



Univerzitet u Zenici

Politehnički fakultet

2022/23

Napredne tehnike programiranja

Prva godina I. ciklus

Dokumentacija prijeka

Videoteka, naručivanje određenih filmova, ocjenjivanje
filmova(C++)

Članovi/ce grupe:

-Ismić Aldina

-Selimović Omar

-Kovač Tarik

Pseudokod

<https://github.com/OmarSelimovic/NTP/blob/main/Videoteka%20Pseudokod.txt>

Originalni plan za izradu prototipa programa za upravljanje videotekom nam je dao osnovne smjernice, olakšao je izradu, planiranje i raspoređivanje obaveza unutar tima.

Iako je, na kraju, gotov c++ kod puno veći i opširniji, ipak olakšava osobi koja ga čita da shvati osnovni koncept i ideju koju smo imali prilikom rada na programu.

Važno je naglasiti da smo imali nekoliko problema koji su usporili samu izradu programa ali smo, zahvaljujući produženom roku za izradu, stigli uraditi sve potrebno nekoliko dana prije novog roka. Najveći problem u izradi predstavljao je gubitak podataka sa kompjutera koji je rezultirao gubljenjem 70% progrsa ostvarenog do tada jer je sigurnosna kopija, urađena 2 dana prije korupcije podataka zbog virusa, nije pokrivala progres ostvaren u toku ta dva dana ektenzivnog rada.

Komentari/objašnjenje koda

Originalni plan je bio da koristimo klase sa private memberima kojii bi bili određene vrijednosti (npr. Ime, Username itd.) i public funkcijama za upravljanje tim varijablama, sto bi omogućilo izmjenu varijabli vezanih za objekte samo korištenjem funkcija specifičnih za tu klasu.

Međutim kada smo saznali da upotreba klasa nije preporučena, klase smo prebacili na strukture i učinili smo sve članove klasa public. Kao rezultat toga, još uvijek imamo Korisnik.cpp i Film.cpp fajlove gdje su funkcije koje su postale jednim dijelom nepotrebne sada kada su Korisnici i Filmovi strukture. Obzirom da je dio koda već koristio te funkcije, odlučili smo ih zadržati da bismo uštedili vrijeme i izbjegli moguće probleme koji bi nastali izmjenom koda.

Unutar samog main.cpp imamo deklaraciju 22 funkcije koje su međusobno uvezane i ostvaruju sve potrebno za ispravan rad programa.

Najlakše bi bilo sve te funkcije podijeliti u 3 grupe:

-Funkcije prije prijavljivanja korisnika

```
void Dobrodosli();  
bool Pocetna_Stranica();  
void PravljenjeKorisnika(vector<Korisnici>& users);  
bool LoginAuth(Korisnici& User, vector<Korisnici> ListaKorisnika , int& redniBroj);  
vector<Korisnici> UcitavanjeKorisnika(const string& imefajla);  
vector<Filmovi> UcitavanjeFilmova(const string& imefajla);
```

-Glavne funkcije programa

```
void DobrodosliNazad(Korisnici TrenutniKorisnik);  
void GlavniMeni(vector<Filmovi>& ListaFilmova, vector<Korisnici>& ListaKorisnika,  
    Korisnici& TrenKorisnik, int RBRTrenKor);  
void PokaziFilmove(vector<Filmovi> ListaFilmova);  
void NarucivanjeFilmova(vector<Filmovi>& ListaFilmova,Korisnici& TrenKorisnik);  
void VracanjeFilmova(vector<Filmovi>& ListaFilmova,Korisnici& TrenKorisnik);  
void PokaziKorisnike(vector<Korisnici> ListaKorisnika);  
void EditujKorisnike(vector<Korisnici>& ListaKorisnika);  
void BrisanjeKorisnika(vector<Korisnici>& ListaKorisnika);  
void EditujFilmove(vector<Filmovi>& ListaFilmova);  
void BrisanjeFilma(vector<Filmovi>& ListaFilmova);  
void DodavanjeFilma(vector<Filmovi>& ListaFilmova);
```

-Funkcije spremanja podataka i gašenja

```
void SpremanjeKorisnika(vector<Korisnici>& ListaKorisnika, Korisnici TrenKorisnik,  
    int RBrTrenKorisnik);
```

```
void SpremanjeListeKorisnika(const vector<Korisnici>& ListaKorisnika,  
    const string& imefajla);  
void SpremanjeListeFilmova(const vector<Filmovi>& ListaFilmova, const string& imefajla);  
void PrevisGresaka();  
void Sekvenca_Gasenja();
```

Funkcije prije prijavljivanja korisnika

```
vector<Korisnici> UcitavanjeKorisnika(const string& imefajla);  
vector<Filmovi> UcitavanjeFilmova(const string& imefajla);
```

Ove dvije funkcije su zadužene da, svaki put kada se program pokrene, učitaju informacije sa fajlova i upišu ih u potrebne vektore struktura.

