Računarska grafika - vežbe

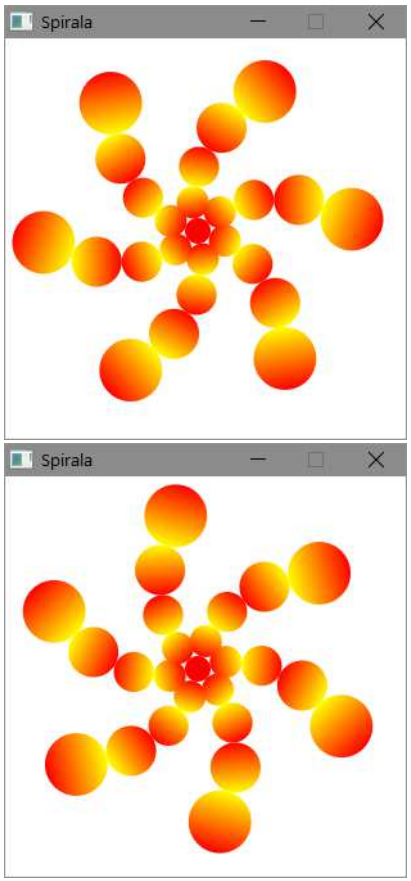
3 – JavaFX

animacija i interakcija

Zadatak 1: 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.



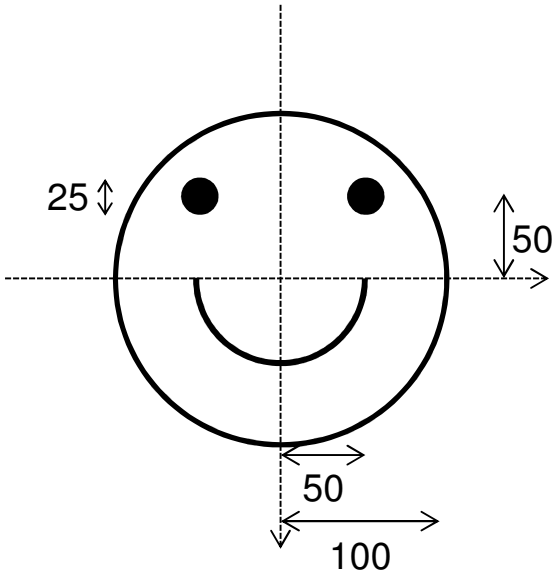
Rešenje: Spirala+

```
// … videti rešenje zadatka Spirala
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);
}

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

Zadatak 2: Trepko

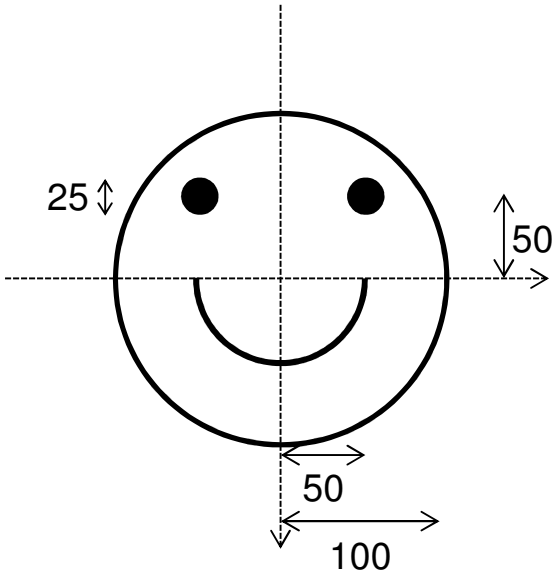
- Napisati program koji u prozoru crta i prikazuje lik "Smeška" koji trepće upotrebom biblioteke JavaFX.

Rešenje: Trepko (1/5)



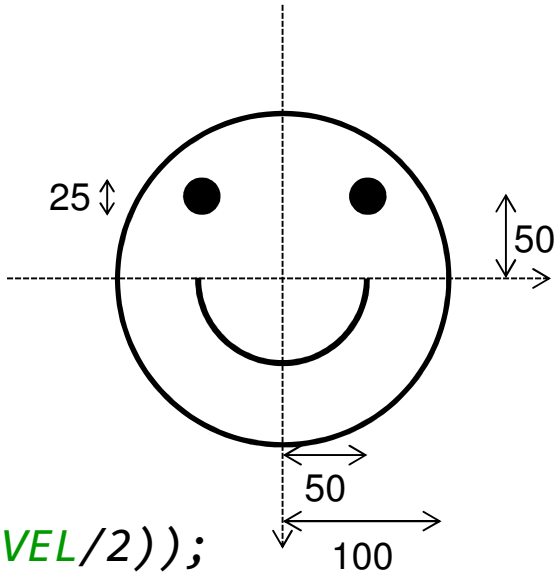

