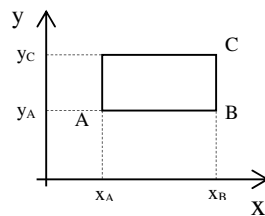


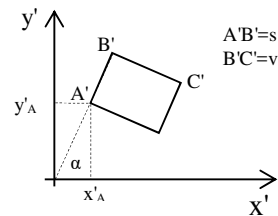
## RAČUNARSKA GRAFIKA

### prvi kolokvijum – teoretski deo

- 1) [20] Moodle test
- 2) [10] U desnom 2D koordinatnom sistemu se nalaze neke geometrijske figure. Potrebno je izvršiti preslikavanje figura iz pravougaonog prozora (*window*) definisanog slikom 2a u prikazni prozor (*viewport*) definisanog slikom 2b, pri čemu se tačke prozora A, B i C preslikavaju u tačke prikaznog prozora A', B' i C', respektivno. Postaviti jednu matričnu jednačinu preslikavanja karakterističnih tačaka figura iz zadatog prozora u zadati prikazni prozor. U sve matrice elementarnih transformacija uvrstiti numeričke ili algebarske vrednosti. Nije potrebno množiti matrice.



Slika 2a



Slika 2b

#### Napomene:

1. Izrada teoretskog dela (Moodle testa i zadatka 2) traje 20 minuta.
2. Rešenje 2. zadatka se predaje na priloženom listu.
3. Nije dozvoljeno korišćenje literature i podsetnika u bilo kom obliku.
4. Nije dozvoljeno uz sebe imati mobilni telefon, bez obzira da li je uključen ili isključen.

**RAČUNARSKA GRAFIKA**  
***prvi kolokvijum – praktični deo***

- 1) [70] Napisati program koji koristi grafičku biblioteku JavaFX i korisniku omogućava da klikom stvara figure na lokaciji kurzora. Veličina figura se menja tokom vremena. Figura se sastoji od 5 koncentričnih kvadrata koji se rotiraju oko centra u smeru kazaljke na satu. Nominalna veličina stranice najmanjeg kvadrata je 20, a svakog sledećeg u seriji za 20 veća. Svaki od njih ima transparentnu unutrašnjost i ivicu slučajne boje. Periodi rotacije svih kvadrata su isti, ali unutrašnji kvadrati pre započinju i pre završavaju sa rotacijom, proporcionalno svojem mestu u seriji koncentričnih kvadrata. Trajanje jednog ciklusa je 2s, a najveći kvadrat kasni sa rotacijom 0.4s u odnosu na najmanji. Početna veličina stranice se kreće u opsegu [0.5, 1.0] od nominalne veličine, a završna u opsegu [1.0, 2.0] od zadate veličine. Priložen je .jar fajl sa demonstracijom programa koji treba napisati.

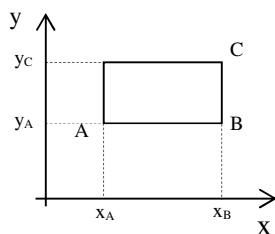
**Napomene:**

1. Izrada praktičnog dela traje 100 minuta.
2. Rešenje zadatka se predaje u obliku .java fajla u predviđenom folderu na računaru.
3. Dozvoljena je upotreba literature koja je stavljena na raspolaganje i pristup Oracle sajtu.
4. Nije dozvoljeno uz sebe imati mobilni telefon, bez obzira da li je uključen ili isključen.

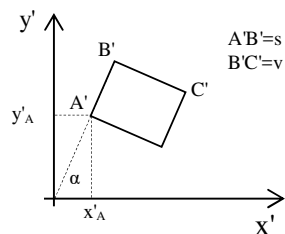
## Rešenja zadatka

### prvi kolokvijum – teoretski deo

2)



Slika 2a



Slika 2b

Kompozitna matrica:

$$M = T(x_A, y_A) \times S(-1, 1) \times R(\pi/2) \times S(v/(y_C - y_A), s/(x_B - x_A)) \times R(\pi/2 - \alpha) \times T(-x'_A, -y'_A)$$

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_A & -y_A & 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{v}{y_C - y_A} & 0 & 0 \\ 0 & \frac{s}{x_B - x_A} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \times \begin{bmatrix} \cos(\frac{\pi}{2} - \alpha) & -\sin(\frac{\pi}{2} - \alpha) & 0 \\ \sin(\frac{\pi}{2} - \alpha) & \cos(\frac{\pi}{2} - \alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x'_A & y'_A & 1 \end{bmatrix}$$

Matrična jednačina kojom se tačka P iz prozora preslikava u tačku P' u prikaznom prozoru:

$$P' = P \times M$$

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \times M$$

# Rešenja zadatka

## prvi kolokvijum – praktični deo

1)

```
package RG_1617_k1;
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.shape.Rectangle;
import javafx.scene.paint.Color;
import javafx.animation.*;
import javafx.util.Duration;

class Kvadrati extends Group {

    public Kvadrati() {
        ParallelTransition parTranz = new ParallelTransition();
        for (int i = 0; i < 5; i++) {
            Rectangle kvadrat = new Rectangle();
            kvadrat.setWidth((i+1)*20);
            kvadrat.setHeight((i+1)*20);
            kvadrat.setLayoutX(-(i+1)*10);
            kvadrat.setLayoutY(-(i+1)*10);
            kvadrat.setStroke(Color.color(Math.random(), Math.random(), Math.random()));
            kvadrat.setFill(Color.TRANSPARENT);

            SequentialTransition sekTranz = new SequentialTransition(kvadrat);
            PauseTransition počPauza = new PauseTransition(Duration.millis(i*100));
            RotateTransition rotTranz = new RotateTransition(Duration.millis(1600));
            rotTranz.setByAngle(360);
            PauseTransition krajPauza = new PauseTransition(Duration.millis((4-i)*100));
            sekTranz.getChildren().addAll(počPauza, rotTranz, krajPauza);
            sekTranz.setCycleCount(Animation.INDEFINITE);

            parTranz.getChildren().add(sekTranz);
            this.getChildren().add(kvadrat);
        }

        ScaleTransition skalTranz = new ScaleTransition(Duration.millis(2000));
        skalTranz.setAutoReverse(true);
        skalTranz.setCycleCount(Animation.INDEFINITE);

        double pocetniSF = 0.5 + Math.random()/2;
        double zavrzniSF = 1.0 + Math.random();

        skalTranz.setFromX(pocetniSF);
        skalTranz.setFromY(pocetniSF);

        skalTranz.setToX(zavrzniSF);
        skalTranz.setToY(zavrzniSF);

        skalTranz.setNode(this);
        skalTranz.play();
        parTranz.play();
    }
}

public class K1_2017 extends Application {
    @Override
    public void start(Stage prozor) {
        Group koren = new Group();
        Scene scena = new Scene(koren, 600, 600);
        scena.setOnMouseClicked(dog -> {
            Kvadrati figura = new Kvadrati();
            figura.setTranslateX(dog.getX());
            figura.setTranslateY(dog.getY());
            koren.getChildren().add(figura);
        });
        prozor.setTitle("Kvadrati");
        prozor.setScene(scena);
        prozor.show();
    }
    public static void main(String[] arg) { launch(arg); }
}
```