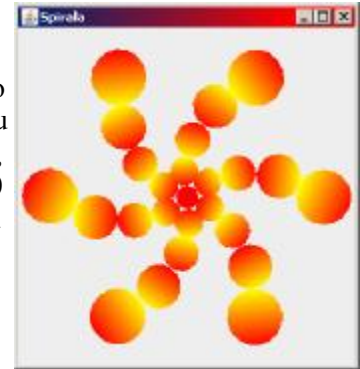


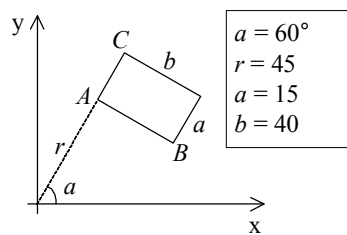
## RAČUNARSKA GRAFIKA prvi kolokvijum

- 1) [25] **Java2D:** Napisati klasu sa potrebnim atributima i metodom `paint()` koja u prozoru crta centralnosimetričnu figuru prikazanu na slici 1. Figura je sastavljena od krugova obojenih valerima od žute do crvene boje, izuzev centralnog kruga koji je potpuno crven. Krugovi su raspoređeni duž 6 krakova tako da se dva susedna kruga u kraku dodiruju, a duž koja spaja centre dva susedna kruga se nalazi pod uglom od 20 stepeni u odnosu na duž koja spaja centre prethodna dva uzastopna kruga. Poluprečnik kruga u kraku je za 25% veći od prethodnog, posmatrano od centra ka periferiji. Poluprečnik kruga u centru je 10. Figura rotira oko svog centra konstantnom ugaonom brzinom. 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.

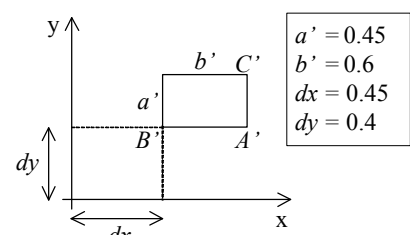


Slika 1

- 2) [25] Posmatraju se objekti u desnom 2D koordinatnom sistemu. Potrebno je izvršiti preslikavanje objekata iz pravougaonog prozora (*window*) definisanog slikom 2a u prikazni prozor (*viewport*) definisanog slikom 2b, pri čemu



Slika 2a



Slika 2b

- se tačke prozora A, B i C preslikavaju u tačke prikaznog prozora A', B' i C', respektivno. Postaviti jednu matičnu jednačinu preslikavanja tačaka iz zadatog prozora u zadati prikazni prozor. U sve matrice elementarnih transformacija uvrstiti konkretne vrednosti. Nije potrebno množiti matrice.

- 3) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Koja su osnovni ciljevi disciplina računarske grafike i računarske vizije (*computer vision*)?
  - Koliko je potrebno bita video memorije po pikselu, a koliko po ulazu tabele boja (VLT – *Video Lookup Table*) da bi se na slici moglo istovremeno predstaviti 256 boja iz palete od 65536 boja?
  - Koji pravougaonici određuju rezultat kopiranja piksel SRGP operacijom CopyPixel?
  - Navesti vrste logičkih grafičkih ulaznih uređaja (prema paketima GKS i PHIGS) i za svaki navesti koju informaciju vraća.
  - Kako se izračunava kompozitna matrica složene transformacije ako se tačka predstavlja kao vektor-kolona? Dokazati.

### Napomene:

- Kolokvijum traje 105 minuta.
- Nije dozvoljena upotreba literature niti programabilnih kalkulatora.
- Dozvoljena je upotreba AWT i Java2D podsetnika.

# Rešenja zadataka

## prvi kolokvijum 2010

### 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 double tekuciUgao;
    private double ugaonaBrzina = 10;
    private int brojLoptica = 4;
    private double inkrementUgla = 20;
    private double koeficijentUvecanja = 1.25;
    private double velicina = 20;
    private AffineTransform tr = new AffineTransform();
    private Ellipse2D krug = new Ellipse2D.Double(-0.5, -0.5, 1, 1);
    private Paint crvena = Color.red;
    private Paint valer1 = new GradientPaint(-0.5f,0,Color.red,
                                           0.5f,0,Color.yellow);
    private Paint valer2 = new GradientPaint(-0.5f,0,Color.yellow,
                                           0.5f,0,Color.red);

    private void crtajSegment(Graphics2D g2d) {
        for(int i = 0; i < brojLoptica; i++) {
            g2d.translate(0.5,0);
            g2d.scale(koeficijentUvecanja, koeficijentUvecanja);
            g2d.translate(0.5,0);
            if( i % 2 == 0 ) g2d.setPaint(valer1);
            else g2d.setPaint(valer2);
            g2d.fill(krug);
            g2d.rotate(inkrementUgla*Math.PI/180);
        }
    }
}
```

```
public void paint (Graphics g) {
    super.paint(g);
    Graphics2D g2d = (Graphics2D) g;
    Dimension dim = getSize();
    AffineTransform prethodnaTransformacija = g2d.getTransform();
    g2d.translate(dim.width/2, dim.height/2);
    g2d.scale(velicina, velicina);
    g2d.setPaint(crvena);
    g2d.fill(krug);
    g2d.rotate(tekuciUgao*Math.PI/180);
    AffineTransform tx = g2d.getTransform();
    for(int i = 0; i < 6; i++){
        g2d.rotate(i*Math.PI/3);
        crtajSegment(g2d);
        g2d.setTransform(tx);
    }
    g2d.setTransform(prethodnaTransformacija);
}

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

$$2) M = R(\alpha-\pi/2) * T(0,r) * S(-b'/b, a'/a) * S(-1,1) * T(-dx-b',dy)$$

$$R(-\alpha) = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0.5 & 0 \\ -0.5 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T(0,r) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -45 & 1 \end{bmatrix},$$

$$S\left(-\frac{b'}{b}, \frac{a'}{a}\right) = \begin{bmatrix} -0.015 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T(-dx-b',-dy) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1.05 & 0.4 & 1 \end{bmatrix}$$