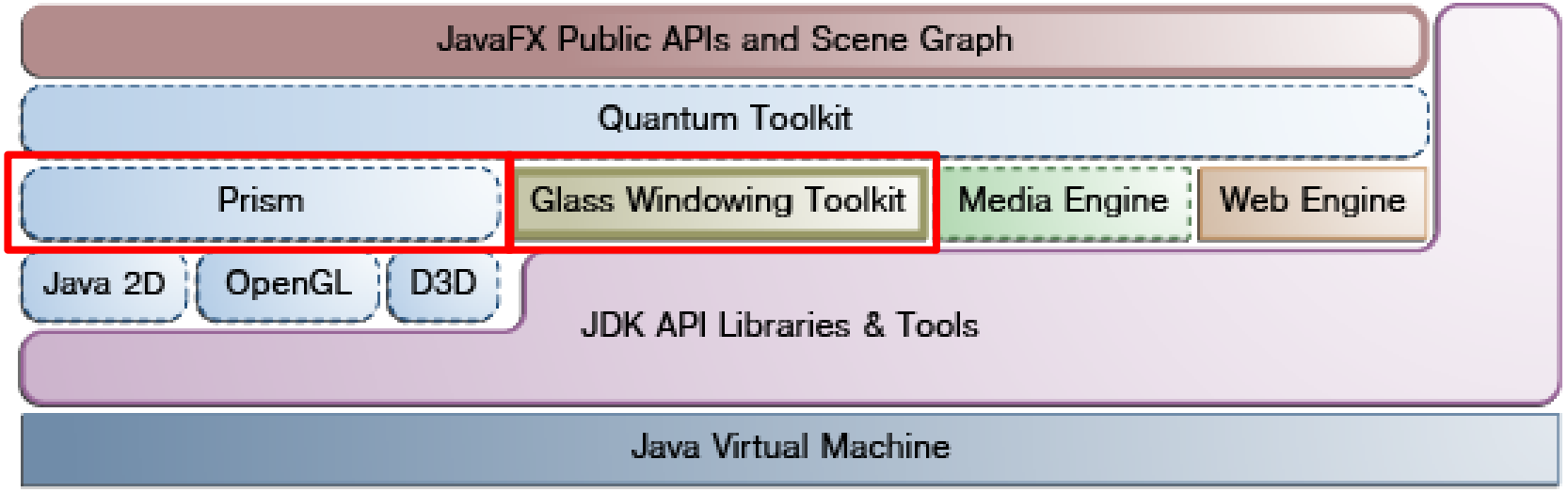
Računarska grafika - vežbe

1 – JavaFX

uvod, primitive, atributi i transformacije

JavaFX - arhitektura



Prism:

- grafička mašina – obavlja crtanje (hardversko ili softversko)

Glass Windowing Toolkit (GWT):

•komunikacija sa OS, upravlja prozorima, tajmerima i površima za crtanje

•upravlja redom za čekanje događaja

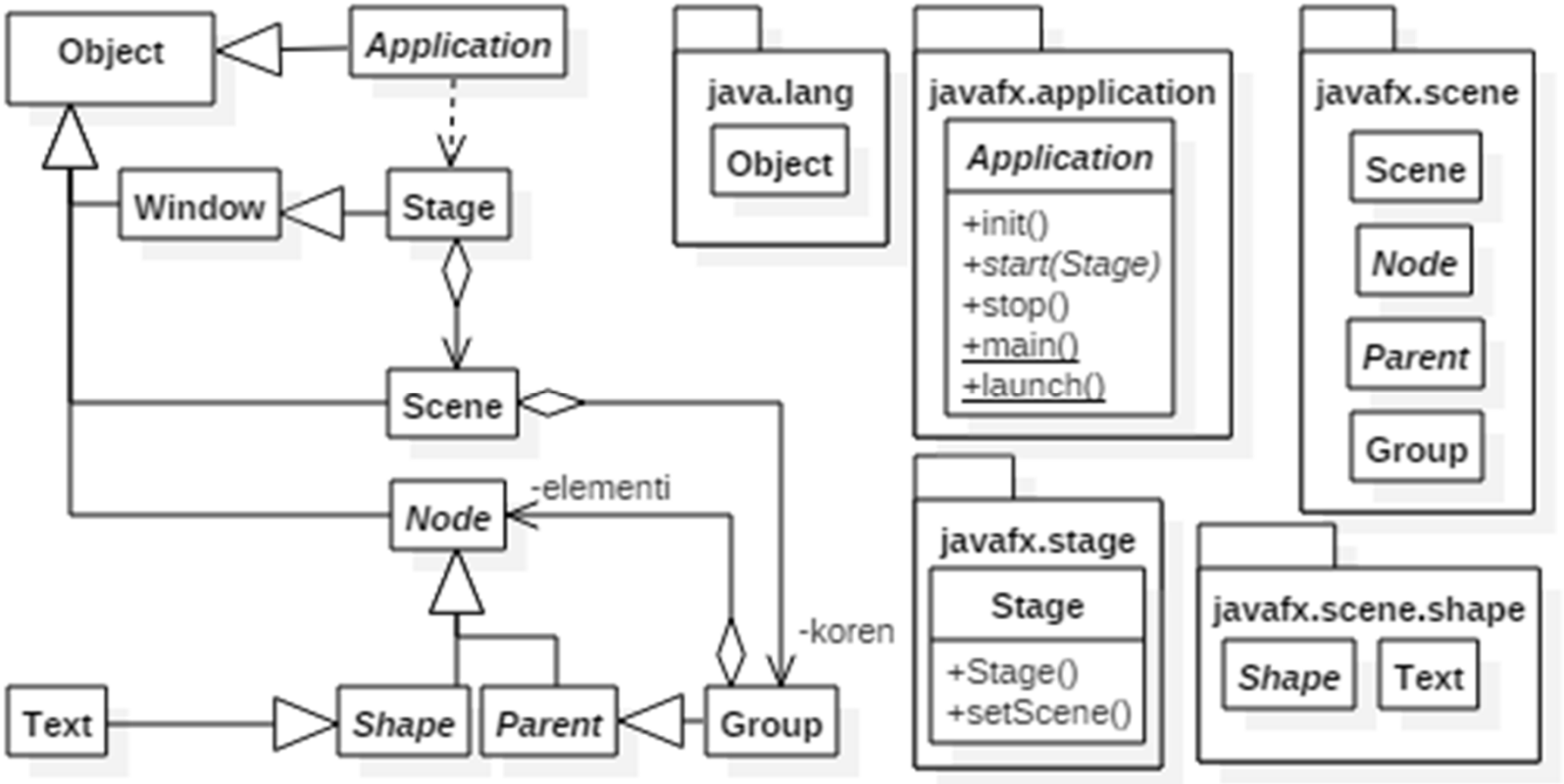
•izvršava događaje pulsa (*Pulse*)

JavaFX – višenitno okruženje

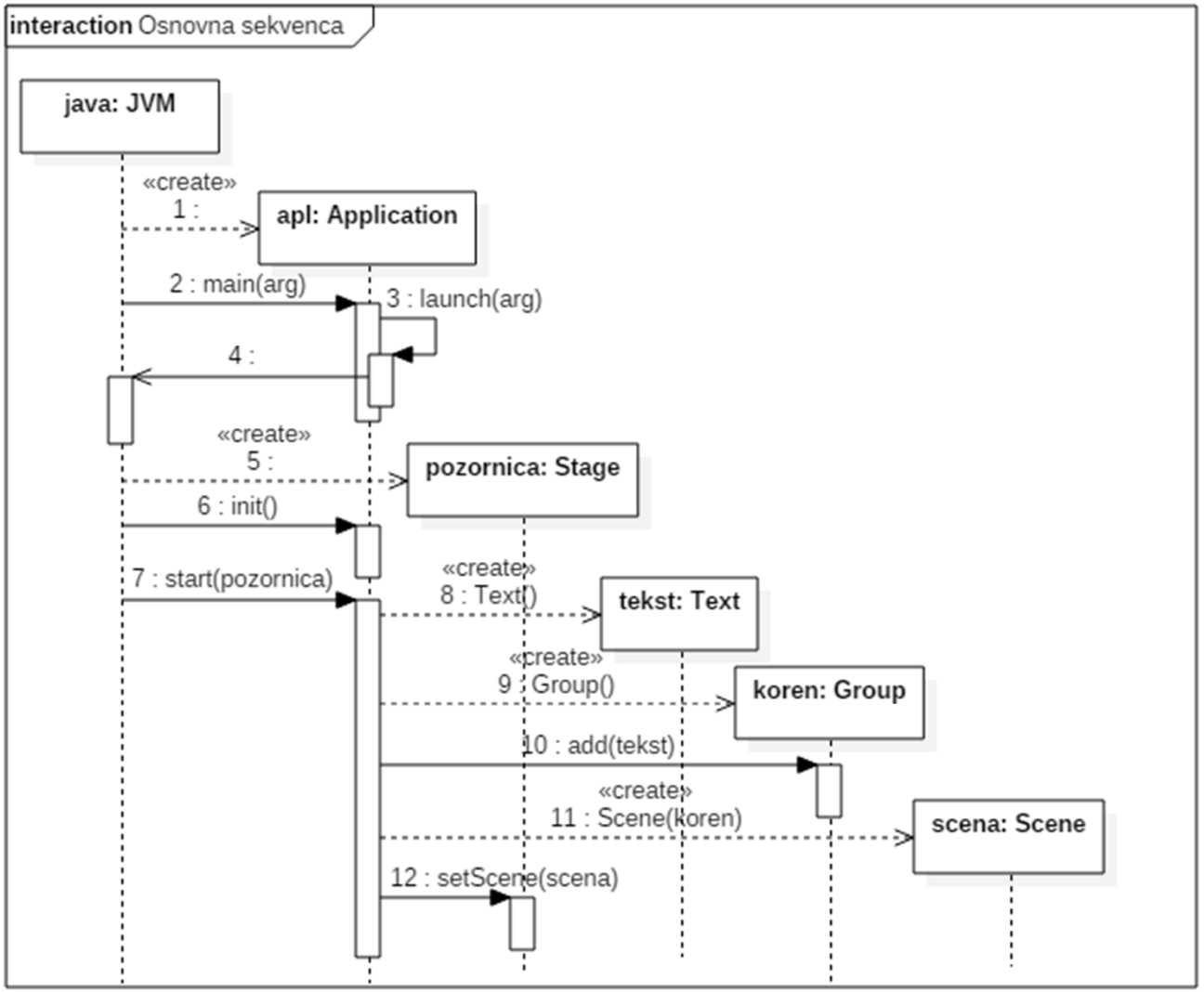
- Više niti koje se konkurentno izvršavaju
  - Nit JavaFX aplikacije
    - primarna nit, izvršava metode `start()` i `stop()`, dispečer događaja
    - svaka JavaFX scena (pridruženi graf scene) **koja se prikazuje** mora biti menjana isključivo kroz ovu nit
    - druge scene mogu biti pripremljene u drugim nitima
  - Nit *Prism* komponente
    - crta nezavisno od dispečera događaja
    - dozvoljava simultano crtanje jedne slike i pripremu naredne
  - Nit za medije
    - sinhronizacija audio/video sadržaja
- Puls (eng. Pulse)
  - Znak da je potrebno sinhronizovati graf scene sa prikazom (nit *Prism*)
  - 60 fps, automatski se okida, sinhronizacija se vrši:
    - ako ima aktivnih animacija u sceni
    - ako ima izmena u sceni (promena stanja ili pozicije čvorova grafa)

