**HÁZI FELADAT**

Programozás alapjai 2.

NHF 2. - Terv  
**Könyvtár**

Brosig Márton János

A0897X

2022.03.30

[Feladat 1](#_Toc222108298)

[Könyvtár 1](#_Toc404120063)

[Feladatspecifikáció 2](#_Toc1200066881)

[Fájlkezelés 2](#_Toc2144511221)

[Kezelt fájl formátuma 2](#_Toc53832280)

[Felhasználói felület, tesztelés 2](#_Toc1701465625)

[Hibakezelés 2](#_Toc2057205897)

[Pontosított funkciók 2](#_Toc349273493)

[Bővíthetőség lehetősége 2](#_Toc726215871)

[Objektum terv és algoritmusok 3](#_Toc337066424)

[Osztálydiagram, osztályok változói 3](#_Toc2075601396)

[Osztály: String 3](#_Toc217818713)

[Osztály: Konyv 3](#_Toc1304448665)

[Osztály: Kalandkonyv- Konyv leszármazott 4](#_Toc113598477)

[Osztály: Konyvtar 4](#_Toc900941359)

[Könyvtár::Fájlba kiírás 4](#_Toc1164220131)

[Könyvtár::rendezésABC, rendezésÉv, rendezésOldalak 4](#_Toc2018961532)

# **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 kielégitésére. 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.