

Domaći zadatak - JUnit+JavaDoc (rok je 30.3.2018.)

Napraviti repozitorijum na GitHub-u koji se zove „VulkanizerskaRadnja“ i postaviti Eclipse projekat VulkanizerskaRadnja 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 **AutoGuma** u paketu **gume** i u nju kopirati sledeći kod:

```
package gume;

public class AutoGuma {

    private String markaModel = null;
    private int precnik = 0;
    private int sirina = 0;
    private int visina = 0;

    public String getMarkaModel() {
        return markaModel;
    }

    public void setMarkaModel(String markaModel) {
        if (markaModel==null || markaModel.length()<3)
            throw new RuntimeException("Morate uneti marku i model");

        this.markaModel = markaModel;
    }

    public int getPrecnik() {
        return precnik;
    }

    public void setPrecnik(int precnik) {
        if (precnik < 13 && precnik > 22)
            throw new RuntimeException("Precnik van opsega");

        this.precnik = precnik;
    }

    public int getSirina() {
        return sirina;
    }

    public void setSirina(int sirina) {
        if (sirina < 135 && sirina > 355)
            throw new RuntimeException("Sirina van opsega");

        this.sirina = sirina;
    }

    public int getVisina() {
        return visina;
    }

    public void setVisina(int visina) {
        if (visina < 25 || visina > 95)
```

```

        throw new RuntimeException("Visina van opsega");

        this.visina = visina;
    }

    @Override
    public String toString() {
        return "AutoGuma [markaModel=" + markaModel + ", precnik=" + precnik +
            ", sirina=" + sirina + ", visina=" + visina + "]";
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        AutoGuma other = (AutoGuma) obj;
        if (markaModel == null) {
            if (other.markaModel != null)
                return false;
        } else if (!markaModel.equals(other.markaModel))
            return false;
        if (precnik != other.precnik)
            return false;
        if (sirina != other.sirina)
            return false;
        if (visina != other.visina)
            return false;
        return true;
    }
}

```

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

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

NAPOMENA: Dokumentaciju treba uraditi što detaljnije tj. napisati neki kratak tekst za svaki atribut i metodu i obavezno uvesti svе 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 AutoGuma (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 svake metode koja se testira (što znači minimum 15 testova ukupno za ovu klasu).
- Testiraju se i toString i equals metode.

- Neke metode **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 **VulkanizerskaRadnja** u paketu **gume.radnja** i u nju kopirati sledeći kod:

```
package gume.radnja;

import java.util.LinkedList;

import gume.AutoGuma;

public class VulkanizerskaRadnja {

    private LinkedList<AutoGuma> gume =
        new LinkedList<AutoGuma>();

    public void dodajGumu(AutoGuma a) {
        if (a == null)
            throw new NullPointerException("Guma ne sme biti null");

        if (gume.contains(a))
            throw new RuntimeException("Guma vec postoji");

        gume.addFirst(a);
    }

    public LinkedList<AutoGuma> pronadjiGumu(String markaModel) {
        if (markaModel == null)
            return null;

        LinkedList<AutoGuma> novaLista = new LinkedList<AutoGuma>();

        for(int i=0;i<gume.size();i++)
            if (gume.get(i).equals(markaModel))
                novaLista.add(gume.get(i));

        return novaLista;
    }
}
```

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

Napisati JavaDoc dokumentaciju za klasu **VulkanizerskaRadnja**, 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 detaljnije 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 VulkanizerskaRadnja (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 (što znači **minimum 7 testova ukupno za ovu klasu**).
- Pri testiranju metode pronadjiGumu, testirati scenarije kad je markaModel null, kad ne može da se nađe nijedna guma koja je te marke i modela, kao i kad vraća jednu ili više guma koje su iste marke i modela ali različitih dimenzija.
- Neke metode **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. 30.3.2018.

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