

## Programski jezici 1

### Projektni zadatak – Simulirana platforma za učenje

U programskom jeziku C++ realizovati simulaciju platforme za učenje u vidu konzolne aplikacije.

**(50%)** Korisnik sistema ima pristupne podatke za sistem (korisničko ime i lozinka). Korisnik može da bude upisan na više kurseva. Na svakom kursu svaki upisani korisnik može biti ili student ili predavač na tom kursu. Student može da se prijavi na kurs i da ga pohađa. Prijava mora biti odobrena od strane predavača na kursu. Svaki korisnik može da bude prijatelj sa drugim korisnicima. Korisnik može da drugom korisniku pošalje zahtjev za prijateljstvo, koji drugi korisnik može da prihvati ili da odbije. Ako je zahtjev prihvaćen, smatra se da se ta dva korisnika poznaju. Korisnik može da razmjenjuje poruke sa korisnicima sa kojima je prijatelj u sistemu. Takođe, svaki student može da razmjenjuje poruke sa predavačima koji su zaduženi na kursevima koje taj student pohađa. Poruke se čuvaju u inboks u korisnika. Na primjer, moguće je da se funkcionalnost za razmjenu poruka realizuje tako da se može demonstrirati sljedećom sekvencom akcija: 1) aplikacija se pokreće, 2) korisnik A se prijavljuje, 3) korisnik A šalje poruku korisniku B i odjavljuje se sa sistema i 4) korisnik B se prijavljuje i u inboks u vidi poruku od korisnika A.

**(20%)** Administrator sistema može dodati, modifikovati ili ukloniti kurseve i korisnike. Omogućena je i perzistencija podataka u sistemu, tako da je stanje sistema očuvano ako se aplikacija zatvori. Nakon otvaranja aplikacije, serijalizovani podaci se učitaju i podaci u sistemu budu isti kao i prije zatvaranja.

**(10%)** Kroz aplikaciju je moguće porediti dva kursa: pronalaženje unije, presjeka i razlike između skupova studenata koji pohađaju kurs, odnosno studenata koji su položili ispit. Takođe, može se vršiti filtriranje poruka u inboks u prema zadatom stringu za pretragu.

**(10%)** Da bi se student mogao prijaviti na kurs potrebno je da je ispunio preduslove tog kursa koji zavise od odslušanih i od položenih kurseva od strane tog studenta. Rješenje treba biti konstruisano tako da programer može da s lakoćom proširi sistem novim tipovima preduslova. Neki primjeri tipova preduslova su: da je student odslušao neki kurs, da je student položio specificiran broj kurseva sa nekim zajedničkim svojstvom, da je student položio određene kurseve s određenom minimalnom ocjenom za svaki od tih kurseva i da je student položio bar po jedan kurs iz nekoliko različitih oblasti.

**(10%)** Moguće je ispitati graf poznanstva između svih osoba u sistemu i utvrditi najkraću putanju u grafu poznanstva između proizvoljne dvije osobe, kao i prosječnu najkraću putanju (realan broj) između osoba koje zadovoljavaju neki uslov odabran od strane korisnika (npr. prva osoba pohađa kurs A, a druga osoba pohađa kurs B).

Program se mora moći kompajlirati, izvršiti i testirati. Pravilno izvršiti modularizaciju. Dozvoljeno je korištenje samo standardne biblioteke i standardne biblioteke šablona (STL). Pridržavati se principa objektno-orijentisanog programiranja, principa pisanja čistog programskog koda, principa ponovne upotrebe programskog koda, SOLID principa i konvencija za programski jezik C++. Izdvojiti u statičku biblioteku korištene komponente koje su zasebno izvan domena problema (npr. osnovna klasa za graf). Formirati testni skup podataka kojim se može inicijalizovati sistem u svrhu demonstracije funkcionalnosti.