

Колоквијум из Објектно оријентисаног програмирања 2

1) (укупно 70 поена) Саставити на језику *Java* следећи пакет класа:

- (10 поена) **Упоредив** појам може да се пореди на једнакост са другим упоредивим појмом.
- **Боја** је упоредив појам који има задате целобројне интензитете црвеног, зеленог и плавог дела, подразумевано су сви 0 и могу да се дохвате. Интензитети су у опсегу 0-255 (вредност већа од 255 или мања од 0 поставља одговарајући интензитет на вредност 255 и 0 респективно). Боја може да се пореди на једнакост само са другим бојама. Грешка је уколико се покуша поређење на једнакост са упоредивим појмом који није боја (*GNeuporedivi*). Боје су једнаке уколико су интензитети одговарајућих појединачних делова боје једнаки. Боји може да се одреди фактор сличности са другом задатом бојом као збир апсолутних вредности разлика интензитета одговарајућих појединачних делова (нпр: $B1(100,200,0)$, $B2(150,100,20)$, $сличност = |100-150|+|200-100|+|0-20|=170$). Може да се састави текстуални опис боје у облику **RGB** (*црвена, зелена, плава*).
- (15 поена) Ускршње **јаје** је упоредив појам који има задату боју и величину (*MALO, SREDNJE, VELIKO*) и садржи аутоматски генерисан јединствен целобројан идентификатор. Све вредности могу да се дохвате. Ускршње јаје може да се пореди на једнакост само са другим ускршњим јајима. Грешка је уколико се покуша поређење на једнакост са упоредивим појмом који није ускршње јаје (*GNeuporedivi*). Јаја су једнака уколико су им једнаке боје и величине. Јаје може да се клонира. Може да се састави текстуални опис јајета у облику **Јаје_ид:боја-величина**.
- (20 поена) Ускршња **корпа** има назив и садржи задати број ускршњих јаја. Може да се дода јаје на прву слободну локацију у корпи. Грешка је уколико је корпа пуна (*GKorpaPuna*) или уколико би се додавањем јајета у корпи нашла два једнака јајета (*GVecPostoji*). Може да се дохвати јаје са задате позиције. Грешка је уколико је позиција ван опсега (*GIndeks*). Може да се дохвати број јаја у корпи и да се из корпе узме јаје које најбоље одговара задатој боји и величини тако што се од свих јаја из корпе нађе јаје које је задате величине и чија је боја најсличнија задатој боји (фактор сличности најближи нули). Ситуација у којој не постоји јаје које одговара критеријумима је регуларна и треба је обрадити. Може да се састави текстуални опис корпе у облику **Корпа: назив** након чега се у наставку исписују садржана јаја у облику [*јаје, јаје, . . . , јаје*].
- (25 поена) Ускршња **изложба** садржи произвољан број ускршњих корпи које се такмиче на изложби и за сваку се памти број победа. Може да се дода корпа изложби, да се дохвати број корпи на изложби и да се започне такмичење. Након почетка такмичења више није могуће додавати корпе изложби (додавање корпе у тој ситуацији се игнорише). Такмичење се састоји из више кругова. Могуће је одржати један круг такмичења (покушај одржавања круга пре започињања такмичења се игнорише) тако што се задају боја и величина, затим се из сваке корпе узима јаје које најбоље одговара задатим критеријумима и на крају од свих њих, по истим критеријумима, бира победничко јаје. Корпи у којој се налазило победничко јаје се увећава број победа. Могуће је дохватити корпу која је у вођству. У случају да више корпи има једнак број победа, у вођству је она корпа која је прва достигла тај број победа. У случају да ниједна корпа нема ниједну победу, у вођству није ниједна корпа. Може да се састави текстуални опис изложбе тако што се у засебним редовима наводе информације о корпама у облику **корпа-Број_победа: број_победа**, при чему се вођство означава навођењем **V**: испред корпе која је у вођству.

(0 поена) Приложена је класа са главном функцијом која испитује основне функционалности пакета класа уз исписивање резултата на стандардном излазу (конзоли).