Osnovna klasna infrastruktura

- UML dijagram važnih klasa i paketa kojima pripadaju

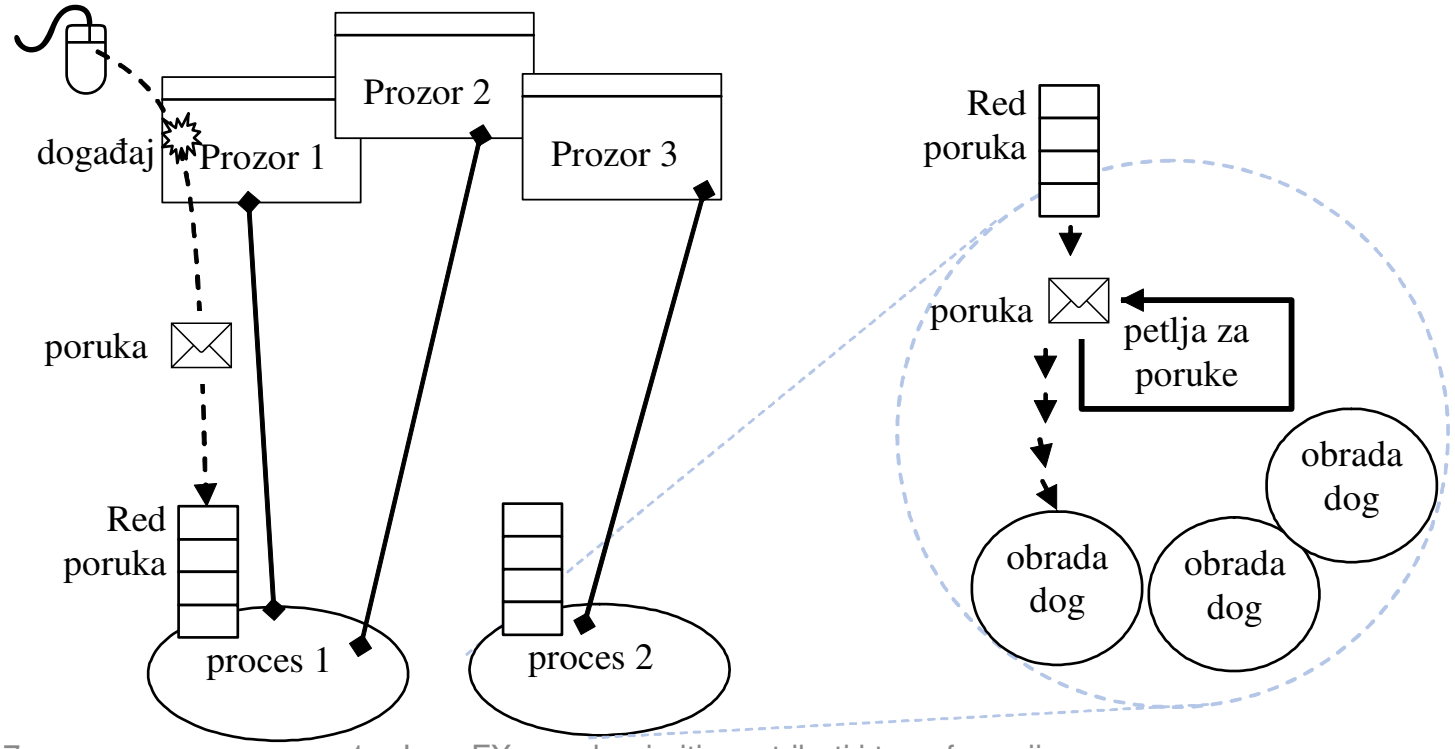


Osnovna sekvenca interakcije



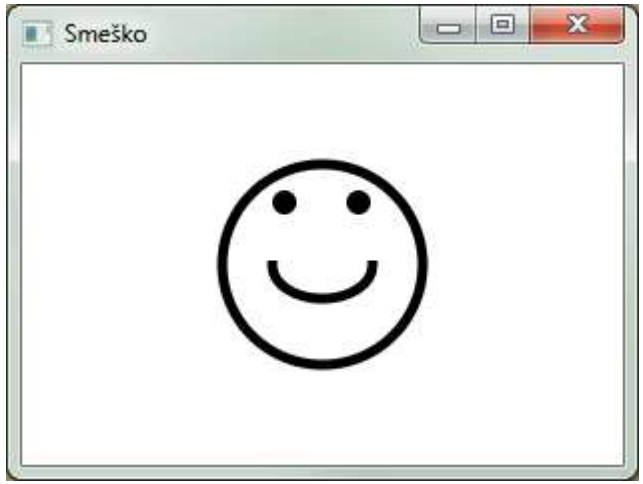
Obrada događaja

- Dobija punu afirmaciju kroz sisteme prozora
- Svaki događaj (*event)* u sistemu praćen je porukom:
  - događaji koje iniciraju korisnici pomoću ulaznih uređaja
  - događaji koje inicira OS u nekim važnim trenucima



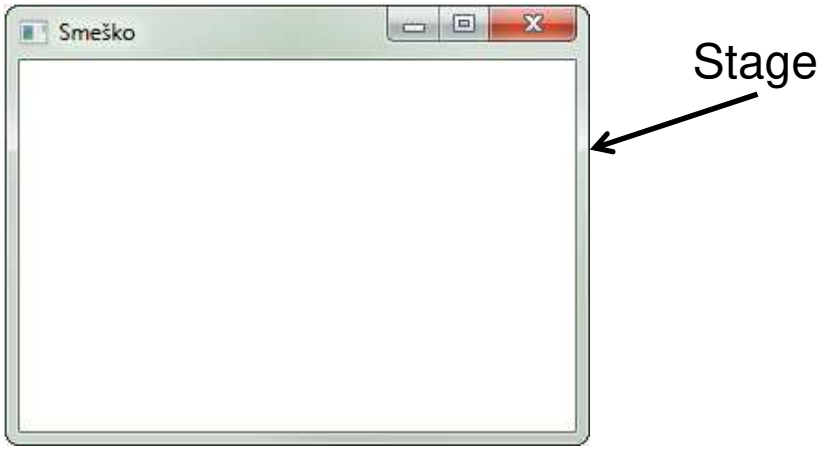
Zadatak 1: Smeško

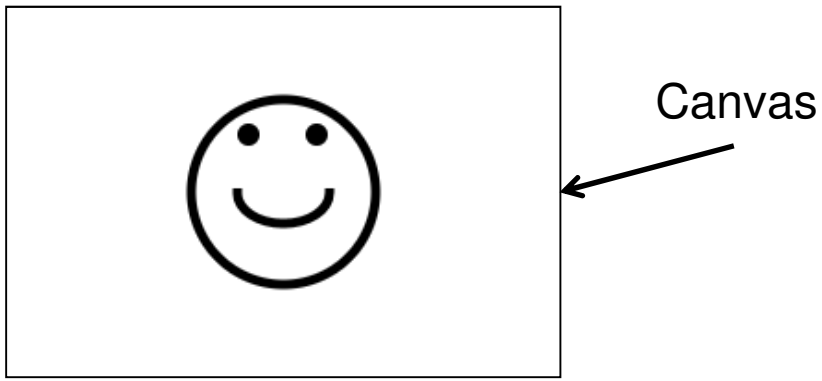
- Napisati program koji u prozoru crta i prikazuje lik "Smeška" upotrebom biblioteke JavaFX

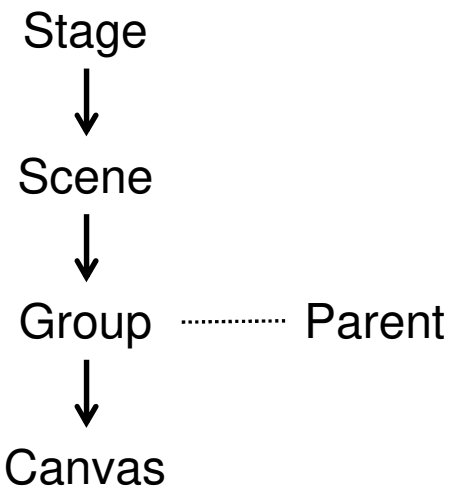


Rešenje – na kanvasu (1/4)

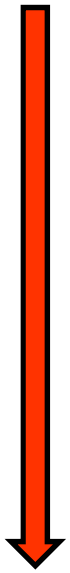
- Upotrebom "platna" za crtanje željenog sadržaja
  - ovo rešenje je u duhu Java2D biblioteke







