

Univerzitet u Kragujevcu
Fakultet inženjerskih nauka



Dokumentacija o projektu: Servis gitare

Ekspertski sistemi

Student:

Jovan Bogdanović 609/2017

Predmetni profesor:

Vesna Ranković

Predmetni saradnik:

Tijana Šušteršič

Sadržaj:

1. Uvod	2
2. Izgled grafičkog interfejsa	3
3. Opis koda	5
3.1. Ulazi programa	5
3.2. Ispis svih pravila	6
3.3. Opis koda grafičkog interfejsa	12
4. Zaključak	21
5. Literatura	22

1. Uvod

Ekspertski sistemi (takođe, **ekspertni sistemi**) su inteligentni računarski programi kojima se emulira rešavanje problema na način na koji to čine eksperti i predstavljaju jednu od najznačajnijih oblasti istraživanja veštačke inteligencije.

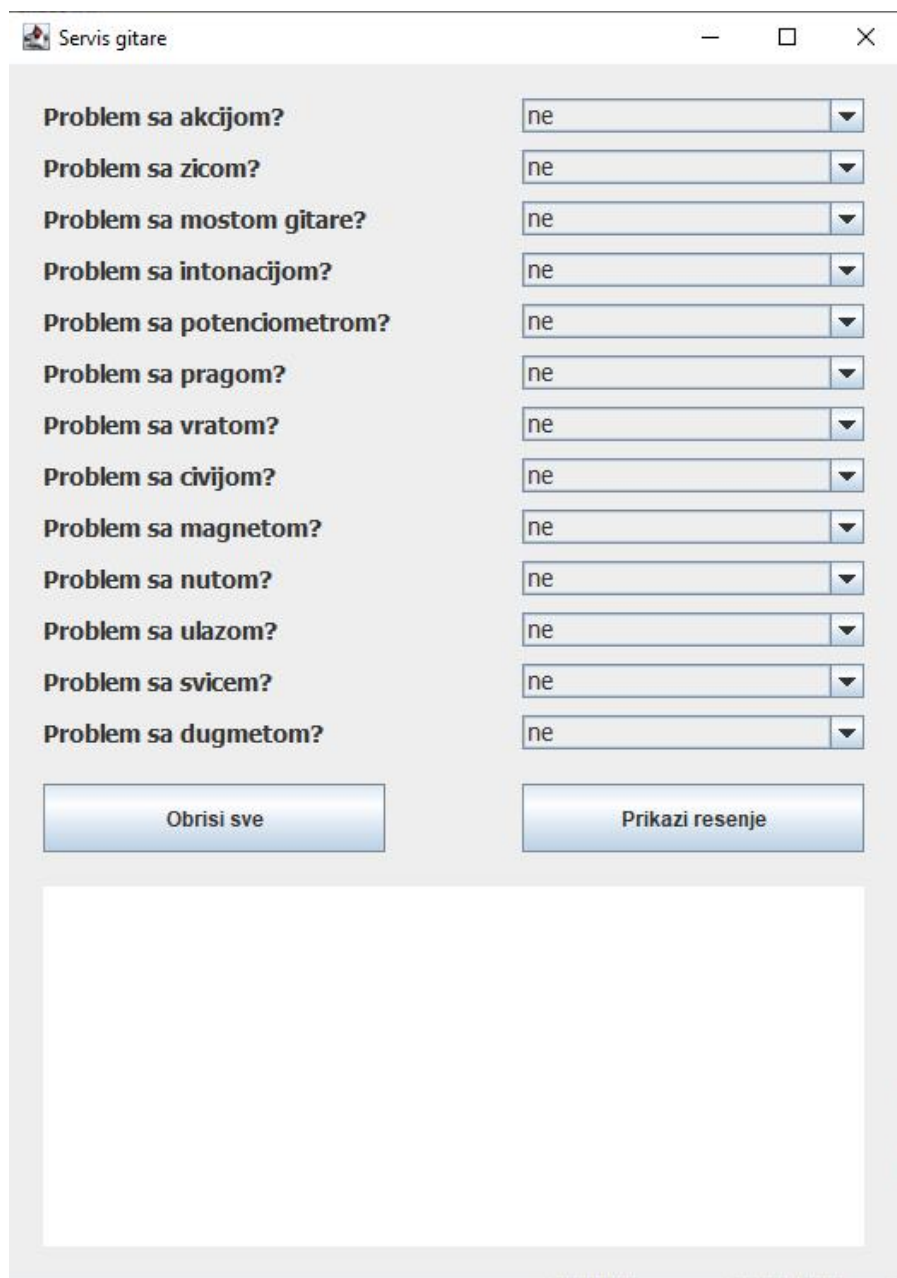
Ekspertski sistemi rešavaju realne probleme iz različitih oblasti, koji bi inače zahtevali ljudsku ekspertizu. Cilj je da uvek računarski program daje korektne odgovore, u datoj oblasti, ne lošije od eksperta, ali je to teško dostižno. Zato se postavlja manje ambiciozan cilj, traži se da sistem pruži pomoć u odlučivanju.

U nastavku rada biće prikazan ekspertski sistem koji služi za popravku električne gitare. Sistem nije idealan niti temeljan ali može pružiti pomoć početnicima.

