

Matematički Fakultet
Univeziteta u Beogradu

SEMINARSKI RAD
IZ PREDMETA
VEROVATNOĆA I STATISTIKA

Profesor:
prof. dr. Vesna Jevremović
Student:
Đorđe Vozarević

Jedini način da zaradite novac na ruletu
je da ga ukradete kada krupije ne gleda
Albert Anštajn

Uvod

Rulet je nastao u 18. veku u Francuskoj. Primitivna verzija ruleta je bila je poznata još u 17. veku a prvi ju je predstavio Blez Paskal. Svoj današnji oblik rulet je dobio 1796. godine u Parizu. Rulet je kockarska igra koja u kojoj se igrač kladu sa vlasnikom ruleta gde će pasti kuglica na točku za rulet. Vlasnik ruleta zapošljava osobu koja rukuje ruletom i koja isplaćuje eventualne dobitke igraču. Na ruletu se može kladiti više igrača odjednom ali su svi igrači međusobno nezavisni. Rulet se sastoji od točka za rulet i stola za rulet. Na stolu za rulet se podnose opklade. Evropski (francuski) točak za rulet se sastoji od 37 ćelija u koje može upasti kuglica. Na jednom stolu za rulet postoji tačno jedna kuglica. Kuglica se u jednom trenutku može nalaziti samo u jednoj ćeliji. Ćelije su numerisane brojevima od 0 do 36. Takođe svaka ćelija ima svoju boju koja može biti crvena ili crna (brojevi od 1 do 36) i zelena (0). Igra se sastoji u tome da igrači saopšte opklade krupijeu (osobi koja rukuje ruletom) nakon čega se točak ruleta zavrti i ispusti se loptica na točak. Nakon što se kuglica zaustavi proglašuje se kraj igre i isplaćuju se eventuale nagrade. Postoji 10 različitih vrsta opklada. Neke od opklada su:



		0		00	
1 to 18	1st 12	1	2	3	
		4	5	6	
		7	8	9	
		10	11	12	
EVEN	2nd 12	13	14	15	
		16	17	18	
		19	20	21	
		22	23	24	
RED	3rd 12	25	26	27	
		28	29	30	
		31	32	33	
		34	35	36	
BLACK	19 to 36				
		2 to 1	2 to 1	2 to 1	

- Obična opklada- opklada koja se sastoji u tome da igrač najavi broj od 0 do 36 i u slučaju da kuglica upadne u ćeliju sa tim brojem igrač dobija svoj ulog pomnožen sa 35.
- Opklada na boju je opklada u kojoj igrač najavljuje boju (crvena ili crna) kojom će biti označena ćelija u koju upadne kuglica. Dobitak je jednak visini uloga.
- Opklada na parnost je opklada u kojoj igrač najavljuje parnost broja ćelije u koju upadne kuglica. Nula u ruletu nije ni paran ni neparan broj. Dobitak je jednak visini uloga.
- Opklada na trećine se sastoji u tome što igrač najavljuje u koju trećinu brojeva će da upadne kuglica (1-12, 13-24, 25-36) Dobitak se nagrađuje sa dvostrukom sumom uloga.

Postoje još 6 različitih vrsta opklada u ruletu ali su sve slične po svom obliku. U svim opkladama na ruletu su odnosi isplata napravljeni kao da se bira između 36 različitih brojeva. Kako na Evropskom ruletu postoji 37 brojeva to daje vlasniku ruleta prednost od 2.7% u odnosu na igrača. Važno je napomenuti da postoji i američki rulet koji ima 38 brojeva od kojih su dve nule (jednostruka i dvostruka nula) što daje prihod vlasniku ruleta od 5.26%. Na stolovima za rulet označeni su minimalni i maksimalni ulog (obično je odnos 1:100).

Na slici sa leve strane prikazan je Američki sto za rulet sa dve nule. Na gornjoj slici prikazan je točak za evropski rulet.

