

# HÁZI FELADAT

## Programozás alapjai 2.

### Dokumentáció

Merena Dénes Sándor

FOBZB7

2025. március 29.

---

#### TARTALOM

1.	Feladat .....	2
2.	Feladatspecifikáció .....	2
2.1.	Bemenetek és az elvárt kimenetek tartalma és formátuma .....	2
2.2.	Adatbázis .....	4
2.3.	A program működésének feltételei .....	4
3.	Programozói dokumentáció .....	4
3.1.	Osztályok .....	4
3.2.	UML ábra .....	9
3.3.	A program fordítása .....	10
4.	Tesztelési dokumentáció .....	10
4.1.	Tesztelési környezet .....	10
4.2.	Tesztek .....	10
4.3.	Memóriaszivárgás-mentesség .....	10
5.	Felhasználói dokumentáció .....	10

# 1. Feladat

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

Egyszerű felhasználói felületet tervezzen! A feladat lényege az objektumorientált megközelítés, ill. modellezés és nem a felhasználói felület szépsége. Használjon heterogén adatszerkezetet! A megoldáshoz **ne** használjon STL tárolót!

## 2. Feladatspecifikáció

### 2.1. Bemenetek és az elvárt kimenetek tartalma és formátuma

- **Bemenetek**

- A felhasználó konzolos környezetben tud menüből számozott opciókat

1. új alapanyag bevitele  
2. új recept bevitele  
3. Recept torlese  
4. Hozzavalok listazasa  
5. Adatbazis mentese fajlba  
6. Adatbazis betoltese fajlbol  
7. Kilepes  
Valassz egy opciot:

választani.

- A felhasználó konzolon keresztül adhatja meg a bemeneteket szövegesen a kettőspontok után. A sorok egyenként jelennek meg, enterrel kell jóváhagyni.
- Példa:

Új alapanyag neve: Víz  
Mértékegység: ml

- Alapanyag bevitele
  - A felhasználó bevihet egy új alapanyagot, amit receptekhez fel lehet használni.
  - Input:
    - Alapanyag neve
    - Mértékegység
- Új recept bevitele
  - A felhasználó megadhat egy új receptet.
  - Input:
    - Recept neve
    - Alapanyag kiválasztása a meglévők közül
    - Mennyiség minden hozzávalóhoz
- Recept törlés
  - A felhasználó törölhet egy receptet.
  - Input:
    - Recept neve
- Recept hozzávalóinak listázása
  - A felhasználó megadhat egy meglévő recept nevet, a program kilistázza az alapanyagokat és a mennyiségeket.
  - Input:
    - Recept neve
- Adatbázis kiírása fájlba
  - Aktuális alapanyagok és receptek elmenthetők a háttértárba egy fájlba.
  - Input:
    - Fájl neve (<fájl neve>.<kiterjesztés>)
- Adatbázis beolvasása fájlból
  - Korábban elmentett receptek és alapanyagok betöltése háttértárból.
  - Input:
    - Fájl neve (<fájl neve>.<kiterjesztés>)

## ● Kimenetek

- Alapanyag bevitele után
  - „Sikeres alapanyag bevitel.”
- Új recept bevitele után
  - „Recept mentve.”
- Recept törlés után
  - „Recept törölve.”
- Hozzávalók listázása
  - Első sorban a recept neve, többi sorban összetevők és mennyiségük.  
Példa:
- Adatbázisba kiírás után

Limonádé recept hozzávalói:  
 -Citromlé: 50 ml  
 -Víz: 300 ml  
 -Cukorszirup: 10 ml

- Fájl neve
- Adatbázisból beolvasás után
  - „Beolvasás sikeres.”

## 2.2. Adatbázis

Az adatbázis mentésnél a felhasználó által megadott formátumban kerül elmentésre, a következő szerkezettel:

- Az első sorban az alapanyagok száma található.
- Az elkövetkezendő x sorban vannak ezek az alapanyagok a következő formátumban:  
<név>;<mértékegység>
- Az alapanyagok felsorolását követő sorban található a receptek darabszáma.
- Ezután fel vannak sorolva a receptek a következő formátumban:
  - Recept neve
  - Receptben található alapanyagok száma
  - Ezután soronként az alapanyag és a mennyiség a következő formátumban:  
<alapanyag neve>;<mennyiség>

Példa:

```
3
Víz;ml
Cukor;g
Citromlé;ml
2
Limonádé
2
Citromlé;50
Cukor;20
Cukros víz
2
Víz;200
Cukor;15
```

## 2.3. A program működésének feltételei

- A felhasználó a menüben megadott lehetőségek közül választ és megfelelő formátumban ad meg szöveges bemenetet.
- A programnak van írási és olvasási jogosultsággal abban a mappában, ahol a fájlokat tárolja.

