Tehnici avansate de programare

Curs -

Cristian Frăsinaru

acf@infoiasi.ro

Facultatea de Informatică

Universitatea ”Al. I. Cuza” Iaşi

JavaFX

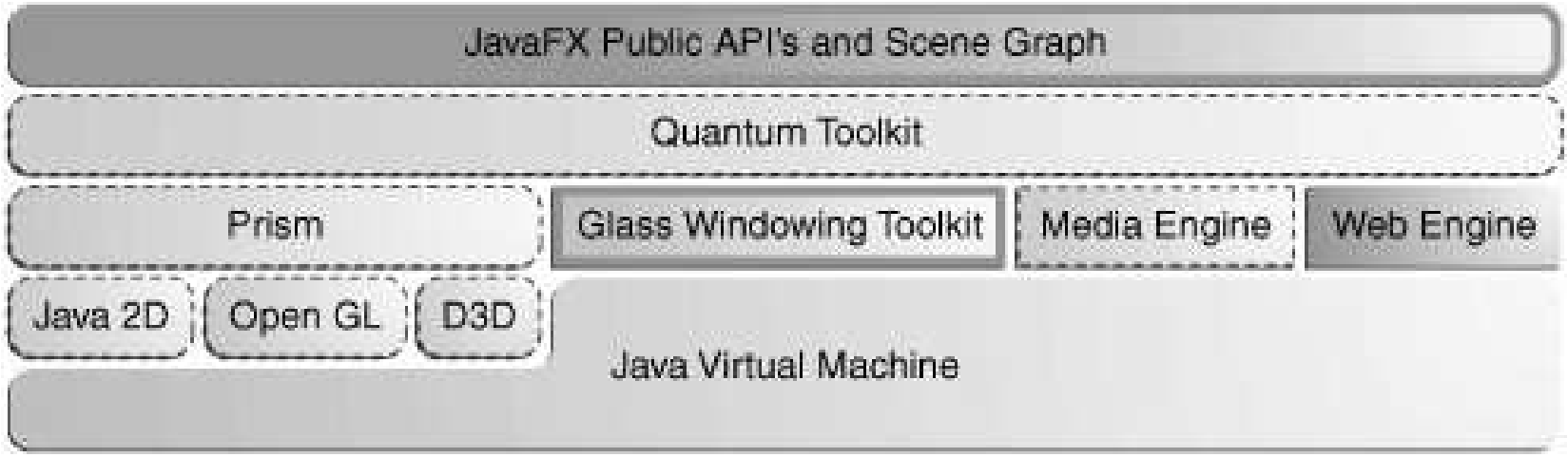
Cuprins

- Ce este JavaFX
- Arhitectura platformei
- Graful scenei
- Lucrul cu componente
- Gestionarea poziţionării
- Tratarea evenimentelor
- FXML

Ce este JavaFX?

- **Platformă** de programare bazată pe Java
- Creare de aplicaţii **RIA (Rich Internet Application)**
- **Portabilitate**: desktop, browser, dispozitive mobile, TV, console jocuri, Blu-ray, etc.
- *Data-driven enterprise client applications*
- Competitori: Adobe AIR, Apache Pivot, OpenLaszlo, Microsoft Silverlight, etc.
- Versiunea curentă: JavaFX 2.0 (oct, 2011)

