JavaFX
https://docs.oracle.com/javase/8/javase-clienttechnologies.htm
Aplicatii propuse
https://docs.oracle.com/javase/8/javafx/get-started-tutorial/animation.htm
https://docs.oracle.com/javase/8/javafx/get-started-tutorial/form.htm
https://docs.oracle.com/javase/8/javafx/get-started-tutorial/css.htm

Tipuri de evenimente

Evenimentele pot fi clasificate în linii mari în următoarele două categori
Foreground Events- evenimente care necesită interacțiunea directă a unui utilizator. Acestea sunt generate ca urmare a interacțiunii cu componentele grafice dintr-o interfață grafică cu utilizatorul.

Background Events- evenimente care nu necesită interacțiunea utilizatorului final, sunt cunoscute ca evenimente de fundal. Întreruperile sistemului de operare, defecțiunile hardware sau software, expirarea temporizatorului, finalizarea operațiunii sunt exemple de evenimente de fundal.

Evenimente în JavaFX

JavaFX oferă suport pentru gestionarea unei game largi de evenimente. Clasa numită Event a pachetului javafx.event este clasa de bază pentru un eveniment.

O instanță a oricăreia dintre subclasele sale este un eveniment. JavaFX oferă o mare varietate de evenimente, exemplu:

Eveniment mouse - este reprezentat de clasa MouseEvent care include acțiuni – apăsarea,eliberarea mouse-ului, mutarea mouse-ului etc.

Key Event - Acest eveniment de intrare indică apăsarea tastei pe un nod. Este reprezentat de clasa KeyEvent.

Drag Event- Acesta este un eveniment de intrare care are loc atunci când mouse-ul este tras. Este reprezentat de clasa DragEvent.

Window Event- Acesta este un eveniment legat de acțiunile de afișare/ascundere a ferestrei. Este reprezentat de clasa numită WindowEvent.

Gestionarea evenimentelor

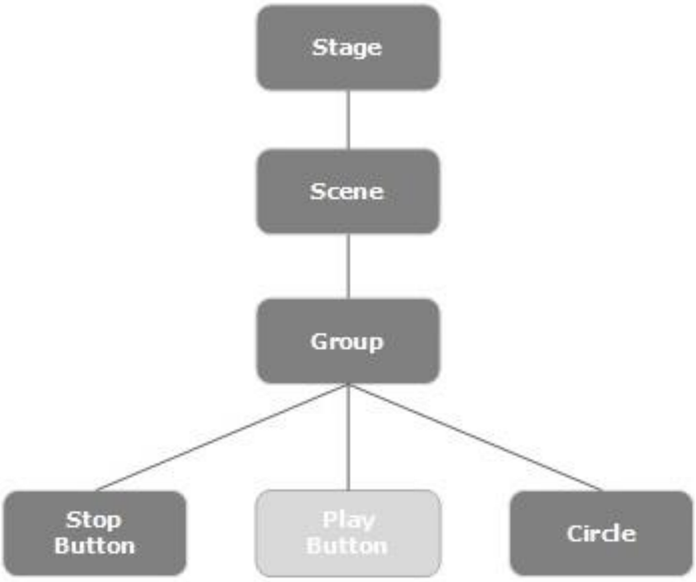
Gestionarea evenimentelor este mecanismul care controlează evenimentul și decide ce ar trebui să se întâmple, dacă are loc un eveniment. Acest mecanism are codul care este cunoscut ca un handler de evenimente care este executat atunci când are loc un eveniment.

JavaFX oferă handlers and filterspentru a gestiona evenimente. În JavaFX fiecare eveniment are:

Target - Nodul pe care a avut loc un eveniment. O țintă poate fi o fereastră, o scenă și un nod.

Sursa - Sursa din care este generat evenimentul va fi sursa evenimentului. În scenariul de mai sus, mouse-ul este sursa evenimentului.

Tip − Tipul evenimentului produs; în cazul unui eveniment mouse – mouse apăsat, mouse eliberat sunt tipul de evenimente.



