

Objektno-orijentisano programiranje, ispit
Matematički fakultet, školska godina 2021/2022

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_Asistent_Prezime_Ime_Indeks` (npr. `oop_VK_Peric_Pera_mi12082`). Pokrenuti *Intelij 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: IR: 2I1A | SS: 2I1B | VK: 2I2A, 2I2B

U tekstu je dat opis klasa, njihovih atributa i metoda. **Dozvoljeno** je dodati nove attribute, klase, metode, enume, interfejsu 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 30 poena**.

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

1. (2 poena) Napraviti klasu `Gost` koja sadrži polja `String ime`, `int budzet`, `double ocena` i polje `premium` koje označava da li je gost `premium` korisnik ili ne. Obezbediti:
 - konstruktor koji prihvata vrednosti za sva navedena polja
 - `get` metode za sva polja
 - `toString` metod koji formira i vraća `String` sačinjen od imena i ocene (u formatu po izboru).
2. (4 poena) Napraviti klasu `Datum` koja sadrži polja `int dan`, `int mesec`. Obezbediti:
 - konstruktor koji prihvata dva cela broja
 - konstruktor koji prihvata `String` (u formatu "dan/mesec")
 - `toString` metod koji formira i vraća `String` u prethodno navedenom formatu
 - poređenje instanci klase `Datum` (implementirati interfejs `Comparable<Datum>` i odgovarajući metod). Porediti prvo mesece, a ako je isti mesec za oba datuma, porediti dane.
3. (5 poena) Napraviti klasu `Termin` koja sadrži dva datuma: `Datum od` i `Datum do`. Obezbediti:
 - metod `boolean validanTermin()` koji proverava da li je termin validan, odnosno da li je "od" manje od "do"
 - konstruktor koji prihvata sva polja i koji vrši proveru validnosti termina (u slučaju da termin nije validan, postaviti datume na vrednosti po izboru)
 - `toString` metod koji formira i vraća `String` u formatu "od - do"
 - metod `boolean preklapaSeSa(Termin)`, 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)
 - poređenje termina (porediti "od" datume dva termina).
4. (3 poena) Napraviti apstraktnu klasu `Soba` koja predstavlja sobu u hotelu. Soba treba da sadrži `cenu (int)`, `broj sobe (int)` i `raspored (Map<Termin, Gost>)` koji sadrži termine u kojima je soba zauzeta (zajedno sa gostom koji je zauzima u tom terminu). Obezbediti:
 - konstruktor koji prihvata i postavlja cenu. Omogućiti da je broj svake naredne kreirane sobe za jedan veći od prethodne (odnosno da brojevi soba idu redom)
 - apstraktni metod `boolean smesti(Termin, Gost)`
 - `toString` metod koji formira i vraća `String` sačinjen od broja sobe u formatu "Soba brSobe" (pogledati slike).
5. (5 poena) Napraviti klasu `ObicnaSoba` koja nasleđuje klasu `Soba`. Implementirati nasleđeni apstraktni metod `boolean smesti(Termin, Gost)` 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)
6. (6 poena) Napraviti klasu `PremiumSoba` koja nasleđuje klasu `Soba`. Implementirati nasleđeni apstraktni metod `boolean smesti(Termin, Gost)` 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:
 - premium korisnicima omogućiti nasumični popust (između 10% i 50%)
 - soba ne sme biti zauzeta u tom terminu
 - ako gost nije premium, njegova ocena mora biti veća od ocene sobe

- gost mora sebi da priušti sobu (njegov budzet treba da bude veći od cene sobe umanjene za popust)

7. (5 poena) Napraviti generičku klasu `Hotel<T>` (T treba da predstavlja klasu koja nasleđuje klasu `Soba`). `Hotel` ima naziv (`String`) i sobe (`List<T>`). Obezbediti:

- konstruktor koji prihvata naziv
- metod `void dodajSobu(T soba)` koji dodaje sobu u listu soba
- `Optional<T> smesti(Termin, Gost)`, koji pokušava da smesti gosta u datom terminu u neku sobu
- `toString` metod koji formira i vraća `String` koji sadrži naziv hotela, zatim redom sobe u formatu "soba (ocena) cena\$", tako da posle svake sobe ispiše i sve zauzete termine (rastuće) u formatu " -> termin gost" (Slika 3).

8. (10 poena) Napraviti klasu `Main` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama.