```
package trepko;

import javafx.scene.Group;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;
import javafx.scene.transform.Translate;
import javafx.util.Duration;
import javafx.animation.*;

class Lice extends Group {
    private final Circle glava;
    private final Circle lOko;
    private final Circle dOko;
    private final Arc usta;
    private static final double VEL = 100.0;
```

Rešenje: Trepko (2/5)




```
public Lice() {
  glava = new Circle(VEL);
  glava.fillProperty().set(Color.YELLOW);
  glava.strokeProperty().set(Color.BLACK);
  glava.strokeWidthProperty().set(5);

  lOko = new Circle(VEL/8);
  lOko.fillProperty().set(Color.BLACK);
  Group lOkoPozicija = new Group();
  lOkoPozicija.getTransforms()
              .setAll( new Translate(-VEL/2, -VEL/2));
  lOkoPozicija.getChildren().add(lOko);
```

Rešenje: Trepko (3/5)

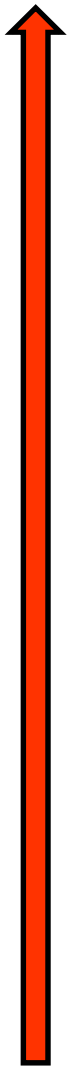



```
    dOko = new Circle(VEL/8);
    dOko.fillProperty().set(Color.BLACK);
    Group dOkoPozicija = new Group();
    dOkoPozicija.getTransforms()
                .setAll( new Translate(VEL/2, -VEL/2));
    dOkoPozicija.getChildren().add(dOko);

    usta = new Arc(0, 0, VEL/2, VEL/2, 180, 180);
    usta.fillProperty().set(null);
    usta.strokeProperty().set(Color.BLACK);
    usta.strokeWidthProperty().set(5);
    usta.setType(ArcType.OPEN);

    getChildren().addAll(glava, lOkoPozicija, dOkoPozicija, usta);
} // Konstruktor
```

Rešenje: Trepko (4/5)

```
public void napraviAnimaciju() {
  ScaleTransition skaliranje =
      new ScaleTransition(Duration.millis(1000), lOko);
  skaliranje.setFromY(1);
  skaliranje.setToY(0.1);
  skaliranje.setCycleCount(Timeline.INDEFINITE);
  skaliranje.setAutoReverse(true);
  skaliranje.play();

  skaliranje = new ScaleTransition(Duration.millis(1000), dOko);
  skaliranje.setFromY(0.1);
  skaliranje.setToY(1);
  skaliranje.setCycleCount(Timeline.INDEFINITE);
  skaliranje.setAutoReverse(true);
  skaliranje.play();
 }
}
```

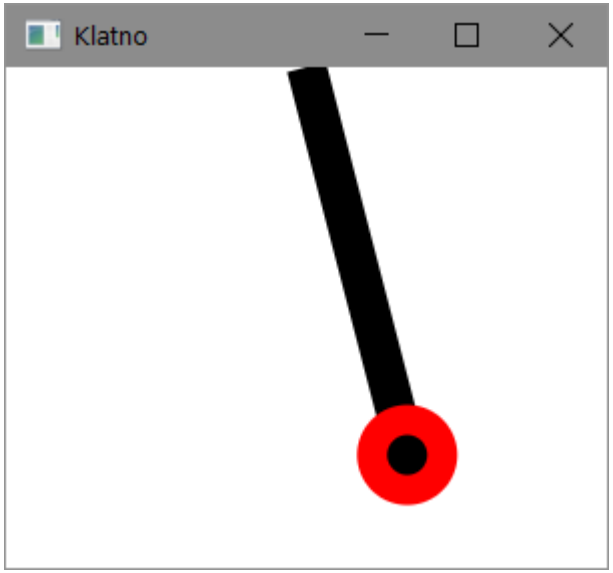
Rešenje: Trepko (5/5)



