

Név: **Drahos Zsolt**

Neptun-kód: **UCZFU3**

# Tartalom

<b>TARTALOM .....</b>	<b>2</b>
<b>FELHASZNÁLÓI DOKUMENTÁCIÓ.....</b>	<b>3</b>
FELADAT: .....	3
KÖRNYEZET:.....	3
HASZNÁLATA: .....	3
<i>A program indítása .....</i>	<i>3</i>
<i>Program bemenete .....</i>	<i>3</i>
<i>Adatok megjelenítése .....</i>	<i>4</i>
<i>Módosít funkció.....</i>	<i>5</i>
<i>Töröl funkció.....</i>	<i>5</i>
<i>Kilépés.....</i>	<i>5</i>
<b>FEJLESZTŐI DOKUMENTÁCIÓ.....</b>	<b>6</b>
FELADAT: .....	6
ADATOK TÁROLÁSA: .....	6
FUNKCIÓK .....	6
ADATSZERKEZET .....	6
A PROGRAM MŰKÖDÉSÉT VEZÉRLŐ FŐ FÜGGVÉNYEK .....	7
<i>Konyv *beolvas(Konyv *eleje);.....</i>	<i>7</i>
<i>Konyv *fajlbol_olvas(Konyv *eleje);.....</i>	<i>7</i>
<i>void mentes(Konyv *eleje);.....</i>	<i>7</i>
A PROGRAM FUNKCIÓT VEZÉRLŐ FÜGGVÉNYEK .....	7
<i>Konyv *torol(Konyv *eleje, int hanyadik);.....</i>	<i>7</i>
<i>Konyv *modosit(Konyv *eleje, int hanyadik);.....</i>	<i>7</i>
<i>void keres_~cim/~szerzo/~ev/~tema(Konyv *eleje);.....</i>	<i>7</i>
<i>void kiir(Konyv *eleje);.....</i>	<i>7</i>
A PROGRAM MENÜRENDSZERÉRÉT FELELŐS FÜGGVÉNYEK.....	8
<i>void menu_kiir();.....</i>	<i>8</i>
<i>void menu_lista(int menu, Konyv *eleje);.....</i>	<i>8</i>
KÖNYVTÁRAK .....	8

# FELHASZNÁLÓI DOKUMENTÁCIÓ

## Feladat:

A feladat egy olyan program készítése, amely szövegfájlból képes beolvasni a könyvek adatait, eltárolni azokat és utána felhasználni különböző funkciókra. Ilyen funkció például a keresés, adatok hozzáadása, törlése és módosítása.

## Környezet:

.exe kiterjesztésű program futtatására alkalmas operációs rendszer (pl.: Windows 10). A program működéshez nem szükséges semmilyen más szoftver megléte.

## Használata:

### *A program indítása*

Ahhoz, hogy a program működjön el kell indítani a `NHF_Konyvtar.exe` fájlt melyet a `~\NHF_Konyvtar\bin\Debug` mappában talál.

### *Program bemenete*

A program az adatokat képes beolvasni fájlból. Ahhoz, hogy a program be tudja olvasni kell lennie egy „`adatok.txt`” nevű fájlnek, melynek pontosan az `NHF_Konyvtar.exe` fájlal kell egy mappában lennie. A fájlban az adatoknak ``;` -vel kell lennie elválasztva.

Továbbá a program **képes adatokat hozzáadni** konzol bemenetről. Ezt a funkciót az „1” szám beütésével érheti el a felhasználó. Ekkor a program megkérdezi egyesével a hozzáadandó könyv címét, szerzőjét, kiadási évét (dátumát) és témáját. Ha a hozzáadás sikerült, akkor a végén egy „üzenet” jelenik meg, hogy sikerült hozzáadni.

```
1
Cím: Egri Csillagok
Szerzo: Gardonyi Geza
Ev: 1899
Tema: Tortenelmi
Hozzáadva!
```

Példa a `NHF_Konyvtar.exe` klaviatúra-inputjára

### Adatok megjelenítése

A program többféle képpen **képes megjeleníteni az adatokat**. Ha a kettes menüpontot választja a felhasználó („2”-es szám begépelése a menüben) akkor az összes könyv adatait láthatjuk kilistázva, egymás után felsorolva és számozva. Továbbá a végén kiírja, hogy összesen hány darab adat található az adatszerkezetben.

```
1. Könyv:
Cím: Variety Lights
Szerzo: Hestia McGilmartin
Ev: 2010
Tema: in

2. Könyv:
Cím: My Little Business
Szerzo: Berenice Birtchnell
Ev: 2001
Tema: dui

3. Könyv:
Cím: Egri Csillagok
Szerzo: Gardonyi Geza
Ev: 1899
Tema: Tortenelmi

A nyilvántartásban 3 darab könyv található!
```

Példa a NHF\_Konyvtar.exe listát funkciójára

Ezen felül van egy funkció, ami képes **keresni az adatok között**. Ha a négyes menüpontot választja a felhasználó („4”-es szám begépelése a menüben) akkor megjelenik egy almenü ahol négy opcióból választva szűkíthetjük a keresési eredményeinket. Az négy opció a „Cím szerinti keresés”, „Szerző szerinti keresés”, „Év szerinti keresés”, „Téma szerinti keresés”.

