HÁZI FELADAT

PROGRAMOZÁS ALAPJAI 2.

Eszes Roland

Tartalom

A Feladat ..... - 3 -
A Specifikáció ..... - 3 -
A terv ..... - 5 -
A számla kiszámító: ..... - 5 -
Dinamikus tömb hozzád adó ..... - 5 -
Tesztprogram algoritmusai ..... - 5 -
Megvalósítás ..... - 5 -
elofizetestar osztály ..... - 5 -
Publikus függvényei ..... - 6 -
tarifa osztály ..... - 6 -
Publikus függvényei ..... - 6 -
ugyfel osztály ..... - 7 -
Publikus függvényei ..... - 7 -
elofizetes osztály ..... - 8 -
Publikus függvényei ..... - 8 -
Osztályon kívüli függvények: ..... - 8 -
A tesztprogram bemutatása ..... - 9 -
A tesztek ..... - 9 -
Memóriakezelés tesztje ..... - 9 -

A Feladat

Az infocpp-n olvasott ötletet valósítottam meg. Egy mobilszolgáltatónál egy egyedi nyilvántartó programmal szeretném kezelni az ügyfeleket. A szolgáltató jelenleg három csomagot biztosít ügyfeleinek: Alap, Mobilnet és SMSMax, de később több csomag is lehet. Minden csomaghoz más perc, SMS és net díj tartozik Az ügyfeleknek van neve és címe, valamint telefonszáma, ami egyben az egyedi azonosítjuk is. Az ügyfelek adatait és a felhasznált perceket, smseket, illetve Megabyte-okat egy fájlból olvasom be.

A Specifikáció

A fő bevitelt egy ugyfel.txt fájl alkotja, melynek a programmal 1 mappában kell lennie. A fájlon keresztül tudjuk bevinni az ügyfelek adatait, melyeket szóközökkel elválasztva, a következő képen kell formázni:

```
sms 201234567 Kis_Pista Nagy_Lajos_utca_12 15 23 5
alap 202020202 Nagy_Pista Bp_Nagy_Lajos_kiraly_utja_8/b 11 11 11
net 06205002256 Kis_Geri Nem_Lakik_Sehol_23 32 12 11132
```

1. Díjcsomag megnevezése, jelenleg sms, alap vagy net, szót kell oda írni, amennyiben hibás lenne a bevitel a program szólni fog, hogy nem találja a megfelelő díjcsomagot.
2. Telefonszám, nem feltétlenül számokat szabad csak írni, fontos a szóköz mentesség.
3. Név, a könnyebb olvashatóság kedvéért ’_’ jelekkel jelöljük a szóközök helyét, ezt akár ponttal is lehet helyettesíteni, nincs megkötés rá.
4. Cím, szintén bármennyi karakter lehet, ahogy a név esetén itt is javasolt ’_’ jel használata.
5. Lebeszélt percek száma, pozitív egész számot várunk.
6. Felhasznált smsek száma, itt is pozitív egész számot várunk.
7. Felhasznált adatforgalom, melyet szintén pozitív egész számként várunk, Megabyte-ba átváltva.

A program elindítása után menü rendszerrel találkozunk, fontos amennyiben nem számot adunk meg, úgy egyből kilép a program. A menü 3 részből áll:

1. Rész: 2 választási lehetőségünk van, 0) kilépünk, 1) Beolvassuk a fájl tartalmát, innen jutunk el a 2. részbe.

```
Udv a Telener rendszereben
Mit szeretnel csinalni?
1) Beolvasni
0) Kilepni
Kerem Valasszon:
```

2. Rész: Itt 3 választási lehetőségünk van, tudunk ügyfelet keresni telefonszám alapján(innen jutunk a 3 részbe), kiírni az ügyfelek adatait és számláit, illetve kilépni, fontos megjegyzés, a menü többszörösen meg tud/ fog jelenni, ez azt jelenti, hogy akárhányszor kereshetünk, kiírhatunk.

```
----------------------------------------------------
Kerem valasszon!

1) ugyfelet keresek telefonszam alapjan
2)Kiirom  az ugyfelek szamlait
0) Kilepek
----------------------------------------------------
```

3. Rész: Az előző menü ügyfél keresője alapján jutottunk ide, itt egy telefonszámot várunk a felhasználótól, a program enterig olvas. Amennyiben nincs találat a rendszer jelzi, újból lehet próbálkozni, ha sikeres a keresés egy 4 pontú menüvel találkozunk: a legutóbb kereset ügyfelet tudjuk tarifa csomagonként megnézni, mennyit kellett volna fizetnie, melyeket az 1, 2, 3-as számokkal kezdeményezhetünk, ez a menü a 2eshez hasonlóan a kilépés gombig (0) nem áll meg.

```
----------------------------------------------------
Kerem valasszon!

1) ugyfelet keresek telefonszam alapjan
2)Kiirom  az ugyfelek szamlait
0) Kilepek
----------------------------------------------------
1
Kerem adja meg a telefonszamot
06205002256

----------------------------------------------------
nev: Kis_Geri
tel: 06205002256
cim: Nem_Lakik_Sehol_23
alapdij: 6000
felhasznalt percek/ingyenes: 32/0
felhasznalt smsek/ingyenes: 12/0
felhasznalt MB-ok/ingyenes: 11132/0
Szamla osszege: 8420Ft
----------------------------------------------------




----------------------------------------------------
 Kerem valasszon
Meg szeretnem nezni masik tarifacsomaggal:
1)alap tarifaval
2)sms tarifaval
3) net tarifaval
0) Kilepek
----------------------------------------------------
```

