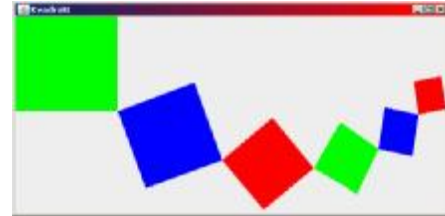


## ISPIT IZ RAČUNARSKE GRAFIKE

### Prvi deo

- 1) [50] **Java2D:** Napisati klasu sa potrebnim atributima, metodom `paint()` koja u prozoru crta niz kvadrata prikazan na slici 1 i glavnim programom. Kvadrati se ciklički boje zelenom, plavom i crvenom bojom. Ulazni parametri za crtanje su broj kvadrata, veličina stranice početnog kvadrata (čije je gornje levo teme smešteno u koordinatnom početku), korak ugla rotacije i faktor promene veličine stranice narednog kvadrata u odnosu na prethodni.

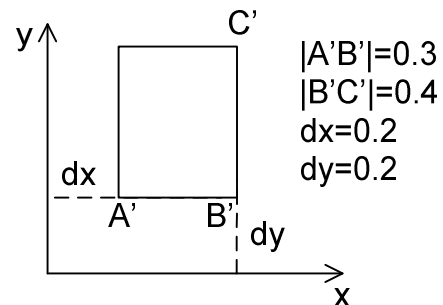


Slika 1

- 2) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- U kojem režimu (`KeyboardMode`) može da radi SRGP tastatura i koje su karakteristike tih režima?
  - Kako se izračunava kompozitna matrica složene transformacije, ako se tačka predstavlja kao vektor-vrsta? Dokazati.

### Drugi deo

- 3) [50] U desnom 3D koordinatnom sistemu, najpre se vrši ogledanje u ogledalu postavljenom na ravan  $z=3$ . Slika proizvoljne tačke dobijene ovim ogledanjem se zatim projektuje sa perspektivom iz centra projekcije  $P(-4,0,-4)$  na projekcionu ravan  $x=0$ . Delove projekcije koji se nalaze u prozoru prikazne ravni definisanim sa  $x \in [-4,4]$ ,  $y \in [-3,3]$ , potrebno je preslikati u prikazni prozor uređaja sa normalizovanim koordinatama, tako da se stranica  $y=-3$  preslika u stranicu prikaznog prozora  $A'B'$ , pri čemu se tačka prozora  $(-4,-3)$  preslikava u tačku prikaznog prozora  $A'$ .



Slika 2

Postaviti matičnu jednačinu preslikavanja proizvoljne tačke iz zadatog 3D prostora u dati prikazni prozor. U sve matrice elementarnih transformacija koje učestvuju u matičnim jednačinama uvrstiti konkretne vrednosti. Nije potrebno množiti matrice.

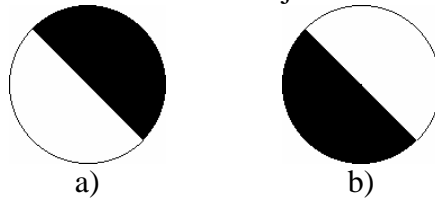
- 4) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Šta su to osne iščezavajuće tačke?
  - Kako se definiše koordinatni sistem prikaza (*Viewing Reference Coordinates, VRC*) u SPHIGS-u?

### Treći deo

- 5) Korišćenjem Bresenham-ovog algoritma za crtanje kružnice kao osnove, potprograma za crtanje linije `Line(x1, y1, x2, y2)` koji crta liniju od tačke `(x1, y1)` do tačke `(x2, y2)`, i potprograma za crtanje tačke `Plot(x, y)` koji crta tačku na koordinatama `(x, y)`, napisati potprogram čija je deklaracija

`void Crtaj(int x1, int y1, int x2, int y2, bool mod);`

gde `mod` određuje da li će se crtati figura 3a ili figura 3b. Na slici 3, **mod** je `true` za figuru 3b. `(x1, y1)` predstavlja donji levi, a `(x2, y2)` gornji desni ugao figure. Pretpostaviti da je pre poziva potprograma izvršena provera validnosti datih koordinata. Crtanje se vrši crnom olovkom na beloј pozadini.



Slika 3

- 6) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- a) Navesti nedostatke Z-bafer algoritma.
  - b) Definirati pojam aditivnog i subtraktivnog modela boja, navesti kada se koji primenjuje i dati primere takvih modela navodeći njihove osnovne boje.

