

Objektno-orijentisano programiranje, , JUN 1, Grupa 2

Matematički fakultet, školska godina 2019/2020

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_Asistent_Prezime_Ime_Indeks` (npr. `oop_NM_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**

Inicijalni asistenti: Biljana - BS, Nemanja - NM, Anja - AB, Denis - DA

U tekstu je dat opis klasa, njihovih atributa i metoda. **Dozvoljeno** je (i ohrabrujemo Vas) dodati nove attribute, klase, metode, enume, interfejsu u slučaju da Vam olakšavaju implementaciju, i/ili smatrate da Vam poboljšavaju kvalitet koda i slično. Nekada će zahtevi u zadatku i zahtevati od Vas da dodate novi atribut ili slično.

Da bi se uspešno položio ispit potrebno je osvojiti **barem 25 poena**.

1. Napraviti apstraktnu klasu `ZaraznaBolest` koja ima atribut `int duzinaBolesti` koji čuva informaciju o dužini trajanja bolesti. Obezbediti konstruktor koji prihvata ceo broj i čuva ga. Implementirati `get` i `set` metod za atribut klase.

2. Implementirati klasu `Grip` koja nasleđuje klasu `ZaraznaBolest`. Implementirati konstruktor koji prima jedan ceo broj, kao i konstruktor kopije.

Implementirati metod:

- `String toString()` - vraća stringovnu reprezentaciju Gripa - prikazati trajanje gripa (pogledati primer).

3. Implementirati klasu `Korona` koja nasleđuje klasu `ZaraznaBolest` koja ima atribut `boolean pokazujeSimptome` koji čuva informaciju o tome da li pacijent pokazuje simptome. Implementirati konstruktor koji prima vrednosti za sva polja, kao i konstruktor kopije. Implementirati potreban `get` metod.

Implementirati metode:

- `boolean test()` - metod vraća `true` ako je pacijent zaražen, inače vraća `false`. Ukoliko pacijent pokazuje simptome, šansa da je zaražen je 80%, u suprotnom 40%. Nasumično odabrati jedan realan broj iz intervala $[0, 1)$ (na primer, korišćenjem metoda `nextDouble()` klase `Random`) i vratiti odgovarajuću vrednost.
- `String toString()` - vraća stringovnu reprezentaciju korone - prikazati dužinu bolesti i informaciju o prisutnosti simptoma (videti primer).

4. Definisati interfejs `Izleciv` koji sadrži metode:

- `void leci(int brojDana)`
- `boolean izlecen()`

5. Implementirati klasu `Pacijent` koja implementira interfejs `Izleciv`. Sačuvati informacije o imenu i prezimenu tipa `String`, identifikacioni broj knjižice, dužinu lečenja pacijenta u danima (`int duzinaLecenja`), da li je zaražen (`boolean zarazen`) kao i dijagnozu bolesti (tipa `ZaraznaBolest`). Implementirati konstruktor koji prima odgovarajuće podatke za sva polja osim za `duzinaLecenja` i `zarazen` koja se pri konstrukciji objekta postavljaju na 0 i `false` redom. Implementirati potrebne `get` i `set` metode.

Metod `boolean izlecen()` vraća vrednost `true` ukoliko je broj dana lečenja pacijenta veći ili jednak broju dana potrebnih za lečenje bolesti koju ima taj pacijent. Metod `void leci(int brojDana)` uvećava broj dana lečenja pacijenta za prosleđeni argument.

Implementirati metod:

- `String toString()` - koji vraća stringovnu reprezentaciju koja sadrži identifikacioni broj, ime, prezime, bolest, kao i vreme (u danima) do izlečenja pacijenta za datu bolest (pogledati primer).

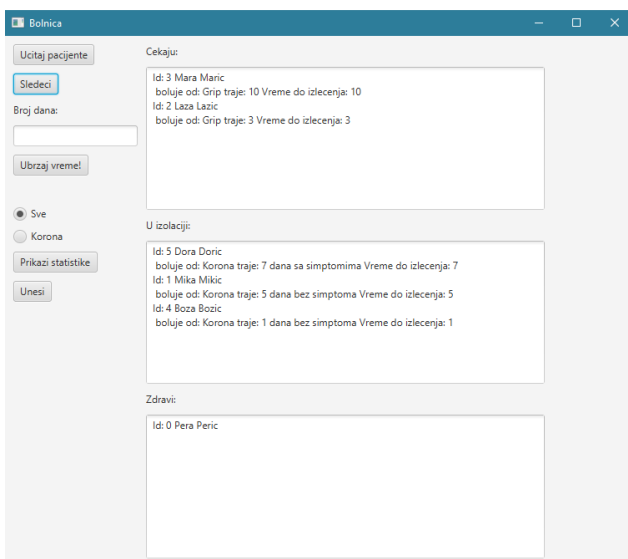
6. Implementirati klasu `Bolnica` koja sadrži tri liste pacijenata: pacijenti u čekaonici, pacijenti u izolaciji i zdravi pacijenti. Implementirati potrebne `get` i `set` metode, kao i naredne metode:

- `void ucitaj()` - iz datoteke "`pacijenti.txt`" učitati podatke u listu čekaonica. Svakom pacijentu je potrebno dodeliti `idKnjizice` redom počevši od 0. Svaki red datoteke sadrži ime, prezime, oznaku bolesti (`g` - grip, `k` - korona) i broj dana potrebnih za lečenje. U slučaju korone postoji dodatan podatak ("da" ili "ne") koji označava da li pacijent pokazuje simptome ili ne.
- `void sledeci()` - izbacuje se prvi pacijent iz liste pacijenata koji su u čekaonici. Ukoliko pacijent boluje od gripa ili je test za koronu pozitivan, dodati ga u listu pacijenata u izolaciji i ažurirati vrednost polja `zarazen`. U suprotnom, ako je test negativan, pacijenta treba dodati u listu zdravih, a dužinu bolesti i dužinu lečenja postaviti na 0.