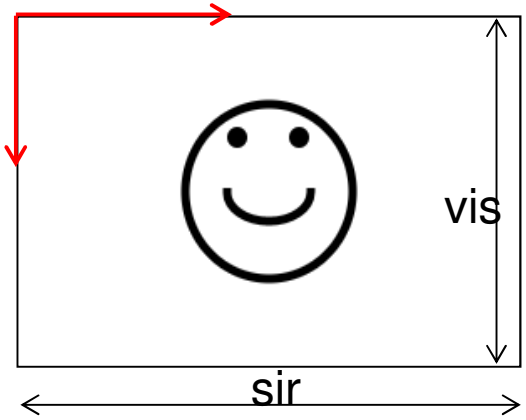
Rešenje – na kanvasu (2/4)



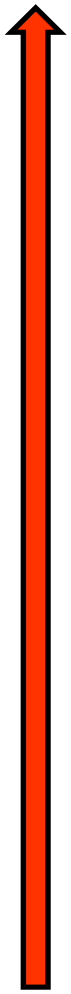
```
package smeško_java2D;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.shape.ArcType;
```

Rešenje – na kanvasu (3/4)

```
public class Smeško1 extends Application{
  @Override
  public void start(Stage prozor) {
    int sir = 300;
    int vis = 200;
    Canvas platno = new Canvas(sir, vis);
    GraphicsContext gk = platno.getGraphicsContext2D();
    gk.setLineWidth(5);
    gk.strokeOval(sir/3, vis/4, sir/3, vis/2);
    gk.strokeArc(sir/2-sir/12, vis/2-vis/12,
                 sir/6, vis/6, 180, 180, ArcType.OPEN);
    gk.fillOval(sir/2-2*sir/24, vis/2-3*vis/16,
                sir/24, vis/16);
    gk.fillOval(sir/2+sir/24, vis/2-3*vis/16,
                sir/24, vis/16);
```



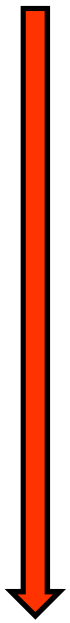
Rešenje – na kanvasu (4/4)



```
        Group koren= new Group();
        koren.getChildren().add(platno);

        Scene scena = new Scene(koren);
        prozor.setTitle("Smeško 1");
        prozor.setScene(scena);
        prozor.show();

    }

    public static void main(String[] arg) {
        launch(arg);
    }

}
```

Rešenje – JavaFX scena (1/3)

- Upotrebom grafa scene za pripremu i crtanje željenog sadržaja
  - ovo rešenje je u duhu JavaFX biblioteke

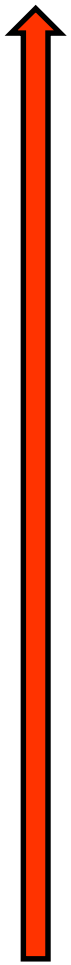


```
package smeško_javaFX;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.shape.Arc;
import javafx.scene.shape.ArcType;
import javafx.scene.shape.Circle;
import javafx.scene.paint.Color;
```

Rešenje – JavaFX scena (2/3)

```
public class Smeško2 extends Application {
  @Override
  public void start(Stage prozor) {
    int sir = 300;
    int vis = 200;
    Circle glava = new Circle(sir/2, vis/2, sir/6);
    glava.setFill(null);
    glava.setStroke(Color.BLACK);
    glava.setStrokeWidth(5);

    Circle lOko = new Circle(21*sir/48, 11*vis/32, sir/48);
    Circle dOko = new Circle(27*sir/48, 11*vis/32, sir/48);
```

Rešenje – JavaFX scena (3/3)



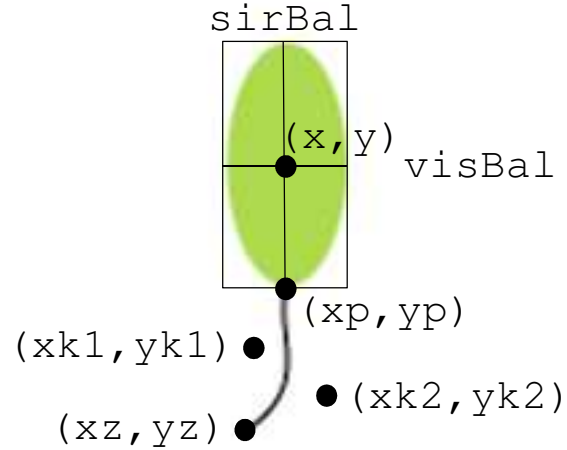
```
        Arc usta = new Arc(sir/2,vis/2,sir/12,vis/12,180,180);
        usta.setFill(null);
        usta.setStroke(Color.BLACK);
        usta.setStrokeWidth(5);
        usta.setType(ArcType.OPEN);

        Group koren = new Group();
        koren.getChildren().addAll(glava, usta, lOko, dOko);

        Scene scena = new Scene(koren);
        prozor.setTitle("Smeško 2");
        prozor.setScene(scena, sir, vis);
        prozor.show();
    }
    public static void main(String[] arg) { launch(arg); }
}
```

Zadatak 2: Baloni

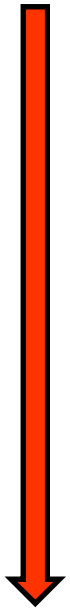
- Prikazati skup balona kao na slici.

Rešenje: Baloni (1/7)

- Balon je:
  - elipsa slučajne veličine
  - u slučajnoj boji, slučajne prozirnosti
  - končić vijori ispod balona
- Končić – kubna kriva sa 4 kontrolne tačke
- Svi baloni su unutar prozora, končići mogu izlaziti iz prozora



Rešenje: Baloni (2/7)



```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.paint.Color;
import javafx.scene.shape.Ellipse;
import javafx.scene.shape.CubicCurve;
import java.util.Random;
```

Rešenje: Baloni (3/7)

```
public class Baloni extends Application {
  @Override
  public void start(Stage prozor) {
    final int n=50;     // broj balona
    double sirProz=600; // sirina scene/prozora
    double visProz=400; // visina scene/prozora
    double maxSirBal=sirProz/10;    // maks sirina balona
    double minSirBal=0.75*maxSirBal;// min sirina balona

    Random rnd = new Random();
    Group koren = new Group();
```

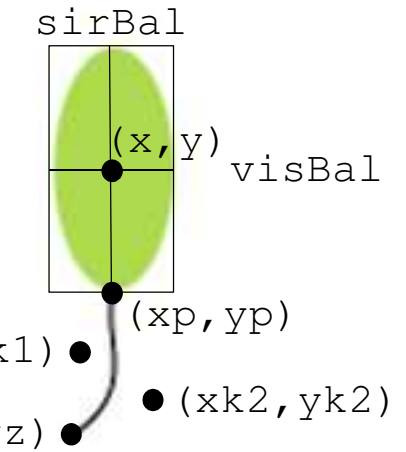
Rešenje: Baloni (4/7)

```
for (int i=0; i<n; i++) {
  // slučajna širina i visina balona
  double sirBal = rnd.nextDouble()*
                  (maxSirBal-minSirBal) + minSirBal;
  double maxVisBal = 2.0*sirBal;
  double minVisBal = 1.5*sirBal;
  double visBal = rnd.nextDouble()*
                  (maxVisBal-minVisBal) + minVisBal;

  //slučajna pozicija balona
  double x = rnd.nextDouble()*(sirProz-sirBal)+sirBal/2;
  double y = rnd.nextDouble()*(visProz-visBal)+visBal/2;
```

Rešenje: Baloni (5/7)

```
//slučajna boja i transparencija (u HSB sistemu)
double ton=rnd.nextDouble()*360;
double zasicenje=rnd.nextDouble()*0.3+0.7;
double sjaj=rnd.nextDouble()*0.3+0.7;
double neprozirnost = rnd.nextDouble()*0.5+0.4;
Color boja = Color.hsb(ton,zasicenje,sjaj,neprozirnost);

Ellipse bal = new Ellipse(x, y, sirBal/2, visBal/2);
bal.setFill(boja);
```

Rešenje: Baloni (6/7)