Zadatak

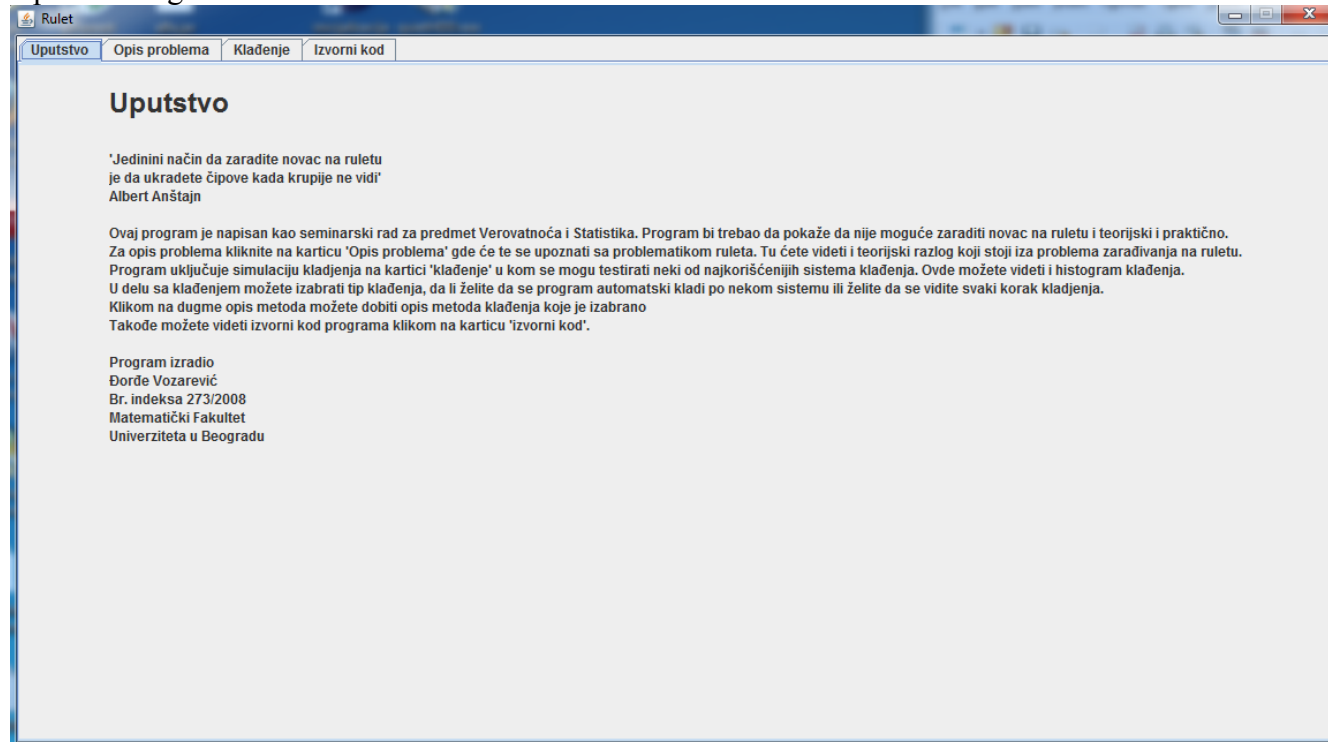
Zadatak ovog seminarskog je da pokaže da ne postoji metoda kladjenja (kombinacija tipova uloga, visina uloga...) kojom se postiže finansijska korist za igrača na ruletu. To sam pokušao da uradim pravljenjem simulacije različitih metoda kladjenja.



Program

Program se sastoji od uputstva, opisa problema, simulacije kladjenja i otvorenog koda.

Uputstvo izgleda ovako:



Opis problema se sastoji od matematičkog modela evropskog ruleta u kome se pokazuje da je matematičko očekivanje za rulet negativno:

Matematički model

Kao primer, možemo ispitati model Evropskog ruleta (rulet sa jednom nulom). Pošto ovaj rulet ima 37 celija sa podjednakom verovatnoćom da budu izabrane, jasno je da je ovo model polja verovatnoće $(\Omega, 2^\Omega, \mathbb{P})$, gde $\Omega = \{0, \dots, 36\}$. $\mathbb{P}(A) = \frac{|A|}{37}$ za sve $A \in 2^\Omega$. Nazvaćemo opkladu S trojkom (A, r, ξ) , gde A za određeni slučajni događaj/događaj $r \in \mathbb{R}_+$, i $\xi : \Omega \rightarrow \mathbb{R}$ slučajna veličina. Događaj A označava dobitak, r u visini opklade (u dinarima, na primer), ξ označava način kladjenja, i matematičko očekivanje $M[\xi]$ koreliraju sa profitabilnošću opklade.

Pravila Evropskog ruleta podrazumevaju 10 vrsta opklada. Prvo ćemo ispitati "običnu" opkladu. Jasno je da u ovom slučaju, $S = (\{\omega'\}, r, \xi)$, gde $\omega' \in \Omega$, i ξ su određeni zakonom:

$$\xi(\omega) = \begin{cases} -r, & \omega \neq \omega' \\ 35 \cdot r, & \omega = \omega' \end{cases}$$

