

Испит из Објектно оријентисаног програмирања II

- 1) (укупно 70 поена) Написати на језику *Java* следећи пакет типова (грешке пријављивати изузецима опремљеним текстовима порука):
- (25 поена) **Фигура** у равни има задату графичку сцену, почетну позицију (x,y) на сцени и боју. Може да се исцрта на графичкој сцени. Може да се дохвати једнословна ознака врсте фигуре и њене координате (појединачно).
 - **Активна фигура** ажурира своју позицију на задатој сцени са задатим периодом (у милисекундама) Може да се покрене, да се привремено и да се трајно заустави.
 - **Сцена** је графичка компонента – `Canvas`, која може да садржи произвољан број фигура. Може да се дохвати фигура са задатим редним бројем. Сцена може да се покрене (покреће све своје активне фигуре), да се привремено заустави (привремено зауставља све активне фигуре) и да се трајно заустави (трајно зауставља све активне фигуре сцене).
 - (30 поена) **Пак** је активна фигура која се исцртава као попуњен круг задатог пречника. Почетна позиција пака је у центру придружене сцене. Пак се креће праволинијски константном брзином, а почетна брзина му се формира на случајан начин, тако да се на сваки период ажурирања позиције, његов положај по x, односно у оси, може променити највише за ± 1 . Уколико пак удари у играча (видети ниже) или у горњу или доњу ивицу сцене, одбија се од препреке (еластични судар), под одбојним углом једнаким упадном и са брзином непромењеног интензитета. Уколико се пак нађе иза леђа једног играча, други играч добија поен. По добијању поена ажурира се резултат игре хокеј, док се пак поставља на почетну позицију и поново се на случајан начин бира његова брзина (као и почетна). Може да се дохвати померај пака по x и у оси за период ажурирања његове позиције, као и пречник пака. Врста пака је **P**.
 - **Играч** је активна фигура која се исцртава као правоугаоник задате висине и ширине. Почетна позиција је по хоризонтали близу одговарајућег краја сцене, а по вертикали на средини висине сцене. Играч може да се креће само дуж у осе и на сваки период ажурирања позиције његов положај може да се промени за ± 1 . Играч ажурира своју позицију зависно од кретања пака. Уколико се пак креће ка играчу, играч тежи да постигне позицију пака. Уколико се пак креће од играча, играч наставља претходно кретање док не удари у једну од ивица, након чега мења смер кретања. Висина и ширина могу да му се дохвате. Врста играча је **I**.
 - (15 поена) **Хокеј** на графичкој корисничкој површи приказује сцену са 2 играча на супротним странама сцене и један пак. У горњем делу прозора приказује се резултат игре. Може да се покрене сцена, да се привремено заустави као и да се трајно заустави. Може да се увећа број поена играча, као и да се поништи тренутни резултат.



- НАПОМЕНЕ:**
- Време за израду задатка је **150** минута.
 - Рад се предаје искључиво на предвиђеном мрежном диску.
 - Називе типова ускладити са називима апстракција из текста задатка, али користити латинично писмо и велико почетно слово.
 - На располагању је приступ *Web* адреси: <https://docs.oracle.com/javase/8/docs/api/>. Није дозвољено имати поред себе друге материјале, нити уз себе имати електронске уређаје, без обзира да ли су укључени или искључени.
 - Резултати колоквијума биће објављени на *Web*-у на адреси: <http://rti.etf.bg.ac.rs/rti/ir2002/index.html>

```

//Figura.java
package hokej;
import java.awt.Color;
public abstract class Figura {
    protected Scena scena;
    protected double x, y;
    protected Color boja;
    protected Figura(Scena s, int xx, int yy,
                    Color b) {
        scena = s; x = xx; y = yy; boja = b; }
    public double dohvX() { return x; }
    public double dohvY() { return y; }
    public abstract void crtaj();
    public abstract char vrsta(); }

// AktFigura.java
package hokej;
import java.awt.Color;
public abstract class AktFigura extends Figura
implements Runnable {
    protected Thread nit = new Thread(this);
    private int tAzur;
    private boolean radi;
    public AktFigura(Scena s, int xx, int yy,
                    Color b, int t) {
        super(s, xx, yy, b); tAzur = t;
    }
    protected abstract void azurirajPolozaj();
    public void run() {
        try {
            while (!Thread.interrupted()) {
                synchronized (this) {
                    if (!radi) wait();
                }
                azurirajPolozaj();
                scena.repaint();
                Thread.sleep(tAzur);
            }
        } catch (InterruptedException ie) {}
    }
    public synchronized void kreni() {
        radi = true; notify();
    }
    public void stani() { radi = false; }
    public void prekini() {
        if (nit != null) nit.interrupt();
    }
}

// Scena.java
package hokej;
import java.awt.*;
public class Scena extends Canvas {
    private Elem prvi, posl;
    private class Elem {
        Figura fig; Elem sled;
        Elem(Figura f) {
            fig = f;
            if (prvi != null) posl.sled = this;
            else prvi = this;
            posl = this; }
    }
    public synchronized void dodaj(Figura f) {
        new Elem(f); }
    public synchronized Figura dohvatiFig(int ind){
        Elem tek = prvi;
        if (ind < 0) return null;
        while (tek != null && ind > 0) {
            tek = tek.sled; ind--;
        }
        if (tek == null) return null;
        else return tek.fig;
    }
}