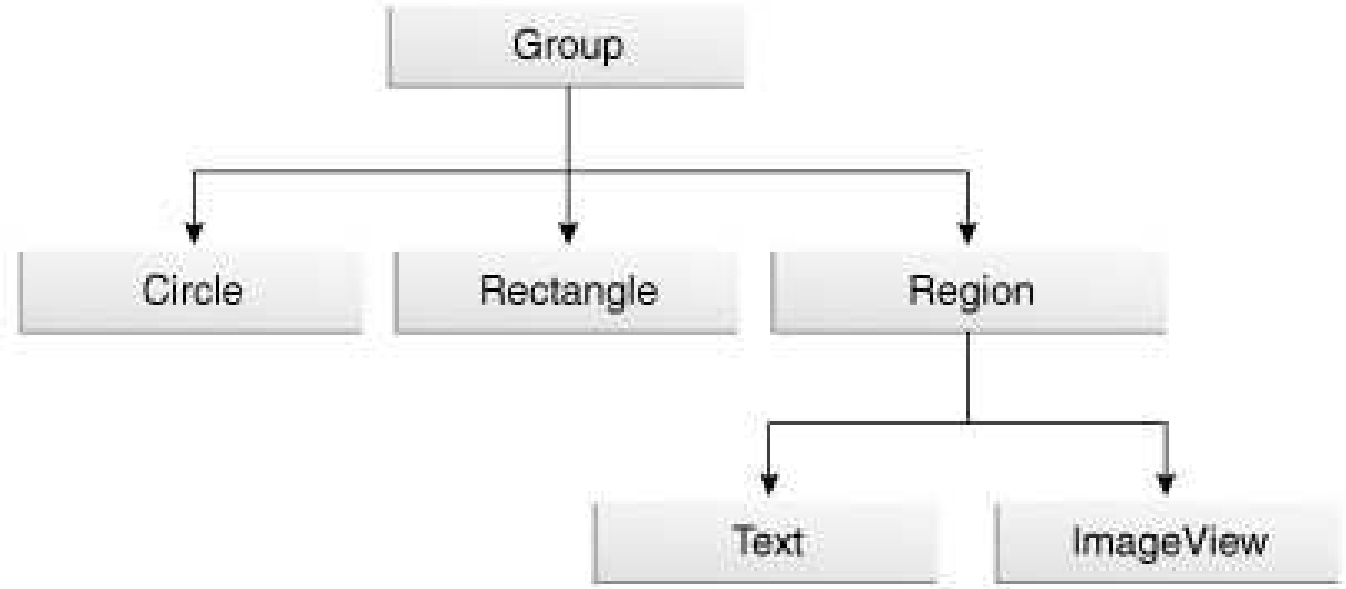
Arhitectura JavaFX



Graful *scenei* (Scene)

- **Retained mode API**
- **Arbore** de componente (`Node`)
- **Nodurile** pot fi **interne** (`Parent`) sau **frunză**
- Nodurile pot fi Shapes (2-D and 3-D), images, media, embedded web browser, text, UI controls, charts, groups, and containers
- Caracteristicile unui nod:
  - proprietăţi: *ID, style class, bounding volume, etc.*
  - efecte (blur, shadow), opacitate, transformări
  - *event handlers* (mouse, tastatură)
  - stare specifică aplicaţiei

Root, Branch, Leaf Nodes



Crearea unei aplicaţii

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class Main extends Application {
   @Override
   public void start(Stage stage) {
       Group root = new Group();
       Scene scene = new Scene(root, 500, 500, Color.BLACK);
       stage.setTitle("Welcome to JavaFX!");
       stage.setScene(scene);
       stage.show();
   }
   public static void main(String[] args) {
       launch(args);
   }
}
```

Adăugarea nodurilor

```
// Cream un nod de tip Group
Group group = new Group();
root.getChildren().add(group);

// Cream un nod de tip Rectangle
Rectangle r = new Rectangle(25,25,250,250);
r.setFill(Color.BLUE);
group.getChildren().add(r);

// Cream un nod de tip Circle
Circle c = new Circle(150, Color.web("blue", 0.5f));
c.setStrokeWidth(4f);
group.getChildren().add(c);
```

Crearea unei animaţii

```
TranslateTransition translate =
        new TranslateTransition(Duration.millis(750));
translate.setToX(390); translate.setToY(390);

FillTransition fill = new FillTransition(Duration.millis(750));
fill.setToValue(Color.RED);

RotateTransition rotate = new RotateTransition(Duration.millis(750));
rotate.setToAngle(360);

ScaleTransition scale = new ScaleTransition(Duration.millis(750));
scale.setToX(0.1); scale.setToY(0.1);