```
package trepko;
import ...
public class Trepko extends Application {
    public void start(Stage prozor) {
        int vel = 300;
        Group koren = new Group();
        Lice lice = new Lice();
        koren.getChildren().addAll( lice );
        koren.getTransforms().setAll( new Translate(vel/2, vel/2));
        lice.napraviAnimaciju();
        Scene scena = new Scene(koren, vel, vel);
        prozor.setTitle("Trepko");
        prozor.setScene(scena);
        prozor.setResizable(false);
        prozor.show();
    }
    public static void main(String[] args) { launch(args);  }
}
```

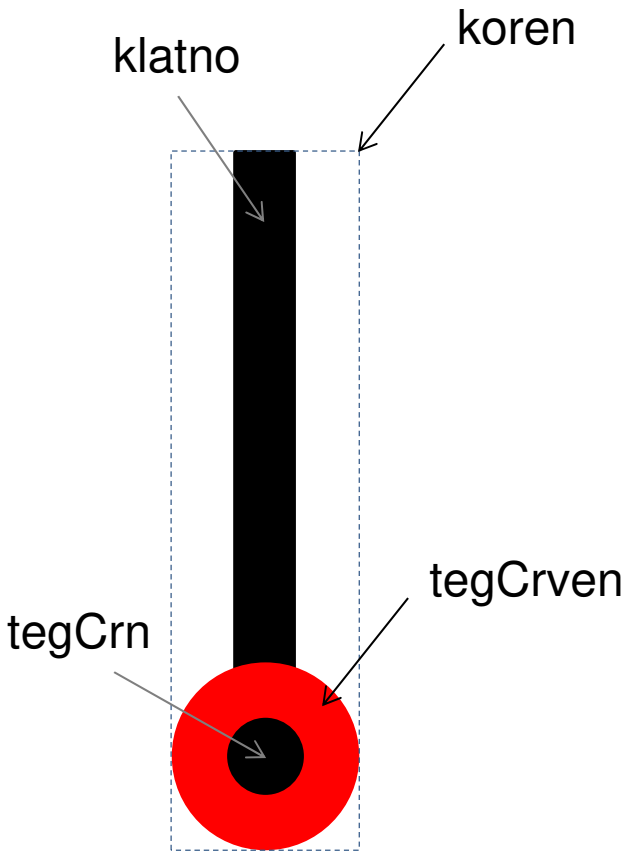
Zadatak 3: Klatno

- Napisati program koji u prozoru crta animirano klatno (prikazano na slici) upotrebom biblioteke JavaFX. Dužina klatna je proporcionalna visini prozora u kojem se klatno prikazuje. Najveći ugao otklona klatna u odnosu na vertikalnu osu je 20 stepeni.



Rešenje: Klatno (1/4)