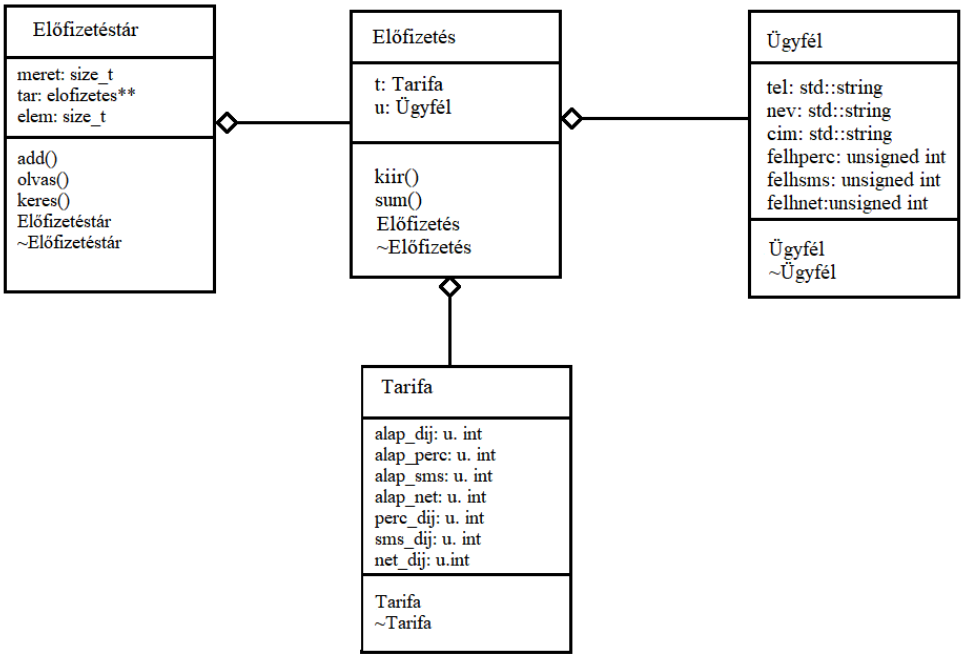
A terv

A számla kiszámító:

Megvizsgálom az ügyfél felhasznált adatainak nagyságát, amennyiben ezek nagyobbak a tarifában foglalt ingyenes mennyiségnél, a különbséget beszorzom a tarifa adott dijával, ezeket összeadom, hozzá adom az alapdíjat, így kapom meg a számla végösszegét.

Dinamikus tömb hozzád adó

Megvizsgálom, van e hely még a tömbben, amennyiben van, az adatot beteszem a helyére, amennyiben nincs, egy ideglenes pointerrel foglalok egy 2x akkora helyet, melyre átmásolom az adatokat, törlöm az eredetit, átmutatom a pointert az újra, és hozzáadom az elemet.



Tesztprogram algoritmusai

A gtest_lite használata segítségével írtam egy függvényt, mely a cporta makró hatására fut le. A függvény ellenőrzi: létrehozást, a dinamikus növelést, a beolvasást, az összeadó függvényt, és a kereső függvényt.

Megvalósítás

elofizetestar osztály

#include <elofizetestar.hpp>

Publikus függvényei

- **elofizetestar** (size_t meret)
  Konstruktor. Adattag nélkül hívható, az adatokat később tesszük bele.
- size_t **size** ()
  meghatározza a következő elem helyét a tárolóban
- size_t **capacity** ()
  meghatározza a tároló aktuális méretét.
- **elofizetes** * **operator[]** (unsigned int idx)
  operátor[] átdefiniálás, így kívülről is elérjük a tároló tömb elemeit, ellenőrzi az indexelés helyességét
- void **add** (**elofizetes** *p)
  a tömbhöz hozzáadd elofizetes elemet, amennyiben van hely, ha nincs növeli a tároló nagyságát.
- void **olvas** ()
  Az adatok olvassa be a fájlból, jelzi ha nem találja, továbbá az add függvény segítségével a tömbbe is rakja őket
- **elofizetes** * **keres** (std::string keresett)
  a stringként kapott telefonszámot kikeresi a számlák közül, és kiírja, ha talált ilyet.
- void **mindki** ()
  A tárolóban összes ügyfelet kiírja a szabványos kimenetre.