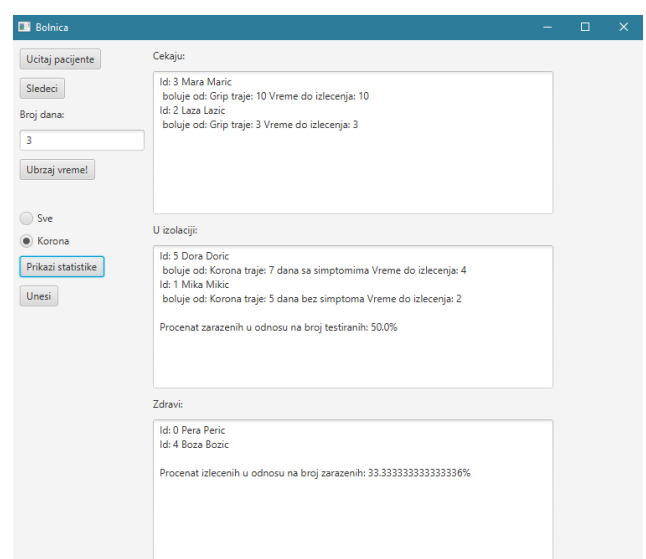
- `void unesi()` - u datoteku "izvestaj.txt" uneti sadržaj sve tri liste redom, tako da sadržaj datoteke odgovara ispisu u `TextArea` elementima.
- Implementirati klasu `Main` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama. Obezbediti da je uvek odabrano tačno jedno radio dugme kao i da je podrazumevano odabrano radio dugme koje ima vrednost `Sve`. Klasa sadrži jedno polje tipa `Bolnica`.
 - U klasi `Main` implementirati statički generički metod `void ispis(List<T>, TextArea ta)` koji bruše sadržaj `TextArea` elementa, a zatim u njega ispisuje sadržaj liste tako da svaki element bude u zasebnom redu.
 - Klikom na dugme `Ucitaj pacijente` učitavaju se podaci o pacijentima u bolnici. Zatim je potrebno sortirati pacijente tako da su na početku liste pacijenti koji potencijalno boluju od korone, zatim pacijenti koji boluju od gripa. U okviru grupe pacijenata koji boluju od iste bolesti, na početku je potrebno da budu pacijenti čija bolest je najduža. Tako sortirane pacijente je potrebno ispisati u prvi `TextArea` (koji predstavlja čekaonicu). Obezbediti da se nakon uspešnog učitavanja više ne može vršiti čitanje.
 - Klikom na dugme `Sledeci` poziva se metod `sledeci()` nad bolnicom. Zatim je potrebno ažurirati prikaz pacijenata u svakom od `TextArea` elemenata tako da prikazuje trenutni sadržaj svake od lista. Ukoliko nema pacijenata u čekaonici ispisati `Nema vise nikoga u cekaonici!`.
 - Klikom na dugme `Ubrzaj vreme!` čita se broj dana (pozitivan ceo broj) iz polja `Broj dana` i simulira se prolazak tog broja dana za pacijente u izolaciji. Potrebno je ažurirati broj dana lečenja za svakog pacijenta u izolaciji. Ukoliko je neko izlečen (proveru vršiti metodom `boolean izlecen()`) ukloniti ga iz izolacije i dodati u listu zdravih pacijenata. Zatim je potrebno ažurirati prikaz pacijenata u svakom od `TextArea` elemenata tako da prikazuje trenutni sadržaj svake od lista. Ukoliko pročitani broj dana nije validan, ispod dugmeta `Ubrzaj vreme!` crvenim slovima ispisati `"Greska!"`.
 - Klikom na dugme `Prikazi statistike` prikazuje se procenat zaraženih u drugom `TextArea` elementu i procenat izlečenih u trećem `TextArea` elementu. U zavisnosti od odabrane opcije, statistika se prikazuje
 - za grip i koronu - procenat zaraženih računa se u odnosu na ukupan broj pacijenata koji nisu u čekaonici, a broj izlečenih računa se u odnosu na ukupan broj zaraženih pacijenata (onih čije polje `zarazen` ima vrednost `true`).
 - samo za koronu - procenat zaraženih računa se u odnosu na ukupan broj testiranih na koronu, a broj izlečenih računa se u odnosu na ukupan broj zaraženih koronom (onih čije polje `zarazen` ima vrednost `true`).
 - Klikom na dugme `Unesi` poziva se metod `unesi()` nad bolnicom.

Primer sadržaja datoteke:

```
Pera, Peric, k, 2, da
Mika, Mikic, k, 5, ne
Laza, Lazic, g, 3
Mara, Maric, g, 10
Boza, Bozic, k, 1, ne
Dora, Doric, k, 7, da
```



Slika 1: Učitavanje i sortiranje pacijenata, proveravanje za 4 pacijenta



Slika 2: Ubrzavanje vremena za 3 dana i prikaz statistike za koronu