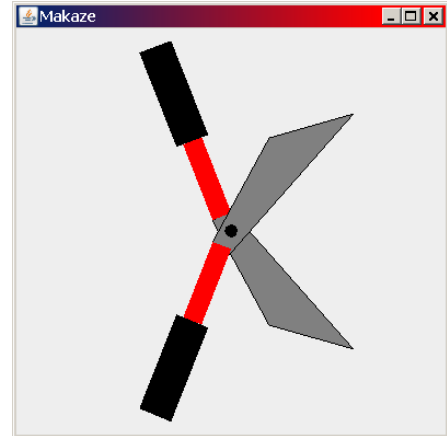


RAČUNARSKA GRAFIKA

prvi kolokvijum

- 1) [25] **Java2D:** Napisati klasu sa potrebnim atributima i metodom `paint()` koja u prozoru crta figuru prikazanu na slici 1. Figura predstavlja baštenske makaze. Sečiva su obojena sivo, kraci su crveni, a drške crne. Sečiva su dodatno oivičena crnom. Oštra ivica sečiva se nalazi pod uglom od 20° u odnosu na produženu spoljašnju ivicu kraka. Centar figure (osa rotacije makaza) se nalazi u centru prozora. Veličina makaza je proporcionalna najmanjoj dimenziji prozora. Tekuća vrednost ugla pod kojim se sečiva međusobno nalaze 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 kvadratne Bezejeove krive definisane nad tačkama P_0 , P_1 , i P_2 , u zavisnosti od realnog parametra t . Tačke P_0 i P_2 su početna i krajnja tačka krive, respektivno. Napisati matrični oblik kvadratne Bezejeove krive. Kako zavisi oblik krive od kontrolne tačke P_1 , ako je $P_1 = (P_0 + P_2)/2$?
- 3) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Koje vrste dinamike postoje u računarskoj animaciji?
 - Koje vektorski zadate nepopunjene geometrijske primitive podržava SRGP?
 - Navesti osobine ekranskog kanvasa SRGP-a.
 - Koje tipove eha lokatora podržava SRGP i koja je namena svakog od tih tipova?
 - Napisati elementarnu matricu rotacije tačke (prema konvenciji pokretnog objekta) oko koordinatnog početka desnog koordinatnog sistema za ugao α u smeru kazaljke na časovniku.

Napomene:

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

Rešenja zadatka

prvi kolokvijum 2012

1) Rešenje

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
public class Kl_1112 extends JPanel implements Runnable{
    private int ciljno_vreme_azuriranja = 40; // 40 ms; -> 25 fps
    private double tekuciUgao;
    private double ugaonaBrzina = 2;
    private static final int x[] = {-3, 25, 14, -3};
    private static final int y[] = {-2, -2, 5, 1};
    private Polygon secivo = new Polygon(x, y, 4);
    private static final double duzina = 60;
    private Stroke olovka = new BasicStroke(0);

    private void crtajPolovinu(Graphics2D g2d) {
        Paint p = g2d.getPaint();
        Stroke s = g2d.getStroke();
        AffineTransform at = g2d.getTransform();

        g2d.setStroke(olovka);
        g2d.setPaint(Color.gray);
        g2d.fill(secivo);
        g2d.setPaint(Color.black);
        g2d.draw(secivo);

        g2d.scale(-1, 1);
        g2d.translate(3,1);
        g2d.rotate(Math.toRadians(-20));
        g2d.setPaint(Color.red);
        g2d.fillRect(0, -3, 12, 3);
        g2d.translate(12, -2);
    }
}
```

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

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

Ako je $P_1 = \frac{P_0 + P_2}{2}$, ne dobija se kvadratna Bezeova kriva, već prava linija: $P(t) = (1-t)P_0 + tP_2, t \in [0,1]$

```
g2d.setPaint(Color.black);
g2d.fillRect(0, -2, 15, 5);

g2d.setStroke(s);
g2d.setTransform(at);
g2d.setPaint(p);
}

public void paint (Graphics g) {
    super.paint(g);
    Graphics2D g2d = (Graphics2D) g;
    Dimension dim = getSize();
    double velicina = Math.min(dim.width, dim.height)/duzina;
    g2d.translate(dim.width/2, dim.height/2);
    g2d.scale(velicina, velicina);
    AffineTransform prethodnaTransformacija = g2d.getTransform();
    g2d.rotate(tekuciUgao);
    crtajPolovinu(g2d);
    g2d.setTransform(prethodnaTransformacija);
    g2d.scale(1,-1);
    g2d.rotate(tekuciUgao);
    crtajPolovinu(g2d);
    g2d.setPaint(Color.black);
    g2d.fillOval(-1,-1, 2, 2);
    g2d.setTransform(prethodnaTransformacija);
}

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