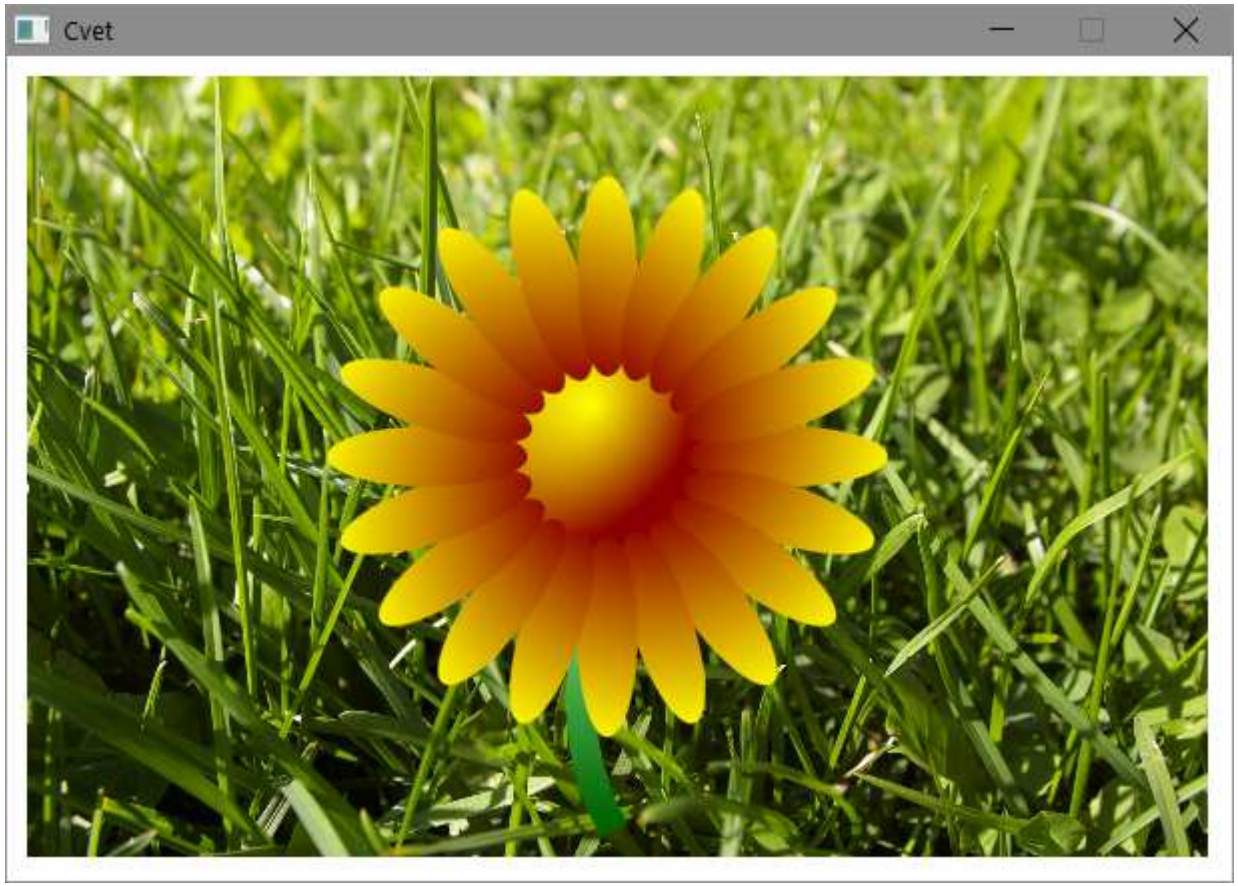

```
    //konac koji se slučajno vijori ispod balona
    double xp=x;
    double yp=y+visBal/2;
    double xk1=xp-sirBal/2+rnd.nextDouble()*sirBal;
    double yk1=yp+visBal/5;
    double xk2=xp-sirBal/2+rnd.nextDouble()*sirBal;
    double yk2=yp+2*visBal/5;
    double xz=xp-sirBal/2+rnd.nextDouble()*sirBal;
    double yz=yp+3*visBal/5;
    CubicCurve rep=new CubicCurve(xp,yp,xk1,yk1,xk2,yk2,xz,yz);
    rep.setFill(null);  rep.setStroke(Color.BLACK);
    koren.getChildren().addAll(bal,rep);
}
```

Rešenje: Baloni (7/7)

```
        Scene scena = new Scene(koren, sirProz, visProz);
        prozor.setTitle("Baloni");
        prozor.setScene(scena);
        prozor.setResizable(false);
        prozor.show();
    }

    public static void main(String[] arg) {
        launch(arg);
    }
}
```

Zadatak 3: Cvet

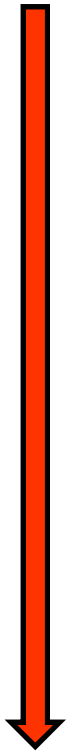
- Prikazati cvet u travi prema slici.



Rešenje: Cvet (1/13)

- Cvet se sastoji od delova:
  - glava
  - latice
  - stabljika
- Pozadina je slika trave
- Glava – krug u crveno-žutom valeru sa radijalnim prelazom i fokusom koji nije u centru glave
- Latice – elipse u crveno-žutom valeru sa linearnim prelazom
  - rotirane i translirane
- Stabljika – izvijena, u zelenom valeru sa linearnim prelazom
  - kubna kriva, nije popunjena, već je valer primenjen na liniju

Rešenje: Cvet (2/13)



```
import javafx.application.Application;

import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;

import javafx.scene.shape.Rectangle;
import javafx.scene.shape.Ellipse;
import javafx.scene.shape.Circle;
import javafx.scene.shape.CubicCurve;
```

Rešenje: Cvet (3/13)

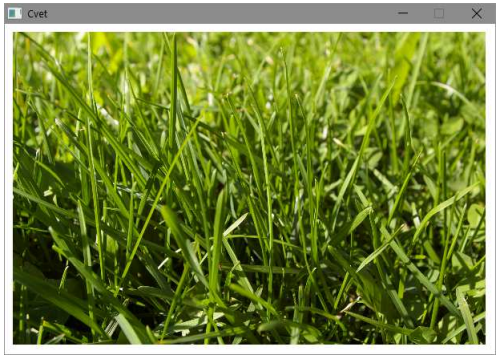
```
import javafx.scene.paint.Color;
import javafx.scene.paint.LinearGradient;
import javafx.scene.paint.RadialGradient;
import javafx.scene.paint.CycleMethod;
import javafx.scene.paint.Stop;
import javafx.scene.paint.ImagePattern;

import javafx.scene.transform.Translate;
import javafx.scene.transform.Rotate;
import javafx.scene.transform.Affine;

import javafx.scene.image.Image;
import java.util.Random;
```

Rešenje: Cvet (4/13)

```
public class Cvet extends Application {
  @Override
  public void start(Stage prozor) {
  final double sirProz=600;        // sirina scene/prozora
  final double visProz=400;        // visina scene/prozora
  final int nLatica=20;            // broj latica
  final double glavaR=50;          // poluprečnik glave
  final double latDuz=100;         // duzina latice cveta
  final double latSir=30;          // sirina latice cveta

  Random rnd = new Random();
  Group koren = new Group();
```

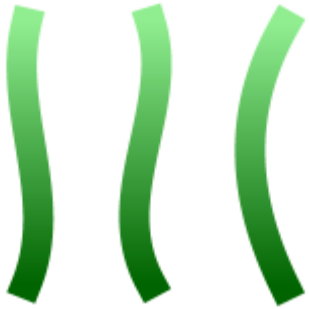
Rešenje: Cvet (5/13)

```
//crtanje pozadine
Image slika = new Image("teksture/trava.jpg");
ImagePattern teksturaPoz =
        new ImagePattern(slika, 0, 0, 1, 1, true);
Rectangle pozadina =
        new Rectangle(10, 10, sirProz-10, visProz-10);
pozadina.setFill(teksturaPoz);
koren.getChildren().add(pozadina);
```



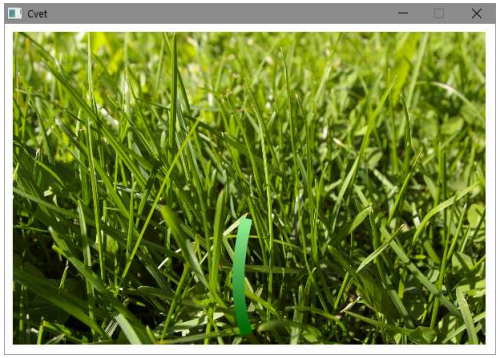
Rešenje: Cvet (6/13)

```
//crtanje stabljike
double xp=sirProz/2;
double yp=visProz/2+glavaR;
double xz=sirProz/2;
double yz=visProz-20;
double xk1=xp-glavaR/2+rnd.nextDouble()*glavaR;
double yk1=yp+(yz-yp)/3;
double xk2=xp-glavaR/2+rnd.nextDouble()*glavaR;
double yk2=yp+2*(yz-yp)/3;
```



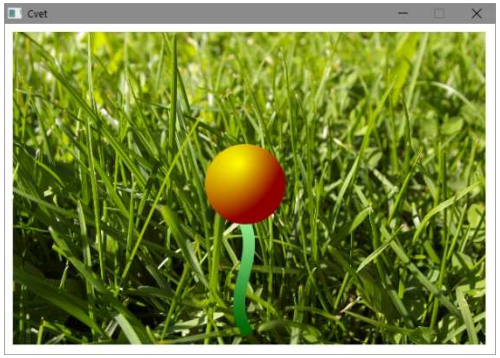
Rešenje: Cvet (7/13)

```
Stop[] staniceStabljika = new Stop[] {
        new Stop(0, Color.LIGHTGREEN),
        new Stop(1, Color.DARKGREEN)};
LinearGradient bojaStabljike =
        new LinearGradient(0.5, 0, 0.5, 1, true,
            CycleMethod.NO_CYCLE, staniceStabljika);
CubicCurve stabljika=
        new CubicCurve(xp,yp,xk1,yk1,xk2,yk2,xz,yz);
stabljika.setFill(null);
stabljika.setStroke(bojaStabljike);
stabljika.setStrokeWidth(15);
koren.getChildren().add(stabljika);
```



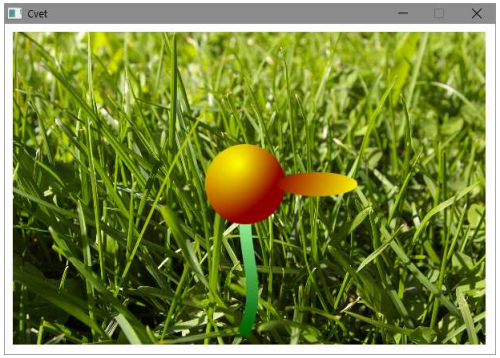
Rešenje: Cvet (8/13)

```
//crtanje glave cveta
Stop[] staniceGlaveCveta = new Stop[] {
        new Stop(0, Color.YELLOW),
        new Stop(1, Color.DARKRED)};
RadialGradient bojaGlaveCveta =
        new RadialGradient(30, 0.5, 0, 0, 1, true,
        CycleMethod.NO_CYCLE, staniceGlaveCveta);
Circle glavaCveta = new Circle(sirProz/2,visProz/2,glavaR);
glavaCveta.setFill(bojaGlaveCveta);
koren.getChildren().add(glavaCveta);
```



Rešenje: Cvet (9/13)

```
//crtanje latica
for (int i=0; i<nLatica; i++) {

  Stop[] staniceLatica = new Stop[] {
       new Stop(0, Color.DARKRED),
       new Stop(1, Color.YELLOW)};
  LinearGradient bojaLat = new LinearGradient(0, 0, 1, 1,
       true, CycleMethod.NO_CYCLE, staniceLatica);

  double ugao = (360/nLatica)*i;
```