2. Izgled grafičkog interfejsa

Iz padajućih menija se biraju problemi koji nastaju na određenim delovima gitare.

Postoje dva dugmeta, jedno za prikaz rešenja za zadate probleme koje kada se pritisne daje rezultat u prostoru za ispis teksta (na dnu programa). I drugo dugme koje služi za resetovanje svih stavki padajućih menija i brisanje teksta koji je dobijen kao rezultat.



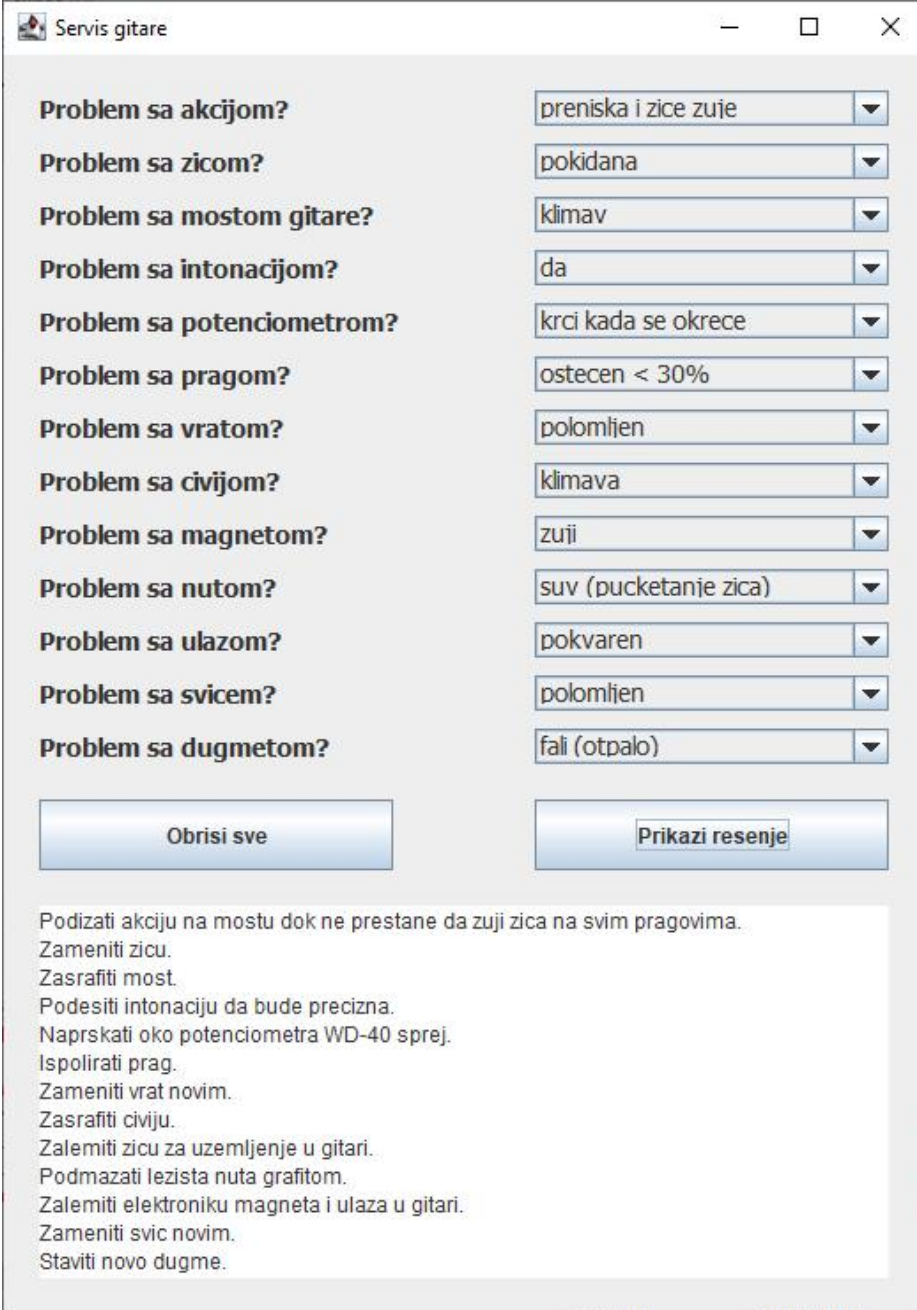
The screenshot shows a window titled "Servis gitare" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a list of 12 guitar-related problems, each followed by a dropdown menu. All dropdown menus currently display the word "ne". Below the list are two buttons: "Obrisi sve" (Clear all) and "Prikazi resenje" (Show solution). At the bottom of the window is a large, empty rectangular area for displaying the solution.