U slučaju da su fajlovi prazni, unutar main(); programa imamo fiksni kod koji će inicijalizirati jednog Administrator korisnika i jednog običnog korisnika za svrhe postavljanja i testiranja programa.

Za sljedeće dvije funkcije se može reći da rade zajedno, samim tim se i pozivanju jedna za drugom unutar main(); funkcije.

```
void Dobrodosli();
```

-Na display ispisuje logo videoteke

```
bool Pocetna_Stranica();
```

-Bool funkcija čija povratna vrijednost određuje da li će biti pokrenuta funkcija za pravljenje novog korisnika ili funkcija prijavljivanja već postojećeg korisnika

```
void PravljenjeKorisnika(vector<Korisnici>& users);
```

-Funkciji se proslijeđuje potpuna lista korisnika po referenci što će omogućiti da lista bude mijenjana unutar same funkcije.

U ovoj funkciji se uzimaju 3 unosa korisnika: username, ime i prezime, password.

Pri unosu username-a program će provjeriti da li je taj username već u upotrebi, da ne bi došlo do preklapanja prilikom upotrebe programa.

Da bi posao bio olakšan, te vrijednosti se prvo učitavaju u privremeni UserVideoteke koji će biti spremljen u glavni vektor koji predstavlja potpunu listu korisnika. Nakon unosa i pravljenja novog usera, korisnik biva proslijeđen na Log-in meni.

```
bool LoginAuth(Korisnici& User, vector<Korisnici> ListaKorisnika, int& redniBroj);
```

-Bool funkcija na koju prosljeđujemo potpunu listu korisnika kao i potpuno novog korisnika i int redni broj.

U ovoj funkciji se od korisnika traži da unese username i password, u slučaju da podudarni podaci budu pronađeni u listi korisnika potpune informacije o tom korisniku bivaju kopirane na User da bismo izbjegli potrebu da svaki put listamo potpunu listu korisnika kada želimo uciniti neku promjenu. Također i redni broj na kojem se taj korisnik nalazi u ukupnoj listi biva spremljen da bismo sebi olakšali spremanje promjena kasnije u toku programa. U slučaju da korisnik unese pogrešne podatke 3 puta pokreću se funkcije

```
void PrevisGresaka(); i void Sekvenca_Gasenja();
```

 koje ćemo objasniti kasnije.

Glavne funkcije programa

Nakon uspješne autentikacije pokreću se dvije funkcije koje nas uvode u glavnu funkcionalnost programa. Te funkcije su:

```
void DobrodosliNazad(Korisnici TrenutniKorisnik);
```

```
void GlavniMeni(vector<Filmovi>& ListaFilmova, vector<Korisnici>& ListaKorisnika,  
Korisnici& TrenKorisnik, int RBRTrenKor);
```

Funkcija DobrodosliNazad je funkcija koja će označiti uspješnu autentikaciju i ispisati ime i prezime korisnika, filmove koje su uzeli do sada ili u slučaju da nemaju uzetih filmova odgovarajuću poruku. Ako je korisnik upozoren od strane administratora odgovarajuća poruka će biti ispisana i automatski obrisana nakon ispisa ali broj upozorenja ostaje.

U slučaju da je korisnik administrator ispis će reći da se radi o administratorskom računu.

Funkcija glavnog menija se grana na dvije zasebne strane, jedna za korisnički meni a druga za administratorski meni. Naime, radi se o petlji koja će se ponavljati sve dok korisnik ne izabere EXIT, u slučaju odabira neke druge opcije, ta radnja će biti obavljena i petlja glavnog menija ponovno pokrenuta. U slučaju odabira neke specifične opcije unutar glavnog menija, korisnik biva upućen na sljedeći sporedni meni koji ima svoje zasebne opcije, ovaj sporedni meni je također loop koji će se ponavljati sve dok korisnik ne izabere opciju 'NAZAD'. Isti princip objašnjen gori važi i za administratora i običnog korisnika. Važno je naglasiti da unutar same liste korisnika i Admin i User imaju istu strukturu podataka, međutim bool Admin; odlučuje koji meni će biti pokazan. Samim tim se Administratoru zabranjuje upotreba korisničkog menija i Korisniku zabranjuje kontrola nad sistemom.

Da bismo dalje razradili funkcionalnosti, podijelit ćemo objašnjenje na dva dijela: Korisnički meni i Administratorski meni.

Korisnički meni:

Korisniku se daje izbor između 4 opcije koje pokreću svoje zasebne funkcije. Prva opcija pod brojem [1] je ispis filmova. Kada korisnik izabere ovu opciju pokreće se funkcija

```
PokaziFilmove(ListaFilmova);
```

 .

