

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 *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. U napravljenom projektu, paket takođe nazvati tako.

Kod **ne sme** imati sintakasnih 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, interfejse u slučaju da olakšavaju implementaciju i/ili poboljšavaju kvalitet koda i slično.

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

Za potrebe održavanja softvera imate zadatak da kreirate sistem za održavanje softvera — praćenje problema i bagova, otvaranje diskusija ili novih ideja za implementaciju i slično.

- (2 poena)** Napraviti enumeracioni tip `Labela` koji se sastoji od narednih konstanti: `ZA_KASNIJE`, `POTREBNO_OBJASNJENJE`, `HITNO`
- (6 poena)** Napraviti apstraktnu klasu `Stavka` koja sadrži polja `korisnickoIme` (`String`), `naslov` (`String`), `sadrzaj` (`String`), `labela` (`Labela`) i `razresena` (`boolean`). Implementirati:
 - konstruktor koji prima vrednosti za polja `korisnickoIme`, `naslov`, `tekst`;
 - `get` metode za sva polja;
 - metod `void razresi()` koji razresava trenutnu stavku tako što postavlja vrednost polja `razresena` na `true`;
 - metod `void labeliraj(Labela)` koja postavlja vrednost polja `labela`;
 - metod `toString` koji formira i vraća nisku sačinjenu od kućice koja je markirana ako je stavka razrešena, labele (ukoliko postoji), korisničkog imena i pratećeg sadržaja (videti sliku 1):
[] (`labela`) `korisnickoIme: naslov`
`sadrzaj`
- (4 poena)** Napraviti klasu `Pitanje` koja nasleđuje klasu `Stavka`. Klasa se dodatno karakteriše poljem `odgovor` koje, ukoliko je stavka razresena, čuva odgovor na pitanje koje ta instanca predstavlja. Implementirati:
 - konstruktor koji prima vrednosti za nasledjena polja;
 - metod `void razresi(String)` koji razrešava pitanje i argument čuva kao odgovor na pitanje;
 - `get` metod za polje `odgovor` koje, ukoliko je pitanje razrešeno vraća odgovor na pitanje, a inače tekst *"Pitanje nema odgovor."*;
 - metod `toString` koji formira i vraća nisku sačinjenu od kućice koja je markirana ako je pitanje razrešeno, labele (ukoliko postoji), korisničkog imena, pratećeg sadržaja i odgovora (ako postoji) (videti sliku 1):
(?) [] (`labela`) `korisnickoIme: naslov`
`sadrzaj - odgovor`
- (5 poena)** Napraviti klasu `Bag` koja nasleđuje klasu `Stavka`. Klasa se dodatno karakteriše poljima `id` (`int`), `ozbiljnost` (`int`) i `zaduzen` (`String`). Implementirati:
 - konstruktor koji prima vrednosti za sva polja;
 - odgovarajuće `get` metode za polja;
 - metod `void zaduzi(String)` koji ovaj bag dodeljuje korisniku sa imenom koje je prosledjeno metodi `zaduzi` — drugim rečima, modifikuje vrednost polja `zaduzen`;
 - `get` metod za polje `zaduzen` koje, ukoliko je bag zadužen nekom korisniku, vraća korisničko ime osobe koja je zadužena za taj bag, a inače tekst *"Bag nije dodeljen nijednom programeru."*;
 - metod `toString` koji formira i vraća nisku sačinjenu od identifikatora boga, njegove ozbiljnosti, imena programera zaduženog za bag (ako je dodeljen nekom programeru), kućice koja je markirana ako je bag razrešen, labele (ukoliko postoji), korisničkog imena osobe koja je postavila bag i pratećeg sadržaja (videti sliku 1):
(!3 10 `zaduzen`) [] (`labela`) `korisnickoIme: naslov`
`sadrzaj`
- (3 poena)** Napraviti generički interfejs `KolekcijaStavki<T>` (obezbediti da `T` mora biti klasa koja nasleđuje tip `Stavka`) koji se karakteriše metodima:
 - `List<T> listaj()` — vraća listu stavki iz kolekcije;
 - `boolean dodaj(T)` — dodaje stavku u kolekciju i vraća indikator uspeha dodavanja;