Problem	Value
Problem sa akcijom?	ne
Problem sa zicom?	ne
Problem sa mostom gitare?	ne
Problem sa intonacijom?	ne
Problem sa potencijometrom?	ne
Problem sa pragom?	ne
Problem sa vratom?	ne
Problem sa civijom?	ne
Problem sa magnetom?	ne
Problem sa nutom?	ne
Problem sa ulazom?	ne
Problem sa svicem?	ne
Problem sa dugmetom?	ne

Buttons: **Obrisi sve**, **Prikazi resenje**

Izgled grafičkog interfejsa pri startovanju aplikacije

Sledi primer unešenih problema i datih rešenja:



Problem sa akcijom?	preniska i zice zuje
Problem sa zicom?	pokidana
Problem sa mostom gitare?	klimav
Problem sa intonacijom?	da
Problem sa potencijetrom?	kraci kada se okreće
Problem sa pragom?	ostecen < 30%
Problem sa vratom?	polomljen
Problem sa civijom?	klimava
Problem sa magnetom?	zuji
Problem sa nutom?	suv (pucketanje zica)
Problem sa ulazom?	pokvaren
Problem sa svicem?	polomljen
Problem sa dugmetom?	fali (otpalo)

Obrisi sve Prikazi resenje

Podizati akciju na mostu dok ne prestane da zuji zica na svim pragovima.
Zameniti zicu.
Zasrafiti most.
Podesiti intonaciju da bude precizna.
Naprsinati oko potencijetra WD-40 sprej.
Ispolirati prag.
Zameniti vrat novim.
Zasrafiti civiju.
Zalemiti zicu za uzemljenje u gitari.
Podmazati lezista nuta grafitom.
Zalemiti elektroniku magneta i ulaza u gitari.
Zameniti svic novim.
Staviti novo dugme.

Primer rada aplikacije

3. Opis koda

Program se sastoji od:

- klase **Gitara.java** u kojoj su definisani delovi gitare, akcije i njihovi odgovarajući geteri i seteri,
- drools fajla **Sample.drl** u kome su definisana pravila na osnovu kojih radi program i
- klase **GUI.java** u kojoj je implementiran kompletan kod grafickog interfejsa.

Kod je kreiran u okruženju Eclipse Java 2019-06.

3.1. Ulazi programa

Sledi deo koda klase Gitara, koji predstavlja ulaze programa:

```
public class Gitara {  
    private String akcija;           // previsoka, preniska, dobra  
    private String zica;             // pokidana, zardjala, zuji, dobra  
    private String most;             // klimav, hrapav, dobar  
    private String intonacija;       // losa, dobra  
    private String potencijometar;   // polomljen, krci, dobar  
    private String prag;             // ostecen>30%, ostecen<30%, dobar  
    private String vrat;             // kriv, polomljen, prljav, suv, dobar  
    private String civija;           // pokvarena, klimava, dobra  
    private String magnet;           // necujan, zuji, tih, glasen, pokvaren, dobar  
    private String nut;              // polomljen, suv, dobar  
    private String ulaz;             // klimav, pokvaren, dobar  
    private String svic;             // polomljen, pokvaren, dobar  
    private String dugme;            // klimavo, fali, dobro
```

pod komentarima su navedena moguća stanja ulaza.

3.2. Ispis svih pravila

Pravilo 1: ulazi koji aktiviraju pravilo su žica (zuji) i akcija (preniska).

```
rule "1. Zica previse blizu vrata i zuji"
when
    o:Gitara(zica == "zuji", akcija == "preniska")
then
    o.setZica("dobra");
    o.setAkcija("dobra");
    o.setAkcija_akcija1("Podizati akciju na mostu dok ne prestane da zuji zica na
svim pragovima.");
end
```

Pravilo 2: ulaz koji aktivira pravilo je akcija (previsoka).

```
rule "2. Zica previse odajena od vrata"
when
    o:Gitara(akcija == "previsoka")
then
    o.setAkcija("dobra");
    o.setAkcija_akcija2("Spustati akciju na mostu koliko god moze a da ne dodje do
zujanja zice.");
end
```

Pravilo 3: ulazi koji aktiviraju pravilo mogu biti žica (pokidana) ili žica (zardjala).

```
rule "3. Pokidana ili zardjala zica"
when
    o:Gitara(zica == "pokidana" || zica == "zardjala")
then
    o.setZica("dobra");
    o.setZica_akcija1("Zameniti zicu.");
end
```

Pravilo 4: ulaz koji aktivira pravilo je most (klimav).

```
rule "4. Klimav most"
when
    o:Gitara(most == "klimav")
then
    o.setMost("dobar");
    o.setMost_akcija1("Zasrafiti most.");
end
```

Pravilo 5: ulaz koji aktivira pravilo je most (hrapav).

```
rule "5. Most hrapav, cesto pucaju zice"
when
    o:Gitara(most == "hrapav")
then
    o.setMost("dobar");
```

```
        o.setMost_akcija2("Osmirglati most.");  
end
```

Pravilo 6: ulaz koji aktivira pravilo je intonacija (loša).

```
rule "6. Losa intonacija"  
when  
    o:Gitara(intonacija == "losa")  
then  
    o.setIntonacija("dobra");  
    o.setIntonacija_akcija1("Podesiti intonaciju da bude precizna.");  
end
```

Pravilo 7: ulaz koji aktivira pravilo je potencijometar (polomljen).

```
rule "7. Polomljen potencijometar"  
when  
    o:Gitara(potencijometar == "polomljen")  
then  
    o.setPotencijometar("dobar");  
    o.setPotencijometar_akcija1("Zameniti potencijometar novim.");  
end
```

