

# NHF Dokumentáció

## 1. Pontosított feladatspecifikáció

A program egy telefonkönyvet fog megvalósítani. A nevet (vezetéknév, keresztnév), a becenevet, a címet, a munkahelyi és a privát számot fogja eltárolni egy listában.

### **A program funkciói:**

Adatok felvétele: Új adatok bevitele az adatbázisba. Ilyenkor a program sorban az adatokat bekéri.

Adatok törlése: Ki is törölhető a telefonkönyvből az adott név a hozzátartozó adatokkal együtt a vezetéknév és a keresztnév megadásával. Ekkor a program visszajelzést ad a törlés sikerességéről (a törlés sikeres/nincs a keresésnek megfelelő elem).

Adatok listázása: Kilistázhatjuk az összes adatot. Ekkor a nevek a hozzátartozó adatokkal együtt egymás alatt jelennek meg.

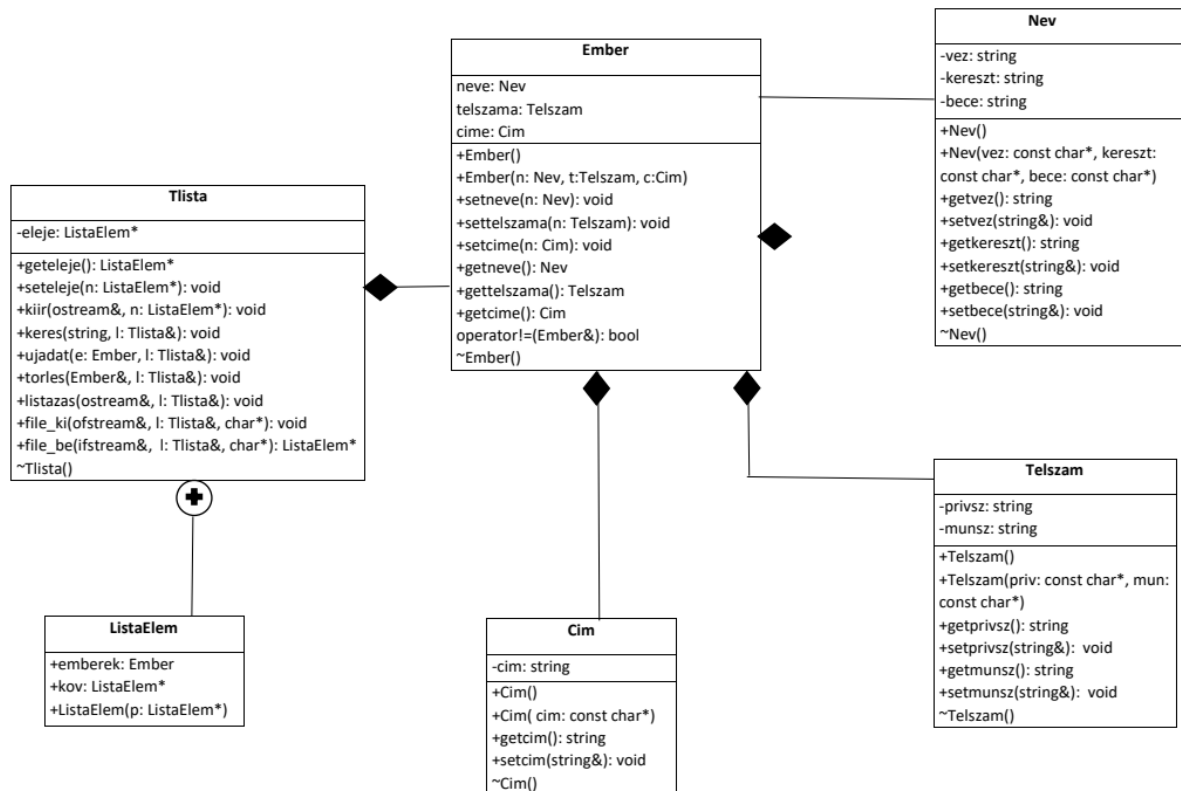
Keresés: Vezetéknév alapján kereshetünk is. Ekkor a program kilistázza a kívánt adatokat (ha több személyre illik az adott vezetéknév akkor az összeset kilistázza). Ha nincs a keresésnek megfelelő elem azt a program jelzi (nincs a keresésnek megfelelő elem).

### **Felhasználói felület:**

A program szöveges formában fog kommunikálni a felhasználóval. Indítás után a konzolanlakban egy menüből választhatjuk ki, hogy melyik funkciót kívánjuk igénybe venni. Az adott funkció kiválasztása után szöveges formában vihetjük be a kívánt adatokat, ha ez szükséges.

