

Balog Péter NHF

Név:Koktélos könyv(Italbolt)

Típus: Menüvezérelt program

Karakterkódolás: UTF-8

A program célja:

Egy egyszerű menüvezérelt program létrehozása, ami képes koktél recepteket és összetevőket eltárolni, azokat listázni és bizonyos kritériumú kereséseket ezeken a recepteken lefuttatni¹.

Feladat:

A NHF-om az infocpp oldalon található Italbolt feladat egy kicsit kibővített változata. A Prog 1 nagyházim egy nagyon hasonló Koktélos könyv nevű program volt ami szinte megfelel az Italbolt feladatnak. Ennek az objektum orientált kivitelezése a célom.

Italbolt:

Készítsen C++ programot italreceptek nyilvántartására! A program alapanyagokból építse fel a recepteket. Elegendő 3-4 receptet kezelnie, de legyen bővíthető. A programmal minimálisan a következő feladatokat kell ellátni:

- új összetevő bevitele az adatbázisba
- új recept bevitele az adatbázisba
- adatbázis kiírása fájlba
- adatbázis beolvasása fájlból
- recept törlése
- kiválasztott ital hozzávalóinak listázása

Kiegészítések:

- Keresés a receptek között
- Összetevők listázása
- Összetevők törlése

Feladatspecifikáció:

Az elkészült program képes lesz összetevők felvételére, listázására és törlésére. Ezenkívül képes lesz recepteket felvenni, listázni és törölni. A recepteken lehet majd egyszerű kereséseket futtatni. A program menüjében egész számok megadásával lehet majd lépkedni.

¹ A fenti sorok a Prog 1 NHF-em specifikációjából lettek átemelve.

A receptek a következő adatokat fogják tárolni:

1. Recept név(string)
2. Összetevők(string lista)
3. Mennyiségek(int lista)
4. Elkészítési leírás(string lista)

Megkötések:

A receptek és az összetevők listáját nem lehet majd kiüríteni a program helyes használatával. Így egy elem mindig fog maradni. (Azaz csak úgy lehet tönkretenni, hogy „kívülről” belenyúlunk. Például a .txt fájlt szerkesztjük.) Ha a lista mégis üres lenne az hibákat okozhat.

A program a stringeket részlegesen fogja ellenőrizni. Azaz ha valahol 200 karaktert lehet megadni akkor a string hosszát ellenőrzi, de a tartalmát nem.

Ha a felhasználó hibásan navigál a menüben (azaz túl- vagy alulindexel) akkor a menü „Hibás érték!” errort dob és újratölti az adott menüpontot.

Ha a felhasználó egy listát hibásan indexel akkor „Hibás index!” errort fog kapni és új értéket kell megadnia.

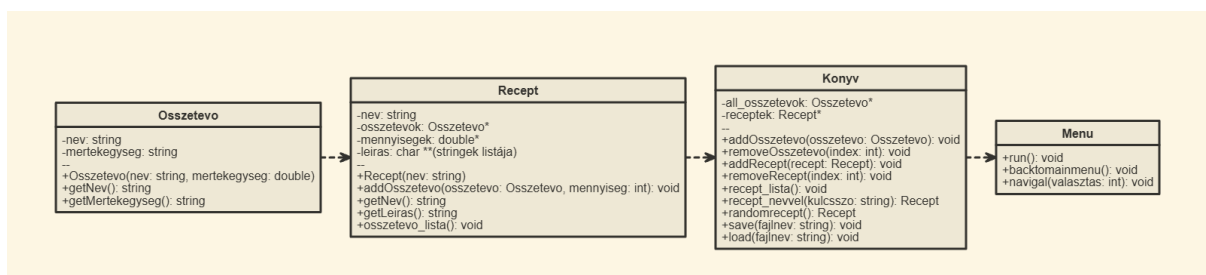
Bementek:

A NHF-om nagyrésze egész számokat, stringeket és objektumokat fog bemenetként kapni. A különleges eseteket a Terv résznél külön részletezem.

Teszt:

A teszt fájl ellenőrizni fog minden menüpontot és hozzá tartozó funkciót. Ellenőrizni fog helyes kiírást, helyes létrehozást és elpusztítást és fájlból/fájlba írást.

Objektumok:



Osszetevo:

Az összetevő classból lesznek példányosítva az összetevő objektumok amiket majd az összetevő lista tárol.

Recept:

A recept classból lesz példányosítva a recept objektum ami a recepteket tárolja

Könyv:

A könyv class fogja tárolni a recepteket és összetevőket és ez a class fog keresni a receptekben.

