

# Projektni zadatak

**Predmet:** Objektno - orijentisano programiranje 1

**Generacija:** 2018/2019.

**Smer:** Softversko i informaciono inženjerstvo

**Univerzitet Singidunum, Centar Novi Sad**

## Specifikacija zadatka

### Tehnička specifikacija

Projekat kreirati upotrebom Objektno-orijentisanih principa u programskom jeziku Java verzije 11.0.2. Za grafički korisnički interfejs koristiti biblioteku JavaFX verzije 11. Projekat kreirati u okviru okruženja Eclipse Enterprise edicije verzije 2018-12 R. Podatke čuvati u tekstualnim datotekama.

### Dodatne napomene

Projekat predati kao Eclipse projekat (eksportovan). **Projekat prilikom kreiranja imenovati u formatu oop\_indeks\_ime\_prezime\_1 (a za drugi deo projekta oop\_indeks\_ime\_prezime\_2).** Eksportovani projekat zapakovati u arhivu sa istim nazivom i okačiti na deljeni disk (Dropbox, Google Drive, ... i sl.) ili repozitorijum (Gitlab ili Github, kao *javni projekat*). Na stranici predmeta će biti objavljen link ka google formi u kojoj je potrebno prijaviti projekat unošenjem podataka u formu, gde će se uneti i linkovi ka projektima jedan i dva. **Projekat jedan i projekat dva se predaju odvojeno.**

**Prvi termin za odbranu projekata je XV nedelja nastave,** a nakon toga se može braniti u svakom narednom ispitnom roku.

### Tekst zadatka

**Kreirati aplikaciju sa grafičkim korisničkim interfejsom koja će predstavljati elektronski indeks (E-indeks) studenta. Aplikacija manipuliše nad sledećim tipovima objekata:**

- **Korisnik** koji je opisan imenom, prezimenom, korisničkim imenom, lozinkom, podatkom da li je obrisani ili ne.
- **Student** je specijalni tip korisnika koji je opisan još i email-om, brojevima telefona i indeksom.
- **Nastavnik** je specijalni tip korisnika koji je opisan još i email-om i zvanjem.
- **Administrator** je specijalni tip korisnika, koji ne dodaje dodatna polja, ali ima specifične operacije.
- **Predmet** ima id, naziv, silabus (opis predmeta), godinu (u kojoj se izvodi, npr. 2019.), godina studija (kojoj godini se izvodi, npr. prvoj), fond predavanja (broj časova predavanja), fond vežbi (broj časova vežbi).
- **Ispit** je opisan indeksom (na osnovu toga se dobija student koji ga polaže), bodovima, ocenom, datumom polaganja, podatkom da li je poništen i predmetom (za koji ocena važi).

➤ **Indeks** je opisan brojem (jedinstven), univerzitetom, fakultetom, smerom, godinom upisa, godinom studija (na kojoj je trenutno godini studija student, npr. prvoj), ispitima, i studentom čiji je taj indeks.

➤ **Zvanje** je enumeracija koja sadrži sledeće konstante: saradnik, asistent, docent, vanredni profesor, rednovan profesor.

➤ **IspitniRok** je enumeracija koja sadrži sledeće konstante: januarski, februarski, martovski, aprilski, majski, junski, julski, septembarski, oktobarski, novembarski, decembarski.

#### **Obezbediti sledeće funkcionalnosti:**

➤ **Prijavljivanje na sistem** vrši odmah pri pokretanju aplikacije. Bilo koji korisnik unošenjem parametara (korisničko ime i lozinka) vrši prijavu, ukoliko korisnik unese ispravne podatke, u zavisnosti od toga koji je konkretan tip korisnika mu se nude odgovarajuće funkcionalnosti. Ukoliko korisnik ne unese ispravne podatke, tražiti ponovni unos, sve dok korisnik ne odabere izlazak iz aplikacije ili unese ispravne podatke. Korisniku koji se prijavi nuditi samo one funkcionalnosti koje on može da radi (date u nastavku teksta). Prijavu na sistem ne može raditi logički obrisani korisnik, čak iako se poklapaju podaci uneti za prijavu.

➤ **Registrowanje korisnika** vrši Administrator. Administrator bira kog konkretnog korisnika unosi, i u zavisnosti od konkretnog tipa se popunjavaju svi podaci koji opisuju tog korisnika. Administrator može dodavati i druge administratore. Inicijalnog administratora obezbeđuje student prilikom izrade projekta. Sve registrovane korisnike čuvati u datoteci.

➤ **Brisanje korisnika** vrši Administrator. Prvo se odabere odgovarajući korisnik (pri odabiru se može vršiti pretraga korisnika), a zatim se odabere opcija obriši. Brisanje zahteva dodatnu potvrdu Administratora. Administrator ne može obrisati samog sebe. Brisanje korisnika je logičko (ne brišu se podaci iz datoteke već se menja status na obrisani).