Mind a négy keresés, nagyon hasonlóan működik. A megfelelő kiválasztása után a program megkérdezi a keresési feltételt, hogy mi alapján szűrjön és kilistázza a talált adatokat. A keresés külön figyel arra, hogyha esetleg nem tudja a felhasználó pontosan mit keres csak egy két szót vagy akár betűt a címből, témából, évből vagy szerzőből akkor is megtalálja az adatokat. Tehát nem kell pontosan begépelni a felhasználónak az keresési feltétel elég csak az első pár betűt és akkor is kiadja az eredményt.

```
---Keresés---
```

```
1. Cím alapján
2. Szerző alapján
3. Év alapján
4. Téma alapján

Kérem válasszon az alábbi felsorolásból!
```

```
1
Cím: Egri

20. Könyv:
Címe: Egri Csillagok
Írója: Gardonyi Geza
Megjelenési éve: 1899
Témaja: Tortenelmi
```

Példa a NHF\_Konyvtar.exe keresés funkciójára

### Módosít funkció

További funkció a programban, hogy a felhasználó képes az adatokat módosítani. A hármas menüpontot választva a felhasználó meghatározhatja, hogy melyik adatot szeretné módosítani. Ezt az adat sorszámának begépelésével teheti meg.

Miután kiválasztotta, hogy hányadikat szeretné módosítani ismét egy felhasználói interakció jelenik meg. Itt megjelenik a kiválasztott sorszámú könyv és még egyszer megkérdezi, hogy biztos ezt szeretné-e módosítani. Ezek után ha „igen”-t gépel be akkor megjeleni az Új adat felvételéhez nagyon hasonló input.

```
Hányadikat szeretné módosítani? 20

20. Könyv:
Címe: Egrí Csillagok
Szerző: Gardonyi Geza
Megjelenési éve: 1899
Témája: Történelmi

Biztosan ezt szeretné módosítani? igen

Cím: Harry Potter
Szerző: J.K.Rowling
Ev: 2008
Téma: Fantasy

Sikeres módosítás!
```

Példa a NHF\_Konyvtar.exe módosít funkciójára

### Töröl funkció

A módosításon, keresésen, listázáson és új hozzáadásán kívül van még egy nagyon hasznos funkció, a törlés. A menüben az 5-ös szám begépelésével lehet előhívni. Itt is hasonlóan mint a módosításnál a felhasználó adhatja meg, hogy hányadikat szeretné törölni az adatbázisból. Miután kiválasztotta, van még egy végső ellenőrzés, hogy biztosan ezt szeretné-e törölni, ha igen-t választ akkor már meg is történt az adat törlése az adatbázisból.

```
Hányadikat szeretné törölni? 20

20. Könyv:
Címe: Harry Potter
Szerző: J.K.Rowling
Megjelenési éve: 2008
Témája: Fantasy

Biztosan ezt szeretné törölni? igen

Sikeres törlés!
```

Példa a NHF\_Konyvtar.exe módosít funkciójára

### Kilépés

Kilépés menüpontban a felhasználó be tudja zárni a programot és elmenteni a módosításokat fájlba.

**FONTOS!!** Ha módosít, vagy töröl az adatbázisból a felhasználó akkor minden képpen a kilépés funkcióval (6-os szám begépelésével) tegye ezt meg. Ha nem ezzel történik a kilépés akkor a fájl (adatok.txt) nem fog frissülni a törölt és/vagy módosított adattal.

# FEJLESZTŐI DOKUMENTÁCIÓ

## Feladat:

A feladat egy olyan program készítése, amely szövegfájlból képes beolvasni a könyvek adatait, eltárolni azokat és utána felhasználni különböző funkciókra. Ilyen funkció például a keresés, adatok hozzáadása, törlése és módosítása.