Menu:

Ez a class fogja vezérelni a menüt.

Tesztfájl:

A tesztfájlom tervezetten a következő funkciókat fogja tesztelni, ebben a sorrendben.

- Fájkezelés(fájl nyitás, betöltés)
- Menü bejárása
- Összetevő funkciók(hozzáad,listáz,töröl)
- Recept funkciók(hozzáad,listáz, töröl)
- Kereső funkciók
- Fájkezelés(fájlbairás, fájlzárás)

Menü:

A menüm tervezett felépítése:

1. Összetevők

- >1.1 Új összetevő hozzáadása
I: Új összetevő megadása
- >1.2 Összetevő törlése
I: Összetevő törlése
O: Biztos?(y/n)
- >1.3 Összetevők listája
O: Összetevők listája

2. Új koktéltrecept felvétele

- > O: "Van e új összetevő?"
- > 2.1 Van
I: Új összetevő megadása
<-Ismételt 2.
- > 2.2 Nincs
I: Receptnév megadása
I: Összetevő lista megadása
I: EL megadása
- >2.3 Nem tudom
-> O: Ismert összetevők listázása
-> Ismételt 2.1 || 2.2

3. Receptek

-> 3.1 Recept módosítása

->3.1.1 Recept nevének módosítása

I: Új receptnév

-> 3.1.2 Összetevő módosítása

I: Új összetevők megadása

-> 3.1.3 Elkészítési leírás módosítása

I: Új EL megadása

-> 3.2 Recept törlése

I: Recept kiválasztása

I: Biztos(y/n)

->3.3 Receptek listázása

O:Receptek listázása

4. Keresés

->4.1 Nincs ötletem

O: Kidob egy random receptet

I: Jó lesz?(y/n)

->4.2 Név szerinti keresés

O: Kidobja a receptet ha van ilyen nevű recept

A menüfa felépítésében az alábbi rövidítések vannak használva:

- EL: Elkészítési leírás
- I: Input
- O: Output
- (y/n): Eldöntendő igen-nem kérdés

A menü és az almenük(az almenük x.x vagy x.x.x formában vannak megadva, ezzel jelezve a menün belüli elérési útvonalat) navigálása az adott egész szám megadásával lehetséges. Például az „Összetevő módosítása” a következő bementéssel érhető el: 3(ezzel belépünk a Receptek menübe), 1(Ezzel belépünk a a Recept módosítása almenübe), 1 (Ezzel kiválasztjuk az Összetevő módosítása lehetőséget).

A program célja:

Egy egyszerű menüvezérelt program létrehozása, ami képes koktél recepteket és összetevőket eltárolni, azokat listázni és bizonyos kritériumú kereséseket ezeken a recepteken lefuttatni.

A program használata:

Menü kezelése: A menüben a képernyőn levő egész számok megadásával lehet lépkedni.

Összetevők megadása: Az összetevőknél az új összetevőt az angol abc karaktereit használva lehet megadni.

Recept megadása és módosítása:

Név: A recept nevét az angol abc betűivel lehet megadni.

Összetevő lista: A recept elemeit a listából lehet kiválasztani egész számokkal, a mennyiséget, pedig lebegőpontos számokkal lehet megadni.

Elkészítési leírás: Az elkészítési leírást az angol abc betűivel lehet megadni.

Összetevők ,receptek törlése és listázása, keresés: A felsorolt műveleteket a képernyőn megjelenő utasításokat követve lehet használni.

A programom felépítése:

.cpp és .h fájlok:

- main.cpp
 - A main függvény beállítja a konzolt utf-8ra és meghívja a főmenüt.
- osszetevo.cpp + osszetevo.h
 - Ebben a fájlban található az Osszetevo class és a hozzá tartozó operátorok.
- recept.cpp + recept.h
 - Ebben a fájlban található a Recept class és a hozzá tartozó operátorok.
- menu.cpp + menu.h
 - A Menu class hívja meg a többi class objektumait és teszi lehetővé a köztük való navigálást.

Kiegészítő fájlok:

- receptek.txt
 - Ez a text fájl tárolja a recepteket a következőképpen:

- Recept|2|2|s;d,s;d,|42,42,|Leir1,Leir2, \n
- Megjegyzés: A program nem engedi, hogy ez a fájl üres legyen. Az üres fájl a programban hibákat okozhat.
- osszetevok.txt
 - Ez a fájl tárolja az összetevőket ilyen formában:
 - Összetevő1 | Mennyiség1 \n
 - Megjegyzés: A program nem engedi, hogy ez a fájl üres legyen. Az üres fájl a programban hibákat okozhat.
 - **Kiegészítő könyvtárak:**
A saját moduljaimon kívül a következő standard könyvtárakat használom.