Pravilo 8: ulaz koji aktivira pravilo je potencijometar (krči).

```
rule "8. Krči potencijometar kada se okreće"  
when  
    o:Gitara(potencijometar == "krči")  
then  
    o.setPotencijometar("dobar");  
    o.setPotencijometar_akcija2("Naprskati oko potencijometra WD-40 sprej.");  
end
```

Pravilo 9: ulaz koji aktivira pravilo je prag (oštećen>30%).

```
rule "9. Prag ostecen preko 30%"  
when  
    o:Gitara(prag == "ostecen>30%")  
then  
    o.setPrag("dobar");  
    o.setPrag_akcija1("Zameniti prag novim.");  
end
```

Pravilo 10: ulaz koji aktivira pravilo je prag (oštećen<30%).

```
rule "10. Prag ostecen ispod 30%"  
when  
    o:Gitara(prag == "ostecen<30%")  
then  
    o.setPrag("dobar");  
    o.setPrag_akcija2("Ispolirati prag.");  
end
```


Pravilo 11: ulazi koji aktiviraju pravilo su akcija (previsoka) i vrat (kriv).

```
rule "11. Kriv vrat i zice previse udaljene od vrata"
when
    o:Gitara(akcija == "previsoka", vrat == "kriv")
then
    o.setVrat("dobar");
    o.setAkcija("dobra");
    o.setVrat_akcija1("Zasrafiti sipku u vratu.");
end
```

Pravilo 12: ulaz koji aktivira pravilo je vrat (polomljen).

```
rule "12. olomljen vrat"
when
    o:Gitara(vrat == "polomljen")
then
    o.setVrat("dobar");
    o.setVrat_akcija2("Zameniti vrat novim.");
end
```

Pravilo 13: ulaz koji aktivira pravilo je vrat (prljav).

```
rule "13. Prljav vrat"
when
    o:Gitara(vrat == "prljav")
then
    o.setVrat("dobar");
    o.setVrat_akcija3("Ocistiti vrat.");
end
```

Pravilo 14: ulaz koji aktivira pravilo je vrat (suv).

```
rule "14. Ispucao vrat, suv, ostecen"
when
    o:Gitara(vrat == "suv")
then
    o.setVrat("dobar");
    o.setVrat_akcija4("Podmazati vrat uljem za odrzavanje.");
end
```

Pravilo 15: ulaz koji aktivira pravilo je čivija (klimava).

```
rule "15. Klimava civija"
when
    o:Gitara(civija == "klimava")
then
    o.setCivija("dobra");
    o.setCivija_akcija1("Zasrafiti civiju.");
end
```

Pravilo 16: ulaz koji aktivira pravilo je čivija (pokvarena).

```
rule "16. Pokvarena civija, zica se ne moze namotati"
when
    o:Gitara(civija == "pokvarena")
then
    o.setCivija("dobra");
    o.setCivija_akcija2("Zameniti civiju novom.");
end
```

Pravilo 17: ulaz koji aktivira pravilo je magnet (nečujan).

```
rule "17. Magnet ne daje zvuk"
when
    o:Gitara(magnet == "necujan")
then
    o.setMagnet("dobar");
    o.setMagnet_akcija1("Zalemiti elektroniku magneta u gitari.");
end
```

Pravilo 18: ulaz koji aktivira pravilo je magnet (zuji).

```
rule "18. Magnet zuji konstantno"
when
    o:Gitara(magnet == "zuji")
then
    o.setMagnet("dobar");
    o.setMagnet_akcija2("Zalemiti zicu za uzemljenje u gitari.");
end
```

Pravilo 19: ulaz koji aktivira pravilo je magnet (tih).

```
rule "19. Magnet se cuje preslabo"
when
    o:Gitara(magnet == "tih")
then
    o.setMagnet("dobar");
    o.setMagnet_akcija3("Odsrafljivati magnet dok se ne dobije zeljena jacinatona.");
end
```

Pravilo 20: ulaz koji aktivira pravilo je magnet (glasen).

```
rule "20. Magnet se cuje prejako"
when
    o:Gitara(magnet == "glasen")
then
    o.setMagnet("dobar");
    o.setMagnet_akcija4("Zasrafljivati magnet dok se ne dobije zeljena jacinatona.");
end
```

Pravilo 21: ulaz koji aktivira pravilo je magnet (pokvaren).

```
rule "21. Magnet ne radi"
when
    o:Gitara(magnet == "pokvaren")
then
    o.setMagnet("dobar");
    o.setMagnet_akcija5("Zameniti magnet novim.");
end
```

Pravilo 22: ulaz koji aktivira pravilo je nut (suv).

```
rule "22. Nut suv, zice pucketaju kada se dotezu"
when
    o:Gitara(nut == "suv")
then
    o.setNut("dobar");
    o.setNut_akcija1("Podmazati lezista nuta grafitom.");
end
```

Pravilo 23: ulaz koji aktivira pravilo je nut (polomljen).

```
rule "23. Nut polomljen"
when
    o:Gitara(nut == "polomljen")
then
    o.setNut("dobar");
    o.setNut_akcija2("Zameniti nut novim.");
end
```

