

Први колоквијум из Објектно оријентисаног програмирања II

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

- (10 поена) **Тачка** има назив и целобројне координате x и y , који могу да се дохвате. Може да се одреди реално растојање између две задате тачке и да се састави текстуални опис тачке у облику *naziv (x, y)*.
- (10 поена) **Круг** има тачку која представља његов центар и целобројну дужину полупречника, који могу да се дохвате. Може да се одреди да ли се површи два задата круга преклапају и да се састави текстуални опис круга у облику **Krug [центар, полупречник]**. Сматра се да се кругови који се само додирују не преклапају; нпр. круг са центром у тачки (1, 1) и полупречником 1 и круг са центром у тачки (3, 1) и полупречником 1 се не преклапају.
- (20 поена) **Цртеж** има име и може да садржи задати број кругова (сматрати да се број кругова увек коректно задаје, не треба проверавати његову исправност). Може да се дода задати круг на цртеж и уклони круг са задатим индексом (индекси почињу од 1, индекс не мора бити коректно задат) са цртежа, само уколико су испуњени одговарајући услови за одређену врсту цртежа. Повратна вредност додавања и уклањања кругова је индикатор успешности операције. Може да се дохвати круг са задатим индексом (индекси почињу од 1, индекс не мора бити коректно задат) са цртежа. Може да се дохвати назив врсте цртежа, да се дохвати број кругова на цртежу и да се састави текстуални опис цртежа у облику *име [назив_врсте]*, иза чега следе текстуални описи кругова, сваки у посебном реду (уколико на некој позицији у цртежу не постоји круг, прескочити је, без измене описа).
- (15 поена) **Преклапајући цртеж** је цртеж у којем за сваки круг мора да важи да му се површ преклапа са површи барем неког другог круга на цртежу. Ствара се са задатим именом, бројем кругова које може да садржи и једним задатим кругом. Није могуће уклонити једини круг са цртежа. Назив врсте цртежа је **PREKLAPAJUCI**.
- (15 поена) **Непреклапајући цртеж** је цртеж у којем за свака два круга мора да важи да им се површи не преклапају. Ствара се са задатим именом и бројем кругова које може да садржи. Назив врсте цртежа је **NEPREKLAPAJUCI**.

(0 поена) Приложена је класа са главном функцијом која направи три тачке, а затим испише податке о првој тачки, испише другу тачку и испише вредност растојања између прве и друге тачке. Након тога направи три круга, а затим испише податке о првом кругу, испише други круг и испише податак о томе да ли се први и други круг преклапају. Након тога направи један преклапајући цртеж са трећим кругом, дода му други и први круг, уклони му круг са позиције 1, а затим испише информацију о броју кругова на цртежу и цртеж. Након тога направи један непреклапајући цртеж, дода му редом трећи, други и први круг, уклони му круг са позиције 1, а затим испише круг на позицији 1, испише круг на позицији 2, испише круг на позицији 5 и испише цртеж. Сва исписивања врше се на стандардном излазу (конзоли).

НАПОМЕНЕ: а) Израда решења задатка траје **100** минута.

б) Рад се предаје искључиво на предвиђеном мрежном диску.

в) Називе типова и метода ускладити са називима из приложене класе са главном функцијом.

г) На располагању је приступ *Web* адреси: <https://docs.oracle.com/javase/8/docs/api/>.

д) Није дозвољено имати поред себе друге материјале, нити уз себе имати електронске уређаје, без обзира да ли су укључени или искључени.

ђ) Резултати колоквијума биће објављени на *Web*-у на адреси: <http://rti.etf.rs/rti/ir2oo2/index.html/>

```

//main.java
package test;

import crtezi.*;

public class Main {

    public static void main(String[] args) {
        Tacka t1 = new Tacka("T1",3,9); // x=3, y=9
        Tacka t2 = new Tacka("T2",7,7); // x=7, y=7
        Tacka t3 = new Tacka("T3",5,4); // x=5, y=4
        System.out.println(t1.dohvNaziv() + "(" + t1.dohvX()+ ", " + t1.dohvY() + ")");
        System.out.println(t2);
        System.out.println(Tacka.d(t1, t2));

        Krug k1 = new Krug(t1, 2); // c=t1, r=2
        Krug k2 = new Krug(t2, 2); // c=t2, r=2
        Krug k3 = new Krug(t3, 3); // c=t3, r=3
        System.out.println("Krug[" + k1.dohvC() + ", " + k1.dohvR() + "]");
        System.out.println(k2);
        System.out.println("Krugovi se " + (Krug.preklapajuSe(k1, k2)?"":"ne ") +
            "preklapaju");

        PreklapajuciC pCrtez = new PreklapajuciC("Moj_crtez_1", 4, k3);
        pCrtez.dodaj(k2);
        pCrtez.dodaj(k1);
        pCrtez.ukloni(1);
        System.out.println("Crtez sadrzi " + pCrtez.brojKrugova() + " krugova");
        System.out.println(pCrtez);

        NepreklapajuciC nCrtez = new NepreklapajuciC("Moj_crtez_2", 3);
        nCrtez.dodaj(k3);
        nCrtez.dodaj(k2);
        nCrtez.dodaj(k1);
        nCrtez.ukloni(1);
        System.out.println(nCrtez.dohvati(1));
        System.out.println(nCrtez.dohvati(2));
        System.out.println(nCrtez.dohvati(5));
        System.out.println(nCrtez);
    }
}

