

# HÁZI FELADAT

## Programozás alapjai 2.

### Tervezés

Sági Barnabás  
HUBX6D

2021. április 12.

---

#### TARTALOM

1.	Feladat .....	2
2.	Feladatspecifikáció .....	2
3.	Terv .....	4
3.1.	Osztálydiagram .....	4
3.2.	Osztályok és függvények .....	5
4.	Tesztek .....	5
4.1.	Osztály tesztek .....	5
4.2.	Memoria teszt .....	5

# 1. Feladat

## Telefonkönyv

Tervezze meg egy telefonkönyv alkalmazás egyszerűsített objektummodelljét, majd valósítsa azt meg! A telefonkönyvben kezdetben az alábbi adatokat akarjuk tárolni, de később bővíteni akarunk:

- Név (vezetéknév, keresztnév)
- becenév
- cím
- munkahelyi szám
- privát szám
- azonosító(szám)

Az alkalmazással minimum a következő műveleteket kívánjuk elvégezni:

- adatok felvétele
- adatok törlése
- listázás
- keresés
- módosítás

## 2. Feladatspecifikáció

A program menüvezérelt, mely több menüre oszlik. Ezek között a meghatározott számok beírásával lehet navigálni, a program mindig vár egy bemenetet. Az összes menüből lehetőség van kilépni a programból, illetve visszalépni az előző menübe. A menü cmd-ben jelenik meg. Amennyiben a program **nem** adat bevitellel kapcsolatos, akkor előtte mindig kiírja a megfelelő választható menüpontokat. A program a TELEFONKONYV.txt-ből olvassa be az adatokat, illetve ebbe is írja vissza azokat. A TXT-ben „adat adat ...” formában vannak benne a bejegyzések, minden sorban 10, de ceg bejegyzés esetén csak 7 érvenyes adat, három pedig 0. A program nem érzékeny a nagy-kis betűkre.

A program be/kimentére a következő formátumok vannak érvényben:

- **Embernév:**
  - Vezetéknév: Bármilyen hosszú szöveg, nem szükséges nagy betűvel kezdődnie.
  - Keresztnév: Bármilyen hosszú szöveg, nem szükséges nagy betűvel kezdődnie.
- **Becenév:** Bármilyen hosszú szöveg, nem szükséges nagy betűvel kezdődnie.
- **Cím:**
  - Irányítószám: Bármilyen hosszú szöveg.
  - Városnév: Bármilyen hosszú szöveg, lehet több szó is.
  - Utca név: Bármilyen hosszú szöveg, lehet több szó is.
  - Házszám: Bármilyen hosszú szöveg.
- **Emberszám:**
  - Munkahelyi szám: Bármilyen hosszú szöveg.
  - Privát szám: Bármilyen hosszú szöveg.
- **Telefonszam(ceg):** Bármilyen hosszú szöveg.
- **Nev(ceg):** Bármilyen hosszú szöveg.
- **Azonosító:** Bármilyen hosszú szöveg.

Menü szerkezete:

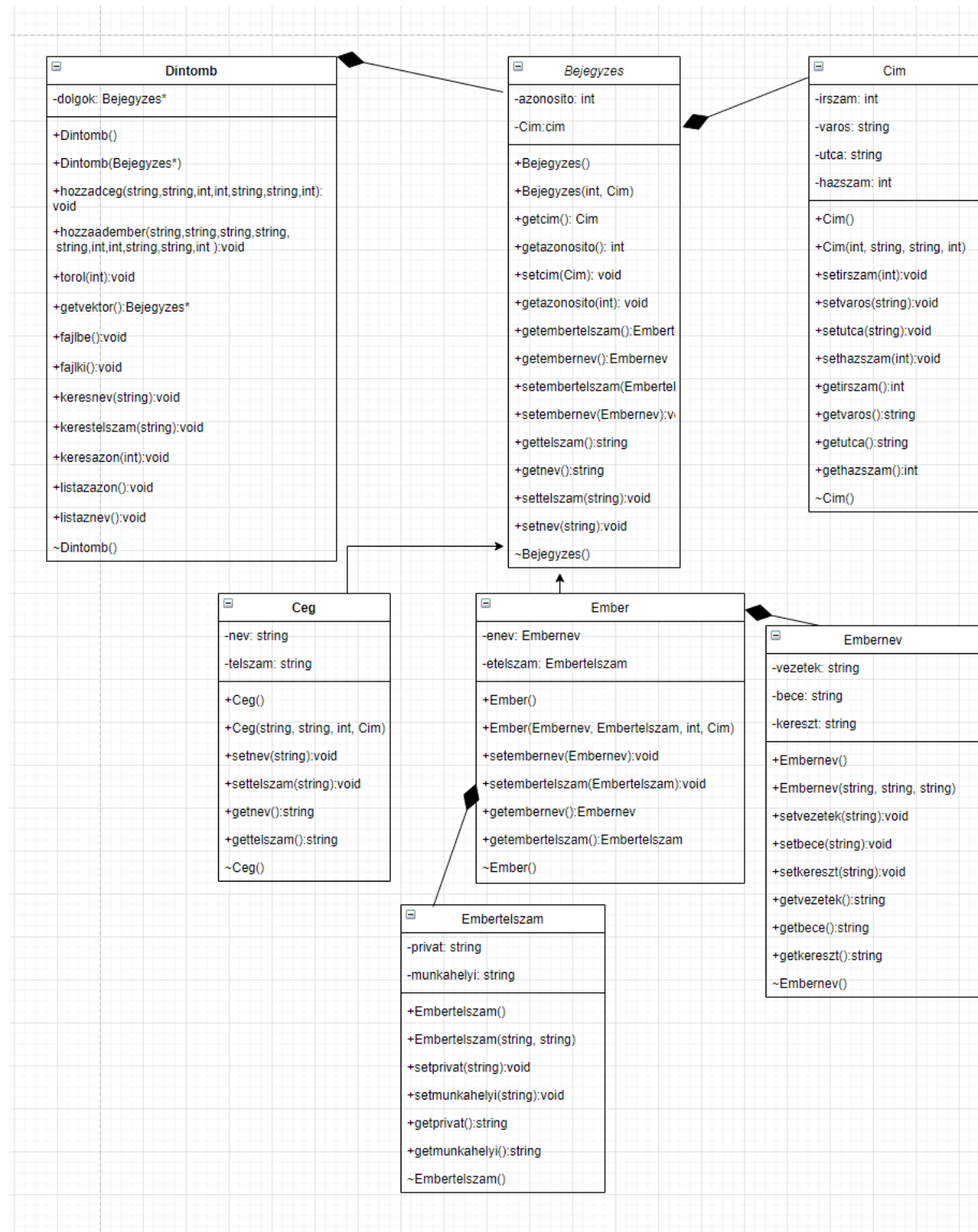
1. **Adatok felvétele:** Ezt a menüpontot kiválasztva lehet új bejegyzést tenni, minden adatot sorban be kell írni a programnak, időközben nem lehet megszakítani egy

