

Domaći zadatak - JUnit+JavaDoc (rok je 12.4.2017.)

Napraviti repozitorijum na GitHub-u koji se zove „Twitter“ i postaviti Eclipse projekat Twitter koji je povezan sa ovim repozitorijumom.

Napomena: ako commit-ova bude i više nego što je naznačeno, nema veze. Uvođenje novih grana i tagova je opciono. Takođe, sa vremena na vreme treba uraditi push na GitHub repozitorijum.

Napraviti javnu klasu **TwitterPoruka** u paketu **com.twitter.poruke** i u nju copy-paste-ovati sledeći kod:

```
package com.twitter.poruke;

public class TwitterPoruka {

    private String korisnik;
    private String poruka;

    public String getKorisnik() {
        return korisnik;
    }

    public void setKorisnik(String korisnik) {
        if (korisnik==null || !korisnik.isEmpty())
            throw new RuntimeException(
                "Ime korisnika mora biti uneto");

        this.korisnik = korisnik;
    }

    public String getPoruka() {
        return "poruka";
    }

    public void setPoruka(String poruka) {
        if (this.poruka==null || this.poruka == new String("") ||
            this.poruka.length()>140)
            throw new RuntimeException(
                "Poruka mora biti uneta i mora imati najvise 140 znakova");

        this.poruka = poruka;
    }

    public String toString(){
        return "KORISNIK:"+korisnik+" PORUKA:"+poruka;
    }
}
```

Uraditi commit sa jasno naznačenom porukom o tome šta je urađeno do sada.

Napisati JavaDoc dokumentaciju za klasu TwitterPoruka, njene attribute i metode (na osnovu toga šta kod radi).

NAPOMENA: Dokumentaciju treba uraditi što detaljno tj. napisati neki kratak tekst za svaki atribut i metodu i obavezno uvesti **sve tagove koji imaju smisla u tom konkretnom slučaju i opise tj. tekst uz te tagove** (za klasu @autor i @version, za metode @param ako ta metoda ima parametre, @throws

ako baca neki izuzetak, @return ako metoda vraća nešto itd.)

Uraditi commit sa jasno naznačenom porukom o tome šta je novo urađeno.

Napisati JUnit testove za klasu TwitterPoruka (za sve set metode, kao i za toString metodu), i proveriti da svi testovi prolaze.

NAPOMENE:

- Koristiti setUp i tearDown metode tj. metode sa @Before i @After anotacijama.
- Potrebno je napisati **makar po jedan test za svaki scenario** izvršavanja iz svake metode koja se testira.
- Neke metode klase TwitterPoruka **sadrže namerno uvedene mane** koje treba testiranjem otkriti (napisati JUnit testove koje izazivaju njihovo manifestovanje) a tek onda i ispraviti.

Uraditi commit sa jasno naznačenom porukom o tome šta je novo urađeno.

Napraviti javnu klasu **Twitter** u paketu **com.twitter** i u nju copy-paste-ovati sledeći kod:

```
package com.twitter;

import java.util.LinkedList;
import com.twitter.poruke.TwitterPoruka;

public class Twitter {

    private LinkedList<TwitterPoruka> poruke =
        new LinkedList<TwitterPoruka>();

    public LinkedList<TwitterPoruka> vratiSvePoruke() {
        return poruke;
    }

    public void unesi(String korisnik, String poruka) {
        //Pravi se nova poruka i puni podacima.
        TwitterPoruka tp = new TwitterPoruka();
        tp.setKorisnik("korisnik");
        tp.setPoruka(poruka);

        //Poruka se unosi u listu na kraj
        poruke.addLast(tp);
    }

    public TwitterPoruka[] vratiPoruke(int maxBroj, String tag) {
        if (tag==null || tag.isEmpty())
            throw new RuntimeException("Morate uneti tag");

        //Ako je maxBroj <=0, vraca maxBroj se postavlja na 100 poruka
        if (maxBroj<=0)
            maxBroj = 100;

        //Pomocna promenljiva koja predstavlja brojac upisanih poruka
        int brojac = 0;

        //Pomocni niz koja predstavlja rezultat pretrage tj. sadrzace
        //sve poruke koje u sebi imaju zadati tag
    }
}
```

```

TwitterPoruka[] rezultat = new TwitterPoruka[maxBroj];

//Pretražuju se poruke i traže se one koje sadrže tag.
//Ako se nađe neka takva, i ako nije prekoračen maxBroj
//ona se upisuje u niz. Ako je prekoračen maxBroj, pretraga
//se prekida.
    for (int i = 0; i < poruke.size(); i++)
        if (poruke.get(i).getPoruka().indexOf(tag)!=-1)
            if (brojac < maxBroj){
                rezultat[brojac+1]=poruke.get(i);
                brojac++;
            }
            else break;

    return rezultat;
}
}

```

Uraditi commit sa jasno naznačenom porukom o tome šta je novo urađeno.

Napisati JavaDoc dokumentaciju za klasu Twitter, njene atribute i metode (na osnovu toga šta metode rade).

Generisati JavaDoc HTML strane. Proveriti da li je generisani JavaDoc u redu. Staviti da git ignoriše doc folder sa javaDoc HTML dokumentacijom (Team -> Ignore). Ovaj folder ne treba da ide na GitHub.

NAPOMENA: Dokumentaciju treba uraditi što detaljno tj. napisati neki kratak tekst za svaki atribut i metodu i obavezno uvesti **sve tagove koji imaju smisla u tom konkretnom slučaju i opise tj. tekst uz te tagove** (za klasu @autor i @version, za metode @param ako ta metoda ima parametre, @throws ako baca neki izuzetak, @return ako metoda vraća nešto itd.)

Uraditi commit sa jasno naznačenom porukom o tome šta je novo urađeno.

Napisati JUnit testove za klasu Twitter (za sve metode), i proveriti da svi testovi prolaze.

NAPOMENE:

- Koristiti setUp i tearDown metode tj. metode sa @Before i @After anotacijama.
- Potrebno je napisati **makar po jedan test za svaki scenario** izvršavanja iz svake metode koja se testira.
- Neke metode klase Twitter **sadrže namerno uvedene mane** koje treba testiranjem otkriti (napisati JUnit testove koje izazivaju njihovo manifestovanje) a tek onda i ispraviti.

Uraditi commit sa jasno naznačenom porukom o tome šta je novo urađeno.

Po završetku domaćeg zadatka, uraditi konačni push na GitHub repozitorijum (sa svim commit-ovima, opcionim granama i opcionim tagovima) i poslati email **sa linkom na GitHub repozitorijum** Bojanu Tomiću (tomicb@fon.bg.ac.rs). Rok za izradu ovog domaćeg zadatka je dve nedelje tj. 12.4.2017.

NAPOMENA: za ovaj domaći se dodeljuju 2 bedža, ali je potrebno dostaviti samo jedan mail sa linkom ka repozitorijumu (ne treba dvaput).