Univerzitet u Kragujevcu
Fakultet inženjerskih nauka


Projekat iz Objektno-orijentisanog programiranja

# Tema:
# Kalkulator za upoređivanje parametara pneumatika

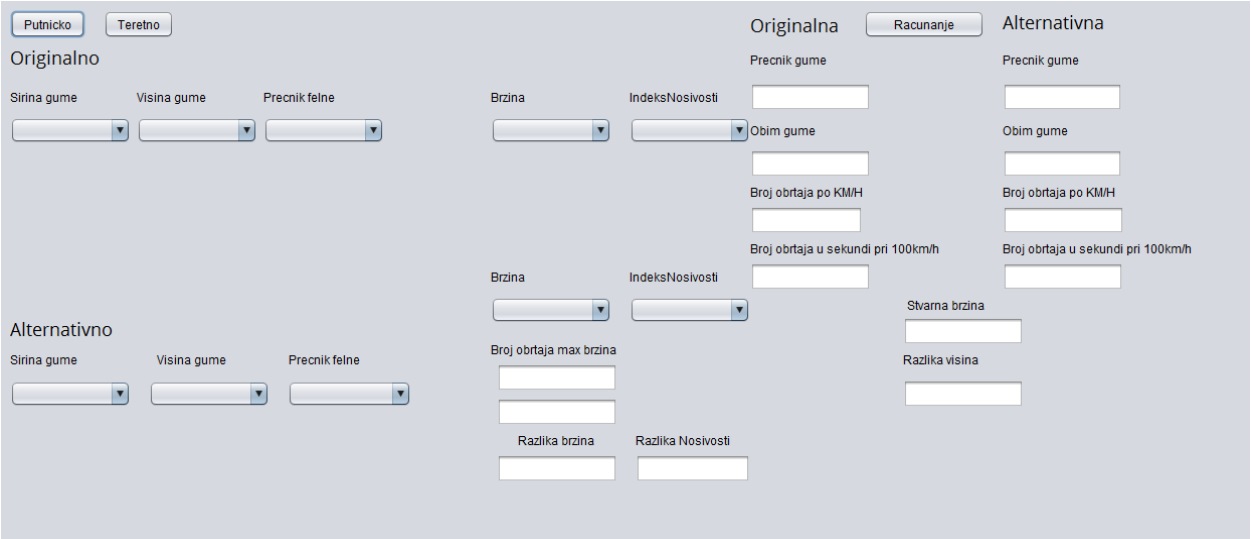
Student:
Aleksa Veličković 576/2015

Predmetni profesor:
Aleksandar Peulić

# SADRŽAJ

# 1. UVOD

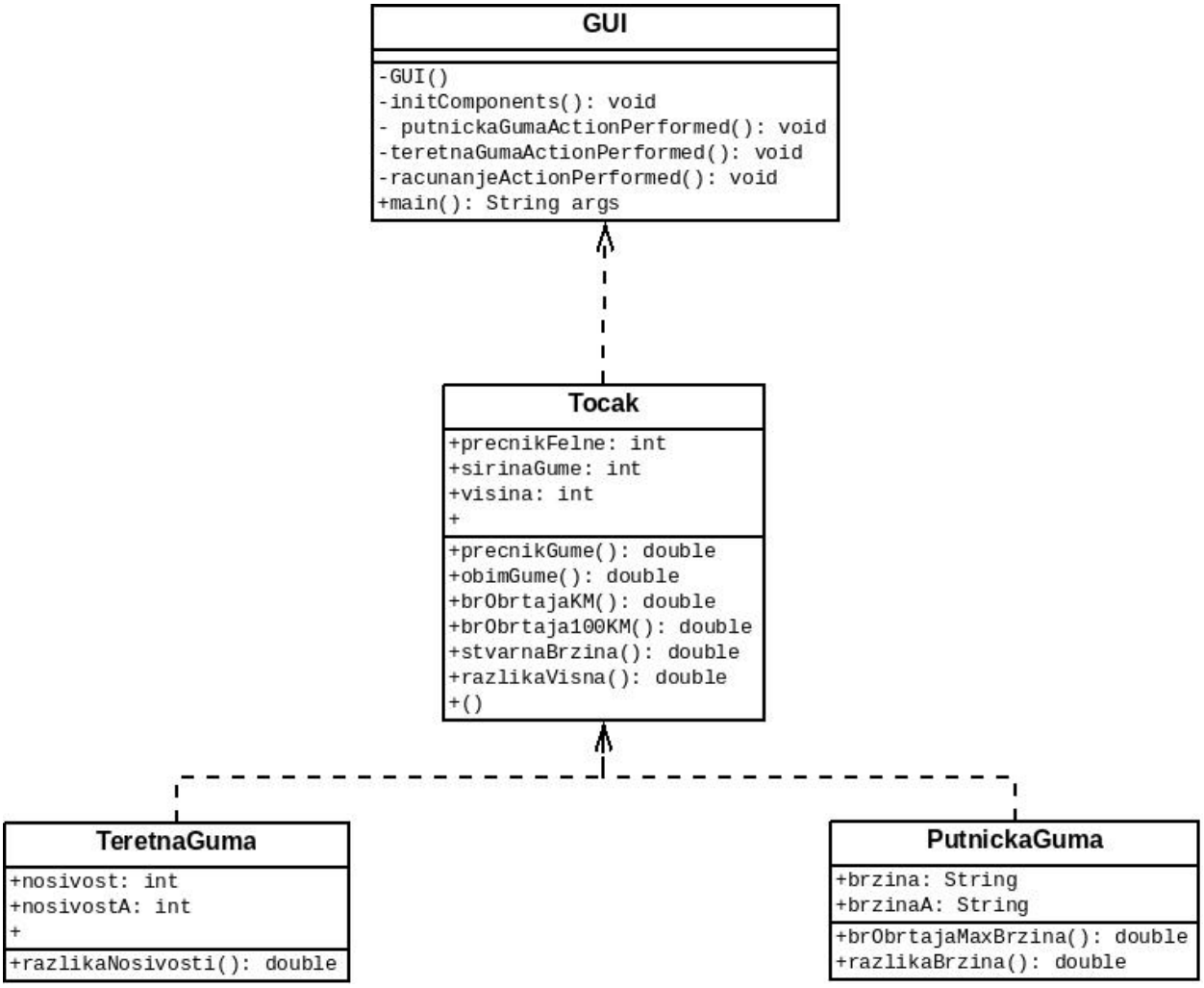
U ovom radu ce biti opisano kako se u programskom jeziku JAVA konstruiše aplikacija, koja radi kao kalkulator za uporedjivanje parametara različitih auto guma, tj. poredi i izračunava parametere originalne i alternativne gume. Biće opisan postupak za pravljenje GUI, grafičkog interfejsa u razvojnom okruženju NetBeans IDE 8.2, a uz to i objasnjenje svakog od izračunatih parametara gume. Tržište auto guma je uvek aktuelno, zbog sve veceg broja vozila na putevima, a parametri pneumatika računati ovom aplikacijom bi trebalo da budu znatno bolji uskoro, zbog pojave guma sa konstrukcijom, tj. bez vazduha. Aplikacija izračunava parametere pneumatika putničkih i teretnih vozila.




Na početku programa korisnik treba da izabere da li se radi o gumi namenjenoj putničkom ili teretnom vozilu, a zavisno od tipa vozila ponuđeni su različiti opsezi za parametre. Sada se unose dimenzije originalnog pneumatika. U prvu padajuću listu treba uneti prečnik felne u inčima, u drugu širinu gume izraženu u mm, u treću padajuću listu se unosi profil gume (procenat širine gume), a u zavisnosti od tipa vozila četvrta padajuća lista će sadržati listu za unos indeksa nosivosti, ako se radi o teretnom vozilu, odnosno listu brzinskih oznaka, ako se radi o putničkom vozilu. Klikom na dugme “Računanje”, aplikacija računa sledeće vrednosti: prečnik gume, obim gume, broj obrtaja po km, broj obrtaja u sekundi pri brzini od 100km/h, broj obrtaja u sekundi pri maksimalnoj brzini. A rezultati koje dobijamo kao poređenje originalne I alternativne gume su: stvarna brzina kada je na brzinometru 100km/h, razlika u prečniku originalne I alternativne gume I teret koji moze da izdži sa datim indeksom nosivosti.