felvételt, csak törölni a létrehozás után. Azonosítót nem kell megadni, automatikusan létrehozza a program(sorban következő, vagy ha van „üres” a régebbiek között). Ha valamelyik adat megadása nem sikerült, hibaüzenet keletkezik és újból azt az adatot várja bemenetre a program.

2. **Adatok törlése:** A program egy pontos azonosítót vár bementre, amennyiben ez megtalálható a bejegyzések között, úgy törli a bejegyzést.
  3. **Adatok módosítása:** A program egy pontos azonosítót vár bementre, amennyiben ez megtalálható a bejegyzések között, úgy az azonosítón kívül minden adat módosítása lehetséges, olyan formában, ahogy az „Adatok felvétele” résznél volt.
  4. **Listázás:**
    1. **Név szerint(ABC):** A program kilistázza a bejegyzések ABC szerinti sorrendben, kiírja az összes adatot róluk. Miután ez megtörtént, a „Listázás” menübe lép vissza a program.
    2. **Azonosító szerint(növekvő):** A program azonosító szerint, növekvő sorrendben kilistázza a bejegyzéseket, kiírja az összes adatot róluk. Miután ez megtörtént, a „Listázás” menübe lép vissza a program.
    3. **Kilép a főmenübe.**
    4. **Kilépés a programból:** megáll a futás.
  5. **Keresés:**
    1. **Név szerint:** Nem teljes név beírásával lehet keresni egy adott bejegyzésre, ha több találat is van, akkor mindet kiírja az összes adattal róluk, amennyiben nem talált egyet sem, visszajelzést ad a program erről.
    2. **Azonosító szerint:** Pontos azonosítót vár bementre, kiírja az egyetlen találat összes adatát, amennyiben nem talált, visszajelzést ad a program erről. Ezek után újból lehet keresni azonosító szerint.
    3. **Telefonszám szerint:** Nem teljes telefonszámot vár bementre, ha több találat is van, akkor mindet kiírja az összes adattal róluk, amennyiben nem talált egyet sem, visszajelzést ad a program erről.
    4. **Kilépés a főmenübe.**
    5. **Kilépés a programból:** megáll a futás.
  6. **Kilépés a programból:** megáll a futás.
- EOF:** megáll a futás.

## 3. Terv

### 3.1. Osztálydiagram



## 3.2. Osztályok és függvények

- **Bejegyzes:** Összeköti az osztályokat, ez egy bejegyzés, az azonosító és a cím tarolodik benne.
- **Embernev:** Tartalmazza a kereszt-, vezeté- és becenevet.
- **Embertelszam:** Tartalmazza a munkhelyi és privat telefonszámot.
- **Cím:** Tartalmazza az irányítószámot, a várost, az utcát és a házszámot.
- **Ember:** Tartalmazza az embernevet és a az embertelszamot, a Bejegyzes leszarmazottja.
- **Ceg:** Tartalmazza a nevet és a telefonszámot, a Bejegyzes leszarmazottja.
- **Dintomb:** Tartalmazza az adat tarolo vektort, itt vannak a program alapvető függvényei.

Minden osztálynak vannak megfelelő konstruktorai, destruktorai, illetve set és get függvényei.

További fontosabb függvények:

- **hozzaadember(string, string,string,string,string,int,int,string,string,int):** Új *Ember* típusu elemet vesz fel.
- **hozzaadceg(string,string,int,int,string,string,int):** Új *Ceg* típusu elemet vesz fel.
- **torol(int):** Töröli a kapott indexu elemet.
- **fajlbe():** Beolvassa a fájlt a *Dintomb*be.
- **fajlki():** Fájlbba írja a *Dintomb*ot.
- **keresazon(int):** Azonosító alapján keres.
- **keresnev(string):** Név alapján keres.
- **kerestelszam(string):** Telefonszám alapján keres.
- **listazon():** Kiírja azonosító szerint az embereket.
- **listaznev():** Kiírja név szerint az embereket.

## 4. Tesztek

### 4.1. Osztaly tesztek

A main.cpp-ben a *tesztem* makro 1-re allitasaval lehet elindítani az osztalyok tesztelese reszt. Hibát nem tapasztaltam a teszteles során.

### 4.2. Memoria teszt

A memoria teszteleset a MEMTRACE modullal vegeztem. Hibát nem tapasztaltam a futtatás során.