

Feladat:

A feladat egy telefonkönyv megvalósítása, ami eltárolja egy adott ember nevét, becenevét, lakcímét, privát- munkahelyi telefonszámát. A felhasználó három lehetséges parancs közül választhat. Adatok felvétele, adatok törlése, listázás.

Feladatspecifikáció:

Ha a felhasználó az adatok felvétele opciót választja akkor soronként megadhatja egy személyhez tartozó adatokat, amik közül nem kötelezőek az alábbiak: becenev, munkahelyi telefonszám, lakcím. Már a telefonkönyvben létező személlyel azonos nevű új személyt nem adhat hozzá a felhasználó.

Abban az esetben, ha a felhasználó a törlés mellett dönt, megadhat egy nevet és a program név alapján kitörli az megfelelő személyt a telefonkönyvből, törlés előtt a rendszer megerősítést kér a végrehajtásról.

Listázás parancs esetén a program névsor szerint felsorolja a telefonkönyvben már szereplő személyeket és adataikat soronként.

Ezekén kívül megnyitáskor betölti a már korábban elmentett személyeket, bezárás esetén elmenti a változtatásokat.

Hibakezelés:

A telefonkönyv mellé lesz egy külön modulban lefutó teszt is, ami különböző adattípusokkal és bevitelekkel teszteli a program helyességét.

A program helytelen beviteli formátum esetén a felhasználót tájékoztatja, szól a hibáról.

Sikertelen betöltés/mentés esetén is tájékoztat.

Pontosított specifikáció:

A feladat egy generikus láncolt lista megvalósítása, mely elemeinek típusát a sablon paramétereként tudjuk megadni és ezeket dinamikusan el is tudjuk tárolni. A generikus lista, mint osztály ezeket a nem implicit generált műveleteket tudja:

- Létrehozás
- Megszüntetés
- Elem hozzáadása
- Összes elem kiírása
- Egy adott elem törlése

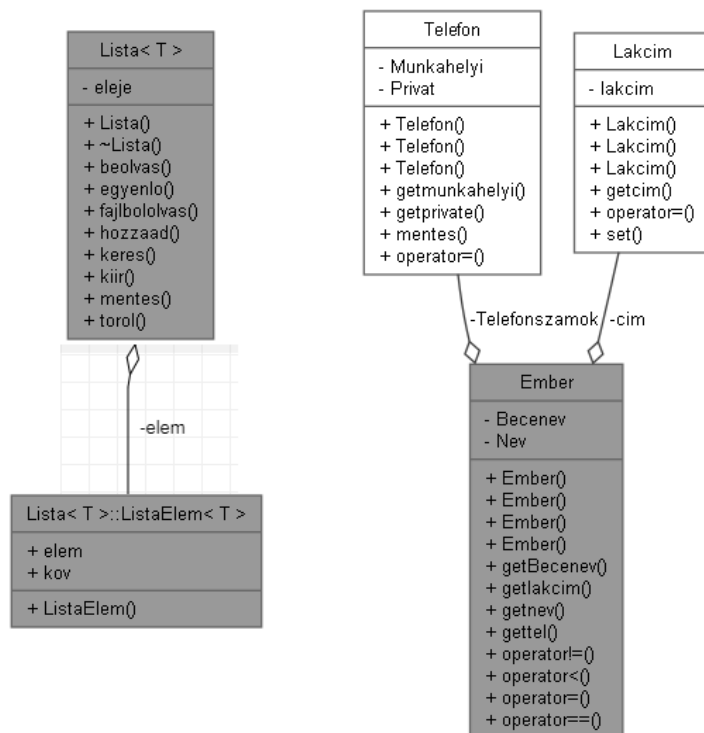
SZAKÁCS OLIVÉR
PROGRAMOZÁS ALAPJAI 2.
SPECIFIKÁCIÓ

- Elem keresése
- Két elem összehasonlítása

A teszteléshez olyan programot fogok írni, ami különböző műveleteket, különböző adattípusokkal műveletek végez testben lesz hibás mutató hiba is.

Hibák esetén a függvények kivételt dobnak, amik különböző függvényeknél eltérő.

Objektum terv:



Algoritmusok:

A program egyetlen összetettebb algoritmus a hozzáad függvény.

Ez a függvény rendezve épít, ami azt jelenti, hogy az új elem hozzáadása előtt összehasonlítja a már létező elemekkel (ebben az esetben az adott emberek nevét hasonlítja össze) és a feltételeknek megfelelő helyre illeszti be.

A feladat abc sorrend szerinti rendezést használ.

Megvalósítás:

A feladat egy sablon osztály és egy Ember adatait tároló 3 osztályt és a hozzájuk tartozó teszt elkészítését igényelte. A fájlok `std::string`, `std::stringstream`, `std::ctype` és `std::fstream` osztályokat használnak. A megvalósítás a tervtől csak pár apróbb lépésben lett más, illetve pár új függvény is kellett a feladat elkészítéséhez. Például egy függvény, ami az adott stringről eldönti, hogy csak számokat tartalmaz-e, ezzel a telefonszám megadását lehet ellenőrizni. A tesztelés és a telefonkönyv között egy makró definiálásival lehet váltani, hogy éppen melyik fusson le.

Tesztprogram

A tesztprogram 6 darab résztesztből áll, amik ezeket tesztelik:

1. Ebben a tesztben a program létrehoz egy integer listát és teszteli a hozzáadás függvényt, viszont mivel nincs visszatérési értéke ezért a keres függvény bizonyítja, hogy tényleg sikerült-e hozzáadni az elemet.
2. Ez a teszt a listából való törlést teszteli.
3. Itt a fájlba mentést, illetve a betöltést teszteljük egy integer listára.
4. Karakter listára teszteli a lista kiír függvényét.
5. A megvalósított Ember osztály operátorait teszteli különböző esetekre.
6. Az utolsó teszt a nem Listában alkalmazott, ellenőrző függvényeket nézi meg, mint pl. a fentebb említett csak számokat tartalmaz-e függvény.

Memóriakezelés

A program a már ismert MEMTRACE modult használja, ezzel ellenőrzi a helyes memória foglalást, illetve felszabadítást. Futások során nem merült fel memóriakezelési hiba.