НАПОМЕНЕ:

- Израда решења задатка траје **100** минута.
- Рад се предаје искључиво на предвиђеном мрежном диску.
- Називе типова и метода ускладити са називима из приложене класе са главном функцијом.
- 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.rs/rti/ir2oo2/index.html/>

```

package main;

import uskrs.*;

public class Main {
    public static void main(String[] args) {

        Boja crna = new Boja(); Boja crvena = new Boja(255, 0, 0); // R, G, B
        Boja zelena = new Boja(0, 255, 0); Boja plava = new Boja(0, 0, 255);
        // dohvatanje pojedinačnih delova boje
        Boja bela = new Boja(crvena.dohvR(), zelena.dohvG(), plava.dohvB());

        Jaje j1 = new Jaje(crvena, Jaje.Velicina.MALO);
        Jaje j2 = new Jaje(zelena, Jaje.Velicina.SREDNJE);
        Jaje j3 = new Jaje(plava, Jaje.Velicina.SREDNJE);
        Korpa k1 = new Korpa("Uskrsnja radost", 3);
        try{
            k1.dodajJaje(j1); k1.dodajJaje(j2); k1.dodajJaje(j3);
            // Neuspesno dodavanje jajeta - korpa je popunjena
            k1.dodajJaje(new Jaje(crvena, Jaje.Velicina.VELIKO));
        }
        catch(GKorpaPuna g) {}
        catch(Exception g) {}

        // Uzimanje jajeta iz korpe koje najbolje odgovara zadatim kriterijumima
        Jaje naj = k1.uzmiNajbolje(new Boja(0, 100, 200), Jaje.Velicina.SREDNJE);
        System.out.println("Jaje koje najbolje odgovara kriterijumima: " + naj);

        Jaje j4 = new Jaje(bela, Jaje.Velicina.MALO);
        Jaje j5 = new Jaje(crna, Jaje.Velicina.SREDNJE);
        Korpa k2 = new Korpa("Minimalisticka korpa", 2);
        try{
            k2.dodajJaje(j4); k2.dodajJaje(j5);
            // Neuspesno dodavanje jajeta - jaje vec postoji u korpi
            k2.dodajJaje(j4.clone());
        }
        catch(GVecPostoji g) {}
        catch(Exception g) {}

        Izlozba i = new Izlozba();
        i.dodajKorpu(k1); i.dodajKorpu(k2);

        i.zapocni();

        System.out.println("*** Ispis korpi u izlozbi pre prvog kruga ***");
        System.out.print(i);
        Jaje pobednik1 = i.odrziKrug(crvena, Jaje.Velicina.SREDNJE);
        System.out.println("*** Ispis pobednickog jajeta i korpi u izlozbi nakon prvog kruga ***");
        System.out.println("Pobednik: " + pobednik1);
        System.out.print(i);
    }
}

```

```
// Primer izvršavanja
```

```

Jaje koje najbolje odgovara kriterijumima: Jaje_3:RGB(0,0,255)-SREDNJE
*** Ispis korpi u izlozbi pre prvog kruga ***
Korpa:Uskrsnja radost[Jaje_1:RGB(255,0,0)-MALO,Jaje_2:RGB(0,255,0)-SREDNJE]-Broj_pobeda:0
Korpa:Minimalisticka korpa[Jaje_4:RGB(255,255,255)-MALO,Jaje_5:RGB(0,0,0)-SREDNJE]-Broj_pobeda:0
*** Ispis pobednickog jajeta i korpi u izlozbi nakon prvog kruga ***
Pobednik: Jaje_5:RGB(0,0,0)-SREDNJE
Korpa:Uskrsnja radost[Jaje_1:RGB(255,0,0)-MALO]-Broj_pobeda:0
V:Korpa:Minimalisticka korpa[Jaje_4:RGB(255,255,255)-MALO]-Broj_pobeda:1