- `void sortiraj()` — sortira stavke kolekcije;
6. (20 poena) Kreirati klase `ListaPitanja` (implementira interfejs `KolekcijaStavki<Pitanje>`) i `SkupBagova` (implementira interfejs `KolekcijaStavki<Bag>`) tako da ne postoje dva boga sa istim identifikatorom u skupu). `ListaPitanja` pitanja čuva u proizvoljno implementiranoj implementaciji liste, dok `SkupBagova` čuva bagove u proizvoljno odabranoj implementaciji skupa. Implementirati metode interfejsa `KolekcijaStavki` tako da se održi semantika interfejsa:
- `dodaj()` za listu pitanja uvek uspeva;
 - `dodaj()` za skup bagova treba da vrati `false` ako se bag sa datim identifikatorom već nalazi u skupu;
 - `sortiraj()` sortira pitanja po razrešenosti — nerazrešena pitanja imaju veći prioritet od razrešenih;
 - `sortiraj()` sortira bagove po razrešenosti (kao pitanja iznad) i po ozbiljnosti opadajuće u okviru obe grupe;
7. (20 poena) Napraviti klasu `PracenjeStavki` koja nasleđuje klasu `Application` biblioteke `JavaFX` i izgleda kao na slikama. Obezbediti da je unapred selektovano prvo radio dugme i da u svakom trenutku može biti selektovano tačno jedno radio dugme. Klasa sadrži polja `pitanja` (`ListaPitanja`), `bagovi` (`SkupBagova`).

Klikom na dugme `Dodaj` dodaje se nova stavka (pitanja ili bag, u zavisnosti od selekcije) sa atributima uzetim iz polja za unos. Obavezna polja za sve stavke su: `korisnik`, `naslov`, `sadrzaj`. Za bag, dodatno je obavezno uneti `id` i `ozbiljnost`. Dodavanje se odvija pozivanjem metoda `dodaj` nad odgovarajucom kolekcijom stavki. Nakon dodavanja, ukoliko je ono bilo uspesno, ispisuje se poruka kao na slici uz rezultat poziva `toString()` metoda nad unetom stavkom. Dodavanje ne uspeva ukoliko:

- ukoliko fali obavezno polje
- identifikator boga ili ozbiljnost boga nisu validni celi brojevi
- već postoji bag sa unetim identifikatorom u skupu bagova

U svakom od navedenih slucajeva ispisati odgovarajuću poruku o grešci u oblast sa leve strane.

Klikom na dugme `Izlistaj` se ispisuje trenutno stanje kolekcija `pitanja` i `bagovi`. Prvo se ispisuju svi bagovi pa zatim i sva pitanja (nije neophodno sortirati).

Klikom na dugme `Sacuvaj` se ispisuje sortirani spisak svih bagova i pitanja u datoteku `spisak.txt` (prepisati je ukoliko već postoji). Prvo ispisati sve bagove a zatim pitanja, i poštovati poredak sortiranja opisan u ranijim stavkama. Nakon ispisa, ukoliko je ispis bio uspešan, napisati odgovarajuću poruku. Ukoliko se desio izuzetak prilikom ispisa, ispisati poruku *"Neuspelo pisanje u fajl"*. Primer rezultata ispisa je dat ispod.

```
Bagovi:
(! 6 10 ) [ ] Zika: OZBILJAN BAG
Nasao sam veoma ozbiljan bag
(! 1 4 ) [ ] Zika: Bag
Nasao sam bag
(! 4 10 Zika) [x] Pera: Novi bag
Dodeljujem Ziki
(! 5 6 Zika) [x] Zile: Bag
Dodeljujem Peri

Pitanja:
(?) [ ] Pera: Pitanje
Kako se ovo koristi?
(?) [x] Pera: HELP
Pomozite pls
- OK
```

Primeri izgleda aplikacije se mogu naći na narednoj strani.