Pravilo 24: ulaz koji aktivira pravilo je ulaz (klimav).

```
rule "24. Ulaz klimav"
when
    o:Gitara(ulaz == "klimav")
then
    o.setUlaz("dobar");
    o.setUlaz_akcija1("Zasrafiti ulaz.");
end
```

Pravilo 25: ulaz koji aktivira pravilo je ulaz (pokvaren).

```
rule "25. Magnet ne daje zvuk, ulaz pokvaren"
when
    o:Gitara(magnet == "necujan", ulaz == "pokvaren")
then
    o.setUlaz("dobar");
    o.setMagnet("dobar");
    o.setUlaz_akcija2("Zalemiti elektroniku magneta i ulaza u gitari.");
end
```

Pravilo 26: ulaz koji aktivira pravilo je svič (polomljen).

```
rule "26. Svic polomljen"
when
    o:Gitara(svic == "polomljen")
then
    o.setSvic("dobar");
    o.setSvic_akcija1("Zameniti svic novim.");
end
```

Pravilo 27: ulaz koji aktivira pravilo je svič (pokvaren).

```
rule "27. Svic pokvaren, ne menja rad magnetu"
when
    o:Gitara(svic == "pokvaren")
then
    o.setSvic("dobar");
    o.setSvic_akcija2("Zalemiti elektroniku svica u gitari.");
end
```

Pravilo 28: ulaz koji aktivira pravilo je dugme (klimavo).

```
rule "28. Dugme klimavo, gitara spada sa kajisa"
when
    o:Gitara(dugme == "klimavo")
then
    o.setDugme("dobar");
    o.setDugme_akcija1("Zasrafiti dugme.");
end
```

Pravilo 29: ulaz koji aktivira pravilo je dugme (fali).

```
rule "29. Dugme fali"
when
    o:Gitara(dugme == "fali")
then
    o.setDugme("dobro");
    o.setDugme_akcija2("Staviti novo dugme.");
end
```

3.3. Opis koda grafičkog interfejsa

Sledi inicijalizacija svih promenljivih: naziva problema, padajućih menija, dugmadi, prostora za ispis rešenja...

```
public Gitara gitara;
public JFrame frame;
public KieSession kSession;
public JPanel panel = new JPanel();
public String resi = "";

public JLabel lblAkcija = new JLabel("Problem sa akcijom?");
public String[] akcija = {"ne", "preniska i zice zuje", "previsoka i krivi vrat"};
final JComboBox akcijaBox = new JComboBox(akcija);

public JLabel lblZica = new JLabel("Problem sa zicom?");
public String[] zica = {"ne", "pokidana", "zardjala"};
final JComboBox zicaBox = new JComboBox(zica);

public JLabel lblMost = new JLabel("Problem sa mostom gitare?");
public String[] most = {"ne", "klimav", "hrapav (cesto pucaju zice)"};
final JComboBox mostBox = new JComboBox(most);

public JLabel lblIntonacija = new JLabel("Problem sa intonacijom?");
public String[] intonacija = {"ne", "da"};
final JComboBox intonacijaBox = new JComboBox(intonacija);

public JLabel lblPotencijometar = new JLabel("Problem sa potencijometrom?");
public String[] potencijometar = {"ne", "polomljen", "krci kada se okreće"};
final JComboBox potencijometarBox = new JComboBox(potencijometar);

public JLabel lblPrag = new JLabel("Problem sa pragom?");
public String[] prag = {"ne", "ostecen > 30%", "ostecen < 30%"};
final JComboBox pragBox = new JComboBox(prag);

public JLabel lblVrat = new JLabel("Problem sa vratom?");
public String[] vrat = {"ne", "kriv", "polomljen", "prljav", "suv i ostecen"};
final JComboBox vratBox = new JComboBox(vrat);

public JLabel lblCivija = new JLabel("Problem sa civijom?");
public String[] civija = {"ne", "klimava", "pokvarena (ne namotava zicu)"};
final JComboBox civijaBox = new JComboBox(civija);

public JLabel lblMagnet = new JLabel("Problem sa magnetom?");
public String[] magnet = {"ne", "necujan", "zuji", "tih", "glasen", "pokvaren"};
final JComboBox magnetBox = new JComboBox(magnet);

public JLabel lblNut = new JLabel("Problem sa nutom?");
public String[] nut = {"ne", "polomljen", "suv (pucketanje zica)"};
final JComboBox nutBox = new JComboBox(nut);

public JLabel lblUlaz = new JLabel("Problem sa ulazom?");
public String[] ulaz = {"ne", "klimav", "pokvaren"};
```

```
final JComboBox ulazBox = new JComboBox(ulaz);

public JLabel lblSvic = new JLabel("Problem sa svicem?");
public String[] svic = {"ne", "polomljen", "pokvaren (ne menja magnet)"};
final JComboBox svicBox = new JComboBox(svic);

public JLabel lblDugme = new JLabel("Problem sa dugmetom?");
public String[] dugme = {"ne", "klimavo", "fali (otpalo)"};
final JComboBox dugmeBox = new JComboBox(dugme);

public JButton resenje = new JButton("Prikazi resenje");
public JButton obrisi = new JButton("Obrisi sve");
public JTextArea textArea = new JTextArea();
```

