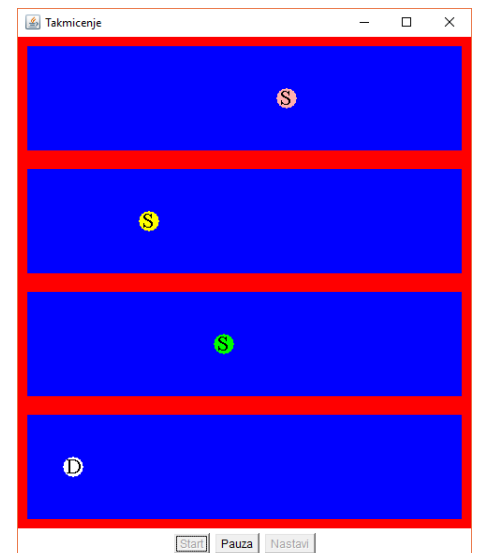


Поправни колоквијум из Објектно оријентисаног програмирања II

1) (укупно 100 поена) Саставити на језику *Java* следећи пакет класа:

- (10 поена) Равномерно **пливајуће** је нешто чему се може дохватити реална константна брзина.
- Пливачке **дисциплине** су LEĐNO, PRSNO, DELFIN, SLOBODNO.
- (20 поена) Активни равномерно пливајући **пливач** има задате дисциплину и константну реалну брзину (m/s) којом плива 100 метара, које могу да се дохвате. Може да му се постави трака (видети ниже) у којој плива. Пливач може да се покрене, након чега ће испливати својих 100 метара и завршити са радом. Док плива, након сваких 40ms јавља пливачкој траци (уколико постоји) своју дисциплину и до тада пређени пут. Тренутни пређени пут може да му се дохвати. Пливање се може паузирати, наставити и трајно прекинути.
- (20 поена) Активна мешовита **штафета** има задате низ пливача и боју. Боја може да јој се дохвати. Штафета може да се покрене, након чега пливачи у њој наступају по редоследу горе наведених дисциплина. Када један пливач преплива своју деоницу, наступа следећи. Након што последњи пливач преплива своју деоницу, штафета завршава са радом. Може да се дохвати пливач који тренутно наступа, да се паузира, настави и трајно прекине наступање пливача у штафети.
- (20 поена) Графичка пливачка **трака** је хоризонтално платно (Canvas) са плавом позадином и црвеним оквиром (ширине 5). Може да му се постави и дохвати штафета чије пливаче приказује. Ширина траке (без оквира) се пресликава у дужину од 50 метара (половина дужине коју пливачи пливају) у реалном свету и приказује напредовање пливача постављенештафете у једном и другом смеру. Пливачи се представљају попуњеним кругом пречника петине висине траке, обојеним бојом штафете са уписаним првим словом дисциплине у круг (као на слици).
- (10 поена) **Базен** је панел који ствара и приказује 4 хоризонталне пливачке траке поређане по вертикали, тако да заузму целу површину панела, као на слици. Може да се дохвати трака са задатим индексом.
- (20 поена) **Такмичење** је прозор који садржи базен испод којег су дугмад Start, Pauza, Nastavi. Омогућена су само дугмад која имају смисла у одговарајућем стању (на почетку само Start, затим за време такмичења само Pauza, а за време паузе само Nastavi). Може да се дохвати трака са задатим индексом и да се постави задата штафета траци са задатим индексом.

(0 поена) Приложена је класа са главном функцијом која испитује основне функционалности пакета класа.



НАПОМЕНЕ:

- Израда решења задатка траје 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/ir2oo2/index.html>

```

package main;

import java.awt.Color;

import plivanje.Disciplina;
import plivanje.Plivac;
import plivanje.Stafeta;
import plivanje.Takmicenje;
import plivanje.Traka;

public class Main {

    public static void main(String[] args) {
        Takmicenje takmicenje = new Takmicenje();

        Traka t1 = takmicenje.dohvTraku(0);
        Plivac p11 = new Plivac(Disciplina.DELFIN, 1 + Math.random());
        p11.postaviTraku(t1);
        Plivac p13 = new Plivac(Disciplina.PRSNO, 1 + Math.random());
        p13.postaviTraku(t1);
        Plivac p12 = new Plivac(Disciplina.LEDJNO, 1 + Math.random());
        p12.postaviTraku(t1);
        Plivac p14 = new Plivac(Disciplina.SLOBODNO, 1 + Math.random());
        p14.postaviTraku(t1);
        takmicenje.postaviStafetu(new Stafeta(new Plivac[] {p11, p12, p13, p14}, Color.PINK), 0);

        Traka t2 = takmicenje.dohvTraku(1);
        Plivac p21 = new Plivac(Disciplina.DELFIN, 1 + Math.random());
        p21.postaviTraku(t2);
        Plivac p22 = new Plivac(Disciplina.LEDJNO, 1 + Math.random());
        p22.postaviTraku(t2);
        Plivac p23 = new Plivac(Disciplina.PRSNO, 1 + Math.random());
        p23.postaviTraku(t2);
        Plivac p24 = new Plivac(Disciplina.SLOBODNO, 1 + Math.random());
        p24.postaviTraku(t2);
        takmicenje.postaviStafetu(new Stafeta(new Plivac[] {p21, p22, p23, p24}, Color.YELLOW), 1);

        Traka t3 = takmicenje.dohvTraku(2);
        Plivac p32 = new Plivac(Disciplina.LEDJNO, 1 + Math.random());
        p32.postaviTraku(t3);
        Plivac p33 = new Plivac(Disciplina.PRSNO, 1 + Math.random());
        p33.postaviTraku(t3);
        Plivac p31 = new Plivac(Disciplina.DELFIN, 1 + Math.random());
        p31.postaviTraku(t3);
        Plivac p34 = new Plivac(Disciplina.SLOBODNO, 1 + Math.random());
        p34.postaviTraku(t3);
        takmicenje.postaviStafetu(new Stafeta(new Plivac[] {p31, p32, p33, p34}, Color.GREEN), 2);

        Traka t4 = takmicenje.dohvTraku(3);
        Plivac p42 = new Plivac(Disciplina.LEDJNO, 1 + Math.random());
        p42.postaviTraku(t4);
        Plivac p44 = new Plivac(Disciplina.SLOBODNO, 1 + Math.random());
        p44.postaviTraku(t4);
        Plivac p43 = new Plivac(Disciplina.PRSNO, 1 + Math.random());
        p43.postaviTraku(t4);
        Plivac p41 = new Plivac(Disciplina.DELFIN, 1 + Math.random());
        p41.postaviTraku(t4);
        takmicenje.postaviStafetu(new Stafeta(new Plivac[] {p41, p42, p43, p44}, Color.WHITE), 3);

        Plivac p5 = new Plivac(Disciplina.SLOBODNO, 1 + Math.random());
        p5.pokreni();
        p5.pauziraj();
        p5.nastavi();
        p5.prekini();

        Stafeta s = new Stafeta(new Plivac[] { new Plivac(Disciplina.DELFIN, 1 + Math.random())}, Color.RED);
        s.pokreni();
        s.pauziraj();
        s.nastavi();
        s.prekini();
    }
}

```

```

package plivanje;
public interface Plivajuca {
    public double dohvBrzinu(); }
}
package plivanje;
public enum Disciplina {
    LEDJNO, PRSNO, DELFIN, SLOBODNO }
}
package plivanje;
public class Plivac extends Thread
    implements Plivajuca {
    private Disciplina d;
    private double v, s;
    private Traka t;
    private boolean radi = true;
    static int DT = 40, L = 100;
    public Plivac(Disciplina dd, double vv)
    { d = dd; v = vv; }
    public Disciplina dohvDisciplinu()
    { return d; }
    public double dohvBrzinu()
    { return v; }
    public double dohvPut() { return s; }
    public void postaviTraku(Traka tt)
    { t = tt; }
    public void pokreni() { start(); }
    public void run() {
        try {
            while (!interrupted() && s < L){
                synchronized(this)
                { if (!radi) wait(); }
                sleep(DT); s += v * DT / 1000;
                if (t != null) t.presao(d, s);
            }
        } catch (InterruptedException e){}
    }
    public void pauziraj() { radi=false; }
    public synchronized void nastavi()
    { radi = true; notifyAll(); }
    public void prekini() { interrupt(); }
}
}
package plivanje;
import java.awt.*;
public class Stafeta extends Thread {
    private Plivac[] plivaci;
    private Color boja;
    private int indTekPlivaca = -1;
    private boolean radi = true;
    public Stafeta (Plivac[] p, Color cc)
    { plivaci = p; boja = cc; }
    public Color dohvBoju() {return boja;}
    public Plivac dohvPlivac() {
        if (indTekPlivaca >= 0)
            { return plivaci[indTekPlivaca]; }
        return null;
    }
    public void pokreni() { start(); }
    public void run() {
        try {
            Disciplina[] d =