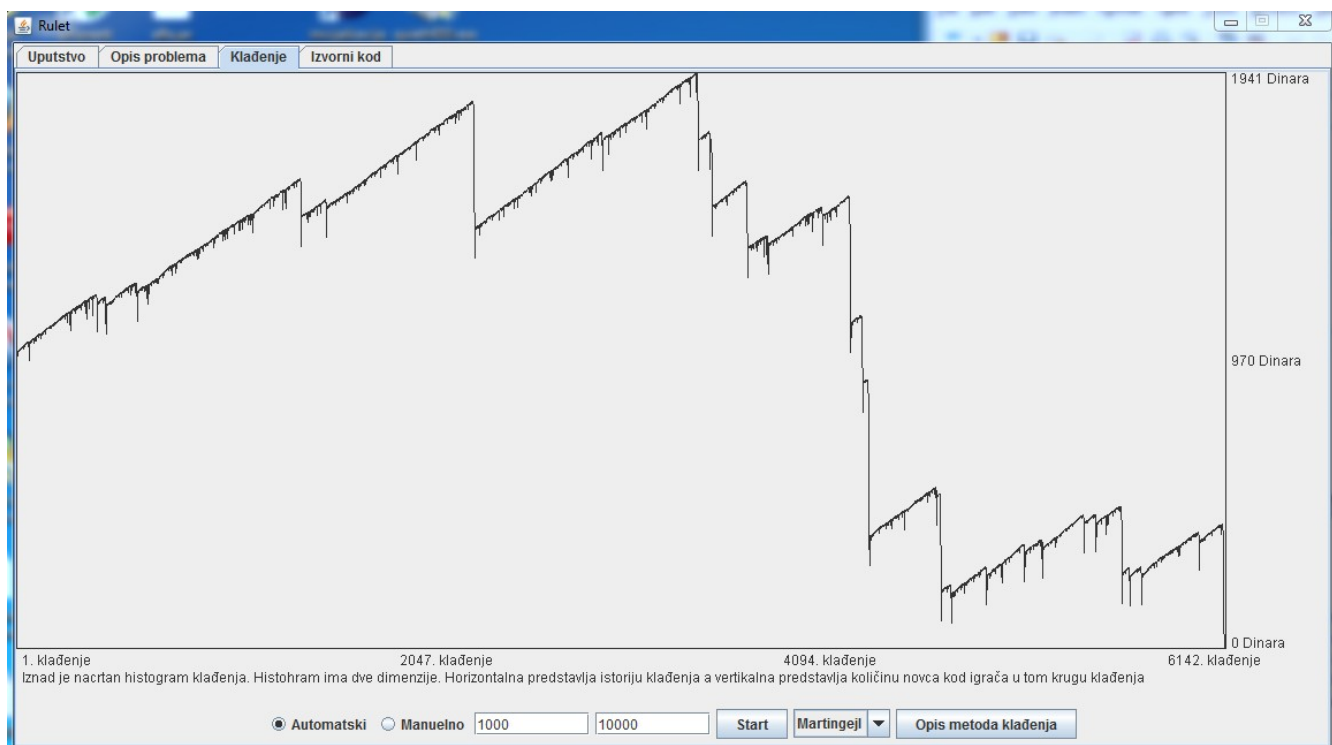
Sada možemo izračunati matematičko očekivanje:

$$M[\xi] = \frac{1}{37} \sum_{\omega \in \Omega} \xi(\omega) = \frac{1}{37} \left(\xi(\omega') + \sum_{\omega \neq \omega'} \xi(\omega) \right) = \frac{1}{37} (35 \cdot r - 36 \cdot r) = -\frac{r}{37} \approx -0.027r.$$

Bez ulaženja u detalje, za opkladu na boju, crvenu ili crnu, pravilo je određeno sa:

$$\xi(\omega) = \begin{cases} -r, & \omega \text{ je crvena} \\ -r, & \omega = \text{nula} \\ r, & \omega \text{ je crna} \end{cases},$$

i isplativost $M[\xi] = \frac{1}{37} (18 \cdot r - 18 \cdot r - r) = -\frac{r}{37}$ Iz sličnih razloga, jednostavno je videti da za sve ostale tipove opklada važi da je isplativost jednaka $-\frac{r}{37}$



Simulacija kladjenja: u ovom delu programa može se pokrenuti simulacija jednog od ponuđenih metoda kladjenja. Prvo se odabere način rada simulacije, potom se unose početna suma novca kojom raspolaže igrač kao i broj ponavljanja simulacije. Zatim se odabere metoda kladjenja. Klikom da dugme opis kladjenja dobija se kratak opis metoda kladjenja. Klikom na dugme Start pokreće se simulacija. Rezultat simulacije prikazuju se na histogramu koji prikazuje promeni količine novca kod igrača kroz vreme.

U delu Izvorni kod može se pogledati kompletan izvorni kod programa. Program je napisan u programskom jeziku java i može se koristiti na različitim operativnim sistemima (Windows, Linux...).

```

Rulet
Uputstvo Opis problema Klađenje Izvorni kod

import javax.swing.*;
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.InputStreamReader;
//import javax.swing.text.TabExpander;
import java.io.*;

public class TabbedPane extends JFrame {
    private static final long serialVersionUID = 1L;
    static String automatski="A";
    JTextField jtfPocSuma;
    JTabbedPane jtp;
    JPanel jp1;
    JPanel jp2;
    JPanel jp3;
    JPanel jp4;
    String[] metodiKladjenja={"Martingelj","FibonaĀci","Dalahmer","Tuce"};
    JTextField jtfBrPonavljanja;
    JLabel label1 = new JLabel();
    Grafik pp=new Grafik();
    JComboBox metodi=new JComboBox(metodiKladjenja);
    JRadioButton jrbAutomatski;
    JRadioButton jrbManuelno;
    ButtonGroup radioGrupa;
    public TabbedPane() {
        // This will create the title you see in the upper left of the window
        setTitle("Rulet");
        setSize(1150, 640); // set size so the user can "see" it
        setLocationRelativeTo(null); // Postavljamo prozor na sredinu ekrana
        // Here we are creating the object
        jtp = new JTabbedPane();
        // This creates the template on the windowed application that we will be using
        getContentPane().add(jtp);
    }
}