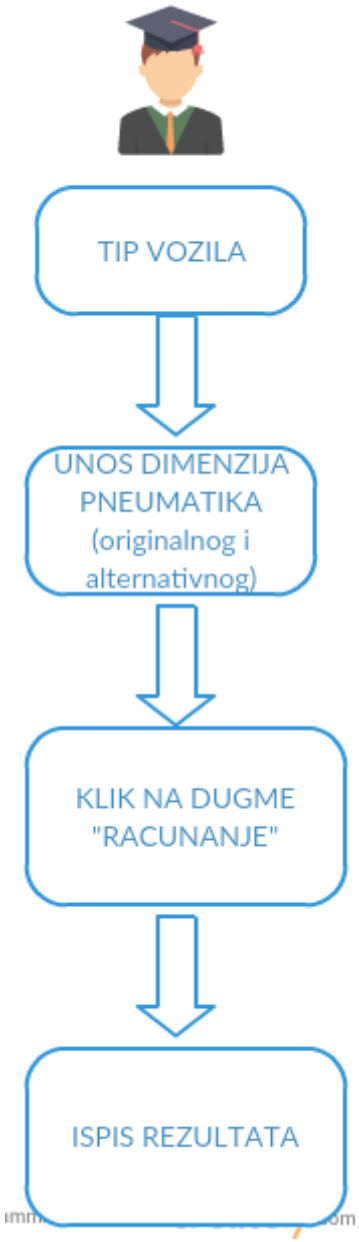
## 2.DIJAGRAM KLASA

Dijagram sa slike, prikazuje strukturu projekta, međusobni odnos klasa, njihovih objekata i metoda.




Na dijagramu se nalaze četiri klase, Tocak, kao zajednička klasa i dve podklase putnickaGuma I teretnaGuma i GUI main klasa.

## 3. *USE-CASE* DIJAGRAM



## 4. IMPLEMENTACIJA KODA

Klasa Tocak sadrži dve podklase, TeretnaGuma i PutnickaGuma. Promenljive ove klase su precnikFelne, sirinaGume i visina. Metode klase Tocak su precnikGume(), koja kao argumente prima pomenute promenljive, obimGume() koja prima prečnik gume kao argument, metoda brObrtajaKM() kao argument uzima obim gume, a metoda brObrtaja100KM takođe uzima kao argument obim gume. Metodom stvarnaBrzina() se rešava se ispravlja greška koju pokazuje brzinometar originalne gume, pa kao ulaze uzima obime originalne i alternativne gume i direktnom proporcijom se računa realna brzina vozila. Metoda razlikaVisina() računa razliku prečnika.

```
public class Tocak {
    public int precnikFelne, sirinaGume, visina; //visina je procenat sirine gume

    public final double inchToCMeter = 2.54;

    public Tocak(int precnikFelne, int sirinaGume, int visina)
    {
        this.precnikFelne = precnikFelne;
        this.sirinaGume = sirinaGume;
        this.visina = visina;
    }

    public double precnikGume(double precnikFelne, double sirinaGume, double visina)
    {
        return (precnikFelne * inchToCMeter) + (2 * sirinaGume * (double) visina/1000);
        //PAZI NA DOUBLE
    }
    public double obimGume(double precnikGume)
    {
        return precnikGume * Math.PI;
    }
    public double brObrtajaKM (double obimGume)
    {
        return (double) Math.pow(10,5)/obimGume;
    }

    public double brObrtaja100kmh (double obimGume)
    {
        return 100/3.6*100 / obimGume;
    }

    //Alternativna guma
    public int precnikFelneA, sirinaGumeA, visinaA;
    //Kada brzinometar pokazuje 100km/h stvarna brzina vozila je
```

