

OBJEKTNO ORIJENTISANO PROGRAMIRANJE

- domaći zadatak broj 1 -

Funkcionalna specifikacija

Na programskom jeziku C++ implementirati statičku biblioteku (.lib) klasa za obradu tabelarnih podataka. Potom napisati glavni program (kao konzolnu .exe aplikaciju) koji testira mogućnosti biblioteke.

Specifikacija klase `TableEditor`

Klasa `TableEditor` predstavlja klasu za učitavanje, obradu i čuvanje tabelarnih podataka.

Propisno stvaranje i uništavanje

Implementirati propisno stvaranje i uništavanje objekata klase `TableEditor`.

Format tekstualne reprezentacije tabele

Tekstualna reprezentacija tabele se sastoji iz proizvoljnog broja **redova**:

- Prvi red u tekstu sadrži **tipove** i nazive **kolona** tabele u formatu `tip:naziv`. Informacije o kolonama su razdvojene zarezima. Naziv kolone sme da se sastoji isključivo iz slova, cifara i donje crte. Tip kolone je predstavljen jednoslovnom oznakom i može da bude tekstualan (t), celobrojan (i) ili decimalan sa tačnošću 0.1 (d).
- Ostali redovi u tekstu predstavljaju skupove povezanih podataka. Svaki podatak u redu naziva se **ćelija**. Vrednosti ćelija u redu su odvojene zarezima. Moguće je izostaviti vrednost neke od ćelija tako što se ne navodi sadržaj između odgovarajućih znakova zareza. Ukoliko vrednost ćelije sadrži znak zareza, cela vrednost ćelije mora da se ogradi znakovima navoda. Tipovi vrednosti u ćelijama odgovaraju tipu kolone u kojoj se nalaze.

Primer stvarne tabele i tekstualne reprezentacije tabele prikazane u gorenavedenom formatu:

naziv	godina	ocena	domaci	
Toma	2021	9.2	DA	<code>t:naziv,i:godina,d:ocena,t:domaci</code> <code>Toma,2021,9.2,DA</code>
Fight Club	1999	8.8	NE	<code>Fight Club,1999,8.8,NE</code>
Monsters, Inc.	2001	8.1	NE	<code>„Monsters, Inc.“,2001,8.1,NE</code>
The Matrix Resurrections	2021		NE	<code>The Matrix Resurrections,2021,,NE</code>

Učitavanje i čuvanje tabela

Implementirati operaciju `int TableEditor::importTable(string table);` koja učitava informacije o tabeli po datom formatu sadržanoj u parametru `table`. Ukoliko je prethodno već učitana tabela, sve informacije o staroj tabeli se brišu i učitava se nova tabela. Povratna vrednost je indikator uspešnosti učitavanja tabele (-2 ukoliko je učitavanje tabele uspešno, odnosno broj reda koji nije sintaksno ispravan ukoliko je došlo do greške, pri čemu je red sa nazivima kolona označen brojem -1, a prvi red sa podacima je označen brojem 0).

Implementirati operaciju `string TableEditor::exportTable();` koja kao povratnu vrednost treba da vrati tabelu u datom formatu.

Selekcije u tabeli

U tabeli je moguće vršiti selekciju reda, kolone ili ćelije. Prilikom obrade podataka u tabeli, neke od operacija vrše obradu nad selektovanim delom tabele. U suprotnom, operacija nema efekta.

Tabela čuva samo poslednju selekciju, tako da nije moguće selektovati više redova, kolona ili ćelija.

Implementirati operaciju `void TableEditor::deselect();` koja eksplicitno poništava prethodnu selekciju.

Implementirati operaciju `void TableEditor::selectRow(int row_position);` koja selektuje red određen parametrom `row_position`. Nije moguće selektovati red sa nazivima kolona.

Implementirati operaciju `void TableEditor::selectColumn(string col_name);` koja selektuje kolonu čiji naziv odgovara parametru `col_name`.

Implementirati operaciju `void TableEditor::selectCell(int row_position, string col_name);` koja selektuje ćeliju u preseku reda određenog parametrom `row_position` i kolone čiji naziv odgovara parametru `col_name`.

Manipulacija nad strukturom tabele

Implementirati operacije koje dodaju nov red/kolonu ispred selektovanog reda/kolone. Ukoliko red/kolona nisu selektovani, red/kolona se dodaju na kraj.

```
void TableEditor::insertRow();
```

```
void TableEditor::insertColumn(string col_name, Type type);
```

Pri dodavanju reda, od korisnika se traži da sa standardnog ulaza unese vrednosti svih ćelija redom kojim su kolone navedene. Pri dodavanju kolone, kroz parametre se zadaju naziv kolone i tip kolone. Tip kolone (*Type*) može da uzme jednu od sledećih vrednosti *INT*, *DECIMAL* ili *TEXT*. Početne vrednosti svih ćelija u koloni su nulte u slučaju celobrojnog i decimalnog tipa, odnosno ne postoje u slučaju tekstualnog tipa.

