HÁZI FELADAT

Programozás alapjai 2.

NHF 4. - Dokumentáció
Könyvtár

Brosig Márton János

A0897X

2022.05.01

Feladat

Könyvtár

Készítsen könyveket nyilvántartó rendszert. Minden könyvnek tároljuk a címét, oldalainak számát és kiadási évét. Bizonyos kalandregényeknél korhatár is van, a szépirodalmi művek esetében egy szöveges leírást is tárolunk. Tervezzen könnyen bővíthető objektummodellt a feladathoz!

Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz **ne** használjon STL tárolót!

Feladatspecifikáció

A program képes könyvek adatait tárolni, konfigurációs fájlból beolvasni. A programban lehetőség van új könyv hozzáadására, illetve a régebben felvett könyvek adatainak megváltoztatására. STL tároló hiányában a program saját dinamikus adatszerkezetekkel dolgozik. (String, Konyvtar)

Fájlkezelés

A program képes .txt fájlba kimenteni az éppen aktuális adatbázist. A program lehetőséget ad régebbi .txt mentés fájl betöltésére is, amennyiben a megadott fájl hibás formátumú, vagy üres a program "const char *" típusú kivételt dob.

Kezelt fájl formátuma

(_ = space, \n = enter)

tipus_könyvneve__oldalakszama_kiadaseve \n opcionalis(korhatar/leiras) \n

Felhasználói felület, tesztelés

A program nem rendelkezik felhasználói felülettel, helyességének tesztelése tesztprogram segítségével történik, amely a program minden funkcionalitását használja, igy minden alrész tesztelésre kerül.

Hibakezelés

Ha a program hibás bementet kap (pld: név nélkül adnak hozzá egy könyvet), szabványos kimenetén jelzi a hiba pontos okát. Fájlkezelés esetén a program feltételezi, hogy a felhasználó csak a program által készített fájlokat olvastatja be.
Minden egyéb esetben a program char * típusú hibát kell dobjon, ami nem elvárt bemenetre utal.

Pontosított funkciók

- **könyv adatbázis betöltése**
- **könyv keresése**
- **könyv adatainak módosítás**
- **könyv hozzáadása**
- **könyv törlése**

A programnak képesnek kell lennie már kimentett adatbázisból könyveket betölteni, azokat adatszerkezetbe beolvasni, majd kezelés után ugyanolyan formában visszamenteni. Képes könyveket bizonyos tulajdonságaik, címük egy részlete alapján megtalálni, majd kilistázni a tulajdonságait. Egy könyv megkeresése után lehetőség van annak adatainak módosítására. Lehetőség van új könyvet hozzáadni az adatbázishoz, ekkor a felhasználó feladata megfelelő formában megadni a könyv adatait. Amennyiben egy könyv elavult vagy elveszett, lehetőség van könyvet törölni a készletből.