## Adatok tárolása:

A program az adatokat a szöveges fájlban a következő formátumban tárolja el:

```
konyv_cime;szerzo;kiadas_ev;tema
```

Az adatok pontosvesszővel vannak elválasztva, de az adatokban lehetnek tetszőleges számú szóközők. Pl.:

```
Egri csillagok;Gárdonyi Géza;1899;történelmi
```

## Funkciók

*Menü, Új könyv hozzáadása, Kilrás, Keresés, Módosítás, Törlés, Fájlban való eltárolás, Fájlba írás*

A program úgy lett kialakítva, hogy egy menü rendszer működik a „Főoldalon”, ahol a megfelelő input megadása után meghív egy függvényt. A beolvasásos függvények (fájlból olvas, konzolról olvas) a `beolvas.c` fájlban míg a funkciók `funkcio.c` fájlban találhatóak. Az összes függvény fejléce és a láncolt lista pedig a `header.h` fejlécfájlban található

## Adatszerkezet

Az adatokat egy egyszerű láncolt listában tárolódnak el melyre a fejléc fájlban található. Mind a cím, szerző, év és téma karakterekből álló dinamikus tömbök.

```
typedef struct Konyv{
    char *cim;
    char *szerzo;
    char *ev;
    char *tema;
    struct Konyv *kov;
}Konyv;
```

## A program működését vezérlő fő függvények

```
Konyv *beolvas(Konyv *eleje);
```

A függvény konzolról képes beolvasni a könyv adatait és hozzáadja a láncolt lista végéhez. Az egyetlen paraméter a láncolt lista, ami könyv struktúrát tartalmazza. A beolvasásnál az adatok bármilyen hosszúak lehetnek, és akárhány darab szóköz lehet köztük. Visszatérési értéke a módosított láncolt lista.

```
Konyv *fajlbol_olvas(Konyv *eleje);
```

A függvény képes az „adatok.txt” fájlból beolvasni adatokat. Az egyetlen paraméter a láncolt lista, ami könyv struktúrát tartalmazza. Ahhoz, hogy a beolvasás működjön az adatoknak a fájlban pontosan követni kell a formátumot. Visszatérési értéke a módosított láncolt lista. A érték maximum 100 karakter hosszú lehet.

```
void mentes(Konyv *eleje);
```

A függvény képes a láncolt lista adatait fájlba menteni. A függvény úgy lett megírva, hogy a eleje paraméterrel meghatározott listát tudja elmenteni az „adatok.txt” -be. A függvényben egy számláló ciklus fut, ami végig megy a listán és egyesével kiírja az adatokat megfelelő formátumban.

## A program funkciót vezérlő függvények

```
Konyv *torol(Konyv *eleje, int hanyadik);
```

A töröl függvény képes törölni adatot az adatbázisból. A két paraméterből az első a láncolt lista, ami könyv struktúrát tartalmazza, második meg hogy hanyadik elemet szeretnénk törölni. Egy ciklussal végig megyünk a listán és ha az index egyezik azzal a számmal, amit a felhasználó adott meg akkor, eltávolítjuk az adatot a listából. Visszatérési értéke a módosított láncolt lista.

```
Konyv *modosit(Konyv *eleje, int hanyadik);
```

A módosít függvény nagyon hasonlóan működik, mint a töröl függvény, annyiban különbözik, hogy a függvény nem töröl, hanem hozzáad egy értéket a meghatározott helyre. Adatbevitel úgy úgy működik, mint az új hozzáadásakor. Visszatérési értéke a módosított láncolt lista.

```
void keres_~cim/~szerzo/~ev/~tema(Konyv *eleje);
```

Ez a függvény képes cím/szerző/év/téma alapján keresni az adatokban. Inputról megkérdezi mit szeretne keresni, begépel a felhasználó és meghatározza a keresési eredményt.

```
void kiir(Konyv *eleje);
```

Ez a függvény a legegyszerűbb függvény. Itt egy számláló ciklussal végig megyünk az adatokon és egyesével kiírjuk őket. A függvény paramétere a láncolt lista.

## A program menürendszerét felelős függvények

```
void menu_kiir();
```

A függvény egy ábrát rajzol ki, ahol a menüpontok vannak megjelenítve. Az ábrák az <https://www.asciart.eu/> weboldalról van, a menü pontok kiegészítésével.

```
void menu_lista(int menu, Konyv *eleje);
```

Ennek a függvénynek pedig a menürendszer megvalósításában nagy szerepe. Az első paraméter meghatározza, hogy a felhasználó melyik menüpontot választja, a második pedig a láncolt lista. Egy több irányú elágazás segítségével határozza meg, hogy éppen mit választott a felhasználó.

## Könyvtárak

A program, hogy színes elemeket tudjon megjeleníteni az `econio` könyvtárat alkalmazza. Továbbá a `debugmalloc` könyvtárat melyet memóriaszivárgás keresésére használ.