

# Objektno orijentisano programiranje 1

## Specifikacija projektnog zadatka 2023/2024

*Poslednja izmena: 29.03.2024.*

Upotrebom tehnologija i tehnika primenjivanih na predavanjima i vežbama isprojektovati programsku podršku za informacioni sistem hotela prema sledećoj specifikaciji.

**Rok za izradu projekta:** sreda, 03.07.2024.

### 1. Opis projekta

Potrebno je implementirati informacioni sistem za rad hotela. Informacioni sistem mora biti implementiran u Java programskom jeziku, a može biti realizovan u dve verzije: sa konzolnim ili grafičkim korisničkim okruženjem. Svaki student treba da osmisli i implementira arhitekturu projekta u skladu sa principima OOP-a. Zadatak svakog studenta je da, uz konsultaciju sa predmetnim asistentom, razvije jedinstvenu implementaciju informacionog sistema prema datoj specifikaciji.

### 2. Specifikacija zahteva

U ovom poglavlju se nalazi opis specifikacije zahteva informacionog sistema, uključujući opise tipova korisnika koji mogu da ga koriste, opise funkcionalnosti sistema za svaki tip korisnika, kao i opis tehničkih ograničenja.

#### 2.1. Korisnici sistema

Sistem vodi evidenciju o četiri tipa korisnika: Administrator, Recepcioner/Agent za usluge gostiju, Sobarica, Gost. Prijava na sistem je omogućena svim tipovima korisnika pri čemu svako od njih ima pristup određenim funkcionalnostima sistema.

- **Administrator** ima pristup svim entitetima i može da izvrši sve operacije nad njima (CRUD). Dodatno, može da dodaje zaposlene hotela u sistem (recepcionere i sobarice).
- **Recepcioner / Agent za usluge gostiju** radi sav posao vezan za pružanje usluga gostima, uključujući potvrđivanje ili odbijanje rezervacija koje gosti mogu da naprave kroz svoj deo sistema, imaju uvid u stanje soba u hotelu i kad su one slobodne ili zauzete; oni vrše *check-in* i *check-out* gostiju na recepciji i dodavanje gostiju u sistem ukoliko oni već nisu u njemu.
- **Sobarica** je zaposlena kao deo osoblja i ima uvid u sobe koje su joj dodeljene za sređivanje. Kada recepcioner odradi *check-out* gosta iz sobe, ta soba se automatski dodeljuje sobarici koja je tog dana imala najmanje raspoređenih soba za sređivanje. Tek kada sobarica sredi određenu sobu i to unese u sistem, soba postaje dostupna za *check-in* gostu. Cilj svake sobarice jeste da na kraju radnog dana nema nijednu dodeljenu sobu u svom pogledu na aplikaciju.
- **Gost** je neko ko je već odsedao u hotelu i koji je registrovan od strane recepcionera na osnovu *email*-a i/ili broja pasoša. Gost može izvršiti rezervaciju za određeni tip sobe i za određeni broj ljudi (nema uvid u pojedinačne sobe, ali ima uvid u to koji tip sobe je

slobodan za izabrane datume). Tip sobe može biti jednokrevetna (1), dvokrevetna sa jednim ležajem (2), dvokrevetna sa dva ležaja (1+1), trokrevetna (2+1) i slično.