ParallelTransition transition = new ParallelTransition(r,
        translate, fill, rotate, scale);
transition.setCycleCount(Timeline.INDEFINITE);
transition.setAutoReverse(true);
transition.play();
```

Conceptul de *puls*

- **Puls** = eveniment care indică faptul că graful scenei trebuie sincronizat cu procesul de desenare propriu-zisă (*Prism*)
- Lansat la maximul 60 cadre pe secundă (fps)
- Lansat atunci când există animaţii în derulare sau este detectată orice schimbare la nivelul GUI (schimbarea poziţiei unui buton, de exemplu)
- Tratare **asincronă** a evenimentelor
- Planificarea execuţiei unui puls se face prin timere native ale SO via *Glass Windowing Toolkit*

Lucrul cu componente

Superclasa **Control**

```
java.lang.Object
    javafx.scene.Node
        javafx.scene.Parent
            javafx.scene.control.Control
```

Crearea unui buton şi tratarea evenimentului generat

```
Image imageOk = new Image(getClass().getResourceAsStream("ok.png"));
Button button = new Button("Accept", new ImageView(imageOk));

button.setOnAction(new EventHandler<ActionEvent>() {
    @Override public void handle(ActionEvent e) {
        label.setText("Accepted");
    }
});
```

Efecte şi stiluri

Adăugarea unui efect de tip "umbră" butonului

```
DropShadow shadow = new DropShadow();

//Adding the shadow when the mouse cursor is on
button.addEventHandler(MouseEvent.MOUSE_ENTERED,
    new EventHandler<MouseEvent>() {
        @Override public void handle(MouseEvent e) {
            button.setEffect(shadow);
        }
});
```

Configurarea "stilului" butonului (CSS)

```
button.setStyle("-fx-font: 22 arial; -fx-base: #b6e7c9;");
```

Gestionarea poziţionării

Superclasa **Pane**

```
    java.lang.Object
        javafx.scene.Node
            javafx.scene.Parent
                javafx.scene.layout.Region
                    javafx.scene.layout.Pane
--> AnchorPane, BorderPane, FlowPane, GridPane,
    HBox, StackPane, TilePane, VBox
```

Crearea unui gestionar de tip `BorderPane`