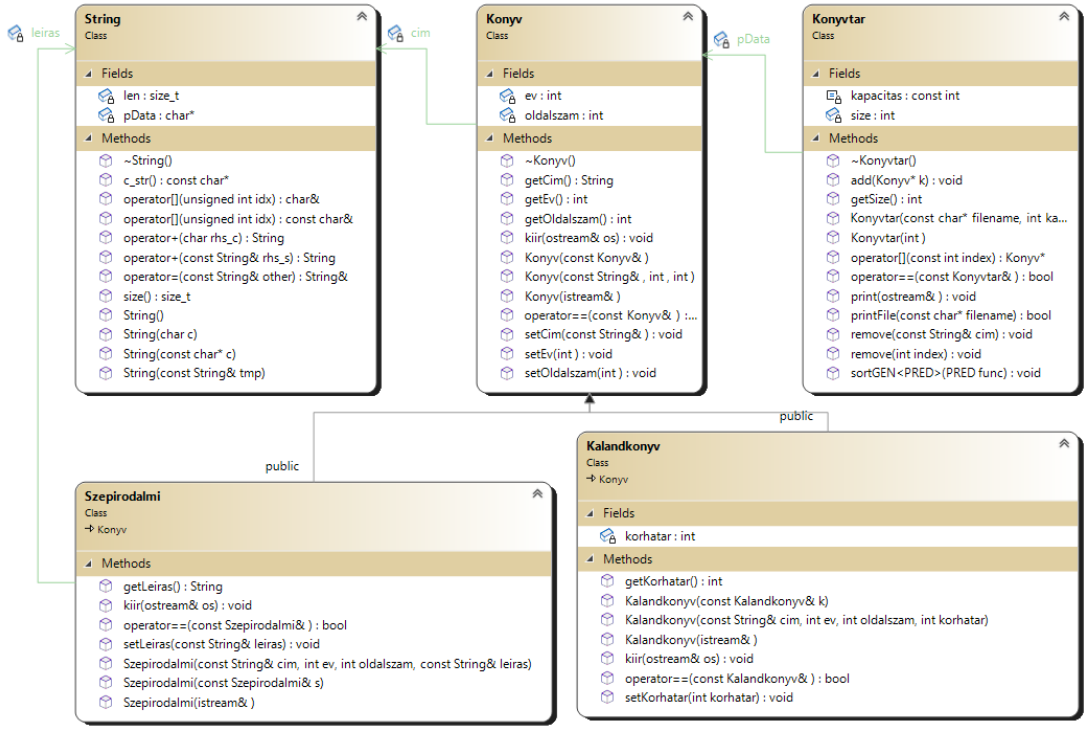
Bővíthetőség lehetősége

A program könnyedén bővíthető új könyv típusokkal, amelyek újabb tulajdonságokkal rendelkezhetnek. A fő adatszerkezet könnyedén bővíthető új funkciókkal. (pld: könyv kölcsönadás / értékelés)

Objektum terv és algoritmusok

Osztálydiagram, osztályok változói

Jelölések: (szürke nyíl: ősosztály felé mutat, zöld nyíl: objektum adattag osztályban)



Osztály: String

A String osztály char * típusú karaktersorozatok dinamikus tárolását fogja elvégezni. Az osztályban felmerülő algoritmusok specifikációját és megoldását az 5. laborfeladat alapján fogom elkészíteni.

Osztály: Konyv

A könyvtípusok ősosztálya, settereken, gettereken és konstruktorokon kívül két virtuális függvénnyel rendelkezik: a kiir(ostream& os) függvény virtualitása a heterogén kollekción belüli használathoz elengedhetetlen, a virtuális függvény virtualitása pedig az öröklődés miatt kötelező.

Osztály: Szepirodalmi - Konyv leszármazott

A szépirodalmi művek tárolására hivatott osztály plusz adattagja egy String típusú leirás, tartalmaz ehhez tartozó gettert, settert illetve a kiir(ostream& os) függvény felüldefiniált változatát.

Osztály: Kalandkonyv- Konyv leszármazott

A kalandkönyvek tarolasara alkalmas osztály az ősosztályhoz képest a korhatár adat taggal bővült, szintén tartalmaz ehhez tartozó settert és gettert illetve tartalmazza a kiir(ostream& os) függvény felüldefiniált változatát.

Osztály: Konyvtar

A konyvtar osztály könyvek tárolására alkalmas. Adattagja a könyvtár maximális mérete, az aktuális mérete, illetve a könyvtár könyveire mutató pointerek tömbje. A könyvtár egy heterogén kollekció. A heterogén kollekció megkönnyíti a könyvtárban való keresést, illetve a bővíthetőség könnyedségét is

növeli.

Fontosabb algoritmusok:

Könyvtár::Fájlba kiírás

Fájlba kiírás esetén, a program végig iterál a könyveket tartalmazó tömbön, és minden elemére meghívja a kiir(ostream& os) függvényt. Minden példány esetén az objektum típusának megfelelő függvény fog meghívni, így a leszármazottak adattagjai is kiírásra kerülnek. Mivel a kiir függvény std::ostream&-na ir, ezért a kiir függvények a szabványos kimeneten kívül fájlba kiírás is használhatóak.

Könyvtár::Fájlból beolvasás

A fájlból való beolvasás az egyik konstruktoron keresztül érhető el, melynek két paramétere a fájl neve, illetve a felveendő könyvtár maximális mérete. Ezután a függvény adott számú sort beolvas és az első szó alapján a megfelelő konstruktort meghívva hozzáadja a heterogén kollekcióhoz.