Ex5 var 1

```
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.event.EventHandler;

import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.input.MouseEvent;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;

import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class Ex5 extends Application {
```

```
@Override
public void start(Stage stage) {
  //Drawing a Circle
  Circle circle = new Circle();

  //Setting the position of the circle
  circle.setCenterX(300.0f);
  circle.setCenterY(135.0f);

  //Setting the radius of the circle
  circle.setRadius(25.0f);

  //Setting the color of the circle
  circle.setFill(Color.BROWN);

  //Setting the stroke width of the circle
  circle.setStrokeWidth(20);

  //Setting the text
  Text text = new Text(" schimba color");

  //Setting the font of the text
  text.setFont(Font.font(null, FontWeight.BOLD, 15));

  //Setting the color of the text
  text.setFill(Color.CRIMSON);

  //setting the position of the text
  text.setX(150);
  text.setY(50);

  //Creating the mouse event handler
  EventHandler<MouseEvent> eventHandler = new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent e) {
      System.out.println("Hello World");
      circle.setFill(Color.DARKSLATEBLUE);
    }
  };
  //Registering the event filter
  circle.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler);

  //Creating a Group object
  Group root = new Group(circle, text);

  //Creating a scene object
  Scene scene = new Scene(root, 600, 300);

  //Setting the fill color to the scene
  scene.setFill(Color.LAVENDER);
```

```
      //Setting title to the Stage
      stage.setTitle("Ex 5 JavaFX");

      //Adding scene to the stage
      stage.setScene(scene);

      //Displaying the contents of the stage
      stage.show();
   }
   public static void main(String args[]){
      launch(args);
   }
}
```

Ex5 var 2

```
import javafx.animation.PathTransition;
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.event.EventHandler;

import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.input.MouseEvent;
import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;
import javafx.scene.shape.LineTo;
import javafx.scene.shape.MoveTo;
import javafx.scene.shape.Path;
import javafx.stage.Stage;
import javafx.util.Duration;

public class Ex5 extends Application {
   @Override
   public void start(Stage stage) {
      //Drawing a Circle
      Circle circle = new Circle();

      //Setting the position of the circle
      circle.setCenterX(300.0f);
      circle.setCenterY(135.0f);

      //Setting the radius of the circle
      circle.setRadius(25.0f);

      //Setting the color of the circle
      circle.setFill(Color.BROWN);
```

```
    //Setting the stroke width of the circle
    circle.setStrokeWidth(20);

    //Creating a Path
    Path path = new Path();

    //Moving to the staring point
    MoveTo moveTo = new MoveTo(208, 71);

    //Creating 1st line
    LineTo line1 = new LineTo(421, 161);

    //Creating 2nd line
    LineTo line2 = new LineTo(226,232);

    //Creating 3rd line
    LineTo line3 = new LineTo(332,52);


    //Adding all the elements to the path
    path.getElements().add(moveTo);
    path.getElements().addAll(line1, line2, line3);

    //Creating the path transition
    PathTransition pathTransition = new PathTransition();

    //Setting the duration of the transition
    pathTransition.setDuration(Duration.millis(1000));

    //Setting the node for the transition
    pathTransition.setNode(circle);

    //Setting the path for the transition
    pathTransition.setPath(path);

    //Setting the orientation of the path
    //pathTransition.setOrientation(
PathTransition.OrientationType.ORTHOGONAL_TO_TAN GENT);

    //Setting the cycle count for the transition
    pathTransition.setCycleCount(50);

    //Setting auto reverse value to true
    pathTransition.setAutoReverse(false);

    //Creating play button
    Button playButton = new Button("Play");
    playButton.setLayoutX(300);
    playButton.setLayoutY(250);
    circle.setOnMouseClicked (new EventHandler<javafx.scene.input.MouseEvent>() {
      @Override
      public void handle(javafx.scene.input.MouseEvent e) {
        System.out.println("Hello World");
        circle.setFill(Color.DARKSLATEBLUE);
      }
    });
    playButton.setOnMouseClicked((new EventHandler<MouseEvent>() {
      public void handle(MouseEvent event) {
        System.out.println("Hello World");
        pathTransition.play();
      }
    }));

    //Creating stop button
    Button stopButton = new Button("stop");
    stopButton.setLayoutX(250);
    stopButton.setLayoutY(250);

    stopButton.setOnMouseClicked((new EventHandler<MouseEvent>() {
      public void handle(MouseEvent event) {
        System.out.println("Hello World");
        pathTransition.stop();
      }
    }));
    //Creating a Group object
    Group root = new Group(circle, playButton, stopButton);

    //Creating a scene object
    Scene scene = new Scene(root, 600, 300);
    scene.setFill(Color.LAVENDER);

    //Setting title to the Stage
    stage.setTitle("Ex 5");

    //Adding scene to the stage
    stage.setScene(scene);

    //Displaying the contents of the stage
    stage.show();
  }
  public static void main(String args[]){
    launch(args);
  }
}
```

