

HÁZI FELADAT

Programozás alapjai 2.

Dokumentáció

Ötvös Róbert - F9ILEX

2024.05.18

Tartalomjegyzék

Feladat	3
Telefonkönyv	3
Feladatspecifikáció.....	3
Pontosított feladatspecifikáció.....	3
A program kezelőfelülete	5
A program főbb funkciói.....	6
1, Bejegyzés hozzáadása.....	6
2, Bejegyzés törlése	6
3, Bejegyzés módosítása	6
4, Telefonkönyv megtekintése.....	6
5, Keresés	6
6, Kilépés	6
Futtatás	6
Megjegyzés	6

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
- munkahelyi szám
- privát szám
- város
- előfizetés

Az alkalmazással minimum a következő műveleteket kívánjuk elvégezni:

- adatok felvétele
- adatok törlése
- listázás

A rendszer lehet bővebb funkcionalitású (pl. módosítás, keresés), ezért nagyon fontos, hogy jól határozza meg az objektumokat és azok felelősségét. Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz **ne** használjon STL tárolót!

Feladatspecifikáció

A feladat egy telefonkönyv készítése. Az alkalmazásom parancssorban fog futni. A megoldással lehet adatokat felvenni/törölni, módosítani, listázni és keresni.

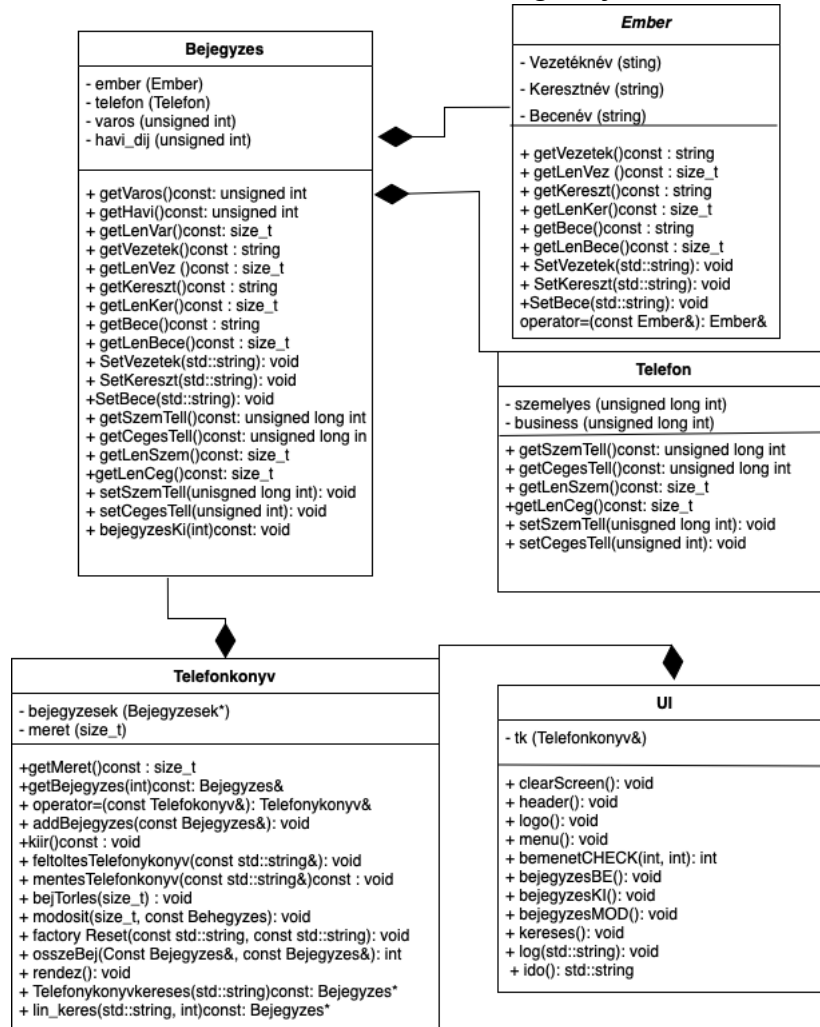
A telefonkönyvem 'bejegyzései' tartalmazni fognak nevet (keresztnév, vezetéknév, becenév) lakhely (város), munkahelyi és személyes telefonszámot, illetve előfizetés (Ft-ban). A felhasználó lépéseit, illetve a program főbb futási elemeit egy .log fájlban lesznek vezetve, illetve időbélyeggel ellátva.

A teszteléshez olyan programot készítek, melyben a konzolos kezelői felület standard inputjáról érkező hibás bemeneteket is helyesen kezel. Ezeket egyaránt jelzi a felhasználónak.

Pontosított feladatspecifikáció

A feladat megoldását az alábbi osztályokkal valósítom meg:

- **Ember:** Eltárolja a személyes adatokat az illetőről.
- **Telefon:** A privát és a cégtelefon számokat tárolja
- **Bejegyzes:** Eltárolja a körzetet (irányító szám), és a havi díját az előfizetőnek (Ft)
- **Telefonkonyv:** eltárolja az összes bejegyzést, illetve ő végzi a főbb műveleteket (pld.: kiírás, bejegyzés hozzáadása a listához)
- **UI:** A felhasználó felület működését szolgáltatja



A programom elindulásakor az adatbázisból (szöveges fájl) beolvassa a telefonkönyv szükséges adatait, és ezzel egyidejűleg kiíródik a konzolra a felhasználói felület. Ha hiányzik a fájl (source.txt) a program megpróbálja azt visszaépíteni (source_backup.txt), ha nem sikerül akkor leáll.