Könyvtár::rendezésABC, rendezésÉv, rendezésOldalak

Lehetőség van a könyv adatbázis rendezésére, például hasznos lehet valamilyen felhasználói kérés kielégítésére. Minden rendezés esetén buborék-rendezést használ a függvény, a Programozás alapjai 1. tárgyban tanult módszerrel.

Tesztprogram működése

A tesztprogram létrehoz egy könyvekből álló könyvtárat, minden könyvfajta minden konstruktorát kipróbálva. Ezt a könyvtárat kimenti egy fájlba, majd a fájlból beolvassa az adatbázist, majd mindhárom rendezést futtatja rá. Minden lépés után ellenőrzi a lépés helyességét. A program továbbá ellenőrzi a memória szivárgást is.

Dokumentáció

konyv.h és konyv.cpp

Rövid leírás: Ősosztály, ami egy könyv adatait tárolja.

class Konyv

String cim - A könyv címe.
int ev - A könyv kiadásának éve.
int oldalszam - A könyv oldalainak száma.

Publikus tagfüggvények:

Konstruktor.
Paraméter: cim - A könyv címe.
Paraméter: ev - A könyv kiadásának éve.
Paraméter: oldalszam - A könyv oldalainak száma.
Konyv(const String&, int, int)

Összefoglaló: Másoló konstruktor, ami a könyv adatait másolja.
Paraméter: k - Könyv, ami másolódik.
Konyv(const Konyv&)

Összefoglaló: Létrehoz egy könyvet a beolvasott adatok alapján. (egy sorból)
Konyv(std::istream&)

Összefoglaló: Getter függvény a könyv címéhez.
Visszatérési érték: String - a könyv címe.
String getCim() const

Összefoglaló: Getter függvény a könyv kiadásának évehez.
Visszatérési érték: int - a könyv kiadásának éve.
int getEv() const

Összefoglaló: Getter függvény a könyv oldalainak számához.
Visszatérési érték: int - a könyv oldalainak száma.
int getOldalszam() const

Összefoglaló: Setter függvény a könyv címéhez.
Paraméter: cim - a könyv címe.
void setCim(const String&)

Összefoglaló: Setter függvény a könyv kiadásának évehez.
Paraméter: ev - a könyv kiadásának éve.
void setEv(int)

Összefoglaló: Setter függvény a könyv oldalainak számához.
Paraméter: oldalszam - a könyv oldalainak száma.

void setOldalszam(int)

Összefoglaló: Kiírja a könyv adatait.
Visszatérési érték: standard kimenet a kiírás helye, nincs visszatérési érték.
virtual void kiir(std::ostream& os) const

Összefoglaló: Egyenlőség operátor, leszármazottakban felül kell definiálni.
Visszatérési érték: true - a két könyv megegyezik
Visszatérési érték: false - a két könyv nem egyezik meg
virtual bool operator==(const Konyv&) const

virtual ~Konyv() {}

A könyv leszármazott osztálya, ami egy kalandkönyv adatait tárolja.

class Kalandkonyv : public Konyv

int korhatar - A könyv korhatárát tárolja.

Publikus tagfüggvények:

Konstruktor.
Paraméter: cim A könyv címe.
Paraméter: ev A könyv kiadásának éve.
Paraméter: oldalszam A könyv oldalainak száma.
Paraméter: korhatar A könyv korhatárát tárolja.
Kalandkonyv(const String& cim, int ev, int oldalszam, int korhatar)

Összefoglaló: Másoló konstruktor, ami egy kalandkönyv adatait másolja.
Paraméter: k - Kalandkonyv, ami másolódik.
Kalandkonyv(const Kalandkonyv& k)

Összefoglaló: Létrehoz egy kalandkönyvet a beolvasott adatok alapján. (egy sorból)
Kalandkonyv(std::istream&)

Összefoglaló: Setter függvény a könyv korhatárához.
Paraméter: korhatar - a könyv korhatára.
void setKorhatar(int korhatar)

Összefoglaló: Getter függvény a könyv korhatárához.
Visszatérési érték: int - a könyv korhatára.
int getKorhatar() const

Összefoglaló: Kiírja a könyv adatait.
Visszatérési érték: standard kimenet a kiírás helye, nincs visszatérési érték.
void kiir(std::ostream& os) const

Összefoglaló: Egyenlőség operátor felüldefiniálása.
Visszatérési érték: true - ha egyező a két könyv

Visszatérési érték: false - ha nem egyező a két könyv
bool operator==(const Kalandkonyv&) const

A könyv leszármazott osztálya, ami egy szépirodalmi adatait tárolja.

class Szepirodalmi : public Konyv

String leiras - A könyv leírását tárolja.

