Objektno-orijentisano programiranje, Jun 2

Matematički fakultet, školska godina 2021/2022

Napomena: Na Desktop-u napraviti direktorijum pod imenom oop_Asistent_Prezime_Ime_Indeks (npr. oop_SS_Peric_Pera_mi12082). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. U napravljenom projektu, paket takođe nazvati tako.

Kod ne sme imati sintaksnih grešaka niti izbacivanje NullPointerException-a.

Vreme za rad: 3 sata

Inicijali: MNV smerovi - SS, RL smerovi - NC

U tekstu je dat opis klasa, njihovih atributa i metoda. **Dozvoljeno** je dodati nove atribute, klase, metode, enume, interfejse u slučaju da Vam olakšavaju implementaciju, i/ili smatrate da Vam poboljšavaju kvalitet koda i slično.

Da bi se uspešno položio ispit potrebno je osvojiti barem 35 poena.

Zadatak je napraviti aplikaciju koja korisniku pronalazi odgovarajući smeštaj u datom terminu.

- 1. [4p] Napraviti klasu Gost koja ima polja imePrezime (String), budzet (int) i polje premium (boolean) koje označava da li je gost premium korisnik ili ne. Obezbediti konstruktor koji prihvata vrednosti za sva navedena polja i postavlja ih, potrebne get metode kao i toString metod koji ispisuje informacije o gostu kao na slici 2.
- 2. [6p] Napraviti klasu Datum koja sadrži polja dan (int), mesec (int). Obezbediti konstruktor koji prihvata vrednosti za sva polja, kao i konstruktor koji prima String (u ocekivanom formatu "dan/mesec"). Implementirati i metod toString koji ispisuje datum u prethodno navedenom formatu. Omogućiti poređenje instanci klase Datum tako što se prvo porede meseci, a ako je isti mesec za oba datuma, porede se dani.
- 3. [6p] Napraviti klasu Termin koja sadrži dva datuma datumOd (Datum) i datumDo (Datum). Obezbediti konstruktor koji prihvata vrednosti za sva polja i koji vrši proveru validnosti termina (da li je datum "od" manji od datuma "do") i u slučaju da termin nije validan izbacuje IllegalArgumentException izuzetak sa odgovarajućom porukom o grešci. Implementirati metod toString koji ispisuje termin u formatu "od do", zatim i metod boolean preklapaSeSa(Termin t), koji proverava da li se termin preklapa sa datim terminom. Npr. termin 4/1 9/1 se preklapa sa 6/1 11/1 (i obrnuto). Obezbediti poređenje termina (porediti "od" datume dva termina).
- 4. [6p] Napraviti apstraktnu klasu Soba koja predstavlja sobu u hotelu. Soba treba da sadrži polja brojSobe (int), strukturaSobe (int, broj kreveta), cena (int), i zauzece (Map<Termin, Gost>) koja čuva termine u kojima je soba zauzeta od strane nekog gosta. Obezbediti konstruktor koji prihvata broj sobe, strukturu i cenu, kao i potrebne get metode. Definisati apstraktni metod boolean smesti(Termin, Gost), i metod toString koji ispisuje sobu navođenjem informacija o sobi u prvom redu u formatu "brojSobe, strukturaSobe, cena", a zatim i sve zauzete termine (rastuće) u formatu " -> termin, gost"(slika 2).
- 5. [7p] Napraviti klasu StandardnaSoba koja nasleđuje klasu Soba. Implementirati nasleđeni apstraktni metod koji proverava da li postoji mogućnost da gost bude smešten u sobu u željenom terminu i ako postoji, smešta ga i vraća true, u suprotnom vraća false. Uslovi su sledeći:
 - soba ne sme biti zauzeta u tom terminu
 - gost mora sebi da priušti tu sobu (njegov budžet treba da bude veći od cene sobe)

Predefinisati nasleđeni metod toString tako da se ispred broja sobe dopisuje tip sobe (slika 3).

- 6. [7p] Napraviti klasu PremiumSoba koja nasleđuje klasu Soba. Implementirati nasleđeni apstraktni metod koji proverava da li postoji mogućnost da gost bude smešten u sobu u željenom terminu i ako postoji, smešta ga i vraća true, u suprotnom vraća false. Uslovi su sledeći:
 - soba ne sme biti zauzeta u tom terminu
 - gost mora sebi da priušti sobu (njegov budzet treba da bude veći od cene sobe, uzimajući u obzir eventualno umanjene za popust)
 - premium korisnicima omogućiti nasumični popust od 15%

Predefinisati nasleđeni metod toString tako da se ispred broja sobe dopisuje tip sobe (slika 3).