Algoritmusok, függvények

1. A get- kezdetű függvények minden esetben visszaadják a rejtett változók értékét, vagy állapotát
2. A set- kezdetű függvényekkel lehet változtatni a privát tagváltozók értékeit
3. bejegyzesKI függvény: kap egy int értéket (sorszám) és kiírja az adatit a console-ra.

4. Telefonkönyv hozzáadó függvénye(addBejegyzes): Képes egy új bejegyzést hozzáadni az adatbázishoz, mely a forrásfájlba is kiírásra kerül a program bezárta után, a beérkező adatokat a UI class testeli
5. kiir – Megépíti a kiíráshoz szükséges adatokkal a táblázatot
6. feltoltesTelefonkonyv – Indításkor a fájlból felépíti a telefonkönyvet, ha nem sikerül futtatja a hibajavító függvényt (factoryReset)
7. bejTorles – kitörli a megadott sorszámú bejegyzést. Az intervallumot a UI class teszteli
8. modosit – Megkapja a sorszámot és egy bejegyzést majd ennek megfelelően módosítja a már ott lévő bejegyzést
9. factoryReset – Biztonsági algoritmus. Amennyiben hiba keletkezik az éterben, akkor ez a függvény hívódik meg. Ha nincs source.txt, ez építi vissza, továbbá a test futásában is fontos szerepe van.
10. osszeBej - A rendezés miatt jött létre, gyakorlatilag megnézi, hogy melyik bejegyzes van előrébb és melyiknek kellene
11. rendez - Az osszeBej függvényt hívja meg, és az alapján eldönti, hogy helyes e az eddig jelen lévő rendezés.
12. Telefonkonyvkereses – vezetéknév alapján keresőfát épít (nyilván ezért is fontos a rendez) és kikeresi a fából a vezetéknévet, amennyiben van.
13. lin_keres – Egy egyszerű lineáris keresést alkalmazó függvény, a többi adat kereséséhez.
14. clearScreen – letörli a kijelzot
15. header() – kiírja a fejléct
16. logo() – kiírja a logót, amennyiben a fájl létezik
17. menu() – a menürendszer megvalósítása, gyakorlatilag ez a HUB
18. bemenetCHECK – Ez egy ellenőrző függvény, vele lehet tesztelni a felhasználó által biztosított adatok helyességét egy intervallumon
19. bejegyzesBE – Akkor hívódik meg, ha a felhasználó új bejegyzést szeretne hozzáadni a telefonkönyvhez, beépített ellenőrzővel van ellátva
20. bejegyzesKI – Bekér egy sorszámot, majd törli az adott bejegyzést a telefonkönyvből
21. bejegyzesMOD – Bekér egy módosítandó sorszámot, majd egy Bejegyzés létrehozásához szükséges adatokat, és lecseréli az ott lévő bejegyzésre
22. kereses – A keresést megvalósító függvény
23. log – Ez végzi a logolás folyamatát a log file-ba, segítségével nyomon követhetjük, mit csinált a felhasználó.
24. ido() – log file időbélyege

A program kezelőfelülete

A felhasználó az alábbi funkciók közül fog majd tudni választani:

- 1. Bejegyzés hozzáadása**
- 2. Bejegyzés törlése**
- 3. Bejegyzés módosítása**
- 4. Bejegyzések listázása**
- 5. Keresés**
- 6. Kilépés**

A program főbb funkciói

1, Bejegyzés hozzáadása

Ha a felhasználó az alábbi menüpontba lép, akkor lehetősége van hozzáadni saját bejegyzést, amennyiben a megfelelő adatokkal teszi azt.

2, Bejegyzés törlése

Bekérünk egy sorszámot, és az ha megfelelő akkor az törlésre kerül az adatbázisból.

3, Bejegyzés módosítása

Ebben a menüpontban lehetősége van a felhasználónak módosítani bizonyos bejegyzéseket.

4, Telefonkönyv megtekintése

Ha a felhasználó az alábbi menüpontba lép, akkor gyakorlatilag 'kinyitja' a telefonkönyvet. Kilstázódik az összes bejegyzés a könyvből.

5, Keresés

Ebben a menüpontban lefut a kereső algoritmusok összessége, ha van találat akkor azt a program egyértelműen jelzi a felhasználónak.

6, Kilépés

A program kilép, és elmenti az adatokat.

Futtatás

A programot ajánlatos olyan terminában futtatni, amely támogatja az ékezeteket, illetve az ANSCII karaktereket. A program MacOS rendszeren lett írva és Warp terminálban lett futtatva.

Az alábbi parancsot ajánlom a futtatáshoz:

```
g++ -o program main.cpp cpp/*.cpp -DMEMTRACE -Wall -Werror
```

Megjegyzés

A skeletonban javasolt javításokat, mind orvosoltam 😊.