UNIVERZITET U TUZLI
FAKULTET ELEKTROTEHNIKE

Objektno-orijentirano programiranje

Zadaća 3

Tuzla, decembar/prosinac 2018.

Sadržaj

Uvod

Cilj zadaće je implementirati funkcionalni imenik koji će pohranjivati osnovne informacije o osobama, odnosno kontaktima. Imenik treba sve kontakte permanentno čuvati u CSV dokumentu. Uz zadaću je priložen CSV dokument (`imenik.csv`) sa inicijalnim stanjem imenika. Također, imenik će moći izlistati pohranjene kontakte, dodati novi kontakt, izmijeniti, pronaći ili obrisati postojeći. Naravno, imenik treba podržavati operacije za čitanje iz CSV dokumenta, kao i upisivanje u isti. Glavna restrikcija imenika jeste da se telefonski brojevi ne smiju duplicirati u više kontakata.

Projekat organizovati na sljedeći način:

Zadaca3 (glavni folder)

Contact.hpp (sadrži strukturu `Contact` te potpise `toString` i `fromString` funkcija)

Contact.cpp (sadrži implementaciju `toString` i `fromString` funkcija)

Storage.hpp (sadrži strukturu `Storage`, potpise funkcija iz zadataka 4, 5, 7, 8, 9 i 10 te implementaciju `createMap` funkcije iz zadatka 6)

Storage.cpp (sadrži implementaciju funkcija iz zadataka 4, 5, 7, 8, 9 i 10)

main.cpp (sadrži implementaciju zadatka 11)

Zadatak 1

Definirati strukturu `Contact` za čuvanje informacija o jednom kontaktu. Ta struktura treba imati polja za pohranu prezimena osobe, imena osobe, starosti osobe, telefonskog broja, e-mail adrese, firme u kojoj osoba radi, te grada gdje osoba živi.

Zadatak 2

Definirati funkcije koje pretvaraju strukturu iz prethodnog zadatka u `std::string (toString)`, kao i `std::string` u strukturu (`fromString`). Tekstualna reprezentacija strukture treba da bude u formatu: *`PREZIME,IME,STAROST,TELEFON,FIRMA,EMAIL,GRAD`*

Ukoliko funkcija `fromString` ne može generisati strukturu `Contact` zbog nevalidnog formata stringa, potrebno je generisati iznimku tipa `std::invalid_argument` iz zaglavlja `stdexcept`.

Zadatak 3

Definirati strukturu `Storage` koja čuva 3 kolekcije:

a) listu svih kontakata,
b) mapu koja mapira ime grada u listu iteratora koji pokazuju na kontakte koji žive u konkretnom gradu,
c) mapu koja mapira starost osobe u listu iteratora koji pokazuju na kontakte konkretne starosti.

Napomena:
Svi kontakti se nalaze u listi iz *a)* dijela zadatka. Mape iz dijelova *b)* i *c)* čuvaju iteratore na kontakte iz liste iz *a)* dijela zadatka.

Zadatak 4

Definirati funkciju koja učitava kontakte iz priloženog CSV file-a u listu svih kontakata. Ovdje trebamo koristiti funkciju iz 2. zadatka. Potpis funkcije:

```
Storage loadFromFile(const std::string& fileName);
```

Zadatak 5

Definirati funkciju koja upisuje sve kontakte u CSV file. Ovdje trebamo koristiti funkciju iz 2. zadatka. Potpis funkcije:

```
void writeToFile(const Storage&, const std::string& fileName);
```

Zadatak 6

Napraviti generičku funkciju `createMap` koja će iz proslijeđene liste, a na osnovu proslijeđenog kriterija, vratiti:

a) mapu koja kao ključ pohranjuje grad iz kojeg osoba dolazi, a kao vrijednost čuva listu iteratora koji pokazuju na osobe koje dolaze iz konkretnog grada, ili
b) mapu koja kao ključ pohranjuje starost osobe, a kao vrijednost čuva listu iteratora koji pokazuju na osobe konkretne starosti.