## 3. Programozói dokumentáció

Doxygen segítségével készült.

### 3.1. Osztályok

#### • Alapanyag

- Az alapanyag egy névből és egy mértékegységből áll, amelyek együtt határozzák meg az alapanyag tulajdonságait.

- Osztályváltozók
  - string nev: Alapanyag neve.
  - string mertekegység: Alapanyag mértékegysége.
- **Konstruktor és destruktorok**
  - Alapanyag()
    - Konstruktor egy új alapanyag létrehozásához.
    - Paraméterek:
      - nev: Az alapanyag neve.
      - mertekegység: Az alapanyag mértékegysége.
- **Tagfüggvények**
  - getMertekegység()
    - Mértékegység getter.
    - Visszaadja:
      - Az alapanyag mértékegysége konstans referenciával
  - getNev()
    - Név getter.
    - Visszaadja:
      - Az alapanyag neve konstans referenciával.
- Operátorok
  - operator<<
    - Kiíró operátor túlterhelése.
    - Paraméterek
      - os: Output stream referencia
      - alapanyag: A kiírandó alapanyag objektum
    - Visszaadja:
      - Referencia a stream objektumra.
      - Az alapanyagot "név (mértékegység)" formátumban írja ki

## • Fajlkezelő

- Fajlkezelő osztály az adatok mentéséhez és betöltéséhez.
- Statikus metódusokat tartalmaz a receptek és alapanyagok fájlba mentéséhez és fájlból való betöltéséhez
- Tagfüggvények
  - betoltes()
    - Adatok betöltése fájlból.
    - Paraméterek
      - receptTarolo: A recepteket tartalmazó tároló
      - alapanyagTarolo Az alapanyagokat tartalmazó tároló
      - fajlnev: A forrásfájl neve
  - mentes()
    - Adatok mentése fájlba.
    - Paraméterek:

- receptTarolo: A recepteket tartalmazó tároló
- alapanyagTarolo: Az alapanyagokat tartalmazó tároló
- fajlnev: A cél fájl neve

## • Hozzavalo

- Egy recept hozzávalóját reprezentáló osztály.
- Osztályváltozók
  - Alapanyag alapanyag: A hozzávaló alapanyaga
  - double mennyiseg: A hozzávaló mennyisége
- Konstruktor és destruktorkok
  - Hozzavalo()
    - Konstruktor egy új hozzávaló létrehozásához.
    - Paraméterek:
      - alapanyag: Az alapanyag érték szerint
      - mennyiseg: A mennyiség értéke
- Tagfüggvények
  - getAlapanyag()
    - Alapanyag getter.
    - Visszaadja:
      - Az alapanyag konstans referenciája.
  - getMennyiseg()
    - Mennyiség getter.
    - Visszaadja:
      - A mennyiség értéke.
- Operátorok:
  - operator<<
    - Kiíró operátor
    - Paraméterek:
      - os: Output stream referencia
      - hozzavalo: A kiírandó hozzávaló
    - Visszaadja:
      - Az output stream referenciája.

## • Menu

- A felhasználói felületet kezelő osztály.
- Kezeli a felhasználói interakciót és a program fő funkcióit.
- Konstruktorok és destruktorkok
  - Menu()
    - Konstruktor
    - Paraméterek:
      - alapanyagTarolo: Referencia az alapanyagok tárolójára
      - receptTarolo: Referencia a receptek tárolójára

- Tagfüggvények
  - clearInputBuffer()
    - Input buffer tisztítása
  - ujAlapanyagBevitel()
    - Új alapanyag bevitele a felhasználótól
  - ujReceptBevitel()
    - Új recept bevitele a felhasználótól
  - receptTorles()
    - Recept törlése a felhasználó választása alapján
  - hozzavalokListazasa()
    - Egy recept hozzávalóinak listázása
  - adatbázisMentes()
    - Adatbázis mentése fájlba
  - adatbázisBetoltes()
    - Adatbázis betöltése fájlból
  - displayMenu()
    - Menü megjelenítése
  - run()
    - A fő programhurok, amely kezeli a felhasználói interakciót