- ❖ <random>
- ❖ <windows.h>
- ❖ <iostream>
- ❖ <cstring>
- ❖ <fstream>
- ❖ <string>
- ❖ <sstream>

Függvények:

A programomban az alábbi függvények vannak:

- **Menü class:**
- void run(void);
 - **Bemenet:** Nincs
 - **Kimenet:** Nincs
 - **Célja:** Navigáció a programban
- **Összetevő class és adatagok:**
- Adattagok
 - **Változók:** std::string nev = Az összetevő neve | std::string mertekegység = a mértékegység neve
- Osszetevo();
 - **Bemenet:** Nincs
 - **Kimenet:** Egy „üres” objektum.
 - **Célja:** Egy új összetevő hozzáadása a listához.
- Osszetevo(std::string nev,std::string db);
 - **Bemenet:** Két std::string, egy név és egy mértékegység.
 - **Kimenet:** Egy Osszetevo objektum.
 - **Célja:** A konstruktor építi meg az objektumot.
- Osszetevo(const Osszetevo& o);
 - **Bemenet:** Egy Osszetevo objektum.
 - **Kimenet:** Egy Osszetevo objektum.
 - **Célja:** Copy konstruktor.
- std::string GetNev() const;

- **Bemenet:** Nincs..
- **Kimenet:** Egy string.
- **Célja:** A nev adattag getterje.
- `std::string GetMertekegyseg() const;`
 - **Bemenet:** Nincs.
 - **Kimenet:** Egy string.
 - **Célja:** A mertekegyseg getterje.
- `void SetNev(std::string knev);`
 - **Bemenet:** Egy string.
 - **Kimenet:** Nincs.
 - **Célja:** A nev setterje.
- `void SetMertekegyseg();`
 - **Bemenet:** Egy string.
 - **Kimenet:** Nincs.
 - **Célja:** A mertekegyseg setterje.
- `void WriteOsszetevo(std::ofstream& os);`
 - **Bemenet:** Egy output filestream.
 - **Kimenet:** Nincs.
 - **Célja:** Ez a függvény beleírja az objektum adatait egy fájlba.
- `Osszetevo ReadOsszetevo(std::ifstream& is);`
 - **Bemenet:** Egy input filestream.
 - **Kimenet:** Egy Osszetevo objektum.
 - **Célja:** Ez a függvény kiolvassa az objektum adatait egy fájlból.
- `~Osszetevo();`
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** A class detruktora.
- **Operátorok:**
 - `Osszetevo operator=(Osszetevo o);`
 - `bool operator==(Osszetevo o);`
 - `std::ostream& operator<<(std::ostream& os, const Osszetevo& o);`
- ***Recept class és adattagok:***
- **Adattagok**
 - ❖ **Változók:**
 - ❖ `std::string nev` = A recept neve.
 - ❖ `Osszetevo* osszetevok` = Az összetevők listája, Osszetevo objektumokat tárol.
 - ❖ `int o_size` = Az osszetevok lista mérete.
 - ❖ `double mennyisegek` = A mértékegységek listája, double-öket tároló tömb.
 - ❖ `std::string* leiras` = Az elkészítési lépések listája, sztringeket tároló tömb.
 - ❖ `int el_size` = A leiras tömb mérete.
- `Recept();`
 - **Bemenet:** Nincs.
 - **Kimenet:** Egy Recept objektum.