9. (20 poena) Klikom na dugme:

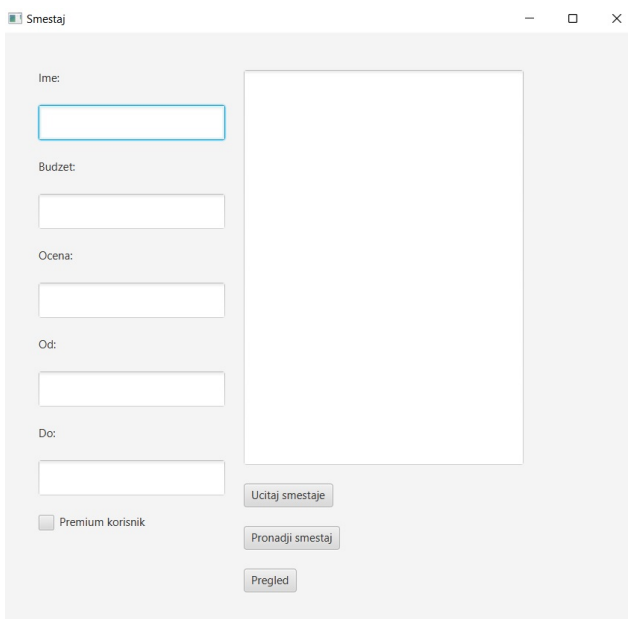
- Učitaj smestaje učitavaju se smeštaji iz datoteke "hoteli.txt" gde svaki red predstavlja hotel i njegove sobe (soba je "cena, ocena" ako hotel ima naznaku "P" kao premium ili samo "cena", pogledati primer). Za svaki hotel započeti brojanje soba ispočetka, uz nasumično odabran početni broj 100, 200, 300, ... 900 npr. 200, 201, 202, ... Hotele čuvati u dve kolekcije (`List<Hotel<PremiumHotel>>` i `List<Hotel<ObicanHotel>>`) ili u jednoj. Sortirati sobe svakog hotela po ceni opadajuće. Nakon učitavanja, omogućiti da ovo dugme postane nevidljivo.
- Pronadji smestaj pokušati pronaći smeštaj u zavisnosti od unesenih podataka u `TextArea`-e sa leve strane. Ako je uspešno pronađen smeštaj, ispisati detalje kao na slici 2, u suprotnom ispisati poruku da smeštaj nije nađen.
- Pregled vrši se pregled svih hotela i njihovih soba - ispisati sve hotele.

Uхватiti sve potencijalne izuzetke (`NumberFormatException`, `IOException`, ...).

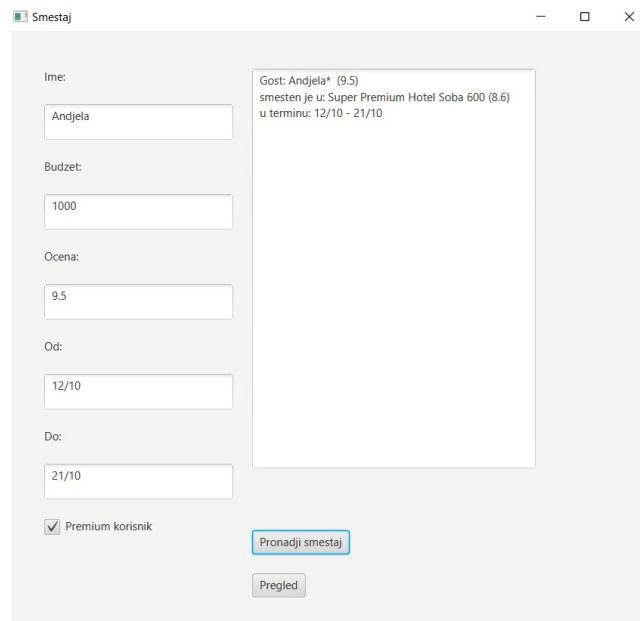
Svi ispisi vrše se u glavni `TextArea` (u primeru se nalazi u desnom delu).

Primer datoteke "hoteli.txt" dat je ispod:

```
Super Premium Hotel, P, 1300, 8.6, 950, 9.0, 1250, 9.4
Premium Hotel, P, 700, 8.1, 1000, 8.9
Najobicniji Hotel, 50, 100
Obican Hotel, 200, 300, 400
```



Slika 1: Ucitavanje.



Slika 2: Trazenje smestaja.

Smestaj

Ime:

Budzet:

Ocena:

Od:

Do:

Premium korisnik

(Super Premium Hotel)
Soba 600 (8.6) 1300\$
-> 12/10 - 21/10 Andjela* (9.5)
-> 22/11 - 25/11 Darko* (8.4)
-> 4/12 - 6/12 Jovan* (9.1)
Soba 602 (9.4) 1250\$
-> 5/12 - 7/12 Jovana* (9.9)
Soba 601 (9.0) 950\$
-> 11/4 - 18/4 Mirko (9.8)

(Premium Hotel)
Soba 901 (8.9) 1000\$
Soba 900 (8.1) 700\$
-> 26/9 - 3/10 Zdravko (9.3)

(Obican Hotel)
Soba 502 400\$
Soba 501 300\$
Soba 500 200\$
-> 9/3 - 19/3 Milica (8.9)

(Najobicniji Hotel)
Soba 901 100\$
Soba 900 50\$

Slika 3: Pregled svih hotela.