➤ **Promena podataka o korisniku** se takođe vrši od strane Administratora. On može da promeni sve ili samo neke podatke bilo kog korisnika, ukoliko se menjaju podaci Nastavnika ili Studenta omogućiti i izmenu njihovih specifičnih polja. Ukoliko se radi delimična promena, onda se za podaci koji nisu promenjeni uzimaju prethodne vrednosti. Promena podataka može biti i aktiviranje obrisanih korisnika.

➤ **Dodavanje predmeta** vrši Administrator tako što unese sve parametre koji opisuju Predmet. Predmet mora imati jedinstveni id. Id predmeta se formira tako što se na dva nasumično kreirana karaktera (a-Z) dodaju poslednje dve cifre godine kada se predmet izvodi i potom se doda skraćenica od smera napisana velikim slovima (npr. Ako je na smeru Softversko i informaciono inženjerstvo i sluša se u 2019. godini može glasiti: Mf19SIII). Voditi računa da i ovaj id mora biti jedinstven (ukoliko spoj nasumično generisanih karaktera, godine i skraćenice smera proizvede već postojeći id, ponovo nasumično generisati karaktere).

➤ **Prijava ispita** se vrši od strane Studenta, tako što odabere predmet za koji prijavljuje ispit i bira ispitni rok u kojem želi da ga prijavi. Prijaviti može samo ispit za predmet koji se izvodi u godini studija u kojoj je i student trenutno (npr. prvoj) i za smer koji student pohađa. Ova prijava se evidentira u Studentovom Indeksu kao novi Ispit. Voditi računa da bodovi, ocena, datum mogu biti prazni (njih popunjava Nastavnik kada upisuje ocenu Studentu, videti zahtev Upis ocena). Student može najviše jednom da prijavi ispit iz odabranog kursa u odabranom roku.

➤ **Poništavanje ocene** vrši Student tako što odabere položeni ispit (ispit koji ima bodove, ocenu i datum popunjene, pored ostalih polja) i odabere opciju poništavanja. Od Studenta zahtevati da potvrdi ili odbije poništavanje ocene. Student ne može poništiti ispit koji je već poništio. Poništavanje je takođe logičko brisanje, dakle ispit se stavlja status na poništen.

➤ **Upis ocena** vrši Nastavnik. Upisuje se ocena tako što Nastavnik pretraži spram ispitnog roka ili naziva predmeta ispitne prijave (Ispit koji je formiran od strane studenta), zatim može dodatno pretražiti taj rezultat spram podataka o Studentu (indeks, ime i prezime) i naći Ispit kojem će dopuniti bodove i ocenu (ocena mora da prati opsege bodova npr. 0-50 je ocena 5, 51-60 je ocena 6, itd.), a datum će biti popunjen trenutnim datumom kada Nastavnik unosi ocenu. Samo nastavnik koji je docent, vanredni profesor ili redovan profesor može upisati ocenu (dakle saradnik ili asistent ne mogu upisati ocenu). Voditi računa da je upis ocena zapravo izmena objekta ispit. Voditi računa da se ocena ne može upisati studentu koji je logički obrisao.

➤ **Pretraga Studenata** se vrši od strane Nastavnika, kako bi video sve podatke o Studentu. Pretragu je moguće vršiti nad sledećim poljima Studenta: ime, prezime, broj indeksa, predmet koji sluša. Pretragu može vršiti i Administrator.

➤ **Pregled indeksa** vrši Student kako bi video sve podatke o sebi i svom elektronskom indeksu. Iz ovog pregleda Student može vršiti sve ostale opcije koje su prethodno za njega navedene. U pregledu omogućiti jasnu razliku između studentovih položenih ispita i poništenih.

Prilikom predaje projekata potrebno je obezbediti inicijalne podatke koji će omogućiti testiranje prethodno navedenih funkcionalnosti sistema. Ovi podaci se nalaze u tekstualnim datotekama.

#### **Za projekat I neophodno je uraditi sledeće zahteve:**

- Implementirati sve tipove objekata;
- Kreirati tekstualne datoteke sa podacima;
- Početni grafički korisnički interfejs koji će omogućiti sledeće funkcionalnosti:
  - Registrovanje korisnika;
  - Prijava na sistem;
  - Brisanje korisnika;
  - Promena podataka o korisniku;
  - Dodavanje predmeta;

#### **Za projekat II uraditi i ostale zahteve (funkcionalnosti).**

*Student bira način za realizaciju grafičkog korisničkog interfejsa, tačnije nemaju već predefinisani widgeti koje mora kreirati, ali je bitno da zna za koju namenu se koristi koji tip widget-a (npr. ako postoji neki spisak koristiti prikaz kao liste ili kao tabele). Potrebno je imati na umu da je potrebno koristiti dijaloge i druge widgete (odnosno ne stavljati sve u jedan jedini widget). Prilikom ocenjivanja će se uzeti u obzir i način na koji je student realizovao korisnički interfejs.*