Exemplu 6var 1

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*; import javafx.scene.layout.*;
import javafx.stage.Stage;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.canvas.*;
import javafx.scene.web.*;
import javafx.scene.Group;

public class Ex6 extends Application {

    // launch the application
    public void start(Stage stage)
    {

        try {

            // set title for the stage
            stage.setTitle("VBox");

            // create a VBox
            VBox vbox = new VBox(15);
              // create a label
            Label label = new Label("Label");
              // add label to vbox
            vbox.getChildren().add(label);
              // add buttons to VBox
            for (int i = 0; i < 5; i++)
            {
                vbox.getChildren().add(new Button("Button " + (int)(i + 1)));
            }
              // create a scene
            Scene scene = new Scene(vbox, 300, 300);
              // set the scene
            stage.setScene(scene);
              stage.show();
        }
          catch (Exception e) {
              System.out.println(e.getMessage());
        }
    }
      // Main Method
    public static void main(String args[])
    {
          // launch the application
        launch(args);
    }
}
```

Exemplu 6 var 2

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.stage.Stage;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.canvas.*;
import javafx.scene.web.*;
import javafx.scene.Group;
import javafx.geometry.Pos;
 public class Ex6 extends Application {
   // launch the application
  public void start(Stage stage)
  {
     try {
       // set title for the stage
      stage.setTitle("VBox");
       // create a VBox
      VBox vbox = new VBox(15);
       // create a label
      Label label = new Label(" VBox examples");
       // add label to vbox
      vbox.getChildren().add(label);
            // set alignment
      vbox.setAlignment(Pos.CENTER);
       // add buttons to VBox
      for (int i = 0; i < 8; i++)
      {
        vbox.getChildren().add(new Button("Button " + (int)(i + 1)));
      }
       // create a scene
      Scene scene = new Scene(vbox, 300, 600);
       // set the scene
      stage.setScene(scene);
       stage.show();
    }

    catch (Exception e) {
       System.out.println(e.getMessage());
    }
  }
   // Main Method
  public static void main(String args[])
  {
     // launch the application
    launch(args);
  } }
```

Ex7

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;
import javafx.scene.text.Font;
import javafx.stage.Stage;

public class Ex7 extends Application {

    private int counter = 0;

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

    @SuppressWarnings("static-access")
    public void start(Stage stage) {
        stage.setTitle("Hello World!");

        GridPane grid = new GridPane();
        grid.setHgap(0);
        grid.setVgap(0);

        Rectangle r1 = new Rectangle();
        r1.setFill(Color.DARKKHAKI);
        r1.setHeight(125);
        r1.setWidth(125);

        Rectangle r2 = new Rectangle();
        r2.setFill(Color.rgb(189, 40, 40));
        r2.setHeight(125);
        r2.setWidth(125);