## • Recept

- Italreceptet reprezentáló osztály.
- A recept névből és hozzávalók dinamikus tömbjéből áll.
- Osztályváltozók
  - string nev: A recept neve
  - Hozzavalo\*\* hozzavalok: A hozzávalók dinamikus tömbje
  - int count: A hozzávalók aktuális száma.
  - int capacity: A tömb kapacitása.
- Konstruktor és destruktorok
  - recept()
    - Konstruktor új recept létrehozásához.
    - Paraméterek:
      - nev: A recept neve
  - ~Recept()
    - Destruktor.
    - Felszabadítja a dinamikusan foglalt memóriaterületet.
- Tagfüggvények
  - resize()
    - Hozzávalók tömbjének átméretezése, akkor hívódik meg amikor nincs elég szabad hely.
  - getHozzavaloCount()
    - Hozzávalók számának lekérdezése
    - Visszaadja:

- A hozzávalók száma.
- `getNev()`
  - Név getter.
  - Visszaadja:
    - A recept neve.
- `hozzavalohozaad()`
  - Hozzávaló hozzáadása a recepthez.
  - Paraméterek:
    - `alapanyag`: Alapanyag referenciája.
    - `mennyiseg`: A hozzávaló mennyisége.
- Operátorok
  - `operator[]`
    - Indexelő operátora hozzávalók eléréséhez.
    - Paraméterek:
      - `index`: A hozzávaló indexe.
    - Visszaadja:
      - A kért hozzávaló pointerre.