```
package klatno;

import javafx.animation.*;
import javafx.application.Application;
import javafx.scene.*;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;
import javafx.scene.transform.*;
import javafx.stage.Stage;
import javafx.util.Duration;

public class Klatno extends Application {
    private Group koren;
    private Rectangle klatno;
    private Circle tegCrven;
    private Circle tegCrn;
    private Scene scena;
```



Rešenje: Klatno (2/4)

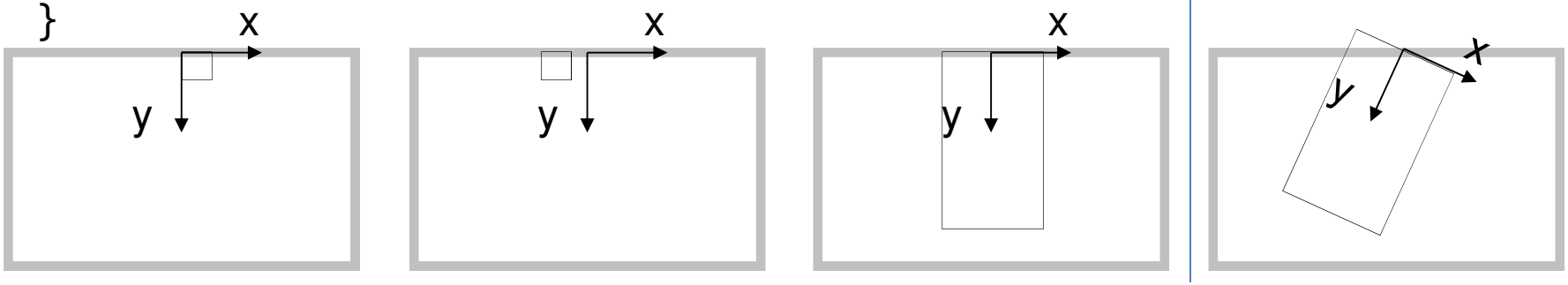
```
@Override
public void start(Stage prozor) {
  koren = new Group();
  klatno = new Rectangle(1, 1);
  tegCrven = new Circle(25);
  tegCrven.fillProperty().set(Color.RED);
  tegCrn = new Circle(10);
  tegCrn.fillProperty().set(Color.BLACK);
  koren.getChildren().addAll(klatno, tegCrven, tegCrn);
  Rotate rotacija = new Rotate();
  koren.getTransforms().add(rotacija);
  Timeline klacenje = new Timeline(
     new KeyFrame(Duration.ZERO,
                  new KeyValue(rotacija.angleProperty(), -20)),
     new KeyFrame(Duration.seconds(2),
                  new KeyValue(rotacija.angleProperty(), 20))
  );
```

Rešenje: Klatno (3/4)

```
    klacenje.setAutoReverse(true);
    klacenje.setCycleCount(Timeline.INDEFINITE);
    klacenje.play();

    scena = new Scene(koren, 300, 250);
    scena.widthProperty().addListener(s -> {promeniVelicinu();});
    scena.heightProperty().addListener(v -> {promeniVelicinu();});
    prozor.setTitle("Klatno");
    prozor.setScene(scena);
    prozor.show();

    promeniVelicinu();
  } // start()

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

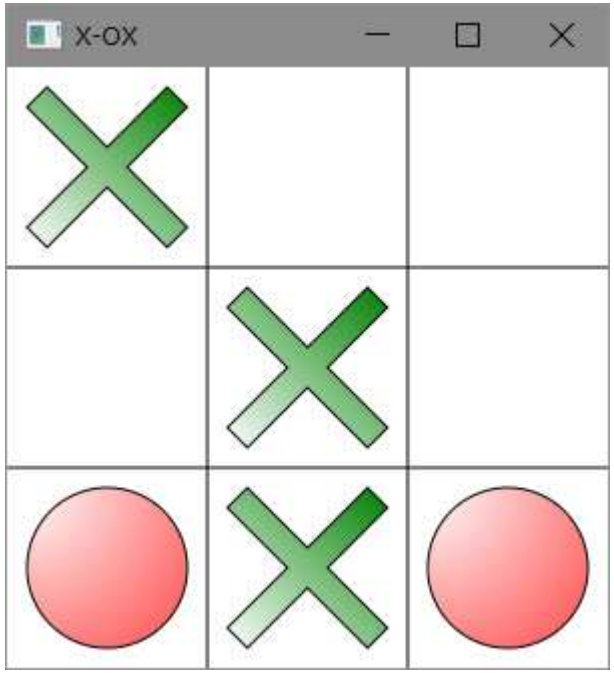
Rešenje: Klatno (4/4)

```
  private void promeniVelicinu() {
    if( scena == null ) return;
    koren.translateXProperty().set(scena.getWidth()/2);

    Transform t = new Translate(-10, 0);
    Transform s = new Scale(20, scena.getHeight()*0.8);
    klatno.getTransforms().setAll(t, s);

    t = new Translate(0, scena.getHeight()*0.8);
    tegCrven.getTransforms().setAll(t);
    tegCrn.getTransforms().setAll(t);
  }
}
```