Kreiranje prozora interfejsa i inicijalizacija:

```
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                GUI window = new GUI();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public GUI() {
    this.gitara = new Gitara();

    initialize();
}
```

Metoda Ks kojom se povezuje sa fajlom Sample.drl:

```
public void Ks(Gitara gitara) {
    try {
        KieServices ks = KieServices.Factory.get();
        KieContainer kContainer = ks.getKieClasspathContainer();
        KieSession kSession = kContainer.newKieSession("ksession-rules");
        FactHandle fh1;

        fh1 = kSession.insert(gitara);
        kSession.fireAllRules();
    }
}
```

```
        catch(Throwable t) {  
            t.printStackTrace();  
        }  
    }  
}
```

Položaji svih objekta koji se vide pri startovanju aplikacije:

```
frame = new JFrame();  
frame.setTitle("Servis gitare");  
frame.setBounds(400, 10, 540, 750);  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
frame.getContentPane().add(panel, BorderLayout.CENTER);  
panel.setLayout(null);  
  
lblAkcija.setFont(new Font("Tahoma", Font.BOLD, 14));  
lblAkcija.setBounds(20, 20, 300, 20);  
panel.add(lblAkcija);  
akcijaBox.setFont(new Font("Tahoma", Font.PLAIN, 14));  
akcijaBox.setBounds(300, 20, 200, 20);  
panel.add(akcijaBox);  
  
lblZica.setFont(new Font("Tahoma", Font.BOLD, 14));  
lblZica.setBounds(20, 50, 300, 20);  
panel.add(lblZica);  
zicaBox.setFont(new Font("Tahoma", Font.PLAIN, 14));  
zicaBox.setBounds(300, 50, 200, 20);  
panel.add(zicaBox);  
  
lblMost.setFont(new Font("Tahoma", Font.BOLD, 14));  
lblMost.setBounds(20, 80, 300, 20);  
panel.add(lblMost);  
mostBox.setFont(new Font("Tahoma", Font.PLAIN, 14));  
mostBox.setBounds(300, 80, 200, 20);  
panel.add(mostBox);  
  
lblIntonacija.setFont(new Font("Tahoma", Font.BOLD, 14));  
lblIntonacija.setBounds(20, 110, 300, 20);  
panel.add(lblIntonacija);  
intonacijaBox.setFont(new Font("Tahoma", Font.PLAIN, 14));  
intonacijaBox.setBounds(300, 110, 200, 20);  
panel.add(intonacijaBox);  
  
lblPotencijometar.setFont(new Font("Tahoma", Font.BOLD, 14));  
lblPotencijometar.setBounds(20, 140, 300, 20);  
panel.add(lblPotencijometar);  
potencijometarBox.setFont(new Font("Tahoma", Font.PLAIN, 14));  
potencijometarBox.setBounds(300, 140, 200, 20);  
panel.add(potencijometarBox);  
  
lblPrag.setFont(new Font("Tahoma", Font.BOLD, 14));  
lblPrag.setBounds(20, 170, 300, 20);
```

```
panel.add(lblPrag);
pragBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
pragBox.setBounds(300, 170, 200, 20);
panel.add(pragBox);

lblVrat.setFont(new Font("Tahoma", Font.BOLD, 14));
lblVrat.setBounds(20, 200, 300, 20);
panel.add(lblVrat);
vratBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
vratBox.setBounds(300, 200, 200, 20);
panel.add(vratBox);

lblCivija.setFont(new Font("Tahoma", Font.BOLD, 14));
lblCivija.setBounds(20, 230, 300, 20);
panel.add(lblCivija);
civijaBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
civijaBox.setBounds(300, 230, 200, 20);
panel.add(civijaBox);

lblMagnet.setFont(new Font("Tahoma", Font.BOLD, 14));
lblMagnet.setBounds(20, 260, 300, 20);
panel.add(lblMagnet);
magnetBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
magnetBox.setBounds(300, 260, 200, 20);
panel.add(magnetBox);

lblNut.setFont(new Font("Tahoma", Font.BOLD, 14));
lblNut.setBounds(20, 290, 300, 20);
panel.add(lblNut);
nutBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
nutBox.setBounds(300, 290, 200, 20);
panel.add(nutBox);

lblUlaz.setFont(new Font("Tahoma", Font.BOLD, 14));
lblUlaz.setBounds(20, 320, 300, 20);
panel.add(lblUlaz);
ulazBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
ulazBox.setBounds(300, 320, 200, 20);
panel.add(ulazBox);

lblSvic.setFont(new Font("Tahoma", Font.BOLD, 14));
lblSvic.setBounds(20, 350, 300, 20);
panel.add(lblSvic);
svicBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
svicBox.setBounds(300, 350, 200, 20);
panel.add(svicBox);

lblDugme.setFont(new Font("Tahoma", Font.BOLD, 14));
lblDugme.setBounds(20, 380, 300, 20);
panel.add(lblDugme);
dugmeBox.setFont(new Font("Tahoma", Font.PLAIN, 14));
dugmeBox.setBounds(300, 380, 200, 20);
panel.add(dugmeBox);
```



```

textArea.setEditable(false);
textArea.setBounds(20, 480, 480, 210);
panel.add(textArea);

resenje.setBounds(300, 420, 200, 40);
panel.add(resenje);

obrisi.setBounds(20, 420, 200, 40);
panel.add(obrisi);

