

Objektno orijentisano programiranje, Ispit JUN1, Grupa 2

Matematički fakultet

Školska godina 2018/2019

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_Asistent_Grupa_Prezime_Ime_Indeks` (npr. `oop_NM_1_Peric_Pera_mi12082`). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. U napravljenom projektu, napraviti paket sa istim nazivom.

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

Vreme za rad: **2,5 sata**

Inicijalni asistenata: Biljana - BS, Nemanja - NM, Anja - AB, Ivan - IR, Rastko - RD

1. Napraviti klasu `Datum` koju karakterišu atributi `dan`, `mesec` i `godina` (svi tipa `int`). Implementirati konstruktor koji prima jednu nisku oblika `DD.MM.GGGG.` gde `DD` predstavlja dan, `MM` mesec, a `GGGG` godinu. Dan i mesec uvek zauzimaju 2 karaktera (na primer, drugi jul zapisuje se kao `02.07`). Možemo pretpostaviti da je niska zadata u ispravnom formatu i ne treba je proveravati. Implementirati potrebne `get` metode, i metod `toString()` koji ispisuje datum u obliku `DD.MM.GGGG.`, kao u primeru:

```
DD.MM.GGGG.  
12.06.2019.  
02.07.2019.
```

2. Napraviti apstraktnu klasu `Poruka` koju karakterišu atributi `datum` (`Datum`, označava kada je poruka poslata ili primljena) i `tekstPoruke` (`String`). Implementirati konstruktor koji prima dve niske, poruku i datum, i konstruktor kopije. Implementirati potrebne `get` metode. Deklarisati apstraktan metod `String sacuvajPoruku()`.
3. Napraviti klasu `PoslataPoruka` koja nasleđuje klasu `Poruka` i sadrži atribut `primalac` (`String`). Implementirati apstraktan metod `sacuvajPoruku()` tako da vraća nisku oblika:

```
s; datum; primalac; tekstPoruke  
s; 12.06.2019.; Ljubica; Srečno na ispitu!
```

Implementirati metod `toString()` koji ispisuje poruku kao u primeru:

```
Primalac: ime datum  
tekstPoruke  
-----  
  
Primalac: Ljubica 12.06.2019.  
Srečno na ispitu!  
-----
```

4. Napraviti klasu `PrimljenaPoruka` koja nasleđuje klasu `Poruka` i sadrži attribute `posiljalac` (`String`) i `procitano` (`boolean`). Implementirati apstraktan metod `sacuvajPoruku()` tako da vraća nisku kao u primeru (`procitano` treba zameniti rečima **da** ili **ne**, u zavistnosti od vrednosti polja):

```
r; datum; posiljalac; tekstPoruke; procitano  
r; 12.06.2019.; Ljubica; Hvala!; ne  
r; 10.06.2019.; Janko; Nece biti generickih klasa.; da
```

Implementirati metod `toString()` koji ispisuje poruku kao u primeru (samo za nepročitane poruke treba dodati tekst u zagradi, a za pročitane ništa):

```
Posiljalac: Ljubica (neprocitano) 12.06.2019.  
Hvala!  
-----  
  
Posiljalac: Janko 10.06.2019.  
Nece biti generickih klasa.  
-----
```

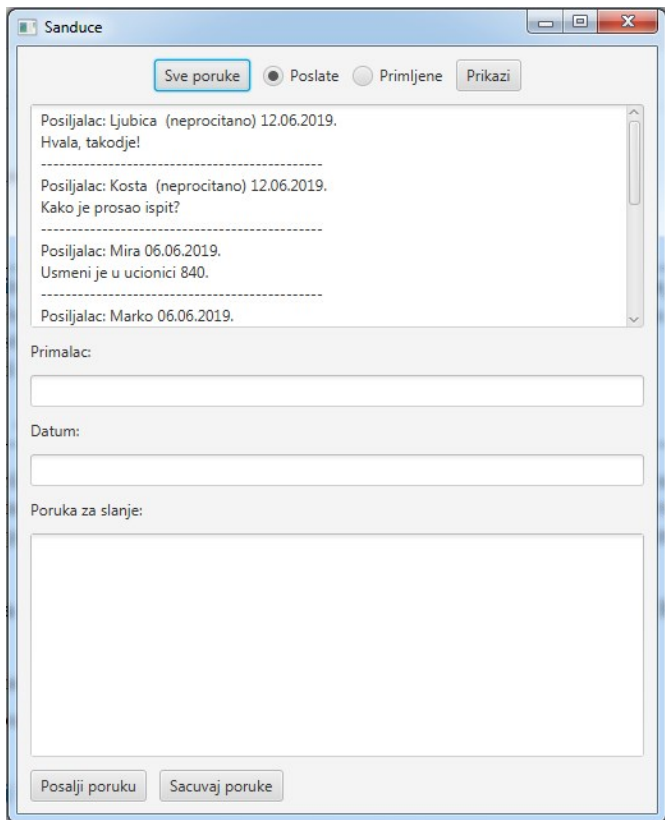
5. Napraviti klasu `Korisnik` koju karakterišu atributi `korisnickoIme` (`String`) i `poruke` (`List<Poruka>`). Implementirati konstruktor koji prihvata vrednost za korisničko ime. Implementirati `get` metode za oba polja. Implementirati metod `void novaPoruka(String podaci)` koji pravi novu poruku na osnovu podataka iz niske `podaci`. Podaci su u obliku kao jedan red datoteke. Implementirati metod `toString()` koji vraća ime korisnika.
6. Napraviti klasu `Sanduče` koja predstavlja `javafx` aplikaciju i izgleda kao na slici 1. Klasa sadrži jedno statičko polje tipa `Korisnik` - instancirati ga sa proizvoljnim imenom. Obezbediti da u svakom trenutku može biti selektovano tačno jedno radio dugme, i da je na početku odabrano "Poslate".