```

```

package uskrs;

public interface Uporediv {
    boolean jednak(Uporediv u) throws GNeuporedivi;
}

public class Boja implements Uporediv {
    private int R, G, B;
    public Boja(int r, int g, int b){
        R = (r >= 0 && r <= 255) ? r : (r < 0) ? 0 : 255;
        G = (g >= 0 && g <= 255) ? g : (g < 0) ? 0 : 255;
        B = (b >= 0 && b <= 255) ? b : (b < 0) ? 0 : 255;
    }
    public Boja(){ this(0, 0, 0); }
    public int dohvR(){ return R; }
    public int dohvG(){ return G; }
    public int dohvB(){ return B; }
    public boolean jednak(Uporediv u) throws GNeuporedivi {
        if(u instanceof Boja){
            return
                R==(Boja)u.R&&G==(Boja)u.G&&B==(Boja)u.B;
        }
        throw new GNeuporedivi();
    }
    public int slicnost(Boja b){
        return Math.abs(R-b.R)+Math.abs(G-b.G)+Math.abs(B-b.B);
    }
    public String toString(){
        return "RGB(" + R + "," + G + "," + B + ")";
    }
}

public class Jaje implements Cloneable, Uporediv {
    public enum Velicina { MALO, SREDNJE, VELIKO; }
    private Boja b; private Velicina v;
    private static int posId= 0; private int id = ++posId;
    public Jaje(Boja b, Velicina v){
        this.b = b; this.v = v;
    }
    public Boja dohvBoju() { return b; }
    public Velicina dohvVelicinu() { return v; }
    public int dohvId() { return id; }
    public boolean jednak(Uporediv u) throws GNeuporedivi {
        if(u instanceof Jaje){
            return b.jednak(((Jaje)u).b)&&v== ((Jaje)u).v;
        }
        throw new GNeuporedivi();
    }
    public Jaje clone() {
        try {
            Jaje j = (Jaje) super.clone();
            j.b = new Boja(b.dohvR(), b.dohvG(), b.dohvB());
            j.id = ++posId; return j;
        } catch (CloneNotSupportedException e) {return null;}
    }
    public String toString() {
        return "Jaje_" + id + ":" + b + "-" + v;
    }
}

public class Korpa {
    private Jaje[] jaja; private int pop;
    private String naziv;
    public Korpa(String n, int k){
        jaja = new Jaje[k]; naziv = n;
    }
    public Korpa dodajJaje(Jaje j) throws GKorpaPuna,
        GVecPostoji {
        if(jaja.length == pop) throw new GKorpaPuna();
        int ind = -1;
        for(int i = 0; i < jaja.length; i++)
            if(ind == -1 && jaja[i] == null) ind = i;
        else
            try {
                if(jaja[i] != null && jaja[i].jednak(j))
                    throw new GVecPostoji();
            } catch (GNeuporedivi e) {}
        jaja[ind] = j; pop++;
        return this;
    }
    public Jaje dohvJaje(int ind) throws GIndeks {
        if(ind<0 || ind>=jaja.length) throw new GIndeks();
        return jaja[ind];
    }
    public int brojJaja(){ return pop; }
    public Jaje uzmiNajbolje(Boja b, Jaje.Velicina v){
        int min = -1; Jaje j = null; int ind = -1;

```

```

        for(int i = 0; i < jaja.length; i++)
            if(jaja[i]!=null && jaja[i].dohvVelicinu()==v)
                if(j==null || jaja[i].dohvBoju().slicnost(b)<min){
                    ind = i; j = jaja[i];
                    min = jaja[i].dohvBoju().slicnost(b);
                }
        if (j != null) { jaja[ind] = null; pop--; }
        return j;
    }
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("Korpa:" + naziv); sb.append("[");
        boolean zarez = false;
        for(int i = 0; i < jaja.length; i++)
            if(jaja[i] != null){
                if (zarez) sb.append(",");
                sb.append(jaja[i]); zarez = true;
            }
        sb.append("]"); return sb.toString();
    }
}

public class Izlozba {
    private static class Elem{
        Korpa korpa; int pobede; Elem sled;
        Elem(Korpa korpa) { this.korpa = korpa; }
    }
    private Elem prvi, posl, vodjstvo;
    private int maxPobeda, pop;
    private boolean zapoceto;
    public Izlozba dodajKorpu(Korpa k){
        if(!zapoceto){
            Elem novi = new Elem(k);
            if(prvi!=null) posl.sled=novi; else prvi=novi;
            posl = novi; pop++;
        } return this;
    }
    public int dohvBrojKorpi() { return pop; }
    public void zapocni() { zapoceto = true; }
    public Jaje odrziKrug(Boja b, Jaje.Velicina v){
        if(!zapoceto) return null;
        Elem tmp=prvi, pob=null; Jaje najbolje = null;
        while(tmp != null){
            Jaje j = tmp.korpa.uzmiNajbolje(b, v);
            if(najbolje==null && j!=null || j!=null &&
                j.dohvBoju().slicnost(b) <
                najbolje.dohvBoju().slicnost(b)){
                najbolje = j; pob = tmp;
            } tmp = tmp.sled;
        }
        if(pob != null) pob.pobede++;
        if(pob != null && pob.pobede > maxPobeda){
            maxPobeda = pob.pobede; vodjstvo = pob;
        } return najbolje;
    }
    public Korpa uVodjstvu() {
        return vodjstvo != null ? vodjstvo.korpa : null;
    }
    public String toString() {
        StringBuilder sb = new StringBuilder();
        Elem tmp = prvi;
        while(tmp != null){
            if(tmp.korpa == uVodjstvu()) sb.append("V:");
            sb.append(tmp.korpa).append("-Broj_pobeda:")
                .append(tmp.pobede).append("\n");
            tmp = tmp.sled;
        }
        return sb.toString();
    }
}

public class GNeuporedivi extends Exception {
    public String toString(){ return "Zadate stvari nije
        moguće uporediti!"; }
}

public class GKorpaPuna extends Exception {
    public String toString(){ return "Niz je popunjen!"; }
}

public class GVecPostoji extends Exception {
    public String toString(){ return "Boja ne sme da se
        ponovi!"; }
}

public class GIndeks extends Exception {
    public String toString(){return "Indeks je van opsega!"; }
}

```