```

Sledi kod akcije koja se dešava kada je pritisnuto dugme Prikaži rešenje u kojem se nalaze svi uslovi na osnovu kojih znamo koje opcije su izabrane iz padajućih menija, u svakom uslovu upisujemo dati ulaz i njegovu osobinu, potom koristimo metodu za prolaženje kroz sva pravila Ks i konačno upisujemo odgovarajuće rešenje za svaki uslov u string reši koji se dodeljuje objektu TextArea.

```

resenje.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        resi = "";
        // akcija
        if(akcijaBox.getSelectedItem().toString() == "ne" ) {
            gitara.setAkcija("dobra");
        }
        if(akcijaBox.getSelectedItem().toString() == "preniska i zice zuje") {
            gitara.setAkcija("preniska");
            gitara.setZica("zuji");
            Ks(gitara);
            resi += gitara.getAkcija_akcija1()+"\n";
        }
        if(akcijaBox.getSelectedItem().toString() == "previsoka i krivi vrat") {
            gitara.setAkcija("previsoka");
            Ks(gitara);
            resi += gitara.getAkcija_akcija2()+"\n";
        }
        // zica
        if(zicaBox.getSelectedItem().toString() == "ne") {
            gitara.setZica("dobra");
        }
        if(zicaBox.getSelectedItem().toString() == "pokidana") {
            gitara.setZica("pokidana");
            Ks(gitara);
            resi += gitara.getZica_akcija1()+"\n";
        }
        if(zicaBox.getSelectedItem().toString() == "zardjala") {
            gitara.setZica("zardjala");
            Ks(gitara);
            resi += gitara.getZica_akcija1()+"\n";
        }
    }
}

```

```

//most
if(mostBox.getSelectedItem().toString() == "ne") {
    gitara.setMost("dobar");
}
if(mostBox.getSelectedItem().toString() == "klimav") {
    gitara.setMost("klimav");
    Ks(gitara);
    resi += gitara.getMost_akcija1()+"\n";
}
if(mostBox.getSelectedItem().toString() == "hrapav (cesto pucaju zice)")
{
    gitara.setMost("hrapav");
    Ks(gitara);
    resi += gitara.getMost_akcija2()+"\n";
}
//intonacija
if(intonacijaBox.getSelectedItem().toString() == "ne") {
    gitara.setIntonacija("dobra");
}
if(intonacijaBox.getSelectedItem().toString() == "da") {
    gitara.setIntonacija("losa");
    Ks(gitara);
    resi += gitara.getIntonacija_akcija1()+"\n";
}
//potencijometar
if(potencijometarBox.getSelectedItem().toString() == "ne") {
    gitara.setPotencijometar("dobar");
}
if(potencijometarBox.getSelectedItem().toString() == "polomljen") {
    gitara.setPotencijometar("polomljen");
    Ks(gitara);
    resi += gitara.getPotencijometar_akcija1()+"\n";
}
if(potencijometarBox.getSelectedItem().toString() == "kraci kada se
okrece") {
    gitara.setPotencijometar("kraci");
    Ks(gitara);
    resi += gitara.getPotencijometar_akcija2()+"\n";
}
//prag
if(pragBox.getSelectedItem().toString() == "ne") {
    gitara.setPrag("dobar");
}
if(pragBox.getSelectedItem().toString() == "ostecen > 30%") {
    gitara.setPrag("ostecen>30%");
    Ks(gitara);
    resi += gitara.getPrag_akcija1()+"\n";
}
if(pragBox.getSelectedItem().toString() == "ostecen < 30%") {
    gitara.setPrag("ostecen<30%");
    Ks(gitara);
    resi += gitara.getPrag_akcija2()+"\n";
}
//vrat

```

```

    if(vratBox.getSelectedItem().toString() == "ne") {
        gitara.setVrat("dobar");
    }
    if(vratBox.getSelectedItem().toString() == "kriv") {
        gitara.setVrat("kriv");
        gitara.setAkcija("previsoka");
        Ks(gitara);
        resi += gitara.getVrat_akcija1()+"\n";
    }
    if(vratBox.getSelectedItem().toString() == "polomljen") {
        gitara.setVrat("polomljen");
        Ks(gitara);
        resi += gitara.getVrat_akcija2()+"\n";
    }
    if(vratBox.getSelectedItem().toString() == "prljav") {
        gitara.setVrat("prljav");
        Ks(gitara);
        resi += gitara.getVrat_akcija3()+"\n";
    }
    if(vratBox.getSelectedItem().toString() == "suv i ostecen") {
        gitara.setVrat("suv");
        Ks(gitara);
        resi += gitara.getVrat_akcija4()+"\n";
    }
    //civija
    if(civijaBox.getSelectedItem().toString() == "ne") {
        gitara.setCivija("dobra");
    }
    if(civijaBox.getSelectedItem().toString() == "klimava") {
        gitara.setCivija("klimava");
        Ks(gitara);
        resi += gitara.getCivija_akcija1()+"\n";
    }
    if(civijaBox.getSelectedItem().toString() == "pokvarena (ne namotava
zicu)") {
        gitara.setCivija("pokvarena");
        Ks(gitara);
        resi += gitara.getCivija_akcija2()+"\n";
    }
    //magnet
    if(magnetBox.getSelectedItem().toString() == "ne") {
        gitara.setMagnet("dobar");
    }
    if(magnetBox.getSelectedItem().toString() == "necujan") {
        gitara.setMagnet("necujan");
        Ks(gitara);
        resi += gitara.getMagnet_akcija1()+"\n";
    }
    if(magnetBox.getSelectedItem().toString() == "zuji") {
        gitara.setMagnet("zuji");
        Ks(gitara);
        resi += gitara.getMagnet_akcija2()+"\n";
    }
    if(magnetBox.getSelectedItem().toString() == "tih") {