- **Célja:** Elkészít egy „üres” recept objektumot.
- Recept(std::string knev, double* kmennyisegek=NULL, int ko_size=0, Osszetevo* kosszetevok=NULL, int kel_size=0, std::string* kleiras=NULL);
 - **Bemenet:** A Recept class adatai.
 - **Kimenet:** Egy Recept objektum.
 - **Célja:** Példányosítja a Recept osztályt.
- Recept(const Recept& r);
 - **Bemenet:** Egy Recept objektum.
 - **Kimenet:** Egy Recept objektum.
 - **Célja:** Copy konstruktor.
- int GetOSize() const | int GetELSize() const | double GetMennyiseg(int idx) const; | Osszetevo GetOsszetevo(int idx) const | std::string GetLeiras(int idx) const;
 - **Bemenet:** Nincs.
 - **Kimenet:** A getternek megfelelő adat.
 - **Célja:** Az adatok getterjei.
- void SetNev(std::string knev);
 - **Bemenet:** Egy string.
 - **Kimenet:** Nincs.
 - **Célja:** A nev adat setterje.
- void WriteRecept(std::ofstream& os);
 - **Bemenet:** Egy output filestream.
 - **Kimenet:** Nincs.
 - **Célja:** Ez a függvény beírja az objektum adatait egy fájlba.
- Recept ReadRecept(std::ifstream& is);
 - **Bemenet:** Egy input filestream.
 - **Kimenet:** Egy Recept objektum.
 - **Célja:** Ez a függvény kiolvassa az objektum adatait egy fájlból.
- void AddOsszetevo(Osszetevo o, double m);
 - **Bemenet:** Egy Osszetevo objektum és egy double.
 - **Kimenet:** Nincs.
 - **Célja:** Hozzáad egy összetevőt a recepthez.
- void RemoveOsszetevo(int idx);
 - **Bemenet:** Egy integer.
 - **Kimenet:** Nincs.
 - **Célja:** Töröl egy összetevőt a receptből.
- void ListOsszetevo();
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** Kiírja a recept összetevőit.
- void AddLeiras(std::string leiras);
 - **Bemenet:** Egy string.
 - **Kimenet:** Nincs.
 - **Célja:** Hozzáad egy lépést a recepthez.

- void RemoveLeiras (int idx);
 - **Bemenet:** Egy integer.
 - **Kimenet:** Nincs.
 - **Célja:** Töröl egy lépést a receptből.
- void ListLeiras ();
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** Kiírja a recept leírását.
- ~Recept();
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** A class detruktora.
- **Operátorok:**
 - **Recept& operator=(const Recept& r);**
 - **std::ostream& operator<<(std::ostream& os, const Recept& r);**
- ***Könyv class és adattagok:***
- **Adattagok**
 - ❖ **Változók:**
 - ❖ **Osszetevo* all_osszetevo** = Az összes összetevő.
 - ❖ **Recept* receptek** = Az összes tárolt recept.
 - ❖ **int o_meret** = Az all_osszetevo aktuális lista mérete.
 - ❖ **int o_capacity** = Az all_osszetevo tömb kapacitása.
 - ❖ **int r_meret** = A receptek lista mérete.
 - ❖ **int r_capacity** = A receptek tömb kapacitása.
- **Konyv()**
 - **Bemenet:** Nincs.
 - **Kimenet:** Egy Konyv objektum.
 - **Célja:** Elkészít egy „üres” Konyv objektumot.
- void AddOsszetevo(Osszetevo o);
 - **Bemenet:** Egy Osszetevo objektum.
 - **Kimenet:** Nincs.
 - **Célja:** Hozzáad egy összetevőt az összetevő listához.
- void RemoveOsszetevo(int idx);
 - **Bemenet:** Egy integer.
 - **Kimenet:** Nincs.
 - **Célja:** Töröl egy összetevőt az összetevő listából.
- void ListOsszetevo();
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** Kiírja az összes összetevőt.
- void AddRecept(Recept r);

- **Bemenet:** Egy .
- **Kimenet:** Nincs.
- **Célja:** Hozzáad egy receptet a tárolt receptekhez.
- void RemoveRecept (int idx);
 - **Bemenet:** Egy integer.
 - **Kimenet:** Nincs.
 - **Célja:** Töröl egy receptet a tárolt receptek közül.
- void ListReceptek ();
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** Kiírja a tárolt recepteket.
- int GetOMeret() const | int GetOCapacity() const | double GetRMeret() const; |
Osszetevo GetRCapacity() const | Osszetevo GetOsszetevok(int idx); | Recept
GetReceptek(int idx);
 - **Bemenet:** Nincs.
 - **Kimenet:** A getternek megfelelő adattag.
 - **Célja:** Az adattagok getterjei.
- void save(std::ofstream& os1, std::ofstream& os2);
 - **Bemenet:** Két output filestream.
 - **Kimenet:** Nincs.
 - **Célja:** Menti a könyv tartalmát.
- void load(std::ifstream& is1, std::ifstream& is2);
 - **Bemenet:** Két input filestream.
 - **Kimenet:** Nincs.
 - **Célja:** Betölti a könyvet az „osszetevo.txt” és „receptek.txt” fájlokból.
- void recept_nevvel(std::string kulcsszo);
 - **Bemenet:** Egy string.
 - **Kimenet:** Nincs.
 - **Célja:** Megkeresi az összes kulcsszo nevű receptet.
- void randomrecept();
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** Kidob egy random receptet.
- ~Konyv();
 - **Bemenet:** Nincs.
 - **Kimenet:** Nincs.
 - **Célja:** A class detruktora.

A projektem itt is elérhető: [Github](#)