

OBJEKTNO ORIJENTISANO PROGRAMIRANJE

- domaći zadatak broj 1 -

Funkcionalna specifikacija

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

Specifikacija klase *SubtitleEditor*

Klasa `SubtitleEditor` predstavlja klasu za učitavanje, obradu i čuvanje sekvence **titlova**.

Propisno stvaranje i uništavanje

Implementirati propisno stvaranje i uništavanje objekata klase `SubtitleEditor`.

Format titla

Format jednog titla u sekvenci koji sistem treba da podrži je sledeći:

- Titl se sastoji iz nekoliko delova međusobno odvojenih sa po jednim znakom za prelaz u novi red.
- Titl počinje numeričkom oznakom koja označava redosled titla u sekvenci. Prvi titl u sekvenci ima numeričku oznaku 1, drugi titl ima oznaku 2, itd.
- U sledećem redu titla nalaze se informacije o vremenima u toku kojih se titl prikazuje. **Vreme** se sastoji iz sata, minuta, sekundi i milisekundi i sledećeg je formata: *sati:minuti:sekunde,milisekunde*. Sati, minuti i sekunde predstavljaju se na širini od dve cifre, a milisekunde na širini od tri cifre. Informacije o vremenima počinju vremenom u kojem se započinje prikaz titla, nakon kojeg sledi znak razmaka, oznaka „-->“, još jedan znak razmaka i na kraju vreme u kojem se završava prikaz titla.
- U sledećih nekoliko redova se nalazi tekst titla. Minimalan broj redova je jedan, dok maksimalan broj redova nije definisan. Svaki red teksta titla može da bude stilizovan tako da slova prilikom prikaza titla budu udebljana, podvučena ili pisana u kurzivu. Stilizovan deo teksta se nalazi između specijalnih oznaka za stilizovanje: „<f>“ za početak stilizovanja i „</f>“ za kraj stilizovanja. Oznaka „f“ za stil može da uzme jednu od sledećih vrednosti: „b“ za podebljan tekst (*bold*), „i“ za tekst pisan u kurzivu (*italic*) ili „u“ za podvučen tekst (*underline*). Nisu podržani ugnedžđeni stilovi.
- Poslednji red titla mora da bude prazan red.
- Poslednji red u sekvenci je takođe prazan red.

Primeri sekvenci od dva titla (levo - bez stilizovanja, desno - sa stilizovanjem):

| | |
|---|--|
| <pre>1 00:26:44,019 --> 00:26:48,230 All right, to a night we'll never remember...</pre> | <pre>1 00:00:47,255 --> 00:00:49,882 DOUG [ON RECORDING]: <i>You've reached Doug. Sorry I missed your call.</i></pre> |
| <pre>2 00:26:50,066 --> 00:26:52,943 ...but the four of us will never forget.</pre> | <pre>2 00:00:50,050 --> 00:00:52,217 <i>Please leave a name and number and I'll get back to you.</i></pre> |

Učitavanje i čuvanje sekvence titlova

Implementirati operaciju `int SubtitleEditor::loadSubtitles(string subtitles);` koja učitava informacije o sekvenci titlova po datom formatu sadržanoj u parametru `subtitles`. Ukoliko je prethodno već učitana sekvenca titlova, sve informacije o staroj sekvenci se brišu i učitava se nova sekvenca. Povratna vrednost je indikator uspešnosti učitavanja titlova (povratna vrednost je `-1` ukoliko je učitavanje titlova uspešno, odnosno redni broj titla u sekvenci koji nije sintaksno ispravan ukoliko je došlo do neke greške).

Implementirati operaciju `string SubtitleEditor::saveSubtitles();` koja kao povratnu vrednost treba da vrati sekvencu titlova u datom formatu.

Selekcija dela sekvence

Editor sekvence titlova ima mogućnost selekcije dela sekvence. Prilikom obrade sekvence titlova, neke od operacija imaju mogućnost obrade nad selektovanim delom sekvence. Ukoliko nijedan deo sekvence nije selektovan, obrada se vrši nad celom sekvencom titlova ili operacija nema efekat.

Implementirati operaciju `void SubtitleEditor::addToSelection(int start, int end);` koja u selekciju dodaje deo sekvence koja obuhvata titlove čiji se redni brojevi nalaze u zadatom opsegu vrednosti, uključujući i granice. Ukoliko je donja granica van opsega, uzeti vrednost `1`. Ukoliko je gornja granica van opsega, uzeti redni broj poslednjeg titla u sekvenci.