U potklasi TeretnaGuma se nalaze dva niza, koja se odnose na nosivost i promenljiva „nosivost“, pomoću ova dva niza i promenljive se indeks nosivosti pretvara u teret koji pneumatik može da izdrži. U ovoj potklasi se nalazi metoda razlikaNosivosti(), koja kao argumente prima pomenuti teret dve gume.

```
public class TeretnaGuma extends Tocak{
    public int nosivost, nosivostA;
    public int precnikFelne, sirinaGume, visina;

    public TeretnaGuma (int precnikFelne, int sirinaGume, int visina, int nosivost, int nosivostA)
    {
        super(precnikFelne, sirinaGume, visina);
        this.nosivost = nosivost;
        this.nosivostA = nosivostA;
    }


    public static final int[] indexNosivost = {62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,
        88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
        115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125,127};

    public static final int[] nosivostKG = {265,272,280,290,300,307,315,325,335,345,355,365,375,387,400,412,425
        545,560,580,600,615,630,650,670,690,710,730,750,775,800,825,850,875,900,925,950,975,1000,1030,1060,1090
        1215,1250,1285,1320,1360,1400,1450,1500,1550,1600,1650,1700};

    public static final double[] BrzinaBroj = {140,150,160,170,180,190,200,210,240,255,270,300};
       public int vratiIndex(int nosivost, int[] indexNosivosti)
    {
        for(int i=0;i<indexNosivosti.length;i++)
        {
            if(indexNosivosti[i]==nosivost)
                return i;
        }
        return 0;
    }

    public int razlikaNosivosti(int nosivost, int nosivostA)
    {
        return nosivostKG[vratiIndex(nosivostA,indexNosivost)]-nosivostKG[vratiIndex(nosivost,indexNosivost)];
    }


}
```

Potklasa PutnickaGuma sadrži promenljive brzina i brzinaA, koje su tipa String, pa se kao i u slučaju nosivosti, pomoću dva niza BrzinaSlovo i BrzinaBroj, dobija vrednost maksimalne brzine, koju pneumatik može da izdrži. Metoda brObrtajaMaxBrzina() uzima pomenutu promenljivu tipa String i vraća broj obrtaja pri maksimalnoj brzini. Metoda razlikaBrzina() kao argumente uzima stringove, koje preko dva niza pretvori u brojčanu vrednost i izračuna njihovu razliku.

```
public class PutnickaGuma extends Tocak {
    public String brzina;
    public String brzinaA;
    public int precnikFelne, sirinaGume, visina;

    public PutnickaGuma (int precnikFelne, int sirinaGume, int visina, String brzina, String brzinaA)
    {
        super(precnikFelne, sirinaGume, visina);
        this.brzina = brzina;
        this.brzinaA = brzinaA;
        //super.precnikGume(precnikFelne, sirinaGume, visina);
        //super.obimGume( super.precnikGume(precnikFelne, sirinaGume, visina));
        //super.brObrtajaKM(super.obimGume(super.precnikGume(precnikFelne, sirinaGume, visina)));
        //super.brObrtaja100kmh(super.obimGume(super.precnikGume(precnikFelne, sirinaGume, visina)));
    }

    public static final String[] BrzinaSlovo = {"N","P","Q","R","S","T","U","H","V","Z","W","Y"};

    public static final double[] BrzinaBroj = {140,150,160,170,180,190,200,210,240,255,270,300};
       public int vratiIndex(String brzina, String[] brzinaSlovo)
    {
        for(int i=0;i<brzinaSlovo.length;i++)
        {
            if(brzinaSlovo[i].equals(brzina))
                return i;
        }
        return 0;
    }

    public double brObrtajaMaxBrzina (String brzina, double obimGume)
    {
        return BrzinaBroj[vratiIndex(brzina,BrzinaSlovo)]/3.6*100 / obimGume;
    }
    public double razlikaBrzina(String brzina, String brzinaA)
    {
        return (BrzinaBroj[vratiIndex(brzinaA,BrzinaSlovo)]-BrzinaBroj[vratiIndex(brzina,BrzinaSlovo)]);
    }
}
```