### Napomene:

1. Ispit traje 180 minuta.
2. Nije dozvoljena upotreba literature niti programabilnih kalkulatora.
3. Dozvoljena je upotreba AWT, Java2D i OpenGL podsetnika.
4. Započeta izrada nekog dela poništava eventualno ranije osvojeni broj poena na tom delu (kolokvijumu).

## Rešenja zadataka JUN 2008

### 1) Rešenje zadatka Java2D

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;

public class K1J extends JPanel
{
    int duzina_ivice, broj_figura;
    double inkrement_ugla, skaliranje;
    Color boje[] = { Color.RED, Color.GREEN, Color.BLUE };
    AffineTransform tr = new AffineTransform();
    Rectangle2D rect1 = new Rectangle2D.Double();

    K1J(int di, int bf, double iu, double s)
    {
        duzina_ivice = di;
        broj_figura = bf;
        inkrement_ugla = iu;
        skaliranje = s;
        rect1.setRect(0, 0, di, di);
    }

    public void paint (Graphics g)
    {
        super.paint(g);

        Graphics2D g2d = (Graphics2D) g;

        Dimension d = getSize();

        tr.setToIdentity();
        for( int i = 1; i <= broj_figura; i++ )
        {
            g2d.setTransform(tr);
            g2d.setPaint( boje[i%3] );

            g2d.fill(rect1);

            tr.translate( duzina_ivice, duzina_ivice );
            tr.scale(skaliranje, skaliranje);
            tr.rotate( -inkrement_ugla*Math.PI/180.0 );
        }
    }

    public static void main(String s[])
    {
        K1J panel = new K1J(128, 6, 20, 0.8);

        panel.setOpaque(true);

        JFrame frame = new JFrame("Kvadrati");
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {System.exit(0);}
        });

        frame.getContentPane().add(panel);
        frame.setSize(new Dimension(800,600));
        frame.setVisible(true);
    }
}
```

## 2) Rešenje zadatka – 3D/2D transformacije

$$[X' \ Y' \ W] = [X \ Y \ Z \ 1] * M$$

$$[X'' \ Y'' \ 1] = [X'/W \ Y'/W \ 1]$$

$$M = T(0, 0, 3) * O_{XY} * T(0, 0, -7) * R_Y(-\frac{\pi}{2}) * P_p(d=4) * T(-4, -3) * S(\frac{0.3}{8}, \frac{0.4}{6}) * T(-0.2, -0.2)$$

$$T(0, 0, 3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 1 \end{bmatrix}, \quad O_{XY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T(0, 0, -7) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 7 & 1 \end{bmatrix}, \quad R_Y(-\frac{\pi}{2}) = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$P_p(d=4) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-4, -3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}, \quad S(\frac{0.3}{8}, \frac{0.4}{6}) = \begin{bmatrix} \frac{0.3}{8} & 0 & 0 \\ 0 & \frac{0.4}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T(-0.2, -0.2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.2 & 0.2 & 1 \end{bmatrix}$$

## 3) Rešenje zadatka sa algoritmom za rasterizaciju figure

```
void crtaJ(int x1, int y1, int x2, int y2, bool mod) {
    int d, x, y;
    int xc = (x1+x2)>>1;
    int yc = (y1+y2)>>1;
    /* OKVIR, nije trazeno u postavci
        line(x1, y1, x2, y1);
        line(x2, y1, x2, y2);
        line(x2, y2, x1, y2);
        line(x1, y2, x1, y1);
        line(x1, y1, x2, y2);
    */
    x = (x2-x1)/2;
    y = 0;
    d = 3-2*x;
    while(x >= y) {
        if( ! mod ) {
            line(xc+y, yc+y, xc+x, yc+y);
            line(xc-y, yc-y, xc+x, yc-y);
            line(xc-y, yc-x, xc+y, yc-x);

            plot(xc-x, yc+y);
            plot(xc-x, yc-y);
            plot(xc-y, yc+x);
            plot(xc+y, yc+x);
        }
    }
}
```

```
    else {
        line(xc-x, yc+y, xc+y, yc+y);
        line(xc-x, yc-y, xc-y, yc-y);
        line(xc-y, yc+x, xc+y, yc+x);

        plot(xc+x, yc+y);
        plot(xc+x, yc-y);
        plot(xc-y, yc-x);
        plot(xc+y, yc-x);
    }

    if( d < 0 ) d += 4*y+6;
    else {
        d += 4 * (y-x)+10;
        x--;
    }
    y++;
}
```