Potpis funkcije:

```
template <typename T>
std::map<T, std::list<iterator_t>> createMap(const std::list<Contact>&,
                                             std::function<T(iterator_t)>);
```

Gdje je `iterator_t`:

```
using iterator_t = std::list<Contact>::const_iterator;
```

Primjer kriterija:

```
auto extractCity = [](iterator_t it) { return it->city; };
auto extractAge = [](iterator_t it) { return it->age; };
```

Refaktorirati funkciju iz zadatka 4 tako da puni mape strukture `Storage` koristeći ovu generičku funkciju.

Zadatak 7

Definirati funkciju `addContact` koja korisniku omogućava dodavanje novog kontakta u listu kontakata. Potpis funkcije:

```
void addContact(Storage&, const Contact&);
```

Funkcija treba dodati i mapiranja za grad i starost.

Zadatak 8

Definirati funkciju `editContact` koja korisniku omogućava izmjenu postojećeg kontakta u listi kontakata.

```
void editContact(Storage&, const std::string& phoneNumber);
```

Zadatak 9

Definirati funkciju koja korisniku omogućava pretragu imenika po dva kriterija:
a) po prezimenu i imenu osobe
b) po telefonskom broju

Potpis funkcije proizvoljan.

Zadatak 10

Definirati funkciju `removeContact` koja korisniku omogućava brisanje određenog kontakta na osnovu telefonskog broja.

```
void removeContact(Storage&, const std::string& phoneNumber);
```

Zašto bi brisanje bilo puno veći problem u slučaju da smo za implementaciju odabrali vektor umjesto std::list? Šta se dešava sa iteratora u jednom, a šta u drugom kontejneru?

Zadatak 11

Kreirati aplikaciju koja će prilikom pokretanja učitati sve kontakte iz CSV dokumenta koristeći funkciju implementiranu u 4. zadatku (putanju do file-a treba unijeti korisnik na početku programa. Ukoliko unese prazan string pokušati učitati file `imenik.csv`), te popuniti strukturu iz 3. zadatka. Ova aplikacija zatim treba prikazati izbornik kao u nastavku i čekati korisnički unos:

```
Odaberite akciju:
       1 - prikazi sve kontakte
       2 - prikazi sve kontakte iz nekog grada
       3 - prikazi sve kontakte iste starosti
       4 - dodaj kontakt
       5 - obrisi kontakt
       6 - pronadji kontakt
       7 - snimi stanje u file
       0 - kraj
 =
```

Dalje, aplikacija treba iskoristiti funkcije iz zadataka 5, 7, 8, 9 i 10 kako bi zadovoljila korisničke naredbe. Ovdje trebamo obratiti pažnju da prilikom unosa novog kontakta, izmjene ili brisanja postojećeg trebamo osigurati konzistentnost kontejnera koji se nalaze u strukturi iz 3. zadatka.

Korisni linkovi

Ova zadaća uključuje rad sa tekstualnim datotekama i parsiranje `std::string`-ova. Sljedeći linkovi sadrže potpunu dokumentaciju, kao i primjere korištenja određenih funkcija za rad sa datotekama i manipulaciju `std::string`-ova:

- Pretvaranje cjelobrojnih i realnih broja u `std::string`: http://www.cplusplus.com/reference/string/to_string/
- Pronalaženje sadržaja unutar `std::string`-a: http://www.cplusplus.com/reference/string/string/find/
- Izdvajanje dijela sadržaja nekog `std::string`-a: http://www.cplusplus.com/reference/string/string/substr/
- Pretvaranje `std::string`-a u `int`: http://www.cplusplus.com/reference/string/stoi/
- Klasa za rad sa file stream-ovima za ulaz/izlaz: http://www.cplusplus.com/reference/fstream/ifstream/ i http://www.cplusplus.com/reference/fstream/ofstream/
    - Ovdje su nam posebno zanimljive metode: `open`, `is_open` i `close`.