```

Otvoreni kod programa:

```
import javax.swing.*;
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.InputStreamReader;
//import javax.swing.text.TabExpander;
import java.io.*;

public class TabbedPane extends JFrame {
    private static final long serialVersionUID = 1L;
    static String automatski="A";
    JTextField jtfPocSuma;
    JTextField suma;
    JTabbedPane jtp;
    JPanel jp1;
    JPanel jp2;
    JPanel jp3;
    JPanel jp4;
    String[] metodiKladjenja={"Martingejl","Fibonači","Dalamber","Tuce",};
    JTextField jtfBrPonavljanja;
    JLabel label1 = new JLabel();
    Grafik pp=new Grafik();
    JComboBox metodi=new JComboBox(metodiKladjenja);
    JRadioButton jrbAutomatski;
    JRadioButton jrbManuelno;
    ButtonGroup radioGrupa;
    public TabbedPane() {
        // This will create the title you see in the upper left of the window
        setTitle("Rulet");
        setSize(1150, 640); // set size so the user can "see" it
        setLocationRelativeTo(null); // Postavljamo prozor na sredinu ekrana
        // Here we are creating the object
        jtp = new JTabbedPane();
        // This creates the template on the windowed application that we will be using
        getContentPane().add(jtp);
        jp1 = new JPanel(); // This will create the first tab
        jp2 = new JPanel(); // This will create the second tab
        jp3 = new JPanel(); // This will create the first tab
        jp4 = new JPanel(); // This will create the second tab
        //sadrzaj prvog taba
        label1.setText("<html><h1>Uputstvo</h1><br> Ovaj program je napisan kao seminarski rad " +
            "za predmet Verovatnoća i Statistika. Program bi trebao da pokaže " +
            "da nije moguće zaraditi novac na ruletu i teorijski i praktično.<br>" +
            "Za opis problema kliknite na karticu 'Opis problema' gde će te se " +
            "upoznati sa problematikom ruleta. Tu ćete videti i teorijski razlog koji stoji " +
            "iza problema zarađivanja na ruletu.<br> Program uključuje simulaciju kladjenja " +
            "na kartici 'kladenje' u kom se mogu testirati neki od najkorišćenijih sistema kladjenja.
```

```

Ovde možete videti i histogram kladenja." +
    "<br>U delu sa kladenjem možete izabrati tip kladenja, " +
    "da li želite da se program automatski kladi po nekom sistemu ili želite da se vidite svaki
" +
    "korak kladenja.<br>Klikom na dugme opis metoda možete dobiti opis metoda kladenja
koje je izabrano" +
    " <br>Takođe možete videti izvorni kod programa klikom na karticu 'izvorni kod.'" +
    "<br><br>Program izradio <br>Đorđe Vozarević<br>Br. indeksa
273/2008<br>Matematički Fakultet<br>" +
    "Univerziteta u Beogradu </html>");
jp2.add(label1);
//sadrzaj drugog taba
JRadioButton jrbAutomatski = new JRadioButton("Automatski",true);
jrbAutomatski.setActionCommand("A");
JRadioButton jrbManuelno = new JRadioButton("Manuelno",false);
jrbManuelno.setActionCommand("M");
radioGrupa = new ButtonGroup();
radioGrupa.setSelected(null, true);
radioGrupa.add(jrbManuelno);
radioGrupa.add(jrbAutomatski);
jp1.add(jrbAutomatski);
jp1.add(jrbManuelno);
RadioListener radioL = new RadioListener();
jrbAutomatski.addActionListener(radioL);
jrbManuelno.addActionListener(radioL);
jtfPocSuma = new JTextField("Početna suma");//pocetna suma
// JTextField jtfKod = new JTextField("ovde ce biti kod programa");

Dimension dimenzijaTextBoxeva=new Dimension();
dimenzijaTextBoxeva.setSize(100, 20);
jtfPocSuma.setSize(dimenzijaTextBoxeva);
jtfPocSuma.setPreferredSize(dimenzijaTextBoxeva);
jp1.add(jtfPocSuma);
JTextField suma=new JTextField(10);
suma.setEditable(false);
suma.setVisible(true);
jtfBrPonavljanja = new JTextField("Broj ponavljanja");//broj ponavljanja
jtfBrPonavljanja.setSize(dimenzijaTextBoxeva);
jtfBrPonavljanja.setPreferredSize(dimenzijaTextBoxeva);
jp1.add(jtfBrPonavljanja);
JButton start = new JButton("Start");
JButton button = new JButton("Opis metoda kladenja");
//Add action listener to button
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        //Execute when button is pressed
        switch (metodi.getSelectedIndex()) {
            case 0:

```

```

        JOptionPane.showMessageDialog(null, "<html>Martingel metoda kladjenja se zasniva
na jednostavnom pravilu:<br>" +
        "ukoliko ste izgubili ovu opkladu sledeća opklada treba da bude dva puta
veća nego sadašnja<br>" +
        "ovo bi trebalo da garantuje da će te uvek povratiti svoj novac i čak
zaraditi. problem je <br>" +
        "u tome što će se skoro sigurno dogoditi niz gubitaka zbog kog ćete
potrošiti sav novac <br>" +
        "primera radi ako bi ste za kockanje imali sve američke dolare koji su u
opticaju izgubili<br>" +
        "bi ste sav novac u slučaju da Vam se dogodi niz gubitaka dužine 43.
Dodatna otežavajuća<br>" +
        "okolnost je to što su maksimalni ulozi u ruletu ograničeni i to uglavnom
na približno 100<br>" +
        "minimalnih uloga.</html>");
        break;
    case 1:
        JOptionPane.showMessageDialog(null, "<html>Fibonači metoda kladjenja je veoma
slična kao Martingel metoda sa <br>" +
        "razlikom u tome kako se povećava opklada u slučaju gubitka. U Fibonači
metodi ulazi nešto sporije rastu<br>" +
        "u skladu sa Fibonačijevim nizom. Ova metoda je podjednako neuspešna
kao i Meringel iz sličnih razloga</html>");
        break;
    case 2:
        JOptionPane.showMessageDialog(null, "<html>Dalamber metoda se zasniva na
predrasudi igrača da je veća verovatnoća<br>" +
        "da nakon dobitka sleduje gubitak i obratno. Ova metoda je, po svojoj
uspešnosti, slična metodi nasumičnog<br>" +
        "kladjenja. Metoda se zasniva na sledećem algoritmu: ukoliko je trenutna
opklada uspešna sledeću opkladu treba<br>" +
        "smanjiti za 1, a u slučaju da ste u ovoj opkladi izgubili sledeću treba
povećati za 1. Ova metoda je neuspešna<br>" +
        "zato što su događaji u ruletu nezavisni</html>");
        break;
    case 3:
        JOptionPane.showMessageDialog(null, "<html>Tuce metoda se bavi opkladom na
trećine. U ruletu igrač se može kladiti<br>" +
        "da će kuglica upasti u brojeve od 1 do 12, 13 do 24 i 24 do 36 što
predstavlja donju, srednju i gornju<br>" +
        "trećinu, respektivno. Metoda se zasniva na brojanju pre koliko krugova
je kuglica poslednji put pala<br>" +
        "u neku trećinu. Opklada se sastoji u tome da se igrač kladi da će kuglica
pasti na trećinu na koju<br>" +
        "najduže nije pala. Ova metoda je neuspešna iz istog razloga kao i
Dalamber metoda a to je da su <br>" +
        "događaji u ruletu nezavisni međusobno</html>");
        break;
    default:

```



```

        break;
    }
}
});
jp1.add(start);
jp1.add(metodi);
jp1.add(button);
//sadrzaj treceg taba
ImageIcon image = new ImageIcon("opis.jpg");
JLabel slika = new JLabel();
slika.setIcon(image);
jp3.add(slika);
//sadrzaj cetrvtog taba
JTextArea kod=new JTextArea();
try{
    String strLine;
    InputStream in = getClass().getResourceAsStream("TabbedPane.txt");
    BufferedReader br = new BufferedReader(new InputStreamReader(in));
    while ((strLine = br.readLine()) != null) {
        kod.append(strLine + "\n");
    }
}
}
catch(Exception e){
    System.out.println("Greska " + e);
}
finally{
    ;
}
kod.setEditable(false);
JScrollPane jspKod=new JScrollPane();
kod.setPreferredSize(new Dimension(1000, 2000));
JLabel nesto=new JLabel();
nesto.setPreferredSize(new Dimension(1000, 500));
jspKod.setViewportView(kod);
pp.repaint();
pp.setVisible(true);
pp.setMinimumSize(new Dimension(100, 100));
jp2.revalidate();
pp.setLayout(new BorderLayout());
pp.add(BorderLayout.SOUTH,jp1);
jtp.addTab("Uputstvo", jp2);
jtp.addTab("Opis problema", jp3);
jtp.addTab("Klađenje", pp);
jtp.addTab("Izvorni kod", jspKod);
ButtonHandler phandler = new ButtonHandler();
start.addActionListener(phandler);
setVisible(true); // otherwise you won't "see" it

```

```

pp.repaint();
}
class ButtonHandler implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        // Igra test = new Igra();
        int maxSuma=1;
        int brojKladjenja=0;
        int sume[]=new int[100000];
        for (int i=0;i<100000;i++)
            sume[i]=0;
        int suma=Integer.parseInt(jtfPocSuma.getText().trim());
        Kladjenje probnoKladjenje = new Kladjenje(suma, metodi.getSelectedIndex()+1);

        for (int i=0; i<=Integer.parseInt(jtfBrPonavljanja.getText()); i++){
            if(TabbedPane.automatski=="M"){
                Icon blueIcon = new ImageIcon("yourFile.gif");
                Object stringArray[] = { " Dalje", "Prekid" };
                String dobitak=new String(" Izgubili ste: "+probnoKladjenje.trenutnaopklada);
                if (probnoKladjenje.dobitak>0)
                    dobitak="Dobili ste: "+(probnoKladjenje.trenutnaopklada*2);
                int test=JOptionPane.showOptionDialog(null,
                    /*jtfPocSuma.getText()+ probnoKladjenje.pocetnaSuma*/ "Broj: " +
probnoKladjenje.igra.brojInt + ", boja: "
                    + probnoKladjenje.igra.boja + ", paran: " + probnoKladjenje.igra.paran
+ " , ulog: "
                    +probnoKladjenje.trenutnaopklada+", suma: " + suma +
dobitak+"test"+probnoKladjenje.low
                    ,"Manuelno kladjenje",
                    JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE,blueIcon, stringArray,stringArray[0]);
                if (test==1)
                    break;
            }
            suma+=probnoKladjenje.izvrši();
            if (suma<=0)
                suma=0;
            sume[i]=suma;
            if (maxSuma<suma)
                maxSuma=suma;
            brojKladjenja++;
            if (suma<=0){
                suma=0;
                break;
            }
        }
        pp.setOstalo(maxSuma,brojKladjenja);
        for (int i=0;i<=brojKladjenja;i++){
            pp.setNovaTacka(i, sume[i]);
        }
    }
}