```

```

public synchronized void paint(Graphics g) {
    Elem tek = prvi;
    while (tek != null) {
        tek.fig.crtaj();
        tek = tek.sled;
    }
}
public synchronized void kreni() {
    for (Elem tek = prvi; tek != null;
         tek = tek.sled)
        if (tek.fig instanceof AktFigura)
            ((AktFigura) tek.fig).kreni();
}
public synchronized void stani() {
    for (Elem tek = prvi; tek != null;
         tek = tek.sled)
        if (tek.fig instanceof AktFigura)
            ((AktFigura) tek.fig).stani();
}
public synchronized void prekini() {
    for (Elem tek = prvi; tek != null;
         tek = tek.sled)
        if (tek.fig instanceof AktFigura)
            ((AktFigura) tek.fig).prekini();
}
}

//Pak.java
package hokej;
import java.util.Random;
import java.awt.*;
public class Pak extends AktFigura {
    private Random rand = new Random();
    private int d;
    private double dx = 1, dy = 0;
    private Hokej hok;
    { rand.setSeed(System.currentTimeMillis());
      Pak(Scena s, Color b, int t, int dd,
          Hokej h) {
          super(s, s.getWidth()/2, s.getHeight()/2,
                b, t);
          d = dd; hok = h; dodeliBrzinu(); nit.start();
      }
    }
    public int dohvPrecnik() { return d; }
    public double dohvDx() { return dx; }
    public double dohvDy() { return dy; }
    private void dodeliBrzinu(){
        // biraj ugao u intervalu [45,60] ili [-60,-45]
        while (Math.abs(Math.tan(dy/dx)) < 1 ||
              Math.abs(Math.tan(dy/dx)) > 1.74){
            dy = -1 + Math.random() * 2;
        }
    }
    private void resetujPoziciju() {
        x = scena.getWidth()/2;
        y = scena.getHeight()/2;
        dodeliBrzinu();
    }
    @Override
    protected void azurirajPolozaj() {
        x+=dx; y+=dy; Figura f;
        Igrac igr1 = null, igr2 = null;
        for(int i = 0;
            (f = scena.dohvatiFig(i)) != null; i++)
            if(f.vrsta()=="I"){
                if(igr1 == null) igr1 = (Igrac)f;
                else {
                    igr2 = (Igrac)f;
                    if(igr1.dohvX()>igr2.dohvX()) {
                        f=igr1; igr1=igr2; igr2=(Igrac)f;
                    }
                }
            }
    }
}