```

```

            Disciplina.values();
            int dIndex = 0, sIndex = 0;
            while (!interrupted() &&
                dIndex <= d.length - 1) {
                synchronized(this) {
                    if (!radi) wait(); }
                    while (sIndex < plivaci.length &&
                        plivaci[sIndex].dohvDisciplinu()
                        != d[dIndex]) { sIndex ++; }
                    if (sIndex == plivaci.length)
                        { sIndex = 0; dIndex++; continue; }
                    indTekPlivaca = sIndex;
                }
            } catch (InterruptedException e) {}
        }
        public synchronized void pauziraj() {
            radi = false;
            if (indTekPlivaca < 0 ||
                indTekPlivaca >= plivaci.length)
                { return; }
            plivaci[indTekPlivaca].pauziraj();
        }
        public synchronized void nastavi() {
            radi = true; notifyAll();
            if (indTekPlivaca < 0 ||
                indTekPlivaca >= plivaci.length)
                { return; }
            plivaci[indTekPlivaca].nastavi();
        }
        public synchronized void prekini() {
            interrupt();
            if (indTekPlivaca < 0 ||
                indTekPlivaca >= plivaci.length)
                { return; }
            plivaci[indTekPlivaca].prekini();
        }
    }
}
package plivanje;
import java.awt.*;
public class Traka extends Canvas {
    private Stafeta stafeta;
    private int xPlivaca;
    private String oznakaPlivaca;
    public Traka()
    { setBackground(Color.RED); }
    public void paint(Graphics g) {
        int w = getWidth() - 10;
        int h = getHeight() - 10;
        g.setColor(Color.BLUE);
        g.fillRect(5, 5, w, h);
        if (stafeta != null) {
            g.setColor(stafeta.dohvBoju());

```

```

            g.fillOval(xPlivaca + 5 - h/10,
                2*h/5 + 5, h/5, h/5);
        }
        if (oznakaPlivaca != null) {
            g.setFont(new Font("TimesRoman",
                Font.PLAIN, h/5));
            g.setColor(Color.BLACK);
            g.drawString(oznakaPlivaca,
                xPlivaca - h/16 + 5, 17*h/30 + 5);
        }
    }
    public void postaviStafetu(Stafeta ss)
    { stafeta = ss; }
    public Stafeta dohvStafetu()
    { return stafeta; }
    void presao(Disciplina d, double s) {
        if (s <= Plivac.L/2) {
            xPlivaca = (int) (((2*getWidth()-20)*s)
                / Plivac.L);
        } else {
            xPlivaca = (int) (((2*getWidth()-20)
                * (Plivac.L - s)) / Plivac.L);
        }
        oznakaPlivaca = d.toString().charAt(0)+"";
        repaint();
    }
}
package plivanje;
import java.awt.*;
public class Bazen extends Panel {
    protected static int N = 4;
    private Traka[] t = new Traka[N];
    public Bazen() {
        setLayout(new GridLayout(0, 1));
        for (int i = 0; i < N; i++)
            { add(t[i] = new Traka()); }
    }
    public Traka dohvTraku(int i) {
        if (i < 0 || i >= N) { return null; }
        return t[i];
    }
}
package plivanje;
import java.awt.*;
import java.awt.event.*;
public class Takmicenje extends Frame {
    private Button start, pauza, nastavi;
    private Bazen bazen;
    private Stafeta[] stafete =
        new Stafeta[Bazen.N];
    public Takmicenje() {
        super("Takmicenje");
        setSize(500, 600);
        setLayout(new BorderLayout());
        Panel buttons = new Panel();
        add(buttons, BorderLayout.SOUTH);
        buttons.add(start =
            new Button("Start"));

```

```

        start.addActionListener(
            new ActionListener() {
                public void actionPerformed(
                    ActionEvent e) {
                    start.setEnabled(false);
                    for (int i=0; i<Bazen.N; i++)
                        { stafete[i].pokreni(); }
                    pauza.setEnabled(true);
                }
            });
        buttons.add(pauza =
            new Button("Pauza"));
        pauza.setEnabled(false);
        pauza.addActionListener(
            new ActionListener() {
                public void actionPerformed(
                    ActionEvent e) {
                    pauza.setEnabled(false);
                    for (int i=0; i<Bazen.N; i++)
                        { stafete[i].pauziraj(); }
                    nastavi.setEnabled(true);
                }
            });
        buttons.add(nastavi =
            new Button("Nastavi"));
        nastavi.setEnabled(false);
        nastavi.addActionListener(
            new ActionListener() {
                public void actionPerformed(
                    ActionEvent e) {
                    nastavi.setEnabled(false);
                    for (int i=0; i<Bazen.N; i++)
                        { stafete[i].nastavi(); }
                    pauza.setEnabled(true);
                }
            });
        add(bazen = new Bazen(),
            BorderLayout.CENTER);
        addWindowListener(
            new WindowAdapter() {
                public void windowClosing(
                    WindowEvent e) {
                    for (int i=0; i<Bazen.N; i++)
                        { stafete[i].prekini(); }
                    dispose();
                }
            });
        setVisible(true);
    }
    public Traka dohvTraku(int i)
    { return bazen.dohvTraku(i); }
    public void postaviStafetu(Stafeta s,
        int i) {
        stafete[i] = s;
        bazen.dohvTraku(i)
            .postaviStafetu(s);
    }
}
}

```