```

```

    }
    pp.setSpremno(true);
    pp.repaint();
}
}
// This is the internal class that defines what the above Action Listener
// will do when the test button is pressed.
// example usage
public static void main(String[]args) {
    TabbedPane tab = new TabbedPane();
    tab.setResizable(false);
    tab.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
class RadioListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        TabbedPane.automatski = e.getActionCommand();
    }
}
}
import java.util.Random;
class Kladjenje {
    boolean automatski=true;
    int tip;
    int trenutnaopklada=1;
    int prethodnaOpklada=0;
    int pocetnaSuma = 0;
    int dobitak;
    int low=0;//pre koliko krugova je kuglica pala na jedan od low brojeva
    int medium=0;//pre koliko krugova je kuglica pala na jedan od medium brojeva
    int high=0;//pre koliko krugova je kuglica pala na jedan od high brojeva
    Igra igra=new Igra();
    Kladjenje(int pocetnaSumad, int tipd) {
        tip=tipd;
        pocetnaSuma= pocetnaSumad;
        trenutnaopklada=1;
        prethodnaOpklada=0;
        pocetnaSuma = 0;
        low=0;//pre koliko krugova je kuglica pala na jedan od low brojeva
        medium=0;//pre koliko krugova je kuglica pala na jedan od medium brojeva
        high=0;//pre koliko krugova je kuglica pala na jedan od high brojeva
    }
    public int izvrsi(){
        igra=new Igra();
        if (trenutnaopklada>100)
            trenutnaopklada=100;
        switch(tip){
            case 1://martingejl dupliranje opklade svaki krug

                if (igra.paran==true){

```

```

dobitak=trenutnaopklada;
trenutnaopklada=1;
}
else{
    dobitak=-1*trenutnaopklada;
    trenutnaopklada*=2;
}
break;
case 2:// fibonacci
if (igra.paran==true){
    dobitak=trenutnaopklada;
    trenutnaopklada=1;
}
else{
    dobitak=-1*trenutnaopklada;
    prethodnaOpklada=trenutnaopklada;
    trenutnaopklada=trenutnaopklada+prethodnaOpklada;
}
break;
case 3://dalamber
if (igra.paran==true){
    dobitak=trenutnaopklada;
    trenutnaopklada=trenutnaopklada-1;
}
else{
    dobitak=-1*trenutnaopklada;
    trenutnaopklada=trenutnaopklada+1;
}
break;
case 4: //dvanaestice

```

```

boolean testDobitka=false;
if (low>medium&&low>high){
    if (igra.brojInt>0&&igra.brojInt<=12){
        dobitak=trenutnaopklada;
        testDobitka=true;
    }
}
if (medium>low&&medium>high){
    if (igra.brojInt>12&&igra.brojInt<=24){
        dobitak=trenutnaopklada;
        testDobitka=true;
    }
}
if (high>medium&&high>low){
    if (igra.brojInt>24&&igra.brojInt<=36){
        dobitak=trenutnaopklada;
        testDobitka=true;
    }
}

```

```

}
if (testDobitka){
    trenutnaopklada=1;
}
else{
    dobitak=-1*trenutnaopklada;
    switch (trenutnaopklada) {
        case 1:
            trenutnaopklada=2;
        case 2:
            trenutnaopklada=3;
            break;
        case 3:
            trenutnaopklada=4;
            break;
        case 4:
            trenutnaopklada=6;
            break;
        case 6:
            trenutnaopklada=9;
            break;
        case 9:
            trenutnaopklada=13;
            break;
        case 13:
            trenutnaopklada=20;
            break;
        case 20:
            trenutnaopklada=30;
            break;
        case 30:
            trenutnaopklada=45;
            break;
        case 45:
            trenutnaopklada=67;
            break;
        case 67:
            trenutnaopklada=100;
            break;

        default:
            break;
    }
}
if (igra.brojInt>0&&igra.brojInt<=12){
    low=0;
    medium++;
    high++;
}

```



```

        if (igra.brojInt>12&&igra.brojInt<=24){
            low++;
            medium=0;
            high++;
        } if (igra.brojInt>24&&igra.brojInt<=36){
            low++;
            medium++;
            high=0;
        }
        break;
    default:
        dobitak=0;
        break;
    }
    return dobitak;
}
}
}
class Igra {
    Random randomGenerator = new Random();
    int brojInt = randomGenerator.nextInt(36);
    String boja;
    boolean paran=false;
    Igra() {
        brojInt = randomGenerator.nextInt(36);
        if(brojInt!=0){
            if (brojInt % 2 == 0) {
                paran = true;
            } else {
                paran = false;
            }
        }
    }
    if(brojInt==0)
        paran=false;
    switch (brojInt) {
        case 1:
        case 3:
        case 5:
        case 7:
        case 12:
        case 14:
        case 16:
        case 18:
        case 19:
        case 21:
        case 23:
        case 25:
        case 27:
        case 30:
        case 32:

```

```

    case 34:
    case 36:
        boja = new String("Crna");
        break;
    case 2:
    case 4:
    case 6:
    case 8:
    case 10:
    case 11:
    case 13:
    case 15:
    case 17:
    case 20:
    case 22:
    case 24:
    case 26:
    case 29:
    case 31:
    case 33:
    case 35:
        boja = new String("Crvena");
        break;

    default:
        boja = new String("Zelena");
        break;
}
}
}
import java.awt.Graphics;
import javax.swing.JPanel;

public class Grafik extends JPanel {

    private static final long serialVersionUID = 1L;
    int brojTacaka=0;
    int[] tackeX=new int[100000];
    int[] tackeY=new int[100000];
    int maxSuma=0;
    boolean spremno=false;
    double korekcijax = 1.0;
    double korekcijay = 1.0;

    void setNovaTacka(int x1, int y1){
        tackeX[brojTacaka]=x1;
        tackeY[brojTacaka]=y1;
        if (x1==0)
            brojTacaka=0;

```

```

    brojTacaka++;
}
void setSpremno(boolean input){
    spremno=input;
}
void setOstalo(int maxSumain,int brojKladnjenja){
    maxSuma=maxSumain;
    korekcijay = -500.0/maxSuma;
}
@Override public void paintComponent(Graphics g) {
    super.paintComponent(g);
    if (spremno==true){
        brojTacaka--;
        brojTacaka--;

        for (int i=1; i<=brojTacaka; i++){
            double fkorigovanaTackay1= (((double)(tackeY[i-1]))*korekcijay);
            double fkorigovanaTackay2= (((double)(tackeY[i]))*korekcijay);
            int korigovanaTackay1=(int) fkorigovanaTackay1+500;
            int korigovanaTackay2=(int) fkorigovanaTackay2+500;
            korekcijax=1049.0/(brojTacaka);
            double fkorigovanaTackax1= (((double)(tackeX[i-1]))*korekcijax);
            double fkorigovanaTackax2= (((double)(tackeX[i]))*korekcijax);
            int korigovanaTackax1=(int) fkorigovanaTackax1;
            int korigovanaTackax2=(int) fkorigovanaTackax2;
            if (i==1)
                korigovanaTackax1=0;
            g.drawLine(korigovanaTackax1,korigovanaTackay1, korigovanaTackax2,korigovanaTackay2);
        }
    }
    g.drawRect(0, 0, 1050, 500);
    g.drawString(Integer.toString(maxSuma)+" Dinara", 1055, 10);
    g.drawString(Integer.toString(maxSuma/2)+" Dinara", 1055, 255);
    g.drawString(Integer.toString(0)+" Dinara", 1055, 500);
    g.drawString(Integer.toString(1)+" . kladenje", 5, 515);
    g.drawString(Integer.toString(brojTacaka/3)+" . kladenje", 1000/3, 515);
    g.drawString(Integer.toString(2*brojTacaka/3)+" . kladenje", 2*1000/3, 515);
    g.drawString(Integer.toString(brojTacaka)+" . kladenje", 1000, 515);
}
}

import java.awt.Graphics;
import javax.swing.JPanel;

public class Grafik extends JPanel {

    private static final long serialVersionUID = 1L;
    int brojTacaka=0;
    int[]tackeX=new int[100000];

```

```

int[] tackeY=new int[100000];
int maxSuma=0;
boolean spremno=false;
double korekcijax = 1.0;
double korekcijay = 1.0;

void setNovaTacka(int x1, int y1){
    tackeX[brojTacaka]=x1;
    tackeY[brojTacaka]=y1;
    if (x1==0)
        brojTacaka=0;
    brojTacaka++;
}
void setSpremno(boolean input){
    spremno=input;
}
void setOstalo(int maxSumain,int brojKladnjenja){
    maxSuma=maxSumain;
    korekcijay = -500.0/maxSuma;
}
@Override public void paintComponent(Graphics g) {
    super.paintComponent(g);
    if (spremno==true){
        brojTacaka--;
        brojTacaka--;

        for (int i=1; i<=brojTacaka; i++){
            double fkorigovanaTackay1= (((double)(tackeY[i-1]))*korekcijay);
            double fkorigovanaTackay2= (((double)(tackeY[i]))*korekcijay);
            int korigovanaTackay1=(int) fkorigovanaTackay1+500;
            int korigovanaTackay2=(int) fkorigovanaTackay2+500;
            korekcijax=1049.0/(brojTacaka);
            double fkorigovanaTackax1= (((double)(tackeX[i-1]))*korekcijax);
            double fkorigovanaTackax2= (((double)(tackeX[i]))*korekcijax);
            int korigovanaTackax1=(int) fkorigovanaTackax1;
            int korigovanaTackax2=(int) fkorigovanaTackax2;
            if (i==1)
                korigovanaTackax1=0;
            g.drawLine(korigovanaTackax1,korigovanaTackay1, korigovanaTackax2,korigovanaTackay2);
        }
    }
    g.drawRect(0, 0, 1050, 500);
    g.drawString(Integer.toString(maxSuma)+" Dinara", 1055, 10);
    g.drawString(Integer.toString(maxSuma/2)+" Dinara", 1055, 255);
    g.drawString(Integer.toString(0)+" Dinara", 1055, 500);
    g.drawString(Integer.toString(1)+". kladenje", 5, 515);
    g.drawString(Integer.toString(brojTacaka/3)+". kladenje", 1000/3, 515);
    g.drawString(Integer.toString(2*brojTacaka/3)+". kladenje", 2*1000/3, 515);
    g.drawString(Integer.toString(brojTacaka)+". kladenje", 1000, 515);
}

