

Objektno-orijentisano programiranje, ispit JUN1
Matematički fakultet, školska godina 2023/2024

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_InicijaliAsistenta.ImePrezime.Alas` (npr. `oop_IA.Marko.Markovic_mi22101`). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. Zadatke sačuvati redom u okviru paketa `zadatak1`, `zadatak2` i `zadatak3`.

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

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

Da bi se uspešno položio ispit potrebno je osvojiti **barem 30 poena**.

1. [20p] Kako bi organizacija koncerta bila jednostavnija potrebno je implementirati sledeće:

- Definisati nabrojivi tip `MuzickiZanr` čije su vrednosti: `POP`, `ROK` i `REP`. Definisati statički metod `String prevediNaEngleski(MuzickiZanr zanr)` koji na osnovu žanra vraća odgovarajući `String` - `"POP"`, `"ROCK"` ili `"RAP"`.
- Definisati apstraktnu klasu `Osoba` koja ima polja `ime (String)` i `zanr (MuzickiZanr)`. Definisati odgovarajući konstruktor.
- Definisati interfejs `EngleskiJezik` koji ima apstraktan metod `String pozdravNaEngleskom()`.
- Definisati klasu `Posetilac` koja nasleđuje klasu `Osoba`. Dodati polja `pocetakSlobodnogVremena (int)` i `krajSlobodnogVremena (int)`. Definisati odgovarajući konstruktor i konstruktor kopije. Predefinisati metod `toString` tako da ispis bude oblika: `"Ja sam posetilac Marko i volim da slusam ROK muziku"`. Interval slobodnog vremena može biti npr. od 0 do 23 časa.
- Definisati klasu `PosetilacKojiZnaEngleski` koja nasleđuje klasu `Posetilac` i implementira interfejs `EngleskiJezik`. Metod `toString` poziva metod `pozdravNaEngleskom()` tako da ispis bude oblika: `"I am visitor Marko and I like to listen to ROCK music"`.
- Definisati klasu `Bend` koja ima polja `naziv (String)`, `vremePocetka (int)`, `vremeKraja (int)`, `strani (boolean)` i `posetioци (List<Posetilac>)`. Definisati odgovarajući konstruktor kao i getere. Definisati metod `void dodajPosetioca(Posetilac p)` koji dodaje posetioca u listu. Predefinisati metod `toString` tako da bude oblika:

```
Bend: nazivBenda, vreme: pocetak-kraj, strani: da/ne  
prviPosetilacIzListe  
drugiPosetilacIzListe  
...
```
- U klasi `Test` napisati `main` metod. Definisati listu od 4 benda (2 strana i 2 domaća) i niz od 6 posetilaca. Posetioce iz niza dodati u listu posetilaca benda ukoliko se njihovo slobodno vreme uklapa u vreme održavanja koncerta. Strani bend može slušati samo posetilac koji govori engleski. Posetilac može slušati najviše jedan bend.
Ispisati bendove iz liste.

2. [20p] Napisati aplikaciju korišćenjem JavaFX biblioteke koja izgleda kao na slikama. U korenom direktorijumu projekta napraviti datoteku `studenti.csv`. Sadržaj jednog reda datoteke je oblika:

```
ime, prezime, ocena1, ocena2, ...
```

Broj ocena po studentu može biti različit. Klikom na dugme `Ucitaj` se učitavaju podaci iz datoteke. Klikom na dugme `Prikazi` se prikazuju informacije o studentima u obliku:

```
Ime Prezime: prosečna.ocena.
```

Ukoliko je u polje pored dugmeta `Prikazi` unet realan broj, ispisuju se informacije o studentima čiji prosek je iznad zadatog. U suprotnom ispisuju se svi podaci.

3. [25p] Implementirati generičku klasu `Paket<T>`, koja ima attribute `sadrzaj` tipa `T` i `tezina` tipa `double`. Implementirati sledeće metode:

- `compareTo` koja poredi pakete svođenjem na poređenje atributa `sadrzaj`.
- `toString` koja vraća informacije o paketu u formatu `"sadrzaj - tezina kg"`.

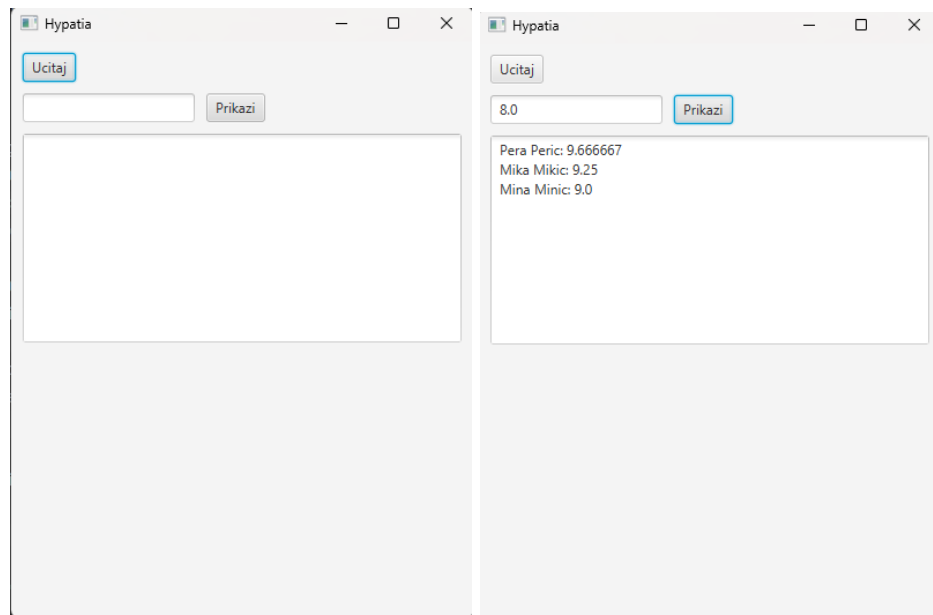
Definisati klasu `Adreser` koja sadrži atribut `adrese` tipa `Map<Paket<String>, String>`. Implementirati sledeće metode:

- Konstruktor koji ne prima ni jedan argument i inicijalizuje atribut `adrese` na instancu klase `TreeMap`.

- Konstruktor koji prima objekat tipa `Comparator` koja inicijalizuje atribut `adrese` na instancu klase `TreeMap` tako da se koristi dati komparator za poređenje ključeva u mapi.
- `dodaj` koja dodaje novi paket i adresu u mapu.
- `print` koja ispisuje sve pakete u mapi i njihove adrese na standardni izlaz.

Napisati klasu *Main* u kojoj treba testirati rad sa klasama *Paket* i *Adreser* na sledeći način:

- Kreirati instancu klase `Adreser` bez argumenata u konstruktoru.
- Dodati nekoliko novih paketa i adresa u adreser metodom `dodaj`. Obratiti pažnju da paketi moraju da imaju različitu vrednost za sadržaj.
- Ispisati sadržaj adresera metodom `print`.
- Kreirati drugu instancu klase `Adreser` tako da za poređenje koristi težine paketa umesto sadržaja.
- Dodati nekoliko novih paketa i adresa u drugi adreser metodom `dodaj`. Obratiti pažnju da paketi moraju da imaju različitu vrednost za težinu.
- Ispisati sadržaj drugog adresera metodom `print`.



Slika uz zadatak 2.