Implementirati operaciju koja briše selektovan red/kolonu/sadržaj ćelije. Ukoliko selekcija ne postoji, operacija je bez efekta.

```
void TableEditor::deleteContent();
```

Implementirati operaciju koja menja sadržaj selektovane ćelije. Ukoliko selekcija ne postoji ili je selektovan red ili kolona, operacija je bez efekta. Pri izmeni sadržaja ćelije, od korisnika se traži da sa standardnog ulaza unese novu vrednost.

```
void TableEditor::editCellValue();
```

Obrada podataka u tabeli

Implementirati operacije koje sabiraju ili množe vrednost selektovane ćelije ili vrednosti svih ćelija selektovane kolone sa vrednošću `value`. Ukoliko nije selektovana ćelija ili kolona, operacija je bez efekta. Operaciju je moguće izvršiti isključivo nad podacima celobrojnog ili decimalnog tipa. Ukoliko je podatak tekstualnog tipa, operacija je bez efekta.

```
void TableEditor::add(double value); void TableEditor::mul(double value);
```

Implementirati operaciju `void TableEditor::sortByValue(string col_name, bool asc);` koja sortira redove u tabeli na osnovu vrednosti ćelija unutar kolone sa nazivom `col_name` rastuće ukoliko je `asc` postavljen na tačnu vrednost, a opadajuće u suprotnom. U slučaju da su podaci tekstualnog tipa, sortiranje se vrši po alfabetskom poretku.

Pretraga podataka u tabeli

Implementirati operacije `int TableEditor::findFirstOf(string value, string col_name);` i `int TableEditor::findLastOf(string value, string col_name);` koje vraćaju indeks prvog, odnosno poslednjeg reda koji u preseku sa kolonom `col_name` sadrži ćeliju sa vrednošću `value`. Ukoliko pretraga nije uspešna, povratna vrednost je -2. Vrednost za pretragu je uvek data u tekstualnoj formi (npr. „tekst“, „132“ ili „2.3“).

Statistički podaci u tabeli

Implementirati operaciju `int TableEditor::countValues(string value, string col_name);` koja vraća broj pojavljivanja vrednosti `value` u ćelijama kolone `col_name`.

Implementirati operaciju `int TableEditor::countDistinctValues(string col_name);` koja vraća broj različitih vrednosti ćelija u koloni `col_name`.

Test funkcija

Javni test sadrži funkciju `void test()`; koja testira obradu podataka u tabeli. Studentima je javno dostupna implementacija funkcije `test` i mogu da je menjaju da bi dodatno testirali svoj kod kao što je opisano komentarima u kodu. Funkcija `test` vrši učitavanje tekstualnih datoteka sa tabelama u opisanom formatu, zatim vrši obradu nad podacima učitane tabele i na kraju kreira nov fajl u istom formatu. Studentima je dat fajl sa tekstualnom reprezentacijom tabele koji predstavljaju ulazni test primer kao i fajlovi sa očekivanim odgovarajućim izlazima radi mogućnosti provere. Test primeri treba da budu smešteni u folder glavnog projekta.

Tehnički zahtevi i smernice za izradu rešenja

Iz kolekcije standardne biblioteke dozvoljeno je koristiti SAMO tip podataka *string*. NIJE DOZVOLJENO koristiti kolekcije iz standardne biblioteke, već isključivo ugrađene i sopstevne tipove podataka i strukture. Sve klase i operacije moraju biti imenovane prema zahtevima iz domaćeg zadatka. Voditi računa o rukovanju dinamičkom memorijom! Programski kod klasa rasporediti u odgovarajuće **.h** i **.cpp** fajlove. Nije dozvoljeno korišćenje globalnih promenljivih za razmenu podataka. Ukoliko u zadatku nešto nije dovoljno jasno definisano, treba usvojiti razumnu pretpostavku i na temeljima te pretpostavke nastaviti izgrađivanje svog rešenja.

VAŽNE NAPOMENE

Za uspešno odbranjen domaći zadatak potrebno je na odbrani pokazati kod podeljen na odgovarajuće projekte, **.h** i **.cpp** fajlove.

1. Klase kojima su implementirani osnovni koncepti treba da budu smeštene u poseban projekat rešenja koji se prevodi kao statička biblioteka (`TableEditor.lib`).
2. Glavni program treba da se nalazi u posebnom projektu koji se prevodi kao Win32 Console Application (`testdz1.exe`) fajl i koji treba povezati sa statičkom bibliotekom. Glavni program je dat u fajlu `Test.cpp` koji je javno dostupan i kojeg treba uključiti u projekat.
3. NIJE DOZVOLJENO SMESTITI CEO KOD U JEDAN PROJEKAT ILI CPP fajl!