```
BorderPane layout = new BorderPane();
layout.setTop(new Rectangle(200, 50, Color.DARKCYAN));
layout.setBottom(...));
layout.setCenter(...);
layout.setLeft(...);
layout.setRight(...);
```

Tratarea evenimentelor

Se face la nivel de arbore: `EventDispatchChain`

- *Target selection*
- *Route construction*
- *Event capturing*
  - Event filters (`addEventFilter`)
  - Event handlers (`setOn`$TipEveniment$))
- *Event bubbling*

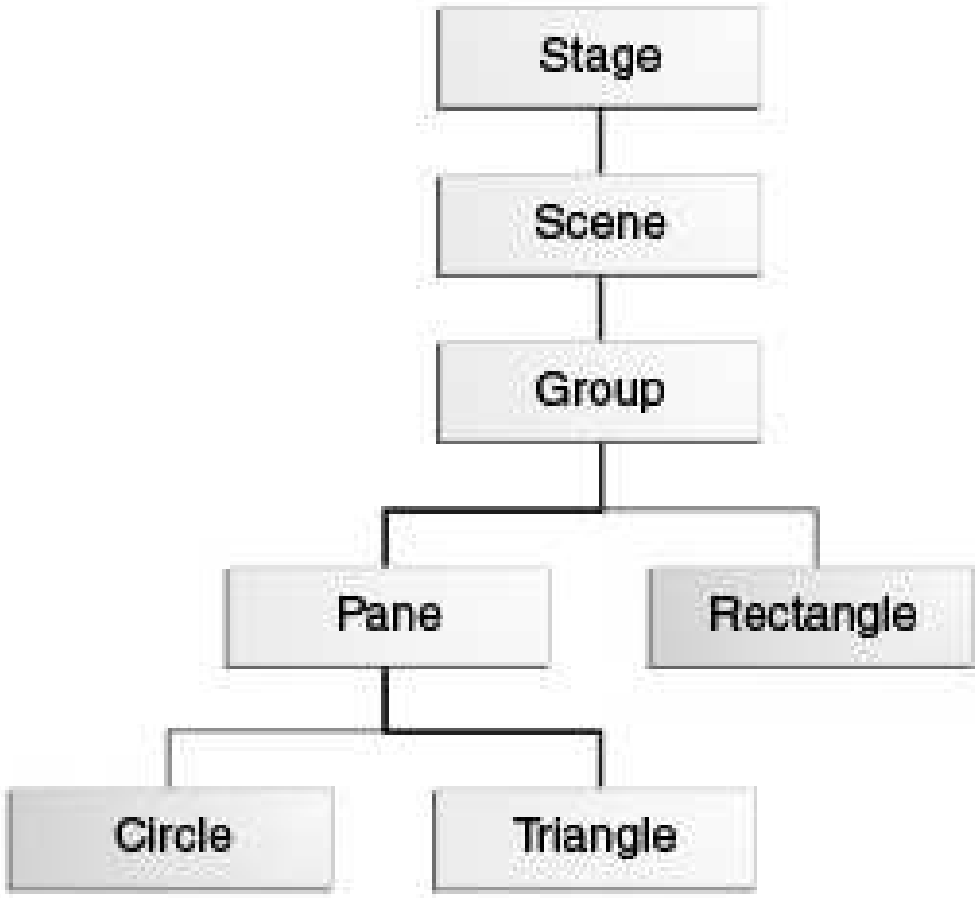
Evenimentele pot fi **consumate**.

Filtre de evenimente

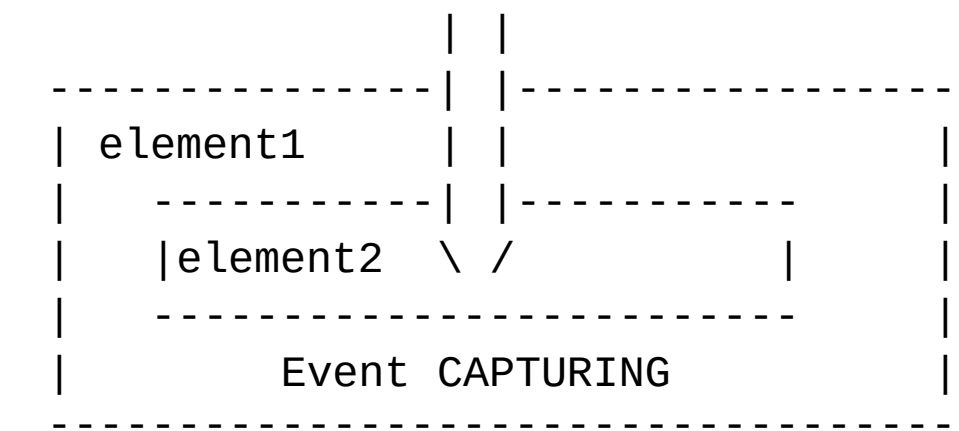
Intercepting Filter Design Pattern

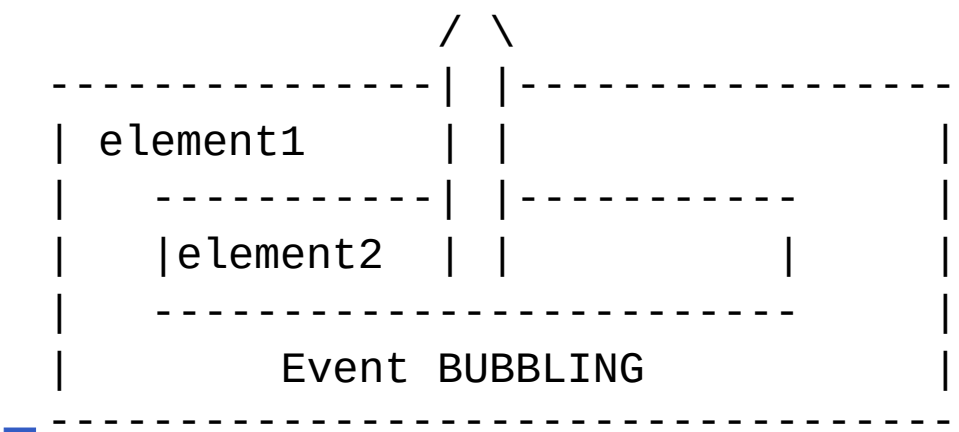
```
// Register an event filter for a single node
node.addEventFilter(MouseEvent.MOUSE_CLICKED,
                    new EventHandler<MouseEvent>() {
                        public void handle(MouseEvent) { ... };
                    });
// Define an event filter
EventHandler filter = new EventHandler(<InputEvent>() {
    public void handle(InputEvent event) {
        System.out.println("Filtering:" + event.getEventType());
        event.consume();
    }
// Register the same filter for two different nodes
myNode1.addEventFilter(MouseEvent.MOUSE_PRESSED, filter);
myNode2.addEventFilter(MouseEvent.MOUSE_PRESSED, filter);
// Register the filter for another event type
myNode1.addEventFilter(KeyEvent.KEY_PRESSED, filter);
```

Event Dispatch Chain



Event Capturing vs. Bubbling





FXML

- **FXML** = limbaj de marcare bazat pe XML dedicat construirii de grafuri de opiecte (scene).
- Tehnică **declarativă** (versus programatică)
- Din categoria: XUL, XAML, etc.
- Arbore de componente $\sim$ arbore DOM XML
- Separarea rolurilor
- Independenţă de limbaj (Java, Scala, Clojure, etc.)
- Suport pentru internaţionalizare

Programatic vs. Declarativ

Programatic