Rešenje: Cvet (10/13)

```
/*
  // Elipsa se definiše u centru glave cveta,
  // zatim se vrši translacija na kružnicu oko glave,
  // pa rotacija oko centra objekta

  Ellipse latica =
      new Ellipse(sirProz/2, visProz/2, latDuz/2, latSir/2);
  double dx=(glavaR-5)*2*Math.cos(2*Math.PI*ugao/360);
  double dy=(glavaR-5)*2*Math.sin(2*Math.PI*ugao/360);
  latica.setTranslateX(dx);
  latica.setTranslateY(dy);
  latica.setRotate(ugao);
*/
```

Rešenje: Cvet (11/13)

```
/*
  // Elipsa se definiše na koncentričnoj kružnici oko glave,
  // t.j. na konačnoj poziciji latice,
  // pa se vrši samo rotacija oko centra objekta

  double dx=glavaR*2*Math.cos(2*Math.PI*ugao/360);
  double dy=glavaR*2*Math.sin(2*Math.PI*ugao/360);
  double x=sirProz/2+dx;
  double y=visProz/2+dy;
  Ellipse latica = new Ellipse(x, y, latDuz/2, latSir/2);
  latica.setRotate(ugao);
*/
```

Rešenje: Cvet (12/13)

```
  // Elipsa se definiše u centru glave cveta,
  // zatim se vrši rotacija oko centra objekta,
  // pa translacija samo po X u rotiranom sistemu objekta
  Ellipse latica =
    new Ellipse(sirProz/2, visProz/2, latDuz/2, latSir/2);
  Rotate rot = new Rotate(ugao,sirProz/2,visProz/2);
  Translate trans = new Translate(2*glavaR-10,0);
  Affine afin = new Affine();
  afin.append(rot); afin.append(trans);
  latica.getTransforms().add(afin);
  latica.setFill(bojaLat);
  koren.getChildren().add(latica);
}
```

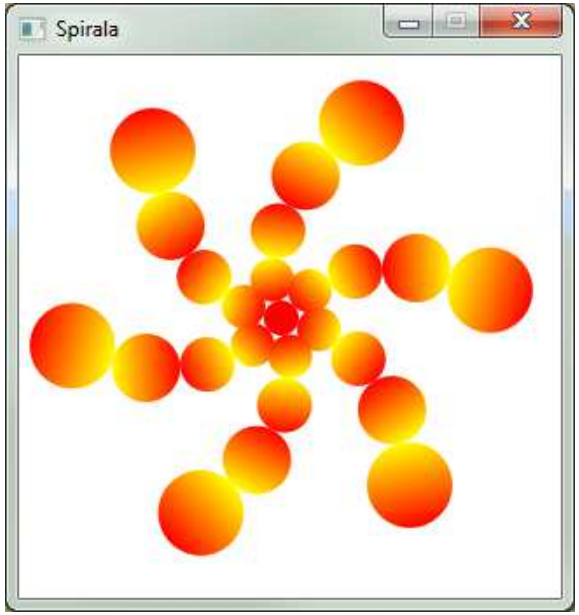
Rešenje: Cvet (13/13)

```
    Scene scena = new Scene(koren, sirProz, visProz);
    prozor.setTitle("Cvet");
    prozor.setScene(scena);
    prozor.setResizable(false);
    prozor.show();
  }

   public static void main(String[] arg) {
     launch(arg);
   }
}
```

Zadatak 4: Spirala

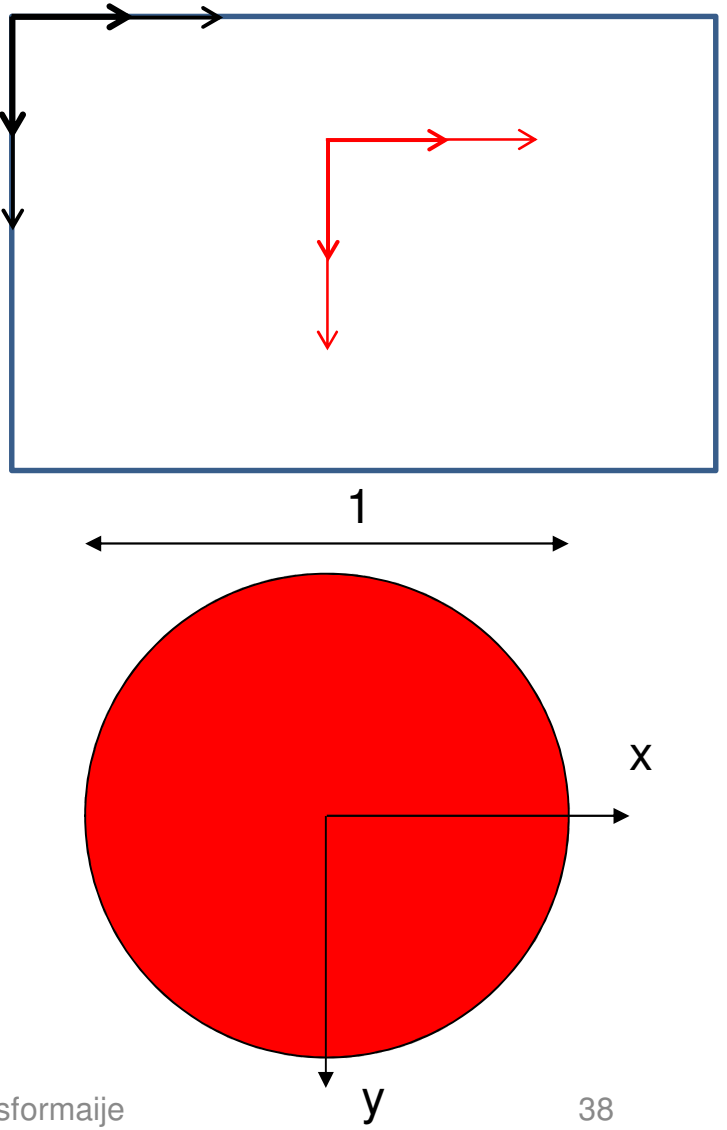
Kolokvijum K1 09/10, zadatak prerađen za JavaFX

Napisati klasu koja sastavlja graf scene za crtanje centralno simetrične figure prikazane na slici. 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 prva kruga u kraku dodiruju, a duž koja spaja centre dva susedna kruga u kraku se nalazi pod uglom od 20° 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.]



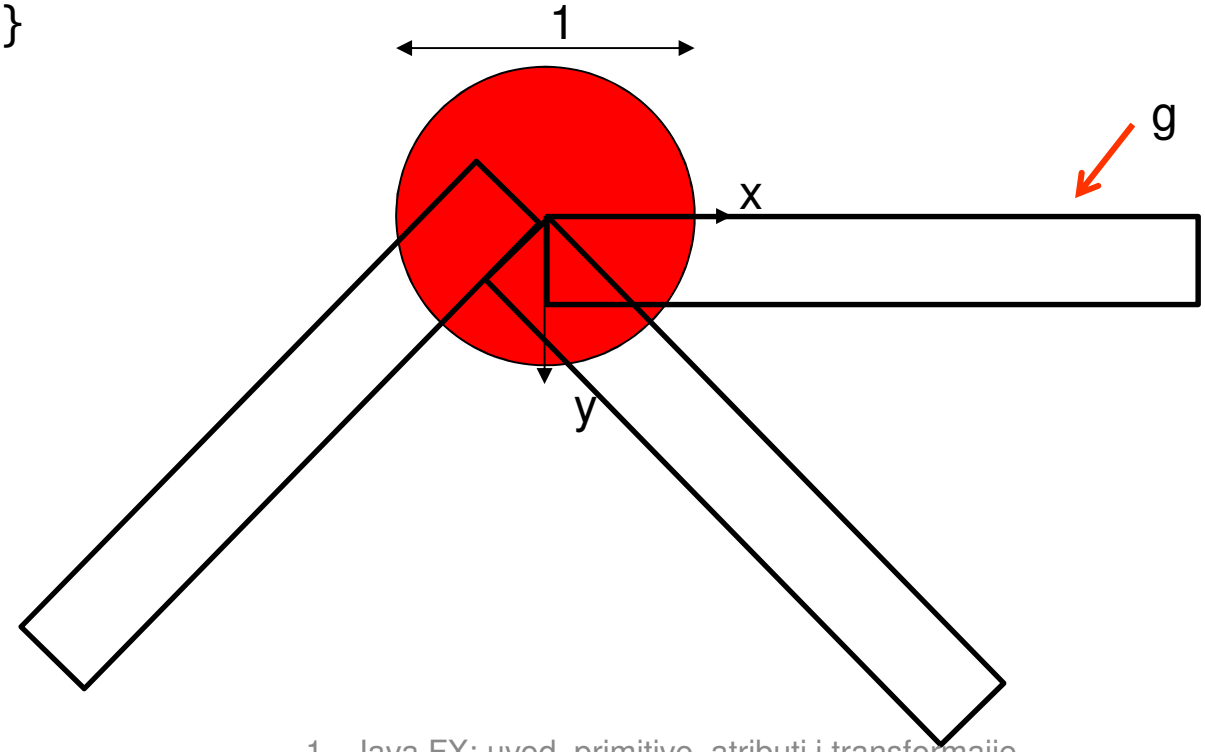
Rešenje: Spirala (1/14)

```
package spirala;

import ...;

public class Spirala extends Application {
    private static final double R = 0.5;
    private static final double S = 20;

    @Override
    public void start(Stage prozor) {
        Group koren = new Group();
        koren.setTranslateX(150);
        koren.setTranslateY(150);
        koren.setScaleX(S);
        koren.setScaleY(S);
        Circle centar = new Circle(R);
        centar.setFill(Color.RED);
        koren.getChildren().add( centar );
```



Rešenje: Spirala (2/14)

