

Programozói Dokumentáció

Szükséges környezet: *C fordító, debugmalloc.h*

Fordítás

```
gcc main.c adatbazis.c menu.c record.c vcard.c -o telkonyv
```

A project felépítése

```
src
├─ adatbazis.c
├─ adatbazis.h
├─ debugmalloc.h
├─ main.c
├─ menu.c
├─ menu.h
├─ record.c
├─ record.h
├─ vcard.c
└─ vcard.h
```

debugmalloc.h

- memóriakezelést segítő könyvtár

adatbazis.h

- adatbáziskezelés (.csv file) (félkészként egy struktúraként menu.c-ben megadva)
- DB_t: adatbázis struct, mely tartalmazza a tárolt adatokat az egyes személyekről

adatbazis.c

```
void db_show_short(DB_t *db, int meret)
```

PARAMÉTEREK: Egy **struktúrát** és egy **méretet** kap paraméterként.

KIMENET: **void**

Ezen a struktúrán meret sorig végigmegy és kiírja a ***legfontosabb adatokat röviden.***

Ezek a **Név**, **Telefonszám**, és **Email** az adatbázis egyes soraiból.

menu.h

- a main függvényben hívott menürendszer kezelése több almenüvel.
- ezekben az almenükben hívódnak meg az egyes függvények (pl: db_show_short).

menu.c

- DB_t db[]: a félkész változatban ebben a fileban van az **adatbázis struktúra**, mely **stringekből** áll

int input_shell(void)

PARAMÉTEREK: void

KIMENET: int

A program shelljeként működik, a bemenetkezelés és rossz bemenet esetén a felhasználó figyelmeztetése, és a bekérés újrapróbálása a feladata.

Helyes bemenet (q, 1-n) esetén a beírt számot adja vissza (q-nál 0-át a visszalépéshez).

int menu(char* s)

PARAMÉTEREK: **string** (menü szövege)

KIMENET: **int** input_shell függvény kimenete

Az input_shell függvényhez egy kiegészítő, mely egy stringet vár paraméterként, ezt kiírja és a kilép lehetőséget, majd meghívja az input_shell függvényt a bemenet helyes lekezeléséért.

int mainmenu(void)

PARAMÉTEREK: void

KIMENET: **int** 0 esetén kilép a program

A program fő függvénye, amiben a többszintes menürendszer van implementálva.

Ebben a függvényben hívódik meg a `menu` függvény különböző paraméterekkel, hogy a következő lépést a menüben tudja a program. Ha a `next` és a `menulevel`(menümélység) változó elérnek egy adott szintet, meghívódnak a program különböző függvényei (pl:
`db_show_short(db, 2);`)

- `next`: a menüben választott menüpont(szám), amire a következő loopban lépni fog a program.
- `menulevel`: segédváltozó a `next`-hez, hogy tudja a program, milyen mélyen van a menüben.
- `prev`: segédváltozó a `next`, és `menulevel`-hez, hogy tudjuk melyik menüből volt választva az egyes almenü.
- `id`: segédváltozó, (`id.`)rekord egyes (`next.`)entry-ének szerkesztéshez.
- `torol`: segédváltozó, rekord törléshez, hogy tényleg szeretné-e törölni azt az adott rekordot a user.

Ha az `id` változónak nem 0 érték van adva, az azt jelenti, hogy az `id.` rekord `next.` entry-ének szerkesztése folyamatban van.

record.h

- egyes rekordokhoz `db[n]` tartozó függvények (pl egy `db` rekord minden adatának kiírása)

record.c

`void record_show_all(DB_t record)`

PARAMÉTEREK: **DB** adatbázis struktúra **egy rekordja**

KIMENET: `void`

Egy adott sort(rekordot) kap meg az adatbázisból, melynek minden entry-ét kiírja. Egy alap hibakezelést tartalmaz, ha `record` első

entry-je (Név) nem létezik, hibát dob nem létező rekordra hivatkozva.

vcard.h

- a vcard formátum előállításához használandó függvényeket tartalmazza

vcard.c

char* vcard_convert(DB_t record)

PARAMÉTEREK: **DB** adatbázis struktúra **egy rekordja**

KIMENET: **string** vcard formázott rekord

Egy adott sort(rekorodot) kap paraméterként és ezt vcard formátumúvá alakítja.

Egy stringet ad vissza, amelyet egy fileíró függvény megkaphat paraméterként egy filenévvel és elkészíthető a vcard export.