```

```

}
if(igr1!=null && igr2!=null) {
    if(x+d/2<=igr2.dohvX()+igr2.dohvSir()/2 &&
       x+d/2>=igr2.dohvX()-igr2.dohvSir()/2 &&
       y>=igr2.dohvY()-igr2.dohvVis()/2 &&
       y<=igr2.dohvY()+igr2.dohvVis()/2)
        dx *= -1;
    else
        if(x-d/2<=igr1.dohvX()+igr1.dohvSir()/2 &&
           x-d/2>=igr1.dohvX()-igr1.dohvSir()/2 &&
           y>=igr1.dohvY()-igr1.dohvVis()/2 &&
           y<=igr1.dohvY()+igr1.dohvVis()/2)
            dx *= -1;
        }
    if(x-d/2<=0) {
        hok.uvecajP(1); dx = 1; resetujPoziciju();
    }
    if(x+d/2>=scena.getWidth()) {
        hok.uvecajP(0); dx = -1; resetujPoziciju();
    }
    if(y-d/2<=0) dy *= -1;
    if(y+d/2>=scena.getHeight()) dy *= -1;
}
public void crtaj() {
    Graphics g = scena.getGraphics();
    g.setColor(boja);
    g.fillOval((int)x-d/2, (int)y-d/2, d, d);
}
public char vrsta() { return 'P'; }
}

//Igrac.java
package hokej;
import java.awt.*;
public class Igrac extends AktFigura {
    private int w, h, dy = 1;
    public Igrac(Scena s, int x, int y, Color b,
                int t, int ww, int hh) {
        super(s, x, y, b, t); w = ww; h = hh;
        nit.start();
    }
    public int dohvSir() { return w; }
    public int dohvVis() { return h; }
    @Override
    protected void azurirajPolozaj() {
        Figura f = null;
        for (int i = 0;
            (f = scena.dohvatiFig(i)) != null; i++)
            if (f.vrsta() == 'P') break;
        if (f == null) return;
        Pak p = (Pak) f;
        // ako pak ide prema igracu
        if(p.dohvX() > x && p.dohvDx() < 0 ||
           p.dohvX() < x && p.dohvDx() > 0){
            // igrac poravnava svoj y polozaj sa pakom
            if(y < p.dohvY()) dy = 1;
            else if(y > p.dohvY()) dy = -1;
        }
        y += dy;
        if (y - h / 2 <= 0){ y = h/2; dy = -dy; }
        else if (y + h / 2 >= scena.getHeight()){
            y = scena.getHeight() - h/2; dy = -dy;
        }
    }
}
public void crtaj() {
    Graphics g = scena.getGraphics();
    g.setColor(boja);
    g.fillRect((int)x-w/2, (int)y-h/2, w, h);
}
public char vrsta() { return 'I'; } }

```

```

//Hokej.java
package hokej;
import java.awt.*;
import java.awt.event.*;
public class Hokej extends Frame {
    private int poeniP, poeniD;
    private Label rezultat;
    private Scena s = new Scena();
    private Button dugKreni = new Button("Kreni"),
                dugStani = new Button("Stani"),
                dugPonisti = new Button("Ponisti");
    public void uvecajP(int igr) {
        if (igr == 0) poeniP++;
        else if (igr == 1) poeniD++;
        rezultat.setText(poeniP + ":" + poeniD);
    }
    private void dodajSever() {
        Panel pan = new Panel();
        pan.setBackground(Color.LIGHT_GRAY);
        rezultat = new Label("0:0");
        rezultat.setFont(new Font(null,Font.BOLD,22));
        pan.add(rezultat);
        add(pan, BorderLayout.NORTH);
    }
    private void dodajCentar() {
        s.setSize(350, 150);
        Pak p = new Pak(s, Color.GREEN, 5, 10, this);
        s.dodaj(p);
        s.dodaj(new Igrac(s, 20, s.getHeight() / 2,
                         Color.BLUE, 8, 5, 30));
        s.dodaj(new Igrac(s, s.getWidth() - 30,
                         s.getHeight() / 2, Color.RED, 8, 5, 30));
        add(s, BorderLayout.CENTER);
    }
    private void dodajJug() {
        Panel pan = new Panel();
        pan.setBackground(Color.GRAY);
        dugKreni.addActionListener(
            (e)->{ s.kreni(); });
        dugStani.addActionListener(
            (e)->{ s.stani(); });
        dugPonisti.addActionListener(
            (e)->{poeniP =poeniD =0;
                  rezultat.setText("0:0"); });
        pan.add(dugKreni);
        pan.add(dugStani);
        pan.add(dugPonisti);
        add(pan, BorderLayout.SOUTH);
    }
    public void dodajKomponente() {
        dodajSever();
        dodajJug();
        dodajCentar();
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                s.prekini(); dispose();
            }
        });
    }
}
public Hokej() {
    super("Hokej");
    setBounds(300, 300, 350, 250);
    setLocation(300, 300);
    setResizable(false);
    dodajKomponente();
    setVisible(true);
}
public static void main(String[] args) {
    new Hokej();
}
}

```