```

```

T1(3, 9)
T2(7, 7)
4.47213595499958
Krug[T1(3, 9), 2]
Krug[T2(7, 7), 2]
Krugovi se ne preklapaju
Crtez sadrzi 2 krugova
Moj_crtez_1[PREKLAPAJUCI]
Krug[T3(5, 4), 3]
Krug[T2(7, 7), 2]
null
Krug[T1(3, 9), 2]
null
Moj_crtez_2[NEPREKLAPAJUCI]
Krug[T1(3, 9), 2]

```

```

package crtezi;

public class Tacka {
    private String naziv;
    private int x; private int y;

    public Tacka(String n, int xx, int yy) {
        naziv = n; x = xx; y = yy;
    }

    public int dohvX() { return x; }
    public int dohvY() { return y; }
    public String dohvNaziv() { return naziv; }
    public static double d(Tacka t1, Tacka t2) {
        return Math.sqrt(Math.pow(t1.x - t2.x, 2)
            + Math.pow(t1.y - t2.y, 2));
    }

    public String toString() {
        return naziv + "(" + x + ", " + y + ")";
    }
}

```

```

package crtezi;

public class Krug {
    private Tacka c;
    private int r;

    public Krug(Tacka cc, int rr) { c=cc; r=rr; }
    public Tacka dohvC() { return c; }
    public int dohvR() { return r; }
    public static boolean preklapajuSe(Krug k1,
        Krug k2) {
        return (k1 != null) && (k2 != null) &&
            (Tacka.d(k1.c, k2.c) < (k1.r + k2.r));
    }

    public String toString() {
        return "Krug[" + c + ", " + r + "]";
    }
}

```

```

package crtezi;

public abstract class Crtez {
    private String ime;
    protected int pop = 0;
    protected Krug[] niz;

    public Crtez(String n, int kap) {
        ime = n; niz = new Krug[kap];
    }

    protected boolean mozeDodaj(Krug k) {
        return brojKrugova() < niz.length;
    }
}

```

```

public boolean dodaj(Krug k) {
    if (mozeDodaj(k)) {
        int i = 0;
        while (i < niz.length && niz[i] != null) {
            i++;
        }
        if (i < niz.length) { niz[i]=k; pop++; }
        return true;
    }
    return false;
}

protected boolean mozeUkloni(int i) {
    return i > 0 && i <= niz.length &&
        niz[i-1] != null;
}

public boolean ukloni(int i) {
    if (mozeUkloni(i)) {
        niz[i-1] = null; pop--; return true;
    }
    return false;
}

public Krug dohvati(int i) {
    if (i > 0 && i <= niz.length) {
        return niz[i-1];
    }
    return null;
}

public abstract String vrsta();
public int brojKrugova() { return pop; }
public String toString() {
    StringBuilder s = new StringBuilder(
        ime + "[" + vrsta() + "]");
    for (int i = 0; i < niz.length; i++) {
        if (niz[i] != null) {
            s.append("\n"); s.append(niz[i]);
        }
    }
    return s.toString();
}
}

```

```

package crtezi;

public class PreklapajuciC extends Crtez {

    public PreklapajuciC(String n, int kk, Krug k) {
        super(n, kk); niz[0] = k; pop = 1;
    }

    protected boolean mozeDodaj(Krug k) {
        if (!super.mozeDodaj(k)) { return false; }
        boolean moze = false; int i = 0;
        while (i < niz.length && !moze) {
            moze = Krug.preklapajuSe(k, niz[i]);
        }
    }
}

```

```

        i++;
    }
    return moze;
}

protected boolean mozeUkloni(int i) {
    if (!super.mozeUkloni(i)) { return false; }
    if (pop == 1) { return false; }
    boolean moze = true; int j = 0;
    while (j < niz.length && moze) {
        if (j != (i-1) &&
            Krug.preklapajuSe(niz[i-1], niz[j])) {
            boolean postoji = false; int k = 0;
            while (k < niz.length && !postoji) {
                postoji = k != (i-1) && k != j &&
                    Krug.preklapajuSe(niz[j], niz[k]);
                k++;
            }
            moze = postoji;
        }
        j++;
    }
    return moze;
}

public String vrsta() {
    return "PREKLAPAJUCI";
}
}

```

```

package crtezi;

public class NepreklapajuciC extends Crtez {

    public NepreklapajuciC(String n, int kap) {
        super(n, kap);
    }

    protected boolean mozeDodaj(Krug k) {
        if (!super.mozeDodaj(k)) { return false; }
        boolean moze = true; int i = 0;
        while (i < niz.length && moze) {
            moze = !Krug.preklapajuSe(k, niz[i]);
            i++;
        }
        return moze;
    }

    protected boolean mozeUkloni(int i) {
        if (!super.mozeUkloni(i)) { return false; }
        niz[i-1] = null; return true;
    }

    public String vrsta() {
        return "NEPREKLAPAJUCI";
    }
}

```