Ova funkcija je zadužena za formatiran i čitak ispis svih filmova, njihovih imena, njihovog žanra, ocjene filma i same dostupne količine kojom videoteka raspolaže.

Pod brojem [2] nalazi se funkcija koja je zadužena [NarucivanjeFilmova\(\)](#); , pri pokretanju ove funkcije, automatski se prvo pokreće funkcija ispisa jer će korisnik birati film na osnovu rednog broja koji se nalazi pored filma.

Korisnik prije svega bira koliko filmova želi naručiti (max 3) što će biti broj ponavljanja komande biranja filma. Nakon odabira svakog zasebnog filma program ispisuje poruku potvrde “Uživajte u gledanju filma ^_^”. Nakon isteka ponavljanja(broja filmova) korisnik biva vraćen na glavni meni.

Pod brojem [3] nalazi se funkcija [VracanjeFilmova\(\)](#); . Ova funkcija uzima i ispisuje formatiranu listu filmova iz člana `vector<string> Uzeti_Filmovi` objekta koji predstavlja trenutnog korisnika. Korisnik je predstavljen sa upitom “Koliko filmova želite vratiti (broj filmova koje korisnik ima)”. Broj koji korisnik unese predstavlja broj ponavljanja petlje koja će brisati film iz liste uzetih filmova, smanjiti broj uzetih filmova i za taj odgovarajući film na glavnom popisu filmova povećati količinu kopija dostupnih.

Nakon uspješnog vraćanja filma korisnik je upitan da li želi ostaviti ocjenu za dati film jer, obzirom da film vraća, trebalo bi značiti da je film pregledan. Naravno postoji opcija da se film ne ocjeni, ali u slučaju da korisnik odluči ocijeniti film to može u vrijednosti od 1 do 10.

Poslije uspješnog ocjenjivanja ili odbijanja istog korisnik biva vraćen na glavni meni.

Pod brojem [4] nalaze se funkcije spremanja informacija i gašenje programa.

Administratorski meni:

Administratorski meni je zadužen da upravljanjem liste filmova i liste korisnika. Pored opcija za ispis [1] i [3] imamo i dvije opcije [2] i [4] koje otvaraju zaseban sporedni meni.

Pod brojem [1] se nalazi identična funkcija kao i u korisničkom meniju za ispis liste filmova.

Pod brojem [3] se nalazi funkcija [PokaziKorisnike\(ListaKorisnika\)](#); koja, kako i ime sugestira pokazuje potpunu listu korisnika, njihov username, ime i prezime, broj upozorenja i listu filmova koji su kod tih korisnika.

Odabirom opcije [2] pokreće se funkcija ispisa svih filmova identično kao i pod opcijom [1], međutim nakon ispisa liste pojavljuje se sporedni meni nazvan “Editovanje Filmova”.

Unutak menija za Editovanje Filmova nalazi se 5 opcija:

[1]-Izbrisi film - [BrisanjeFilma\(vector<Filmovi>& ListaFilmova\)](#) ;

[2]-Dodaj novi film - [DodavanjeFilma\(vector<Filmovi>& ListaFilmova\)](#);

[3]-Povecaj broj filmova -
[4]-Smanji broj filmova -
[5]-<----NAZAD<----

Ove opcije rade ono što im i ime govori međutim bitno je naglasiti da, prve dvije opcije pokreću svoje zasebne funkcije dok su opcije [3] i [4] pisane direktno unutar funkcije

`EditujFilmove(vector<Filmovi>& ListaFilmova);` ova funkcija obuhvata cijeli sporedni meni i sve opcije pokazane iznad koje su u petlji i ponavljat će se sve dok izbor nije [5];

Opcije [3] i [4] ,nakon testiranja valjanosti unosa i izbora korisnika, traže unos količine tj. Broj filmova kojeg želite dodati/oduzeti. Taj broj je ograničenje ponavljanja petlje u kojoj se poziva funkcija `ListaFilmova[broj].povecajBroj_filmova();` ili `ListaFilmova[broj].smanjBroj_filmova();` ove funkcije su specifične za strukturu te sve što one rade kada su pozvane je `Broj_Filmova ++;` ili `Broj_Filmova--;`

Izabirom opcije [5] unutar sporednog menija se vraćamo na glavni Administratorski meni.

Izabirom opcije [4] unutar glavnog menija otvaramo sporedni meni kontrole liste korisnika.

Ova opcija pokreće dvije funkcije u petlji, to jeste pozivom funkcije `EditujKorisnike ();` Prije svega se pokreće funkcija ispisa korisnika `PokaziKorisnike(ListaKorisnika);` koja će biti pokretana zajedno sa sporednim izbornim menijem u petlji sve dok korisnik ne izabere opciju [5].

