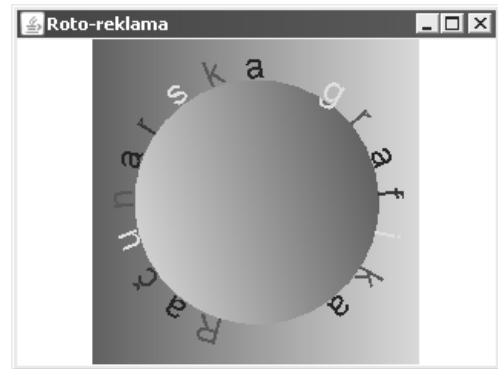


RAČUNARSKA GRAFIKA

prvi kolokvijum

- 1) [25] **Java2D:** Napisati klasu sa potrebnim atributima i metodom `paint()` koja u prozoru crta figuru prikazanu na slici 1. Figura je sastavljena od kvadrata obojenog valerom od tamno sive do svetlo sive, kruga obojenog valerom od tamno sive do zelene i teksta koji se crta po obodu kruga i kod kojeg se slova repetitivno boje crvenom, crnom, plavom i narandžastom. Slova su ravnomerno raspoređena duž luka od 320° . Tekstu koji se ispisuje se pristupa referencom deklarisanom na sledeći način: **private String** tekst; Centar figure se nalazi u centru prozora. Dužina stranice kvadrata odgovara minimumu od širine i visine prozora. Prečnik kruga iznosi $\frac{3}{4}$ dužine stranice kvadrata. Krug rotira konstantnom ugaonom brzinom u suprotnom smeru od smera kretanja kazaljki na časovniku. Ispisan tekst rotira istom tom ugaonom brzinom u smeru kretanja kazaljki na časovniku. Tekuća vrednost ugla rotacije je smeštena u atribut klase deklarisan na sledeći način: **private double** tekUgao; Smatrati da promena vrednosti ugla rotacije i obaveštavanje da je potrebno ažuriranje sadržaja prozora u pravilnim vremenskim intervalima realizovano metodom `run()`, koju ne treba pisati. Nije potrebno pisati ni glavni program.



Slika 1

- 2) [25] Izvesti analitičku formulu kubne Bezejeve krive definisane nad tačkama P_0, P_1, P_2 i P_3 , u zavisnosti od realnog parametra t . Tačke P_0 i P_3 su početna i krajnja tačka krive, respektivno. Napisati matrični oblik kubne Bezejeve krive. Da li se poklapanjem kontrolnih tačaka P_1 i P_2 dobija kvadratna Bezejeva kriva?
- 3) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Navesti elemente konceptualnog okvira interaktivne računarske grafike. Šta je osnovna uloga grafičkog sistema?
 - Koji su atributi SRGP teksta?
 - Na čemu se zasniva efikasnost operacije `SRGP_CopyPixel` i koji su njeni parametri?
 - Koje fizičke uređaje apstrahuju valuatori i šta vraćaju kao meru?
 - Napisati matričnu jednačinu za preslikavanje iz prozora u 2D realnom svetu u prikazni prozor. Prozor je određen tačkama $(0,10)$ i $(20,30)$, a prikazni prozor tačkama $(0.75,0)$ i $(1,0.25)$. Nije potrebno množiti matrice.

Napomene:

- Kolokvijum traje 100 minuta.
- Nije dozvoljena upotreba literature niti programabilnih kalkulatora.
- Dozvoljena je upotreba AWT i Java2D podsetnika.

Rešenja zadatka

prvi kolokvijum 2011

1) Rešenje

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
public class K1_0910 extends JPanel implements Runnable {
    private String tekst = "Računarska grafika";
    private Font f = new Font( "Arial", Font.PLAIN, 24 );
    private double tekuciUgao;
    private double ugaonaBrzina = 25;
    private Ellipse2D krug = new Ellipse2D.Double(-0.5, -0.5, 1, 1);
    private Rectangle2D okvir = new Rectangle2D.Double(-0.5, -0.5, 1, 1);
    private Paint []boje = { Color.red, Color.black, Color.blue, Color.orange };
    private Paint valer =
        new GradientPaint(-0.5f, 0, Color.darkGray, 0.5f, 0, Color.lightGray);
    private Paint valer2 =
        new GradientPaint(-0.5f, 0, Color.darkGray, 0.5f, 0, Color.green);

    public void paint (Graphics g) {
        super.paint(g);
        Graphics2D g2d = (Graphics2D) g;

        Dimension dim = getSize();
        double velicina = Math.min(dim.width, dim.height);
        AffineTransform prethodnaTransformacija = g2d.getTransform();

        g2d.translate(dim.width/2, dim.height/2);
        g2d.scale(velicina, velicina);
        g2d.setPaint(valer);
        g2d.fill(okvir);

        g2d.setTransform(prethodnaTransformacija);
```

$$2) \begin{aligned} P_{01}(t) &= tP_1 + (1-t)P_0 & P_{02}(t) &= tP_{11} + (1-t)P_{01} & P_{03}(t) &= tP_{12} + (1-t)P_{02}, \\ P_{11}(t) &= tP_2 + (1-t)P_1, & P_{12}(t) &= tP_{21} + (1-t)P_{11}, \\ P_{21}(t) &= tP_3 + (1-t)P_2 \\ P(t) &= P_{03}(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3, t \in [0,1] \end{aligned}$$

$$P(t) = [1 \ t \ t^2 \ t^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Ako se poklope P_1 i P_2 , ne dobija se kvadratna Bezejova kriva, već: $P(t) = (1-t)^3 P_0 + 3t(1-t)P_{12} + t^3 P_3, t \in [0,1]$

Analitička formula za kvadratnu Bezejovu krivu je: $P(t) = (1-t)^2 P_0 + 2t(1-t)P_{12} + t^2 P_3, t \in [0,1]$

```
g2d.translate(dim.width/2, dim.height/2);
g2d.scale(velicina*3/4, velicina*3/4);
g2d.rotate( Math.toRadians(-tekuciUgao) );
g2d.setPaint(valer2);
g2d.fill(krug);

g2d.setTransform(prethodnaTransformacija);
g2d.translate(dim.width/2, dim.height/2);
g2d.rotate( Math.toRadians(tekuciUgao) );

g2d.setFont(f);

AffineTransform rotiranCentar = g2d.getTransform();
double inkrementRotacije = 320.0 / tekst.length();
double ugaoRotacije = 0;
for(int i = 0; i < tekst.length(); i++) {
    g2d.setPaint(boje[i%4]);
    g2d.translate(0, -velicina*3/8);
    g2d.drawString(tekst.substring(i, i+1), 0, 0);
    g2d.setTransform(rotiranCentar);
    ugaoRotacije += inkrementRotacije;
    g2d.rotate(Math.toRadians( ugaoRotacije ) );
}
g2d.setTransform(prethodnaTransformacija);
}

public void run() { ... }
public static void main(String s[]) { ... }
```