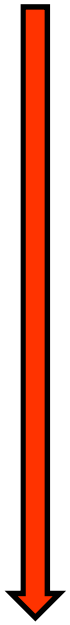


Zadatak 4: X-OX

- Napisati program koji crta tablu za igru X-OX prema priloženoj slici i omogućava postavljanje crvenih simbola O tasterima 1-9 i zelenih simbola X klikom miša. Taster 0 je za čišćenje table. Tabla se prilagođava dimenzijama prozora.



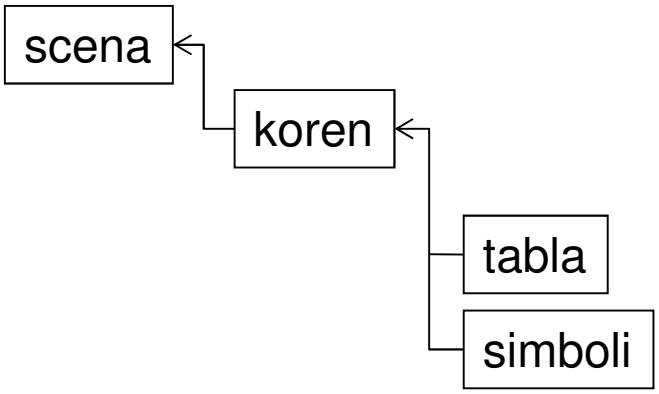
Rešenje: X-OX (1/6)

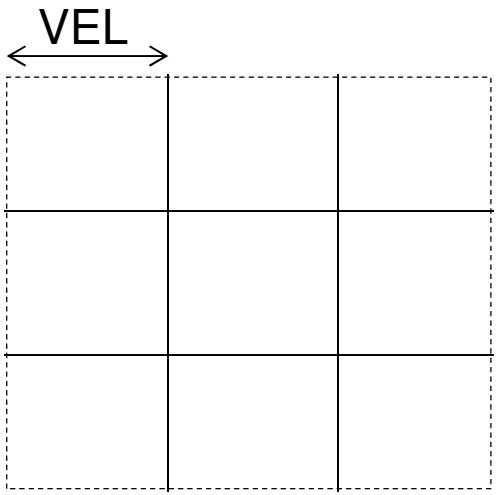


```
package xox;

import javafx.application.Application;
import javafx.scene.*;
import javafx.scene.input.*;
import javafx.scene.paint.*;
import javafx.scene.shape.*;
import javafx.scene.transform.*;
import javafx.stage.Stage;
```

Rešenje: X-OX (2/6)

```
public class XoX extends Application {

    private static final int VEL = 100;
    private Scene scena;
    private Group koren = new Group();
    private Group tabla = new Group();
    private Group simboli = new Group();
```





Rešenje: X-OX (3/6)