tarifa osztály

#include <tarifa.hpp>

Publikus függvényei

- **tarifa** (unsigned int alap_dij=0, unsigned int alap_perc=0, unsigned int alap_sms=0, unsigned int alap_net=0, unsigned int perc_dij=0, unsigned int sms_dij=0, unsigned int net_dij=0)
  Adattal és nélküle is hívható konstruktor.
- **~tarifa** ()
- void **s_dij** (unsigned int p)
  Beállítja az alapdíjat.
- void **s_sms** (unsigned int p)
  Beállítja az ingyenes smsek számát.
- void **s_net** (unsigned int p)
  Beállítja az ingyenes Megabyte-ok számát.
- void **s_perc** (unsigned int p)
  Beállítja az ingyenes percek számát.
- void **s_smsdij** (unsigned int p)
  Beállítja az sms díját.
  void **s_netdij** (unsigned int p)
  Beállítja az internet díját.

- void **s_percdij** (unsigned int p)
  Beállítja az sms díját.
- unsigned int **g_dij** ()
  Visszatér a díjjal.
- unsigned int **g_sms** ()
  Visszatér az ingyenes smsek számával.
- unsigned int **g_net** ()
  Visszatér az ingyenes Megabyte-ok számával.
- unsigned int **g_perc** ()
  Visszatér az ingyenes percek számával.
- unsigned int **g_smsdij** ()
  Visszatér az smsdíjjal.
- unsigned int **g_netdij** ()
  Visszatér a netdíjjal.
- unsigned int **g_percdij** ()
  Visszatér a percdíjjal.

ugyfel osztály

#include <ugyfel.hpp>

Publikus függvényei

- **ugyfel** (std::string tel=" ", std::string nev=" ", std::string cim=" ", unsigned int felh_perc=0, unsigned int felh_sms=0, unsigned int felh_net=0)
  Adattal és nélküle is hívható konstruktor.
- **~ugyfel** ()
- std::string & **g_tel** ()
  Visszatér a telefonszámmal.
- std::string & **g_nev** ()
  Visszatér az ügyfél nevével.
- std::string & **g_cim** ()
  Visszatér az ügyfél címével.
- unsigned int **g_felhperc** ()
  Visszatér a felhasznált percekkel.
- unsigned int **g_felhsms** ()
  Visszatér a felhasznált smsekkel.
- unsigned int **g_felhnet** ()
  Visszatér a felhasznált internettel.
- void **s_tel** (std::string p)
  Beállítja a telefonszámot.

- void **s_nev** (std::string p)
  Beállítja a nevet.
- void **s_cim** (std::string p)
  Beállítja a címet
- void **s_felhperc** (unsigned int p)
  Beállítja a felhasznált percek számát.
- void **s_felhsms** (unsigned int p)
  Beállítja a felhasznált smsek számát.
- void **s_felhnet** (unsigned int p)
  Beállítja a felhasznált Megabyteok számát.

elofizetes osztály

#include <elofizetes.hpp>

Publikus függvényei

- **elofizetes** (const **ugyfel** u, const **tarifa** t)
- **~elofizetes** ()
- **ugyfel** & **get_u** ()
  Visszatér az ügyfél adataival.
- **tarifa** & **get_t** ()
  Visszatér a tarifa adataival.
- int **sum** ()
  Összeszámolja a számla végösszegét.
- void **kiir** ()
  Kiírja az ügyfél adatait a konzolra.

Osztályon kívüli függvények:

- void **menu()**
  A menu függvények első példánya, a többi függvényt közvetlenül ő hívja, itt a beolvasást tudjuk elindítani.
- void **menu1()**
  Itt hajtódik végre a kereső, illetve a mindenkit kiíró függvény.
- void **menu2**(elofizetes* k)
  Ez a kereső almenüje, itt tudjuk megnézni az ügyfeleket másik tarifa csomaggal.
- void teszt()
  Ez a függvény hajtja végre a teszteseteket, csak a cporta makró hatására hajtódik végre.

A tesztprogram bemutatása

A tesztek

A main.cpp fájl tartalmazza, cporta makró hatására aktiválódik, és hajtódik végre.
Összesen 4 része van:

1. létrehozást teszteljük, létrehozunk egy 1 elemű tömböt, ellenőrizzük, majd megpróbálunk beletölteni 3 adatot, mely hatására 4 elemű lesz a tömb.
2. Itt a beolvasást teszteljük, az 1-3 ügyfélre, mindenkinél telefon, név és cím ellenőrzés.
3. A számla eredményét teszteljük, mindhárom ügyfélre.
4. Utolsó sorban a kereső függvényt teszteljük, egy kis trükkel, rákeresünk egy telefonszámra, amikor visszakapunk egy ügyfelet, lekérjük a telefonszámát és ezt ellenőrizzük.

A tesztek a program lényeges részét lefedik.

Memóriakezelés tesztje

A laborgyakorlaton használt MEMTRACE programot használtam, a projektbe definiáltam a makróját, egy pár alkalommal igen, de ezeket kijavítva nem tapasztaltam memóriakezelési hibát.