Publikus tagfüggvények:

Összefoglaló : Konstruktor.
Paraméter: cim A könyv címe.
Paraméter: ev A könyv kiadásának éve.
Paraméter: oldalszam A könyv oldalainak száma.
Paraméter: leiras A könyv leírását tárolja.
Szepirodalmi(const String& cim, int ev, int oldalszam, const String& leiras)

Összefoglaló: Másoló konstruktor, ami egy szepirodalmi mű adatait másolja.
Paraméter: s - Szepirodalmi mű, ami másolódik.
Szepirodalmi(const Szepirodalmi& s)

Összefoglaló: Létrehoz egy szepirodalmi műt a beolvasott adatok alapján. (egy sorból)
Szepirodalmi(std::istream&)

Összefoglaló: Setter függvény a könyv leírásához.
Paraméter: leiras - a könyv leírására.
void setLeiras(const String& leiras)

Összefoglaló: Getter függvény a könyv leírásához.
Visszatérési érték: String - a könyv leírására.
String getLeiras() const

Összefoglaló: Kiírja a könyv adatait.
Visszatérési érték: standard kimenet a kiírás helye, nincs visszatérési érték.
void kiir(std::ostream& os) const

Összefoglaló: Egyenlőség operátor felüldefiniálása.
Visszatérési érték: true - ha egyező a két könyv
Visszatérési érték: false - ha nem egyező a két könyv
bool operator==(const Szepirodalmi&) const

konyvtar.h és konyvtar.cpp

class Konyvtar

Konyv** pData
int size
const int kapacitas

Publikus tagfüggvények:

Konstruktor.

Paraméter: kapacitas A könyvtar kapacitása.
Konyvtar(int)

Összefoglaló: Beolvas egy lemetett könyvtárat egy szöveges fileból.
Paraméter: filename - a file neve.
Konyvtar(const char* filename, int kapacitas)

Összefoglaló: Hozzáad egy könyvet a könyvtarhoz.
Paraméter: k - hozzáadandó könyv.
void add(Konyvk)

Összefoglaló: Eltávolít egy könyvet a könyvtarból, index alapján.
Paraméter: index - az eltávolítandó könyv indexe.
void remove(int index)

Összefoglaló: Eltávolít egy könyvet a könyvtarból, cím alapján.
Paraméter: cim - az eltávolítandó könyv címe.
void remove(const String& cim)

Összefoglaló: Kiírja a könyvtar tartalmát tetszőleges std::ostream& objektumba.
void print(std::ostream&) const

Összefoglaló: Kiírja a könyvtar tartalmát egy fileba.
Paraméter: filename - a fájl neve.
Visszatérési érték: true - ha sikeresen kiírta a fileba.
Visszatérési érték: false - ha nem sikerült kiírni a fileba.
bool printFile(const charfilename) const

Összefoglaló: Visszaadja a könyvtár jelenlegi kapacitását
Visszatérési érték: int - a kapacitás
int getSize() const

Összefoglaló: Destruktor.
~Konyvtar()

Összefoglaló: Index alapján visszaad egy könyvet.
Paraméter: index
Visszatérési érték: Konyv- a könyvre mutató pointer.
Konyv* operator[](const int index) const

Összefoglaló: Egyenlőség operátor könyvtárakra, a könyvek sorrendje is számít.
Visszatérési érték: true - a két könyvtár megegyezik.
Visszatérési érték: false - a két könyvtár nem egyezik.
bool operator==(const Konyvtar&) const

Összefoglaló: Sablon függvény, ami predikátum segítségével rendezi a könyvtárat, a paramétereként átadott függvényt használva.