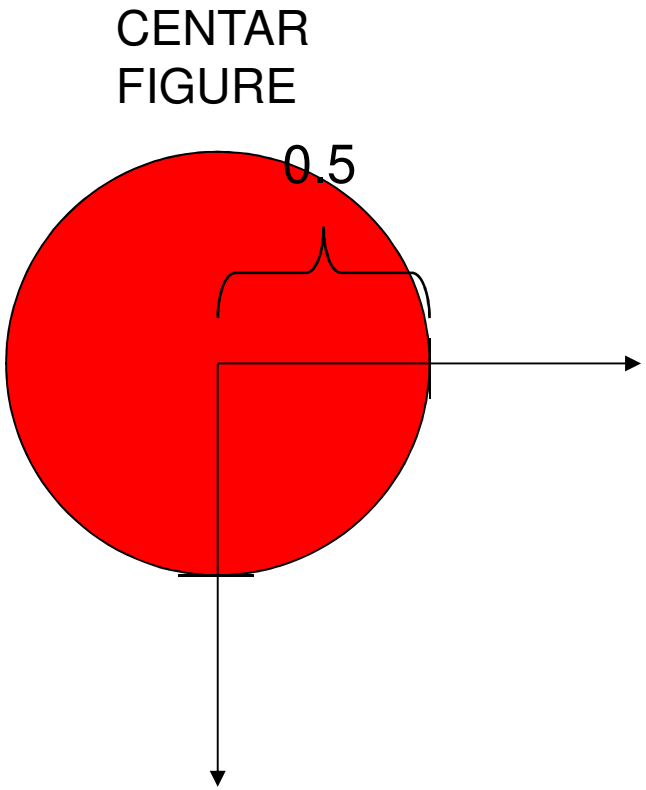
```
for(int i = 0; i < 6; i++) {
    Group g = new Group();
    g.getChildren().addAll( napraviKrak() );
    g.getTransforms().setAll( new Rotate(60*i));
    koren.getChildren().add(g);
}
```



Rešenje: Spirala (3/14)

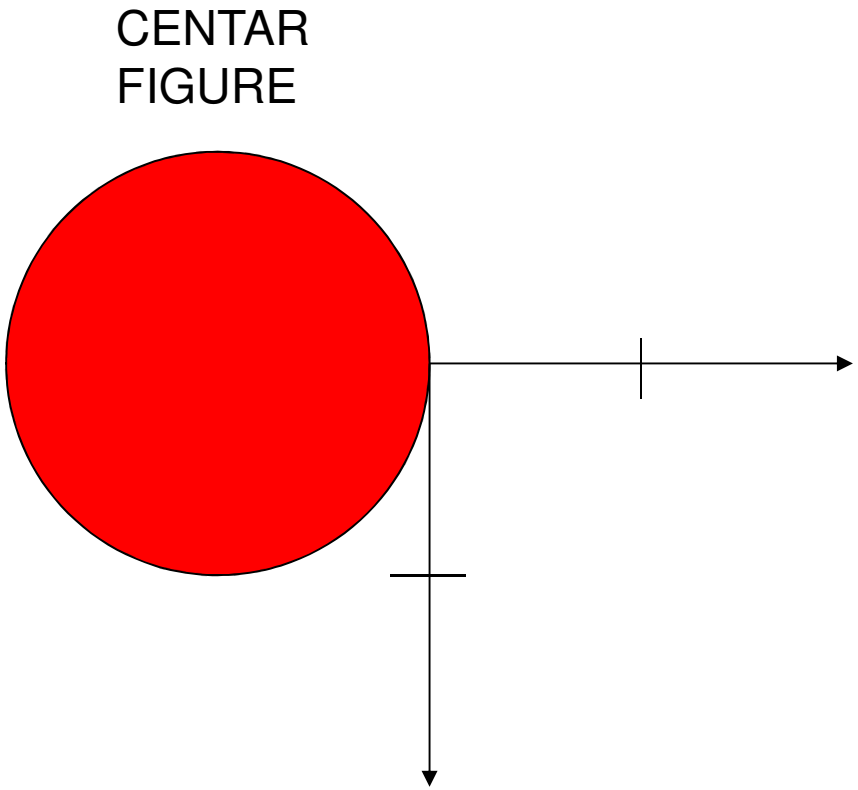
```
        /*
        RotateTransition rt = new
                    RotateTransition(Duration.seconds(5), koren);
        rt.setFromAngle(0); rt.setToAngle(360);
        rt.setInterpolator(Interpolator.LINEAR);
        rt.setCycleCount(Timeline.INDEFINITE);
        rt.play();
        */

        Scene scena = new Scene(koren, 300, 300);
        prozor.setTitle("Spirala");
        prozor.setScene(scena);
        prozor.setResizable(false);
        prozor.show();
    } // start
```

Rešenje: Spirala (4/14)



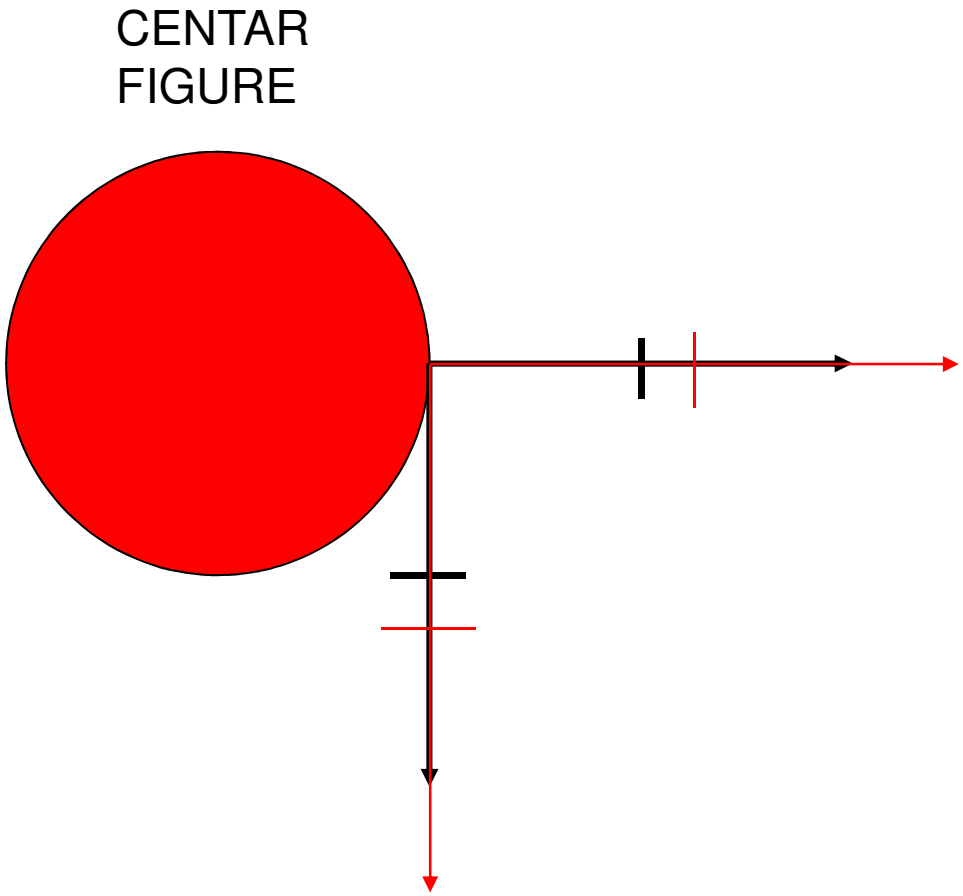
Sledeći korak: Translacija(Δx=0.5, Δy=0)

Rešenje: Spirala (5/14)



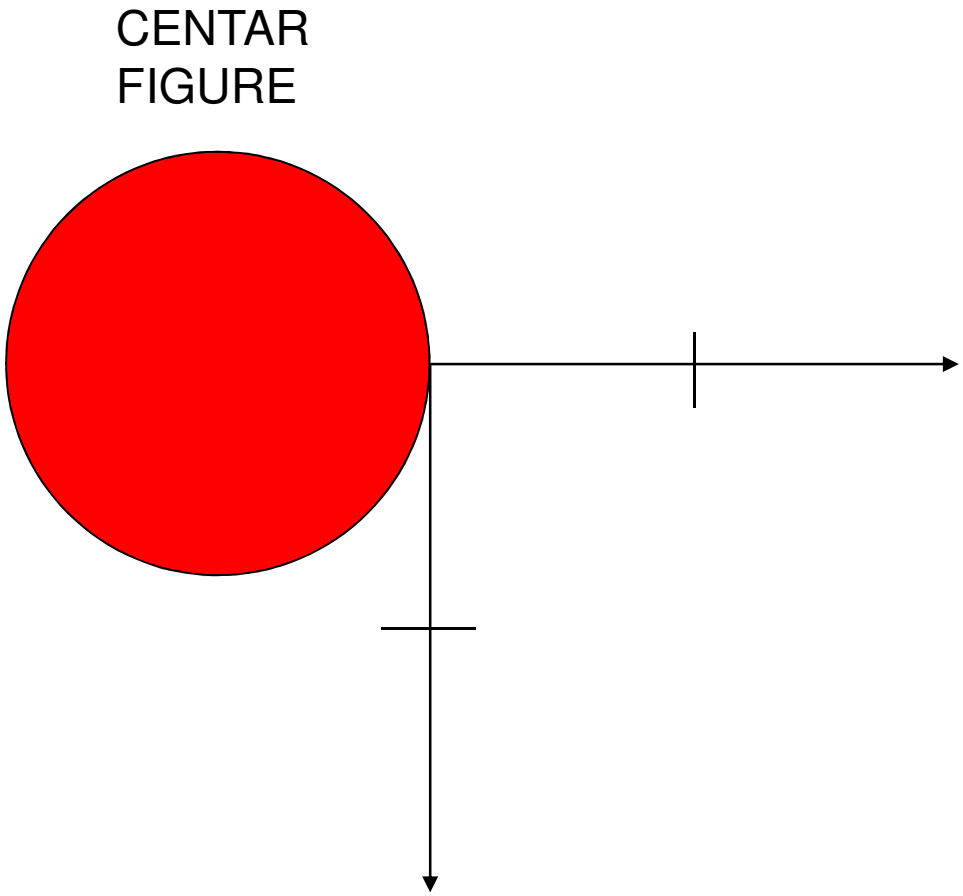
Sledeći korak: Skaliranje(Sx = 1.25, Sy = 1.25)

Rešenje: Spirala (6/14)