```
BorderPane border = new BorderPane();
Label top = new Label("Page Title");
border.setTop(top);
Label center = new Label ("Some data here");
border.setCenter(center);
```

Declarativ

```
<BorderPane>
    <top>
        <Label text="Page Title"/>
    </top>
    <center>
        <Label text="Some data here"/>
    </center>
</BorderPane>
```

Exemplu

```
<AnchorPane fx:controller="demo.LoginController"
            id="Login" xmlns:fx="http://javafx.com/fxml">
  <maxWidth><Double fx:value="-Infinity"/></maxWidth>
  <maxHeight><Double fx:value="-Infinity"/></maxHeight>
  <children>
    <TextField layoutX="68.0" layoutY="58.0"
               prefWidth="126.0" fx:id="userId"
               onAction="#processLogin"/>
    <PasswordField ... />
    <Label ..." />
    <Button ...  onAction="#processLogin"/>
    ...
  </children>
  <properties>
    <backgroundColor>
      <Color blue="1.0" green="1.0" red="1.0" />
    </backgroundColor>
  </properties>
</AnchorPane>
```

FXML Loader

```
URL location = getClass().getResource("example.fxml");
ResourceBundle resources = ResourceBundle.getBundle("messages");

FXMLLoader fxmlLoader = new FXMLLoader();
fxmlLoader.setLocation(location);
fxmlLoader.setResources(resources);
fxmlLoader.setBuilderFactory(new JavaFXBuilderFactory());

InputStream inputStream = null;
Pane root;
try {
    inputStream = location.openStream();
    root = (Pane)fxmlLoader.load(inputStream);
} finally {
    if (inputStream != null) {
        inputStream.close();
    }
}
```

Swing sau JavaFX?

Pro Swing

- Maturitate
- Biblioteci de componente, Framework-uri
- Documentaţie şi resurse bogate

Pro JavaFX

- Tehnologie modernă, nu doar pentru desktop (RIA)
- Orientată spre "spectaculos"
- Suport pentru animaţii, grafică 2D, 3D

Documentaţie

- **Creating a JavaFX GUI** Tutoriale JavaFX

  http://docs.oracle.com/javafx/index.html

- **JavaFX API**

  http://docs.oracle.com/javafx/2.0/api/index.html