Sporedni meni editovanja korisnika izgleda ovako:

Editovanje Korisnika:

```
|-----|
|[1]-Upozori korisnika      |
|[2]-Resetuj upozorenja    |
|[3]-Obrisi korisnika       |
|[4]-Napravi admin-korisnika|
|[5]-<----NAZAD<-----    |
|-----|
```

Izabirom opcije [1] pokreće se kod koji traži redni broj korisnika i poruku koja će biti ispisana korisniku sljedeći put kada se uloguje na svoj račun.

Izabirom opcije [2] pokreće se kod koji će pokrenuti dvije funkcije koje su specifične za strukturu Korisnici. Te dvije funkcije rade ono što im ime predlaže, a one su:

`ListaKorisnika[rednibroj].ocistiUpozorenja();`
`ListaKorisnika[rednibroj].brisanjePorukeUpoz();`

Izabirom opcije [3] Poziva se funkcija `BrisanjeKorisnika(ListaKorisnika);`. Na display se ispisuje strogo upozorenje koje bi trebalo spriječiti ikoga da slučajno obriše korisnika, te se od korisnika traži potvrda da zaista to žele.

Nakon čitanja i potvrde funkcija od korisnika se traži unos rednog broja koji se nalazi pored svakog korisnika na listi iznad sporednog menija. Po uspješnom brisanju korisnika na display se ispisuje poruka “Korisnik uspješno brisan”.

Izabirom opcije [4] pokreće se kod koji će specifičnog korisnika učiniti administratorom.

Izabirom opcije [5] dolazi do prekida petlje te korisnik biva vraćen na glavni meni za administratorske opcije.

Zadnja opcija na glavnom administratorskom meniju je opcija EXIT koja će korisnika pitati da li žele spremi promjene, ukoliko prihvati pokreću se 4 funkcije o kojima ćemo pričati i završnom dijelu objašnjavanja koda. Ukoliko korisnik odbija spremanje, bit će pokrenuta samo funkcija gašenja.

Funkcije spremanja podataka i gašenja

Funkcije koje se pokreću pri gašenju uz prihvatanje spremanja podataka su

```
void SpremanjeKorisnika(vector<Korisnici>& ListaKorisnika, Korisnici TrenKorisnik,
    int RBrTrenKorisnik);
void SpremanjeListeKorisnika(const vector<Korisnici>& ListaKorisnika,
    const string& imefajla);
void SpremanjeListeFilmova(const vector<Filmovi>& ListaFilmova, const string& imefajla);
void Sekvenca_Gašenja();
```

Funkcija spremanja korisnika je zadužena za upisivanje Trenutnog korisnika na mjesto u glavnoj listi korisnika s kojeg je originalno na početku kopiran, pozivom ove funkcije potpuna lista korisnika biva upotpunjena i ažuriranja sa najnovijim promjenama i samim tim spremna za glavno spremnje u CSV fajl.

Funkcija SpremanjeListeKorisnika je zadužena za otvaranje fajla Korisnici.csv i upisa informacija u fajl u formatu “comma separated values” sa delimiterom “\$”, nakon ispisa svake zasebne varijable iz vektora<Korisnici> Korisnik pokreće se `file<<endl;` što informacije o sljedećim varijablama iz vektora upisuje u novu liniju i tako za svaku varijablu.

Funkcija SpremanjeListeFilmova radi apsolutno isto što i funkcija spremanje korisnika sa razlikom što se informacije spremaju u fajl Filmovi.csv.

Posljednja funkcija o kojoj ćemo govoriti u ovoj dokumentaciji je `Sekvenca_Gašenja();` Ova funkcija daje vizuelnu reprezentaciju gašenja programa (iako nepotrebna, dodaje stepen personalnosti samom programu). Nakon ispisa prethodno spomenutih grafika funkcija poziva `exit(0);` što u biti gasi program i vraća 0 povratnu vrijednost.

Završne riječi

Uprkos poteškoćama izrada ovog programa je bila zabavna i dozvolila nam je da testiramo naše vještine timskog rada i programiranja programa u timu. Ovaj program nam je također omogućio da zajedno radimo na traženju grešaka u kodu koje sam compiler ne vidi jer nisu sintaksne greške. Za pronalaženje dodatnih grešaka u kodu smo tražili pomoć grupe “testnih korisnika” iz naše grupe prijatelja sa Discord servera.

Neke od grešaka koje su ostale u kodu je manjak ograničenja tipa varijable za upis, npr. Kada program očekuje integer ukoliko upišemo slovo dolazi do prestanka rada programa.

Ovo se može popraviti korištenjem modificirane `cin.ignore()`; komande što ćemo uraditi ukoliko stignemo obzirom na ostale obaveze priprema ispita.