        Rectangle r3 = new Rectangle();
        r3.setFill(Color.hsb(235, 0.52, 0.36));
        r3.setHeight(125);
        r3.setWidth(125);

        Rectangle r4 = new Rectangle();
```

```
                r4.setFill(Color.web("b894cc"));
                r4.setHeight(125);
                r4.setWidth(125);

                grid.add(r1, 0, 0);
                grid.add(r2, 0, 1);
                grid.add(r3, 1, 0);
                grid.add(r4, 1, 1);

                Label l = new Label();
                l.setFont(new Font("Calibri", 15));
                l.setTextFill(Color.BLACK);

                Button button = new Button();
                button.setFont(new Font("Calibri", 15));
                button.setText("Say 'Hello World'");

                button.setOnAction(new EventHandler<ActionEvent>() {
public void handle(ActionEvent event) {
l.setText(" Hello  world:" + ++counter );
                        }
                });

                BorderPane bp = new BorderPane();
                bp.setBottom(l);
                bp.setAlignment(l, Pos.CENTER);
                bp.setCenter(button);

                StackPane root = new StackPane();
                root.getChildren().add(grid);
                root.getChildren().add(bp);
                stage.setScene(new Scene(root, 250, 250));
                stage.show();

        }

}
```

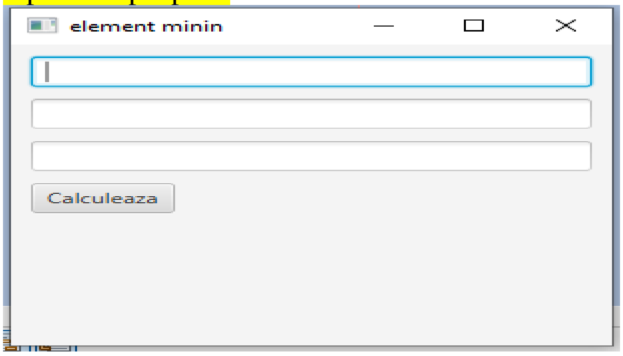
Ca și în Swing, JavaFX permite asocierea unor metode de tip *handler* cu declanșarea unor evenimente. Modul de conectare este foarte similar:

https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
Scene Builder este un instrument de dispunere vizuală care permite utilizatorilor să proiecteze rapid interfețe de utilizare a aplicației JavaFX, fără codificare
https://docs.oracle.com/javafx/scenebuilder/1/use_java_ides/sb-with-nb.htm
https://docs.oracle.com/javase/8/scene-builder-2/work-with-java-ides/sb-with-nb.htm

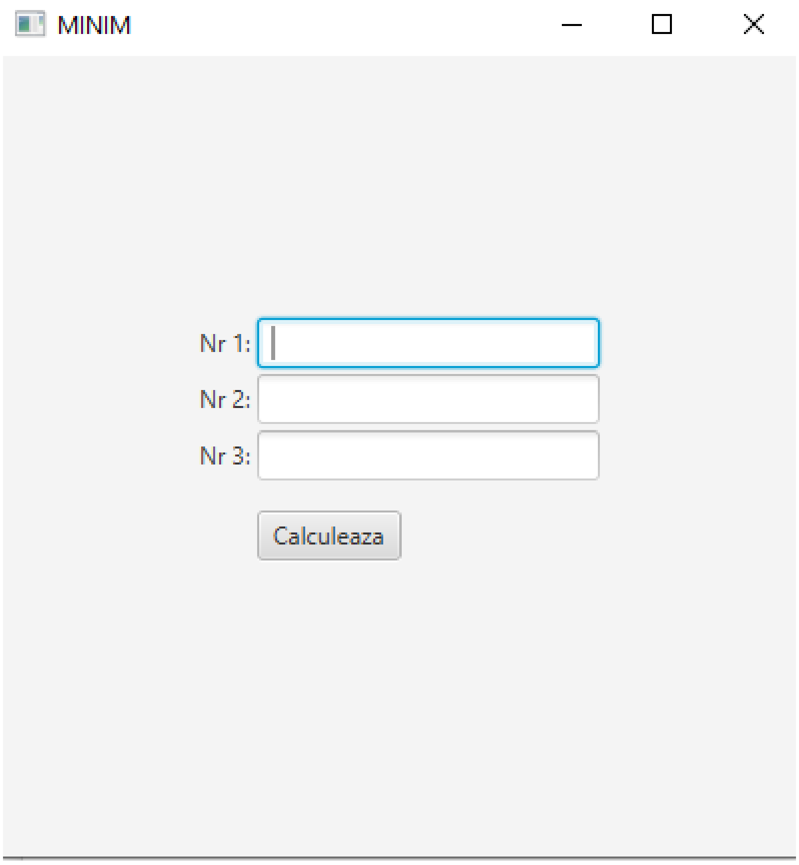
Hello.Java

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

public class JavaFX_2020 extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);

        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }

}
```