Sablonparaméter PRED - predikátum függvény

Paraméter: func

```
template<typename PRED>
void sortGEN(PRED func)
```

Globálisan megvalósított, osztályhoz tartozó függvények:

Összefoglaló: Összehasonlít két könyvet, hogy melyikük van előbb a könyv címe alapján.
Paraméter: k1
Paraméter: k2
Visszatérési érték: true
Visszatérési érték: false
bool compareABC(Konyv constk1, Konyv constk2)

Összefoglaló: Összehasonlítja a két könyvet, hogy melyikük van előbb a könyv kiadási éve alapján.
Paraméter: k1
Paraméter: k2
Visszatérési érték: true
Visszatérési érték: false
bool compareYear(Konyv constk1, Konyv constk2)

Összefoglaló: Összehasonlít két könyvet, hogy melyikük van előbb oldalak száma alapján.
Paraméter: k1
Paraméter: k2
Visszatérési érték: true
Visszatérési érték: false
bool comparePages(Konyv constk1, Konyv constk2)

string.h és string.cpp

class String

char *pData - pointer az adatra
size_t len - hossz lezáró nulla nélkül

Publikus tagfüggvények:

Összefoglaló: Konstruktor, ami létrehozza egy üres stringet.

```
String()
```

Összefoglaló: Megadja a string hosszát.
Visszatérési érték: size_t - a string hossza.

```
size_t size() const
```

Összefoglaló: Visszaad egy \0 val lezárt karaktersorozatot.
Visszatérési érték: const char*

const char* c_str() const

Összefoglaló: Konstruktor, ami egy karakterből hoz létre egy String objektumot
Paraméter: c
String(char c)

Összefoglaló: Konstruktor, ami egy String objektumot hoz létre karaktersorozatra mutató pointer segítségével.
Paraméter: c - a karaktersorozat.
String(const char *c)

Összefoglaló: String osztály desktruktora
~String()

Összefoglaló: Másoló konstruktor a String osztályhoz.
Paraméter: tmp
String(const String &tmp)

Összefoglaló: értékadás operátor a String osztályhoz.
Paraméter: other - a másolandó String objektum.
Visszatérési érték: String&
String& operator=(const String& other)

Összefoglaló: Összeadás operátor a String osztályhoz.
Paraméter: rhs_s - a hozzáadandó String objektum.
Visszatérési érték: String
String operator+(const String& rhs_s) const

Összefoglaló: Karakter hozzáfűzés egy string objektumhoz.
Paraméter: rhs_c - a hozzáfűzendő karakter.
Visszatérési érték: String
String operator+(char rhs_c) const

Összefoglaló: Indexelő operátor a String osztályhoz.
Paraméter: idx
Visszatérési érték: char&
char& operator[](unsigned int idx)

Összefoglaló: Indexelő operátor a String osztályhoz.
Paraméter: idx
Visszatérési érték: char
const char& operator[](unsigned int idx) const

Globálisan megvalósított, osztályhoz tartozó függvények:

Összefoglaló: Inserter operátor a String osztályhoz.
Paraméter: os - a kiírás helye
Paraméter: s0 - a kiírandó String objektum
Visszatérési érték: std::ostream&
std::ostream& operator<<(std::ostream& os, const String& s0)

Összefoglaló: Extractor operátor a String osztályhoz.
Paraméter: is - a beolvasás helye
Paraméter: s0 - a beolvasandó String objektum
Visszatérési érték: std::istream&
std::istream& operator>>(std::istream& is, String& s0)

Összefoglaló: Karakter és string összeadását megvalósító függvény.
Paraméter: ch - a karakter
Paraméter: str - a string
Visszatérési érték: String
inline String operator+(char ch, const String& str)

Összefoglaló: Összehasonlító operátor a String osztályhoz.
Paraméter: s1
Paraméter: s2
Visszatérési érték: true
Visszatérési érték: false
bool operator==(const String& s1, const String& s2)

Összefoglaló: Nagyobb-e operátor a String osztályhoz.
Paraméter: s1
Paraméter: s2
Visszatérési érték: true
Visszatérési érték: false
bool operator>(const String& s1, const String& s2)