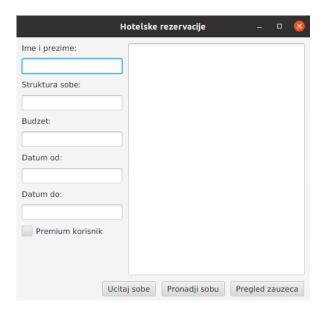
- 7. [9p] Napraviti klasu Hotel koja nasleđuje klasu Application biblioteke javafx i izgleda kao na slikama. Klasa sadrži polje sobe (List<Soba>) koje čuva sve postojeće sobe u hotelu.
- 8. Klikom na dugme:
 - [7p] Ucitaj sobe učitavaju se informacije o sobama u hotelu iz datoteke "sobe.txt", gde svaki red predstavlja jednu sobu sa njenim brojem, strukturom, cenom i naznakom "P" ukoliko se radi o premium sobi. Nakon učitavanja, onemogućiti ponovo klitanje na ovo fugme pomocu metoda setDisable.

- [9p] Pronadji sobu se pokušava sa smeštanjem gosta u neku od soba u zavisnosti od unesenih podataka u TextArea-e sa leve strane (gost moze biti smesten u sobi strukture ≥ od zahtevane). Ako je uspešno pronađen odgovarajući smeštaj, ispisati podatke o rezervaciji kao na slici 2, u suprotnom ispisati poruku da smeštaj nije nađen.
- [4p] Pregled zauzeca vrši se pregled zauzeća svih soba u hotelu (slika 3).

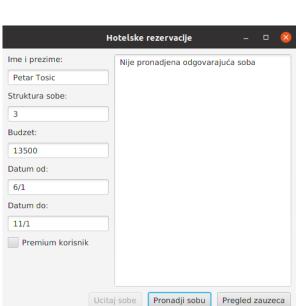
Svi ispisi vrše se u glavni TextArea (u primeru se nalazi u desnom delu).

Pretpostaviti da će sadržaj datoteke biti ispravan. Primer sadržaja datoteke:

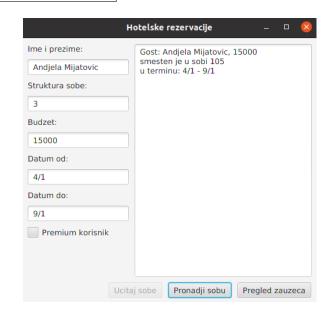
```
313, 2, 21050, P
210, 4, 16600
311, 4, 27800, P
105, 3, 13500
```



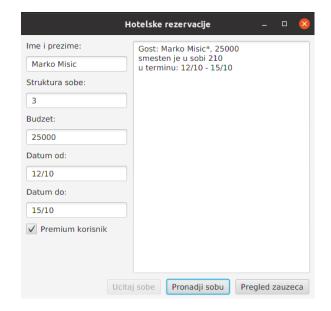
Slika 1: Početni izgled aplikacije



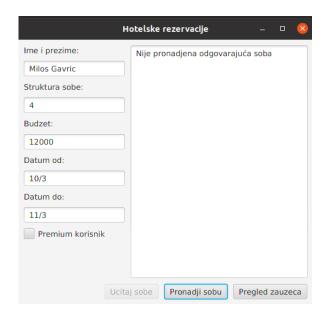
Slika 3: Neuspešna rezervacija usled preklapanja datuma



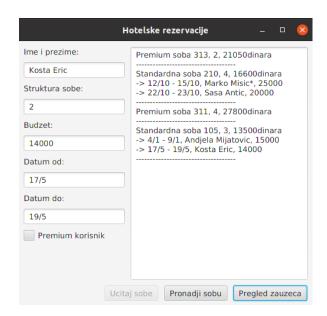
Slika 2: Klik na dugme Uspešna rezervacija



Slika 4: Uspešna rezervacija premijum korisnika



Slika 5: Neuspešna rezervacija usled nedovoljnog budžeta



Slika 6: Pregled zauzeća soba