## 2.2. Funkcionisanje informacionog sistema

- Samostalna registracija korisnika nije moguća, ali administratori imaju mogućnost da dodaju nove zaposlene u sistem (sobarice i recepcionere), a recepcioner može da dodaje goste (pri prvom *check-in*-u se gost registruje od strane recepcionera, gde mu *e-mail* postaje korisničko ime, a broj pasoša lozinka).
- O svim korisnicima se evidentiraju sledeći podaci: ime, prezime, pol, datum rođenja, telefon, adresa, korisničko ime i lozinka.
- Zaposleni imaju dodatne podatke o nivou stručne spreme, stažu i plati. Plata se računa na osnovu koeficijenta vezanog za stručnu spremu, osnove i godina staža.
- Recepcioner pravi nalog za gosta koji se prvi put prijavljuje u hotel (*check-in*). Korisničko ime je *e-mail*, a broj pasoša je lozinka.
- Gost može da napravi zahtev za rezervaciju određenog tipa sobe za datume koje sam odabere. Rezervacija može imati jedan od sledećih statusa: "NA ČEKANJU", "POTVRĐENA", "ODBIJENA" i "OTKAZANA".
- Inicijalni status rezervacije će biti „NA ČEKANJU“ dok god neko od agenata za usluge gostiju (recepcionera) ne potvrdi ili ne odbije rezervaciju (rezervacija dobija status "POTVRĐENA" ili "ODBIJENA").
- Prilikom potvrđivanja rezervacije od strane recepcionera, aplikacija mora da proveri da li postoji bar jedna soba tog tipa koja je slobodna u tom opsegu datuma. Ukoliko nema slobodnih soba, recepcioneru prikazati poruku greške. Rezervacije koje nisu prihvaćene do datuma njihovog početka će se automatski odbiti (rezervacije koje su istekle).
- Kada se gost pojavi u hotelu, recepcioner ulazi u *check-in* proces. On bira slobodnu sobu traženog tipa, dodeljuje sobu rezervaciji i status sobe menja na „ZAUZETO“, čime se formalno završava *check-in* proces.
- Kada gost ide iz hotela i vrši *check-out*, recepcioner menja status sobe u „SPREMANJE“ i sistem tu sobu automatski dodeljuje sobarici koja ima najmanje posla tog dana.
- Nakon čišćenja sobe, spremačica potvrđuje da je završila svoj posao. Status sobe se setuje na „SLOBODNA“.
- Recepcioner ne može izvršiti novi *check-in* za potvrđenu rezervaciju dok god status sobe ne bude postavljen na "SLOBODNA". On može izvršiti *check-in* samo za slobodne sobe, a ne i za zauzete ili sobe koje nisu spremne za *check-in*.
- Recepcioner vidi sve podatke o dnevnim dolascima i odlascima, kao i o dnevnoj zauzetosti hotela.
- Administrator ima uvid u podatke o svim zaposlenim kao i prihode i rashode za određeni period.
- Gost može da vidi svoje rezervacije sa statusom potvrde. Gost može i da otkáže svoju rezervaciju, ali u tom slučaju nema prava na refundaciju novca (status "OTKAZANA"). Ukoliko recepcioner odbije pokušaj rezervacije od strane gosta (status "ODBIJENA"), gost dobija novac nazad i hotel nema prihod od tih rezervacija.
- Prilikom kreiranja rezervacije, gost može odabrati i dodatne usluge (doručak, ručak, večera, ...) i one će mu biti naplaćene po trenutno važećem cenovniku hotela, po svakom noćenju. Dodatne usluge na rezervaciju može dodati i recepcioner prilikom *check-in* procesa, ukoliko to gost usmeno zatraži.
- Cenovnik hotela definiše administrator. Cena se definiše za svaki tip sobe, kao i za svaku dodatnu uslugu, i ima svoj datum početka i kraja važenja. Prilikom računanja

ukupne cene boravka u hotelu, gostu se cena obračunava tako što se traži cena svake pojedinačne stavke za taj datum noćenja i sabira se na konačnu cenu, po važećem cenovniku. Tako se može desiti da cena za recimo prva dva noćenja bude manja nego za druga dva (u slučaju četiri noćenja), ako se desi da cena raste između datuma, što će biti slučaj sa datumima koji su na početku ili kraju sezone, kad se cene i inače menjaju.

- Cena se računa u trenutku kreiranja rezervacije. Kada gost rezerviše smeštaj, u tom momentu mu se obračunava cena po trenutno važećem cenovniku i ta cena je konačna i upisuje se u samu rezervaciju. Ako se cenovnik hotela promeni nakon rezervacije, to neće uticati na već postojeće rezervacije.

### 2.3. Tehnička specifikacija

Potrebno je razviti konzolni ili grafički korisnički interfejs upotrebom Java Swing biblioteke koji će omogućiti prijavu na sistem kao i pristup formama za pregled, dodavanje, izmenu i brisanje podataka o sledećim entitetima:

- Zaposleni
- Gosti
- Rezervacije
- Tip soba
- Sobe
- Dodatne usluge hotela (doručak, ručak, večera)
- Cenovnik

Moguće je proširiti postojeće klase novim atributima, ili napraviti dodatne klase.

## 3. Ocenjivanje projektnog zadatka

U ovom poglavlju su definisane dodatne funkcionalnosti koje zavise od ocene koju želite da dobijete na projektu.

### 3.1. Ocene $\leq 7$

Aplikacija je izrađena u **konzolnom interfejsu** i mora da podrži sve navedeno u specifikaciji zahteva, osim analiza i prikaza *chart*-ova i jediničnih testova. Za ocenu 6 se za čuvanje podataka aplikacije može upotrebiti serijalizacija objekata (Vežbe: Tokovi podataka...).

Potrebno je uraditi izveštaje:

- prihoda i rashoda, koliko je soba svaka sobarica soba spremila za izabrani opseg datuma, kao i koliko rezervacija je potvrđeno za odabrani opseg datuma
- obrađenih zahteva rezervaciju, koliko je rezervacija odbijeno a koliko otkazano za izabrani opseg datuma za upis na kurs, za izabrani opseg datuma.
- za prikaz soba, što podrazumeva prikaz podataka o sobi i njenom tipu, ukupan broj noćenja u toj sobi za izabrani opseg datuma, kao i ukupne prihode iz svake sobe na osnovu izabranog opsega datuma

### 3.2. Ocena 8

Sve kao iz 3.1, ali uz upotrebu GUI-ja (Swing).