U GUI-u se nalazi dugme koje se naziva dugme „Putnicka“, pritiskom na to dugme korisnik može da bira odgovarajuće opsege parametara ovog tipa vozila. Svaki padajući meni je objekat tipa comboBox, koji poprima dobije parametre zavisno od izabranog dugmeta, Putnicko ili Teretno. U okviru metode dugmeta Racunanje vrše se ključne operacije za rad aplikacije.Prvo se uzimaju podaci, koje je korisnik odabrao iz comboBoxa, pa se instanciranjem odgovarajućih klasa i pozivanjem njihovih metoda nad objektima, kao i kastovanjem podataka vrše sva potrebna izračunavanja i prikazuju korisniku u poljima za tekst.

```
private void putnickaGumaActionPerformed(java.awt.event.ActionEvent evt) {
    if(putnickaGuma.isEnabled())
    {String[] niz = {"12","13","14","15","16","17","18","19","20","21","22",
       "23","24","25","26"};
     ComboBoxModel ppe= new DefaultComboBoxModel( niz );
     jComboBox1.setModel(ppe);
     ////////////////////////////////////
     jComboBox1.setVisible(true);
     String[] niz1 = {"145", "155","165","175", "185" ,"195", "205"};
     ComboBoxModel ppe1= new DefaultComboBoxModel( niz1 );
     jComboBox2.setModel(ppe1);
     jComboBox2.setVisible(true);
     ////////////////////////////////////////

     String[] niz2={"20","25","30","35","40","45","50","55","60"};
     ComboBoxModel ppe2= new DefaultComboBoxModel( niz2 );
     jComboBox3.setModel(ppe2);
     jComboBox3.setVisible(true);
     ////////////////////////////////////////////////////////////

     //String[] niz3 = {"N","P","Q","R","S","T","U","H","V","Z","W","Y"};
     //ComboBoxModel ppe3= new DefaultComboBoxModel( niz3 );
     //jComboBox4.setModel(ppe2);
     //jComboBox4.setVisible(true);
     //int precnik = Integer.parseInt(jComboBox1.getActionCommand());
     ////////////////////////////////////////////////////////////
     String[] niz3 = {"12","13","14","15","16","17","18","19","20","21","22",
       "23","24","25","26"};
     ComboBoxModel ppe3= new DefaultComboBoxModel( niz3 );
     jComboBox7.setModel(ppe3);
     jComboBox7.setVisible(true);
     ////////////////////////////////////////////////

     String[] niz4 = { "215", "225", "235", "245", "255", "265", "275", "285", "295", "305", "315", "325", "335", "345", "355"};

     ComboBoxModel ppe4= new DefaultComboBoxModel( niz4 );
     jComboBox5.setModel(ppe4);
     jComboBox5.setVisible(true);
     ------------------------------------------------------------
```

```
private void racunanjeActionPerformed(java.awt.event.ActionEvent evt) {
    racunanje.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){
         String precnikF = (String)jComboBox1.getSelectedItem();
         String sirinaG = (String)jComboBox2.getSelectedItem();
         String visinaG = (String)jComboBox3.getSelectedItem();
         int precnik = Integer.parseInt(precnikF);
         int sirina = Integer.parseInt(sirinaG);
         int visina = Integer.parseInt(visinaG);
         double precnik1 = Double.parseDouble(precnikF);
         double sirina1 = Double.parseDouble(sirinaG);
         double visina1 = Double.parseDouble(visinaG);
         Tocak tocak = new Tocak(precnik,sirina,visina);
         DecimalFormat df = new DecimalFormat("#.00");
         double precnikT = tocak.precnikGume(precnik1, sirina1, visina1);
         String precnikd = df.format(precnikT);
         precnikGume.setText(precnikd);
         String zObim = precnikGume.getText();
         double obim = Double.parseDouble(zObim);
         double Obim = tocak.obimGume(obim);
         String obimd = df.format(Obim);
         obimGume.setText(obimd);
         String gObim = obimGume.getText();
         double gObim1 = Double.parseDouble(gObim);
         double kMH = tocak.brObrtajaKM(gObim1);
         String kMh = df.format(kMH);
         brojKM.setText(kMh);
         double bObrtaj = tocak.brObrtaja100kmh(gObim1);
         String brojObrtaja = df.format(bObrtaj);
         obrtaji.setText(brojObrtaja);
```

## 5. ZAKLJUČAK

Jedino što razdvaja automobil od podloge je pneumatik, pa je uskladjivanje parametara alternativne gume sa originalnom važno, zbog bezbednosti I udobnosti. Ovakva aplikacija je primenljiva u vulkanizerskim radnjama, gde bi se uz još informacija o karakteristikama gume, mogla opisati jasna slika o ponasanju vozila sa pneumaticima različitih dimenzija. Korišćenje funkcionalnog grafičkog interfejsa je moguće bez ikakvog znanja iz oblasti programiranja, pa je ova aplikacija dostupna prosečnom korisniku.