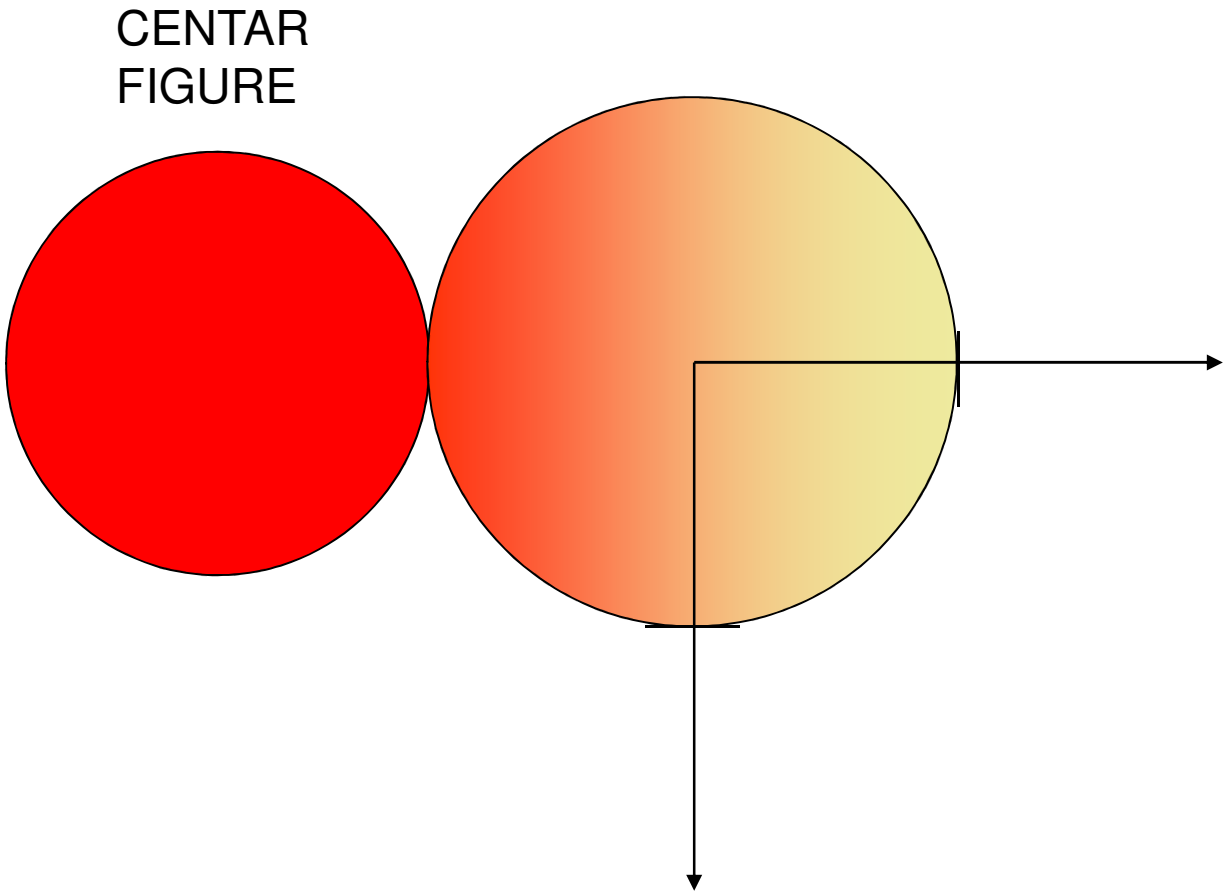
Skaliranje(Sx = 1.25, Sy = 1.25)

Rešenje: Spirala (7/14)



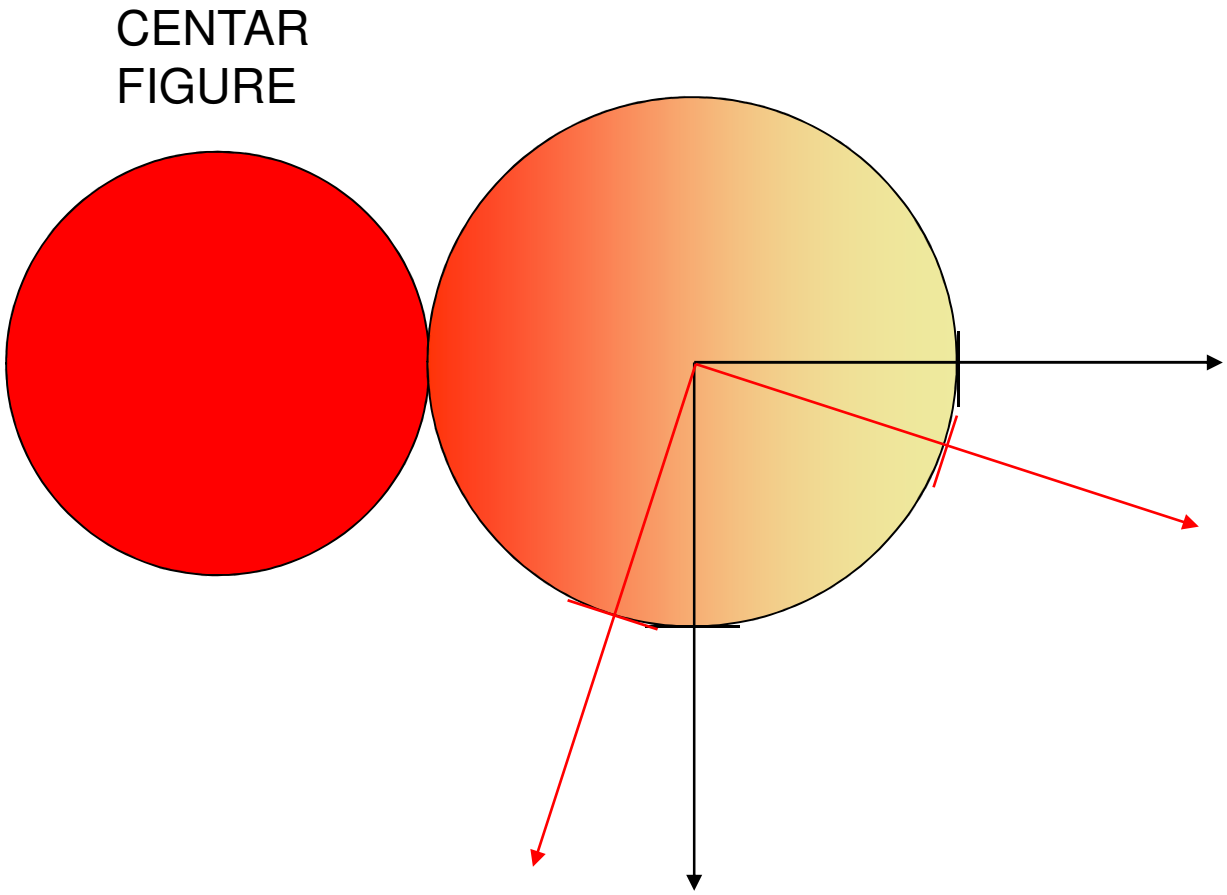
Sledeći korak: Translacija(Δx=0.5, Δy=0)

Rešenje: Spirala (8/14)



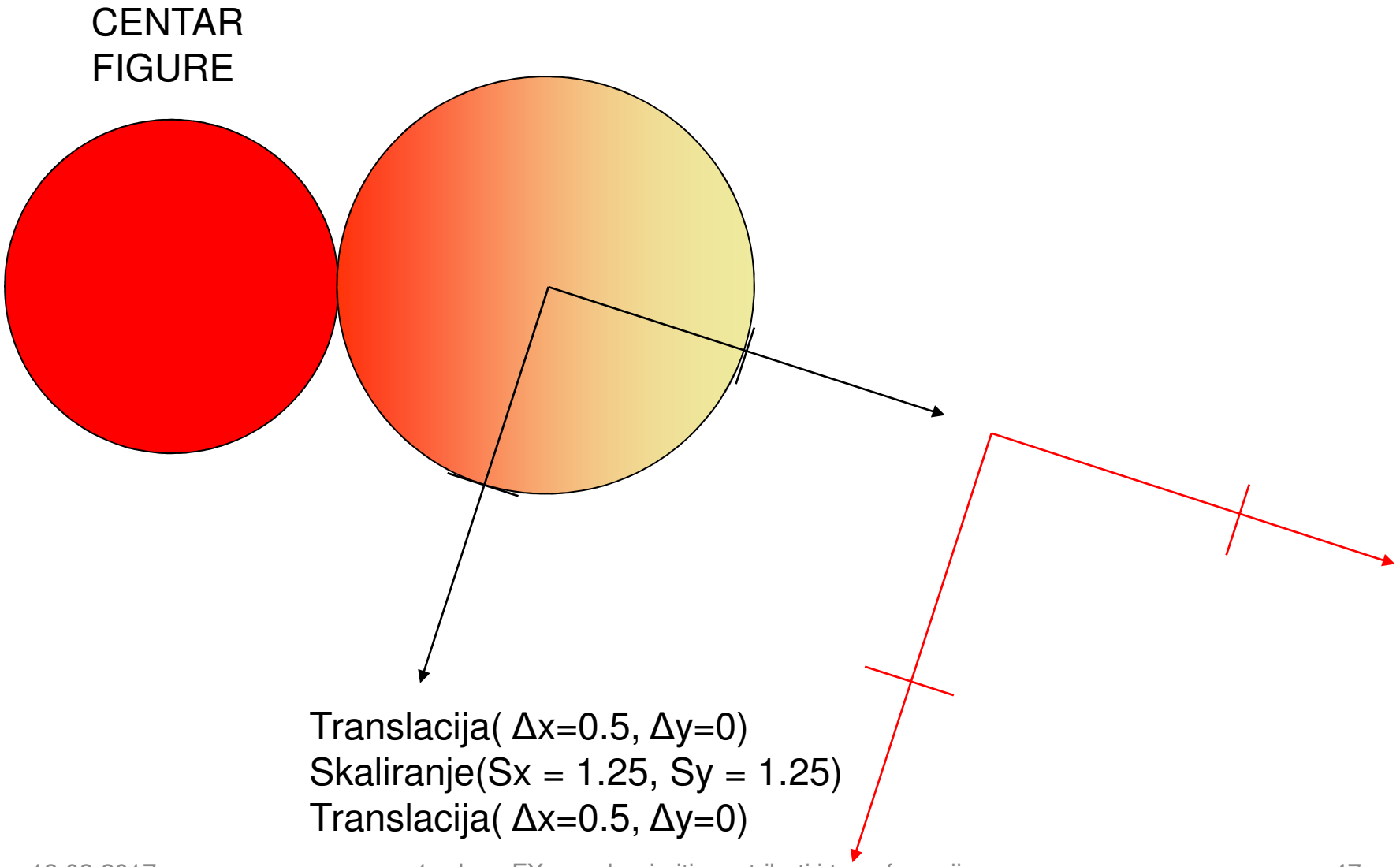
Novi krug (bojenje CRVENA→ŽUTA)