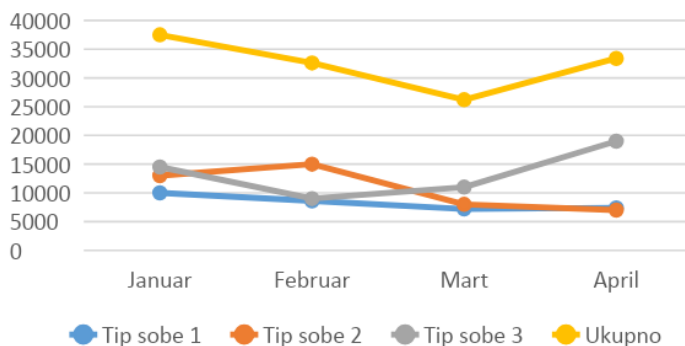
### 3.3. Ocene 9 i 10

Sve kao iz 3.2, uz dodatne funkcionalnosti koje slede.

Aplikacija je urađena u GUI-u (Swing), podržava navedene funkcionalnosti (iz 3.2), sa još dodatnim. Dodatne funkcionalnosti:

- Gost vidi sve svoje rezervacije i njihove statuse, troškove za svaku rezervaciju (imati na umu da nema troškova na rezervacije koje su odbijene od strane hotela) i ukupan trošak za sve rezervacije
- Prilikom rezervacije, omogućiti gostu da unese dodatne kriterijume pretrage mogućih soba (da li soba poseduje klima uređaj, balkon, tv, pušačka/nepušačka soba), pa izlistati samo tipove soba koji imaju konkretne sobe koje zadovoljavaju te kriterijume.
- Recepcioner ima mogućnost da filtrira sve rezervacije po tipu sobe, konkretnoj sobi, opsegu cene koju gost plaća (ako želi prvo da odobri sve najskuplje rezervacije na koje hotel ima najveću zaradu), kao i po dodatnim uslugama koje su navedene u rezervaciji.
- prikaz *chart*-ova
  - o Prikaz prihoda za prethodnih 12 meseci po tipu sobe, kao i ukupan prihod. Iscrtati grafički prikaz putem neke jednostavne biblioteke (npr. <https://knowm.org/open-source/xchart/>) koja se oslanja na Swing.)

Prihodi po tipu sobe



- o Prikaz opterećenja sobarica za prethodnih 30 dana, kao i prikaz statusa svih rezervacija kreiranih u prethodnih 30 dana

Opterećenje sobarica u prethodnih 30 dana



Status rezervacija u prethodnih 30 dana



- Implementirati jedinične testove za sve menadžere (nije potrebno za čiste entitete i GUI)

#### 4. Napomene

- Svaki student samostalno izrađuje svoje rešenje
- Rešenje razvijati u Javi 17, a grafički korisnički interfejs upotrebom Swing biblioteke
- Maksimalan broj bodova koji je moguće osvojiti je 50, u zavisnosti od rešenja i rokova
- Postoje dva roka za predaju – jun i septembar

- Broj bodova zavisi od demonstracije rešenja - u kom roku je predato rešenje, testiranja funkcionalnih zahteva i kvaliteta rešenja
- Za rešenje sa konzolnim korisničkim interfejsom moguće je dobiti maksimalnu ocenu 7.
- Za rešenje bez jediničnog testiranja moguće je dobiti maksimalnu ocenu 8.
- Za ocene 9 i 10 neophodno je implementirati rešenje sa grafičkim korisničkim interfejsom i jediničnim testiranjem.
- Čuvanje podataka vršiti u tekstualnom (ljudski čitljivom) formatu (npr. ".csv").
- Za rešenje koje koristi serijalizaciju za čuvanje podataka moguće je dobiti maksimalno ocenu 6.
- Za podatke koji imaju predefinisani skup mogućih vrednosti potrebno je implementirati odgovarajuće enumeracije.
- Ukoliko postoje dva (ili više) rešenja koja su slična, oba (ili više) studenata dobijaju 0 poena i zabranu predavanja rešenja za ovaj projekat.
- Datumi predaje i odbrane su podložni promeni. Obaveštenja o projektu i predaja rešenja vršiće se putem <https://canvas.ftn.uns.ac.rs/> platforme ili E-nastave.
- Dozvoljeno je korišćenje dodatnih biblioteka i tehnologija koje nisu sastavni deo kursa uz prethodnu konsultaciju sa asistentom.
- Za sve ostale nejasnoće potrebno je konsultovati se sa predmetnim asistentom.

## Pitanja

1. Kako da importujem klase iz java.awt paketa i zašto ne radi autocomplete?

Ovo je greška u Eclipse editoru. Da biste rešili, možete ručno pisati import linije, a možete otići u Window > Preferences > Java > Appearance > Type Filters > uncheck java.awt\*.

2. Da li smem da koristim WindowBuilder?

Da.