A telefonkönyv az adatait egy .txt fájlban fogja eltárolni. Ezt a fájlt fogja a program használni a funkciók elvégzése során.

## 2. Osztálydiagram



## 3. Osztályok és a főbb függvények

**class Nev:** A vezetéknév, keresztnév és a becenév eltárolásához használt osztály. Alapvető getter, setter függvényeket tartalmaz.

**class Nev:** A privát- és munkahelyi szám eltárolásához használt osztály. Alapvető getter, setter függvényeket tartalmaz.

**class Cim:** A cím eltárolásához használt osztály. Alapvető getter, setter függvényeket tartalmaz.

**class Ember:** Ez az osztály köti össze az eddig felsorolt osztályokat, egy bejegyzés a telefonkönyvben. Az alapvető getter, setter függvényeken kívül tartalmaz egy != operátort is ami a törléshez kell.

**class Tlista:** Itt születik meg a telefonkönyv listája és itt vannak a program fő funkcióinak a függvényei is:

- **void kiir(std::ostream &, ListaElem \*n)**

Kiírja a kapott ListaElemet.

- **void keres(std::string, Tlista& l)**

A kapott vezetéknév alapján megkeresi a kívánt elemet és kiírja a kiir függvény segítségével és visszajelzést ad, ha nem találja meg a keresett embert: Nincs a keresésnek megfelelő elem. Ha több bejegyzésre is illik a keresés akkor azokat is kiírja.

- **void ujadat(Ember e, Tlista& l)**

Hozzáfűz a meglévő listához egy új ListaElemet, amit feltölt a paraméterként kapott adattal.

- **void torles(Ember&, Tlista& l)**

Kitörli a már meglévő ListaElemet, ha megegyezik a ListaElem adattal a kapott adattal (itt használja a program a != operátort).

- **void listazas(std::ostream &, Tlista& l)**

Kilistázza az összes ListaElem tartalmát egy for ciklus és a kiir függvény segítségével.

- **void file\_ki(std::ofstream&, Tlista &l, char\*)**

Elmenti az adott .txt fájlba a lista tartalmát.

- **ListaElem\* file\_be(std::ifstream&, Tlista& l, char\*)**

Létrehoz egy listát és abba beolovassa az adott .txt fájl tartalmát az ujadat függvény segítségével. Végül visszatér a létrehozott lista elejére mutató pointerrel.

## 4. Tesztprogram

A tesztprogramot az indítás utáni konzolablakban megjelenő menüből választhatjuk ki. Ez után a standard inputról beolvasott egész szám alapján különböző teszteseteket hajt végre. Ezt addig tehetjük meg amíg a

tesztprogramból való kilépést nem választjuk. A tesztesetek sikerességéről a konzolablakban visszajelzést kapunk. Minden főbb funkciónak van egy tesztesete. Először a program létrehoz egy listát és a későbbiekben ennek a segítségével tesztl.

### **Az egyes teszteseteket megvalósító függvények**

- **void test\_1(Ember e1, Ember e2, Tlista& l)**

1. tesztesetet megvalósító függvény: Ellenőrzi, hogy sikeres volt-e a fájlból való beolvasás.

- **void test\_2(Ember e1, Ember e2, Ember e3, Tlista& l)**

2. tesztesetet megvalósító függvény: Hozzáad az újadat függvény segítségével a listához egy ListaElmenet és megbizonyosodik a művelet sikereességéről.

- **void test\_3(Ember e1, Ember e2, Tlista& l)**

3. tesztesetet megvalósító függvény: A listazas függvényt teszteli. Ehhez a tesztesethez a kiiras függvényt hívja segítségül a program, ami egy Ember osztály kiírását könnyíti meg.

- **void test\_4(Ember e1, Ember e2, Tlista& l)**

4. tesztesetet megvalósító függvény: A listából kitöröl egy elemet és visszajelzést ad a törlés sikerességéről.

- **void test\_5(Ember e1, Ember e2, Tlista& l)**

5. tesztesetet megvalósító függvény: A keresés funkciót teszteli. A konzolablakban megjelenik az elvárt adat alatt az aktuális is, hogy le tudjuk ellenőrizni a keresés funkció megfelelő működését.

- **void test\_6(Ember e1, Ember e2, Ember e3, Tlista& l, char\* testfile)**

6. tesztesetet megvalósító függvény: A mentést teszteli. Hozzáad egy új adatot a listához, majd kimentí az adott .txt fájlba. Ezután onnan beolvassa és kiírja a lista tartalmát, hogy le lehessen ellenőrizni a mentést.