

I parcijalni ispit, septembarski rok

Pozivnica za GitHub Classroom: <https://classroom.github.com/a/DDkRqvGX>

Ukupno bodova: 20 (Bodovi će se dodijeliti proporcionalno broju uspješnih testova.)

Na repozitoriju se nalazi gotov projekat koji sadrži samo praznu Main klasu i testove. Vaš zadatak je da napravite kompletan Java program koji zadovoljava postavku zadatka i prolazi testove.

Napomena: ukoliko se uoče neke nesuglasice između teksta i testova (npr. tekst izuzetka), uraditi zadatak tako da prođe test.

Zadatak 1:

Potrebno je kreirati klasu **Korisnik**. Korisnik je opisan svojim **imenom** (String), **prezimenom** (String) i **nadimkom** (String). Za klasu vrijedi:

- Treba imati konstruktor koji prima sva tri parametra i odgovarajuće gettere i settere.
- Niti jedno od atributa se ne smije moći postaviti na null ili prazan string. Ukoliko se ipak ovo pokuša treba baciti izuzetak tipa **IllegalArgumentException** sa porukom **“Ime, prezime i nadimak ne smiju biti null!”** odnosno **“Ime prezime i nadimak ne smiju biti prazni!”**.
- Nadimak korisnika mora imati 5 ili više karaktera. Ukoliko se zada nadimak koji ne ispunjava ovo treba baciti prethodni izuzetak sa porukom **“Nadimak mora imati 5 ili više karaktera!”**
- Objekat tipa korisnik se ispisuje na ekran u sljedećem formatu **Ime Prezime (nadimak)**

Zadatak 2:

Nakon ovoga, razvijte klasu **Poruka**. Poruka sadrži informacije o **pošiljaocu** (posiljalac, Korisnik), **primaocu** (primalac, Korisnik), datumu kada je poruka poslana (**datumSlanja**, LocalDateTime) te **tekstu** poruke (String). Pored toga postoji atribut koji treba da naznači da li je poruka pročitana ili ne. Taj atribut se naziva **statusPoruke** i tipa je **StatusPoruke** (enum). Ovaj atribut ima jednu od dvije moguće vrijednosti: PROCITANA I NEPROCITANA. Za klasu Poruka vrijedi:

- Treba imati konstruktor koji postavlja sve attribute s tim što ne prima status već ga postavlja na NEPROCITANA
- Treba imati odgovarajuće gettere i settere
- Niti jedan od atributa osim datuma čitanja ne smije biti null. Ukoliko se pokuša postaviti ovakva vrijednost treba baciti korisnički definisan izuzetak

NeispravanFormatPoruke sa porukom “**{Atribut} ne smije biti null!**” (npr. Tekst ne smije biti null!).

- Tekst poruke ne smije biti prazan. Ukoliko se pokuša postaviti takva vrijednost baciti isti tip izuzetka sa porukom “Tekst ne smije biti prazan!”.
- Objekat tipa poruka se prikazuje na ekranu u sljedećem formatu:
[od: Miki Maus (miki123) za: Paja Patak (paja123) tekst: Da li si slobodan za kafu danas?]

Zadatak 3:

Konačno napravite klasu **Messenger**. Ova klasa služi kako bi se vodila evidencija o svim porukama koje se razmjenjuju između korisnika. Klasa sadrži listu poruka (**poruke**, ArrayList). Napravite konstruktor bez parametara koji listu inicijalizira na praznu. Napravite sljedeće metode:

- **dajSvePoruke** koja vraća sve poruke
- **posaljiPoruku** koja prima tri parametra, korisnika koji šalje, korisnika koji prima poruku te tekst poruke. Ova metoda registruje poruku u listu, njen status postavlja na NEPROCITANA, a datum i vrijeme slanje postavlja na trenutačno.
- **posaljiPoruku** koja prima poruku kao parametar i dodaje je u listu poruka
- **posaljiPoruke** koja prima listu poruka kao parametar koje treba evidentirati
- **ponistiSlanje** koja prima kao parametar poruku čije se slanje pokušava poništiti. Ukoliko poruka nije poslana (nije u listi) treba baciti izuzetak tipa **NeispravnaAkcija** sa porukom “**Nije moguće poništiti slanje poruke koja nije nikada poslana!**”. Ukoliko je stanje poruke **NEPROCITANA** tada se može izvršiti poništavanje. U suprotnom baciti isti izuzetak sa porukom “**Nije moguće poništiti slanje poruke koja je već pročitana!**”.
- **procitajPoruku** koja prima kao parametar poruku koja se pokušava pročitati. Ukoliko poruka nije poslana treba baciti isti izuzetak kao u prethodnoj metodi, sa porukom “**Nije moguće pročitati poruku koja nije nikada poslana!**”. Ukoliko je poruka nepročitana, ovom metodom se mijenja njen status na PROCITANA.
- **oznaciKaoNepročitano** koja prima kao parametar listu poruka koje treba označiti kao nepročitane. Provjeriti da je svaka od poruka poslana. Ukoliko bar jedna nije poslana onda baciti izuzetak **NeispravnaAkcija** sa porukom “**Neke od poruka koje pokušavate označiti kao nepročitane nisu poslane!**”. Ukoliko su sve poruke poslane, promijeniti status svih poruka na nepročitano.
- **dajNepročitanePoruke** koja vraća mapu u kojoj je svaki element par Korisnik i lista Poruka. Za svakog korisnika treba pronaći njegove nepročitane poruke (koje su njemu poslane) i na taj način formirati ovu mapu. U mapi se trebaju uzeti u obzir samo korisnici kojima je dostavljena bar jedna poruka (neovisno o tome da li je pročitana ili ne).
- **dajPristiglePorukeZaKorisnika** koja prima kao parametar korisnika a vraća sve njemu poslane poruke. Ukoliko nema takvih, vraća praznu listu.
- **dajPorukeZaKorisnika** koja je slična kao prethodna metoda, ali pored korisnika prima i status tipa **StatusPoruke** te treba da vrati sve njemu poslane poruke a čiji status odgovara vrijednosti drugog parametra metode. Ako takvih nema (ili nema

poruka za korisnika), vratiti praznu listu. Ovu metodu obavezno napisati korištenjem **samo streamova**.

- **filtrirajPoruke** koja prima kao parametar kriterij na osnovu kojeg treba filtrirati sve poruke i vraća listu tako filtriranih poruka.
- **dajStarijeOd** prima korisnika i datum (LocalDateTime) te vraća listu svih poruka koje je korisnik poslao a poslane su prije datuma koji se proslijeđuje kao drugi parametar (računajući i taj datum). **Ovu metodu obavezno napišite tako što ćete pozvati metodu filtrirajPoruke sa odgovarajućom lambda funkcijom.**
- odgovarajuću **toString** metodu koja na ekran ispisuje sve poruke u listi poruka tako da su sve ispisane u novom redu u formatu koji je naznačen u postavci klase Poruka.

Sarajevo, 12. 9. 2020

Vedran Zuborić