```

```

        gitara.setMagnet("tih");
        Ks(gitara);
        resi += gitara.getMagnet_akcija3()+"\n";
    }
    if(magnetBox.getSelectedItem().toString() == "glasan") {
        gitara.setMagnet("glasan");
        Ks(gitara);
        resi += gitara.getMagnet_akcija4()+"\n";
    }
    if(magnetBox.getSelectedItem().toString() == "pokvaren") {
        gitara.setMagnet("pokvaren");
        Ks(gitara);
        resi += gitara.getMagnet_akcija5()+"\n";
    }
    //nut
    if(nutBox.getSelectedItem().toString() == "ne") {
        gitara.setNut("dobar");
    }
    if(nutBox.getSelectedItem().toString() == "polomljen") {
        gitara.setNut("polomljen");
        Ks(gitara);
        resi += gitara.getNut_akcija2()+"\n";
    }
    if(nutBox.getSelectedItem().toString() == "suv (pucketanje zica)") {
        gitara.setNut("suv");
        Ks(gitara);
        resi += gitara.getNut_akcija1()+"\n";
    }
    //ulaz
    if(ulazBox.getSelectedItem().toString() == "ne") {
        gitara.setUlaz("dobar");
    }
    if(ulazBox.getSelectedItem().toString() == "klimav") {
        gitara.setUlaz("klimav");
        Ks(gitara);
        resi += gitara.getUlaz_akcija1()+"\n";
    }
    if(ulazBox.getSelectedItem().toString() == "pokvaren") {
        gitara.setMagnet("necujan");
        gitara.setUlaz("pokvaren");
        Ks(gitara);
        resi += gitara.getUlaz_akcija2()+"\n";
    }
    //svic
    if(svicBox.getSelectedItem().toString() == "ne") {
        gitara.setSvic("dobar");
    }
    if(svicBox.getSelectedItem().toString() == "polomljen") {
        gitara.setSvic("polomljen");
        Ks(gitara);
        resi += gitara.getSvic_akcija1()+"\n";
    }
    if(svicBox.getSelectedItem().toString() == "pokvaren (ne menja magnet)")
{

```

```

        gitara.setSvic("pokvaren");
        Ks(gitara);
        resi += gitara.getSvic_akcija2()+"\n";
    }
    //dugme
    if(dugmeBox.getSelectedItem().toString() == "ne") {
        gitara.setDugme("dobro");
    }
    if(dugmeBox.getSelectedItem().toString() == "klimavo") {
        gitara.setDugme("klimavo");
        Ks(gitara);
        resi += gitara.getDugme_akcija1()+"\n";
    }
    if(dugmeBox.getSelectedItem().toString() == "fali (otpallo)") {
        gitara.setDugme("fali");
        Ks(gitara);
        resi += gitara.getDugme_akcija2()+"\n";
    }

    textArea.setText(resi);
}
});

```

I poslednji deo koda, akcija koja se dešava kada je pritisnuto dugme Obrisi sve, svi padajući meniji se vraćaju na početnu vrednost i briše se tekst rešenja.

```

obrisi.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        textArea.setText(null);
        resi = "";
        akcijaBox.setSelectedItem("ne");
        zicaBox.setSelectedItem("ne");
        mostBox.setSelectedItem("ne");
        intonacijaBox.setSelectedItem("ne");
        potencijometarBox.setSelectedItem("ne");
        pragBox.setSelectedItem("ne");
        vratBox.setSelectedItem("ne");
        civijaBox.setSelectedItem("ne");
        magnetBox.setSelectedItem("ne");
        nutBox.setSelectedItem("ne");
        ulazBox.setSelectedItem("ne");
        svicBox.setSelectedItem("ne");
        dugmeBox.setSelectedItem("ne");
    }
});

```

4. Zaključak

Ekspertski sistemi su korisni jer menjaju ljude eksperte i na taj način im štede dosta vremena, mogu pomoći ljudima koji nisu eksperti i ne mogu doći u kontakt sa ekspertom.

Potrebno je dosta vremena i planiranja da bi se razvio dobar ekspertski sistem koji je precizan do najsitnijih detalja.

Uvek je dobro imati pomoć koju pruža ekspertski sistem pri odlučivanju pogotovo kod poslova u kojima je veoma bitno uvek doneti pravu odluku.

5. Literatura

- 1) https://sr.wikipedia.org/wiki/Ekspertski_sistemi
- 2) <http://moodle.fink.rs>