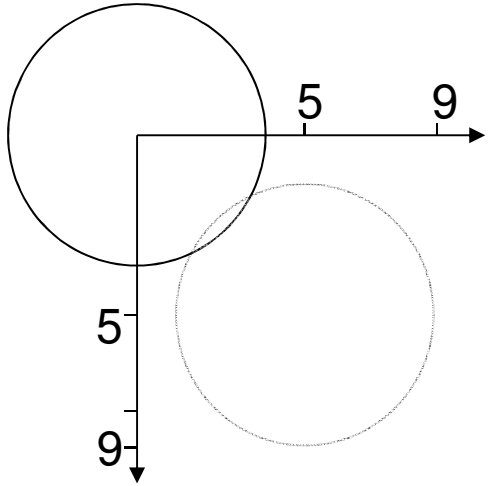
```
private void tabla()    {
  tabla.getChildren().add( new Line(0, VEL, 3*VEL, VEL) );
  tabla.getChildren().add( new Line(0, 2*VEL, 3*VEL, 2*VEL) );
  tabla.getChildren().add( new Line(VEL, 0, VEL, 3*VEL) );
  tabla.getChildren().add( new Line(2*VEL, 0, 2*VEL, 3*VEL) );
}

private void promeniVelicinu() {
  if( scena == null ) return;
  Transform s = new Scale(scena.getWidth()/(3*VEL),
                          scena.getHeight()/(3*VEL));
  koren.getTransforms().setAll(s);
}
```

Rešenje: X-OX (4/6)

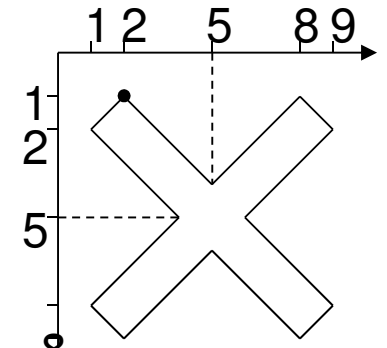
```
@Override public void start(Stage prozor) {
  tabla();
  koren = new Group();
  koren.getChildren().addAll( tabla, simboli );
  koren.setPickOnBounds(true);
  koren.setOnMouseClicked( d -> naKlikMisa(d) );
  scena = new Scene(koren, 3*VEL, 3*VEL);
  scena.widthProperty().addListener(s -> {promeniVelicinu();});
  scena.heightProperty().addListener(v -> {promeniVelicinu();});
  scena.addEventHandler(KeyEvent.KEY_PRESSED, d -> naTaster(d));
  prozor.setTitle("X-OX");
  prozor.setScene(scena);
  prozor.show();
} // start()

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

Rešenje: X-OX (5/6)

```
private void naTaster(KeyEvent e) {
   char znak = e.getText().charAt(0);
   if( znak >= '1' && znak <= '9') {
     int x = (znak - '1')%3;       int y = 2 - (znak - '1')/3;
     Stop []stan={new Stop(0,Color.WHITE),new Stop(1,Color.RED)};
     Circle O = new Circle(4);
     O.setFill( new RadialGradient(0, 0, 0, 0, 2, true,
                                   CycleMethod.NO_CYCLE,stan));
     O.setStroke(Color.BLACK);  O.setStrokeWidth(0.1);
     O.getTransforms().setAll(
       new Translate((x+0.5)*VEL, (y+0.5)*VEL),
       new Scale(VEL/10, VEL/10) );
     simboli.getChildren().add(O);
   } else if( znak == '0' )
     simboli.getChildren().clear();
}
```



Rešenje: X-OX (6/6)




```
private void naKlikMisa(MouseEvent e) {
  int x = (int)e.getX()/VEL, y = (int)e.getY()/VEL;
  Polygon X = new Polygon();
  X.getPoints().setAll( new Double[]
       {  2., 1., 5., 4., 8., 1., 9., 2., 6., 5., 9., 8.,
          8., 9., 5., 6., 2., 9., 1., 8., 4., 5., 1., 2.  }
       );
  Stop []stan = { new Stop(0, Color.WHITE),
                  new Stop(1, Color.GREEN ) };
  X.setFill( new LinearGradient(1, 9, 9, 1, false,
                                CycleMethod.NO_CYCLE, stan));
  X.setStroke( Color.BLACK );
  X.setStrokeWidth(0.1);
  X.getTransforms().setAll( new Translate(x*VEL, y*VEL),
                            new Scale(VEL/10, VEL/10));
  simboli.getChildren().add(X);
 }
}
```