Rešenje: Spirala (9/14)

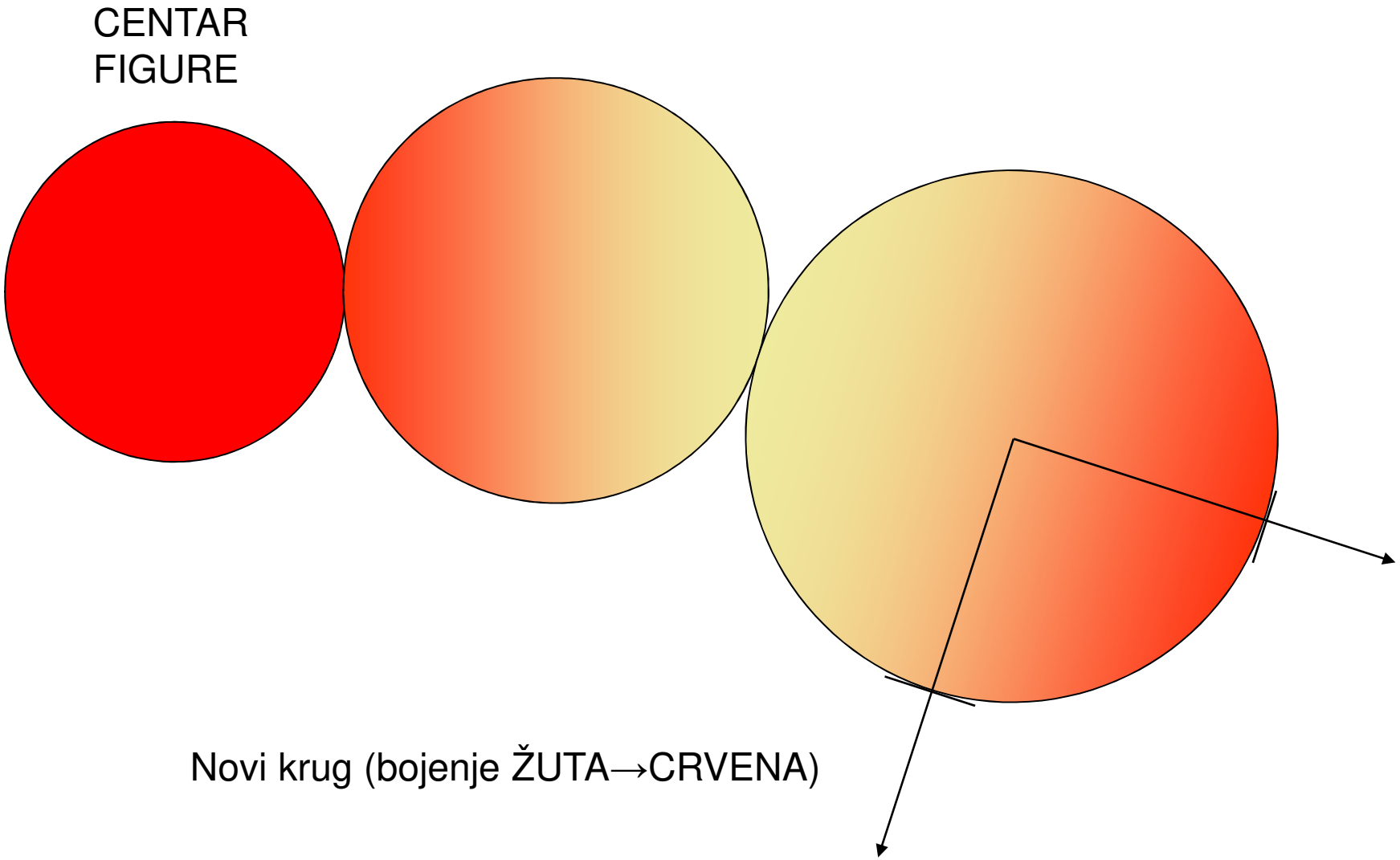


Rotacija(20°)

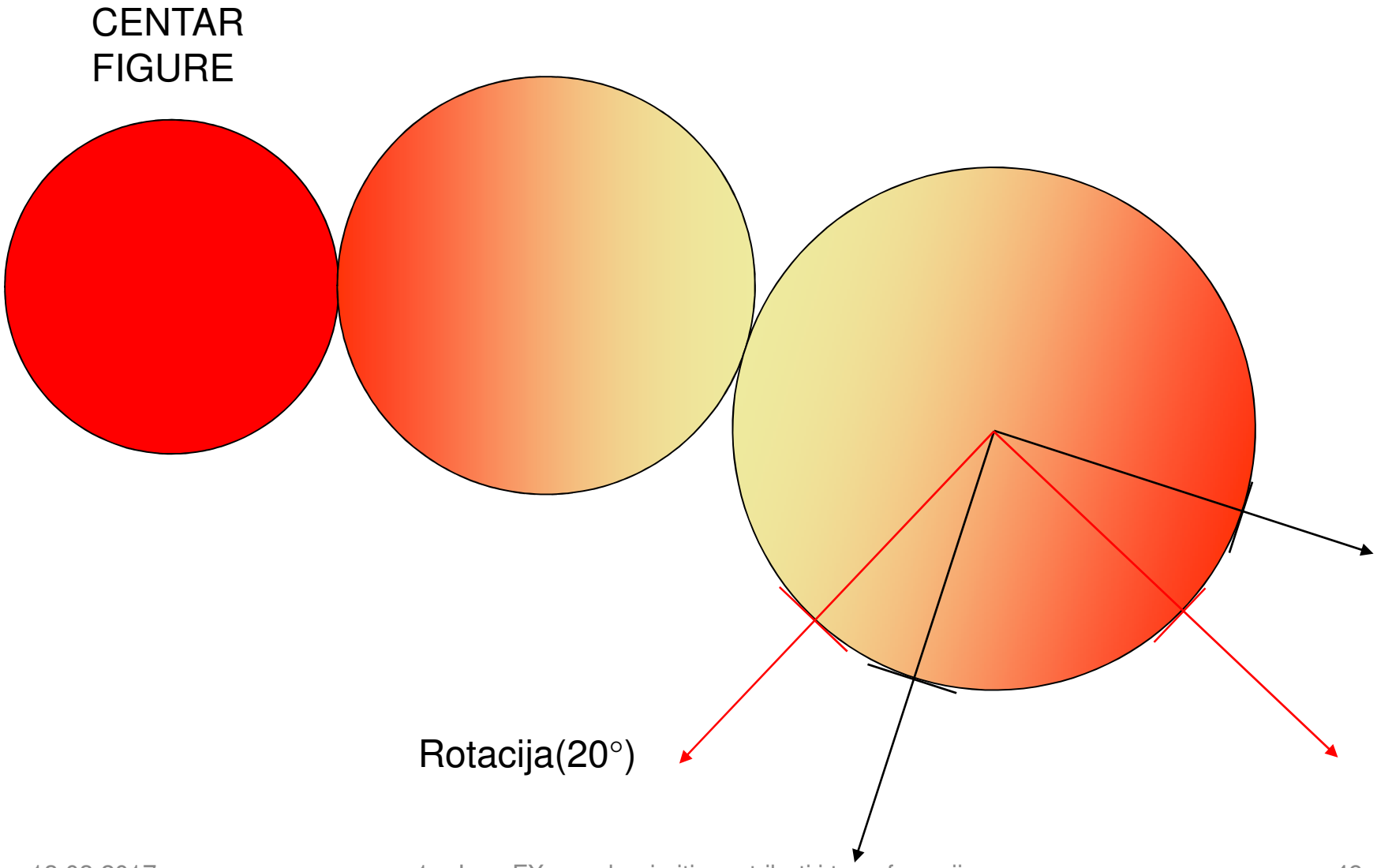
Rešenje: Spirala (10/14)



Rešenje: Spirala (11/14)



Rešenje: Spirala (12/14)



Rešenje: Spirala (13/14)

```
private Group napraviKrak() {
  Group krak = new Group();
  Group g = krak;
  Stop s1 = new Stop(0.0, Color.RED);
  Stop s2 = new Stop(1.0, Color.YELLOW);
  Stop s3 = new Stop(0.0, Color.YELLOW);
  Stop s4 = new Stop(1.0, Color.RED);
  LinearGradient lg1 = new LinearGradient(-0.5, 0, 0.5, 0,
           false, CycleMethod.NO_CYCLE, new Stop[]{s1, s2});
  LinearGradient lg2 = new LinearGradient(-0.5, 0, 0.5, 0,
           false, CycleMethod.NO_CYCLE, new Stop[]{s3, s4});

  LinearGradient []lg = { lg1, lg2 };
```

Rešenje: Spirala (14/14)

```
        for(int i = 0; i < 4; i++) {
            Translate t = new Translate(0.5, 0);
            Scale s = new Scale(1.25, 1.25);
            Rotate r = new Rotate(20);
            g.getTransforms().setAll(r, t, s, t);

            Circle c = new Circle(R);
            c.setFill( lg[i%2] );
            g.getChildren().add(c);
            Group sled = new Group();
            g.getChildren().add(sled);
            g = sled;
        }
        return krak;
    }

    public static void main(String[] args) { launch(args); }
}
```