7. Na klik dugmeta:

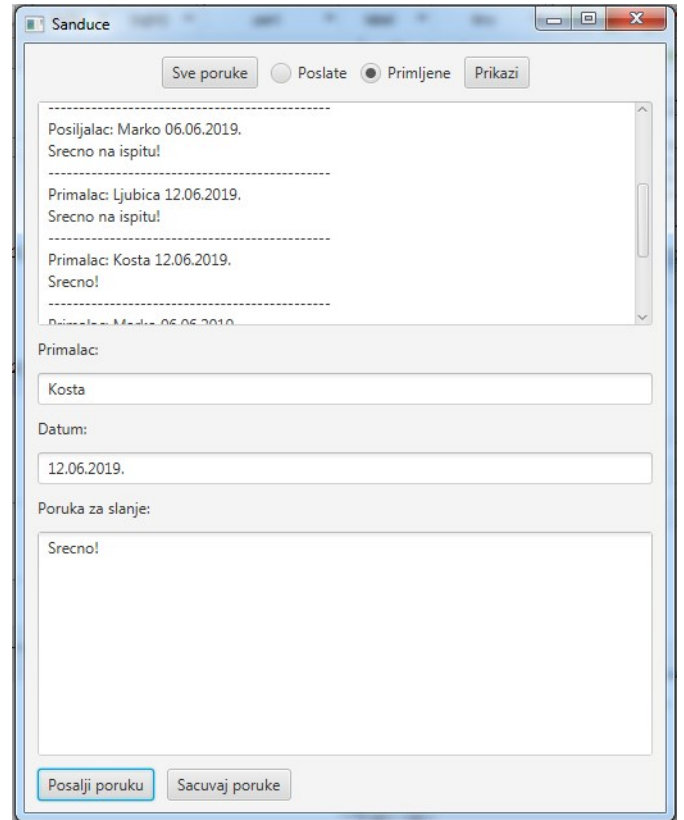
- **Sve poruke** vrši se čitanje iz datoteke `poruke.txt` i podaci se smeštaju u listu poruka korisnika (korišćenjem metoda `novaPoruka`). Lista se sortira i podaci se prikazuju u elementu `TextArea`. Poruke se prvo sortiraju po tipu tako da prvo idu primljene poruke, a zatim poslate (na primer, korišćenjem operatora `instanceof`). U slučaju da postoji više primljenih poruka potrebno je sortirati ih tako da prvo idu nepročitane, pa pročitane poruke, a onda, u slučaju više nepročitanih ili pročitanih poruka, po datumu tako da prvo idu novije poruke pa starije. U slučaju da ima više poslanih poruka, urediti ih samo po datumu tako da prvo idu novije poruke, pa starije.
- **Prikazi** u elementu `TextArea` ispisuju se samo poslate ili samo primljene poruke, u zavisnosti od odabranog radio dugmeta.
- **Posalji poruku** pravi novu poruku i dodaje je u listu poruka instanciranog korisnika. Ukoliko je neki `TextField` element prazan ne vršiti slanje.
- **Sacuvaj** u datoteku `poruke.txt` se upisuju poruke iz liste instanciranog korisnika u formatu koji se dobija primenom metoda `sacuvajPoruku()`. U element `TextArea` upisati poruku `Podaci uspesno sačuvani`.

Primer datoteke:

```
s; 12.06.2019.; Ljubica; Srecno na ispitu!  
s; 06.06.2019.; Marko; HVALA!  
r; 12.06.2019.; Ljubica; Hvala, takodje!; ne  
r; 06.06.2019.; Mira; Usmeni je u ucionici 840.; da  
s; 05.06.2019.; Janko; Sta ce biti na ispitu iz OOP-a?  
r; 12.06.2019.; Kosta; Kako je prosao ispit?; ne  
r; 06.06.2019.; Marko; Srecno na ispitu!; da
```



Slika 1: Prikaz učitanih podataka



Slika 2: Slanje nove poruke