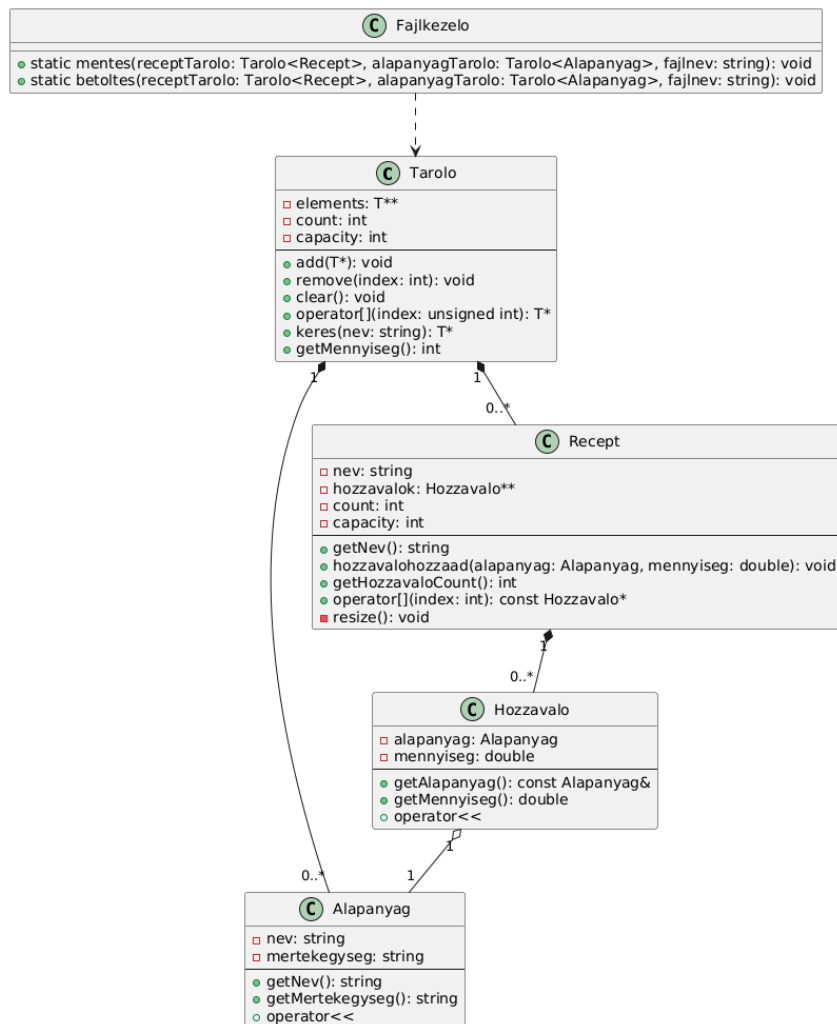
## • Tarolo

- Dinamikus tömbként működő tároló, amely képes elemek hozzáadására, törlésére.
- Template paraméterek
  - `T`: A tárolandó elem típusa
- Osztályváltozók
  - `T** elements`: A tárolt elemek tömbje.
  - `int count`: Az elemek aktuális száma.
  - `int capacity`: A tömb kapacitása.
- Tagfüggvények
  - `add()`
    - Új elem hozzáadása.
    - Paraméterek:
      - `element`: Az új elem pointerre.
  - `getMennyiseg()`
    - Elemek számának lekérdezése.
    - Visszaadja:
      - Az elemek száma.
  - `keres()`
    - Elem keresés név alapján.
    - Paraméterek
      - `nev`: A keresett elem neve
    - Visszaadja:
      - Pointer a talált elemre, vagy `nullptr` ha nincs találat.
  - `remove()`



- Elem törlése index alapján.
- Paraméterek:
  - index: A törlendő elem indexe.
- clear()
  - Az összes elem törlése.
- Operátorok
  - operator[]
    - Indexelő operátor.
    - Paraméterek
      - index: A kért elem indexe.

## 3.2.UML ábra



(plantuml.com segítségével létrehozott ábra.)

### 3.3. A program fordítása

Ha gnu compiler telepítve van, a könyvtárba navigálás után a következő kóddal lehet az exe fájlt létrehozni:

```
g++ -DMEMTRACE -o program main.cpp alapanyag.cpp recept.cpp fajlkezelő.cpp  
tarolo.cpp memtrace.cpp test.cpp hozzavalo.cpp menu.cpp
```

## 4. Tesztelési dokumentáció

### 4.1. Tesztelési környezet

- Fordító: g++
- Memóriaszivárgás-ellenőrzés: memtrace
- Teszt könyvtár: gtest\_lite

### 4.2. Tesztek

- **Alapanyag és Tarolo**
  - Új alapanyagok hozzáadása Tarolo típushoz
  - Hozzáadott elemek lekérdezése index alapján
  - Tarolo dinamikus bővülése tesztelve (több mint 10 elem)
  - Elem eltávolítása a remove() metódussal, memória újraszervezés ellenőrzése
- **Recept és Hozzávalók**
  - Recept létrehozása, név beállítása
  - Hozzávalók hozzáadása recepthez meglévő alapanyagokra hivatkozva
  - Operátor túlterhelések (<<) eredményének ellenőrzése sztring reprezentációval
  - Recept növekedésének ellenőrzése 20 összetevőig
- **Fájlkezelés (mentés és betöltés)**
  - Egy minta recept és hozzávalói elmentése fájlba
  - Fájl tartalmának visszaolvasása új konténerekbe
  - A beolvasott adatok összevetése az eredeti értékekkel
  - Fájl törlése teszt végén

### 4.3. Memóriaszivárgás-mentesség

A memtrace.h segítségével minden memória foglalás és felszabadítás monitorozva van. A memtrace jelentést készít, ha bármilyen memóriakezelési probléma van.

## 5. Felhasználói dokumentáció

Ez a program egy egyszerű konzolos alkalmazás, amely receptek és alapanyagok kezelésére szolgál. A felhasználó a menüből választható opciók segítségével adhat hozzá, törölhet vagy listázhat recepteket, illetve mentheti vagy betöltheti az adatbázist fájlból. A program konzolos

környezetben fut. A menü számozott opciókat kínál. A felhasználónak a kívánt művelet számát kell megadnia, majd az enter billentyűvel megerősítenie. Az egyes műveletek során a program kérdéseket tesz fel, amelyeket a felhasználónak soronként szövegesen kell megválaszolnia.

A program használatának részletesebb felhasználói dokumentálása, megegyezik a specifikációval. A **2.1**-es pontban található.

Adatbázis mentése után a fájl a programmal megegyező könyvtárba kerül. Ha a felhasználó be szeretne tölteni egy adatbázist, a fájlnek a programmal azonos könyvtárban kell lennie.