Implementirati operaciju `void SubtitleEditor::deselect();` koja uklanja trenutnu selekciju.

Stilizovanje titlova

Implementirati operacije:

```
void SubtitleEditor::boldText(int start, int end);  
void SubtitleEditor::italicText(int start, int end);  
void SubtitleEditor::underlineText(int start, int end);
```

koje stilizuju karaktere u opsegu prosleđenih parametara `start` i `end` svih titlova u selekciji. Ukoliko je donja granica van opsega, uzeti vrednost `0`. Ukoliko je gornja granica van opsega, uzeti redni broj poslednjeg karaktera za svaki titl. Operacija je bez efekta ukoliko selekcija ne postoji.

Implementirati operaciju `void SubtitleEditor::removeStyle();` koja uklanja stilizovanje titlova u selekciji. Operacija je bez efekta ukoliko selekcija ne postoji.

Obrada sekvence titlova

Implementirati operaciju `void SubtitleEditor::shiftForward(int ms);` koja pomera sve titlove u sekvenci (ili selekciji, ukoliko postoji) napred za `ms` milisekundi.

Implementirati operaciju `void SubtitleEditor::shiftBackward(int ms);` koja pomera sve titlove u sekvenci (ili selekciji, ukoliko postoji) nazad za `ms` milisekundi.

Implementirati operaciju `void SubtitleEditor::insertSubtitle(int start, int length, string text);` koja dodaje novi titl u trenutku `start` (u milisekundama) i trajanja `length` (u milisekundama) sa tekstom `text`. Voditi računa o ažuriranju rednih brojeva titlova u sekvenci nakon dodavanja novog titla.

Implementirati operaciju `void SubtitleEditor::deleteSubtitles();` koja briše sve titlove u selekciji, ukoliko postoji. Ukoliko selekcija ne postoji, operacija je bez efekta. Voditi računa o ažuriranju rednih brojeva titlova u sekvenci nakon brisanja titlova. Nakon operacije, selekcija više ne postoji.

Implementirati operaciju `void SubtitleEditor::merge(string subtitles);` koja učitava informacije o sekvenci titlova po datom formatu sadžane u parametru `subtitles`, a potom spaja trenutno učitavanu sekvencu sa novom sekvencom vodeći računa o ažuriranju rednih brojeva titlova u sekvenci nakon spajanja.

Rešavanje grešaka u sekvenci titlova

Implementirati operaciju `void SubtitleEditor::fixLongLines(int max);` koja sve titlove u sekvenci (ili selekciji, ukoliko postoji) koji u barem jednom svom redu sadrže veći broj nebelih karaktera od parametra `max` ispravlja tako što tekst titla prelama u veći broj redova tako da svi redovi sadrže jednak ili manji broj karaktera od parametra `max`. Prelom se vrši na granici reči tako da u svakom redu (osim, eventualno, poslednjem) stane maksimalan mogući broj karaktera. Voditi računa da se razmaci između reči koje se prelamaju uklone.

Implementirati operaciju `void SubtitleEditor::fixLongTime(int ms);` koja sve titlove u sekvenci (ili selekciji, ukoliko postoji) čije je vreme trajanja duže od `ms` milisekundi ispravlja tako što ih deli u više uzastopnih titlova koji se prikazuju u uzastopnim, jednakim, vremenskim intervalima. Broj uzastopnih titlova je minimalan, a takav da je trajanje svakog titla manje od parametra `ms`. Deljenje se vrši na granici reči tako da u svakom uzastopnom titlu (osim, eventualno, nekoliko poslednjih) stane isti broj reči. Znakovi za novi red se uklanjaju.

Dodatne opcije

Implementirati operaciju `void SubtitleEditor::undo();` koja uklanja poslednju izmenu nad sekvencom titlova vraćajući je u prethodno stanje.

Test funkcija

Javni test sadrži funkciju `void test();` koja testira obradu sekvence titlova. 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 sekvencom titlova u `.srt` formatu, zatim vrši obradu nad učitanim sekvencom titlova i na kraju kreira nov fajl u `.srt` formatu nakon obrade. Studentima su dati fajlovi sa titlovima koji predstavljaju ulazne test primere kao i fajlovi sa titlovima koji predstavljaju očekivane odgovarajuće izlaze 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 (`subtitleEditor.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!