, amit majd a főprogram fog lekezelni: ez hozzáadásnál abban az esetben fordulhat elő, ha olyan könyvet szeretnénk hozzáadni, ami már szerepel a listában. Törlésnél pedig hibás indexelés, vagy nem létező könyv esetében dob kivételt.

A keres függvény szintén a *bool* függvény segítségével keres a nyilvántartásban. Ha szó alapján kívánunk keresni, azt a cím vagy a szerző megadásával tudjuk megtenni.

A fájlbeolvasás, illetve a fájlkiírás is itt valósul meg. A nyilvántartásban lévő könyveket egy *.txt* fájlból olvassa be, majd mentés esetén oda is menti vissza azokat. Sikertelen fájlbeolvasás esetén a program kivételt dob, amit szintén a menü kezel le.

String.h – string.cpp:

Az osztályokban saját *String* osztályt használok, amelyet az egyik korábbi laborfeladat alapján hoztam létre. Több konstruktorral rendelkezik, melyek különböző típusú adatokból hoznak létre *String*-eket.

Saját másoló konstruktorral, értékadó- és index operátorral, valamint összehasonlító operátorral rendelkezik. Az index operátor hibás index esetén kivételt dob.

A *c_str()* függvény egy *const char** stringet ad vissza, amit elsősorban a kiíratásnál használ a program.

Konyv.h – Sorozat.h – KMK.h:

A heterogén kollekció 4 osztállyal (*könyv*, *sorozat*, *kmk*, *kmksorozat*) dolgozik. Egy közös ősosztály van, a *Konyv*. Mivel a programban *diamond* öröklést használok, ezért a *Sorozat*, illetve a *KMK* osztályok virtuálisan örökölnék az ősosztályból.

```
class Sorozat : virtual public Konyv {  
protected:  
    String nev;  
    double hanyadik;
```

Minden öröklés publikus, tehát a leszármazott osztály az ősosztály minden adattagját és függvényét örökli.

Minden osztálynak 2 konstruktora van: az egyik a default konstruktor, amit minden hozzáadásnál használ a program. A másik konstruktor *std::string*-ekből készít *String*et, amit a fájlbeolvasásnál használok.

Minden osztályban két kiírató függvény található, ahol az egyik a listázásért felel, míg a másik a fájlba való kiíráskor hívódik meg.

Az öröklődés miatt a *Konyv* osztály destruktora, valamint kiírató függvényei virtuálisak.

2023.05.22.