```

g.drawString("Iznad je nacrtan histogram kladenja. Histohram ima dve dimenzije. Horizontalna predstavlja " +

"istoriju kladenja a vertikalna predstavlja količinu novca kod igrača u tom krugu kladenja", 5, 530);

}

import java.util.Random;

class Kladjenje {

boolean automatski=true;

int tip;

int trenutnaopklada=1;

int prethodnaOpklada=0;

int pocetnaSuma = 0;

int dobitak;

int low=0;//pre koliko krugova je kuglica pala na jedan od low brojeva

int medium=0;//pre koliko krugova je kuglica pala na jedan od medium brojeva

int high=0;//pre koliko krugova je kuglica pala na jedan od high brojeva

Igra igra=new Igra();

Kladjenje(int pocetnaSumad, int tipd) {

tip=tipd;

pocetnaSuma= pocetnaSumad;

trenutnaopklada=1;

prethodnaOpklada=0;

pocetnaSuma = 0;

low=0;//pre koliko krugova je kuglica pala na jedan od low brojeva

medium=0;//pre koliko krugova je kuglica pala na jedan od medium brojeva

high=0;//pre koliko krugova je kuglica pala na jedan od high brojeva

}

public int izvrsi(){

igra=new Igra();

if (trenutnaopklada>100)

trenutnaopklada=100;

switch(tip){

case 1://martingejl dupliranje opklade svaki krug

if (igra.paran==true){

dobitak=trenutnaopklada;

trenutnaopklada=1;

}

else{

dobitak=-1*trenutnaopklada;

trenutnaopklada*=2;

}

break;

case 2:// fibonaci

if (igra.paran==true){

dobitak=trenutnaopklada;

trenutnaopklada=1;

}


```

else{
    dobitak=-1*trenutnaopklada;
    prethodnaOpklada=trenutnaopklada;
    trenutnaopklada=trenutnaopklada+prethodnaOpklada;
}
break;
case 3://dalamber
if (igra.paran==true){
    dobitak=trenutnaopklada;
    trenutnaopklada=trenutnaopklada-1;
}
else{
    dobitak=-1*trenutnaopklada;
    trenutnaopklada=trenutnaopklada+1;
}
break;
case 4: //dvanaestice

boolean testDobitka=false;
if (low>medium&&low>high){
    if (igra.brojInt>0&&igra.brojInt<=12){
        dobitak=trenutnaopklada;
        testDobitka=true;
    }
}
if (medium>low&&medium>high){
    if (igra.brojInt>12&&igra.brojInt<=24){
        dobitak=trenutnaopklada;
        testDobitka=true;
    }
}
if (high>medium&&high>low){
    if (igra.brojInt>24&&igra.brojInt<=36){
        dobitak=trenutnaopklada;
        testDobitka=true;
    }
}
if (testDobitka){
    trenutnaopklada=1;
}
else{
    dobitak=-1*trenutnaopklada;
    switch (trenutnaopklada) {
        case 1:
            trenutnaopklada=2;
        case 2:
            trenutnaopklada=3;
            break;
        case 3:

```

```

        trenutnaopklada=4;
        break;
    case 4:
        trenutnaopklada=6;
        break;
    case 6:
        trenutnaopklada=9;
        break;
    case 9:
        trenutnaopklada=13;
        break;
    case 13:
        trenutnaopklada=20;
        break;
    case 20:
        trenutnaopklada=30;
        break;
    case 30:
        trenutnaopklada=45;
        break;
    case 45:
        trenutnaopklada=67;
        break;
    case 67:
        trenutnaopklada=100;
        break;

    default:
        break;
}
}
if (igra.brojInt>0&&igra.brojInt<=12){
    low=0;
    medium++;
    high++;
}
if (igra.brojInt>12&&igra.brojInt<=24){
    low++;
    medium=0;
    high++;
}if (igra.brojInt>24&&igra.brojInt<=36){
    low++;
    medium++;
    high=0;
}
break;
default:
    dobitak=0;
    break;

```

```

    }
    return dobitak;
}
}
class Igra {
    Random randomGenerator = new Random();
    int brojInt = randomGenerator.nextInt(36);
    String boja;
    boolean paran=false;
    Igra() {
        brojInt = randomGenerator.nextInt(36);
        if(brojInt!=0){
            if (brojInt % 2 == 0) {
                paran = true;
            } else {
                paran = false;
            }
        }
        if(brojInt==0)
            paran=false;
        switch (brojInt) {
            case 1:
            case 3:
            case 5:
            case 7:
            case 12:
            case 14:
            case 16:
            case 18:
            case 19:
            case 21:
            case 23:
            case 25:
            case 27:
            case 30:
            case 32:
            case 34:
            case 36:
                boja = new String("Crna");
                break;
            case 2:
            case 4:
            case 6:
            case 8:
            case 10:
            case 11:
            case 13:
            case 15:
            case 17:

```

```
case 20:  
case 22:  
case 24:  
case 26:  
case 29:  
case 31:  
case 33:  
case 35:  
    boja = new String("Crvena");  
    break;
```

```
default:  
    boja = new String("Zelena");  
    break;
```

```
}  
}  
}
```