Aplicatie propusa



```
package javafx_2020;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

import javax.xml.soap.Text;

public class Main extends Application
{

    Stage window;
    Button Calculeaza;
    Label rez=new Label();


    TextField a = new TextField();
    TextField b = new TextField();
    TextField c = new TextField();

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

    @Override
    public void start(Stage primaryStage) throws Exception {
        window = primaryStage;
        window.setTitle("element minin");
        Calculeaza = new Button("Calculeaza");

        Calculeaza.setOnAction(e->{
            int                                    re                                    =
Math.min(Math.min(Integer.parseInt(a.getText()),Integer.parseInt(b.getText())),Integer.parse
Int(c.getText()));
            rez.setTextFill(Color.rgb(32,36,200));
            rez.setText("Numarul cel mai mic este : " + Integer.toString(re));

        });

        VBox layout = new VBox();
        layout.setPadding(new Insets(10,10,10,10));
        layout.setSpacing(10);
            layout.getChildren().addAll(a,b,c);
        layout.getChildren().add(Calculeaza);
        layout.getChildren().add(rez);
        Scene scene = new Scene(layout, 300, 250);

        window.setScene(scene);
        window.show();
    }


}
```



```
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.GridPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;


public class Ex_2 extends Application
{
```

```
Stage window;
Button Calculeaza;
Label rez=new Label();
TextField a = new TextField();
TextField b = new TextField();
TextField c = new TextField();

Label N1 = new Label("Nr 1:");
Label N2 = new Label("Nr 2:");
Label N3 = new Label("Nr 3:");
  int min = 0;


@Override

public void start(Stage primaryStage)  {
    window = primaryStage;
    window.setTitle("MINIM");
    Calculeaza = new Button("Calculeaza");

    GridPane pane = new GridPane();
    pane.setAlignment(Pos.CENTER);
    pane.setHgap(3);
    pane.setVgap(3);
    pane.setPadding(new Insets(25,25,25,25));

    a.setPrefColumnCount(14);
    b.setPrefColumnCount(14);
    c.setPrefColumnCount(14);

    pane.add(N1, 0, 1);
    pane.add(a, 1,1);
    pane.add(N2,0,2);
    pane.add(b,1,2);
    pane.add(N3,0,3);
    pane.add(c,1,3);

    pane.add(Calculeaza,1,8);
    pane.add(rez,1,9);

    BorderPane borderPane = new BorderPane();
    borderPane.setCenter(pane);

    Calculeaza.setOnAction(e->{
        min = Integer.parseInt(a.getText());
        int A1=Integer.parseInt(b.getText());
        int B1=Integer.parseInt(c.getText());
        int C1=Integer.parseInt(a.getText());
        if (B1 <= min)
        { min=B1;}
```

```
           if (C1 <= min)
          { min=C1;}
          rez.setTextFill(Color.rgb(32,36,200));

                rez.setText("Elementul minim este " + min );




        });
        Scene scene = new Scene(borderPane, 400, 400);

        window.setScene(scene);
        window.show();

    }



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