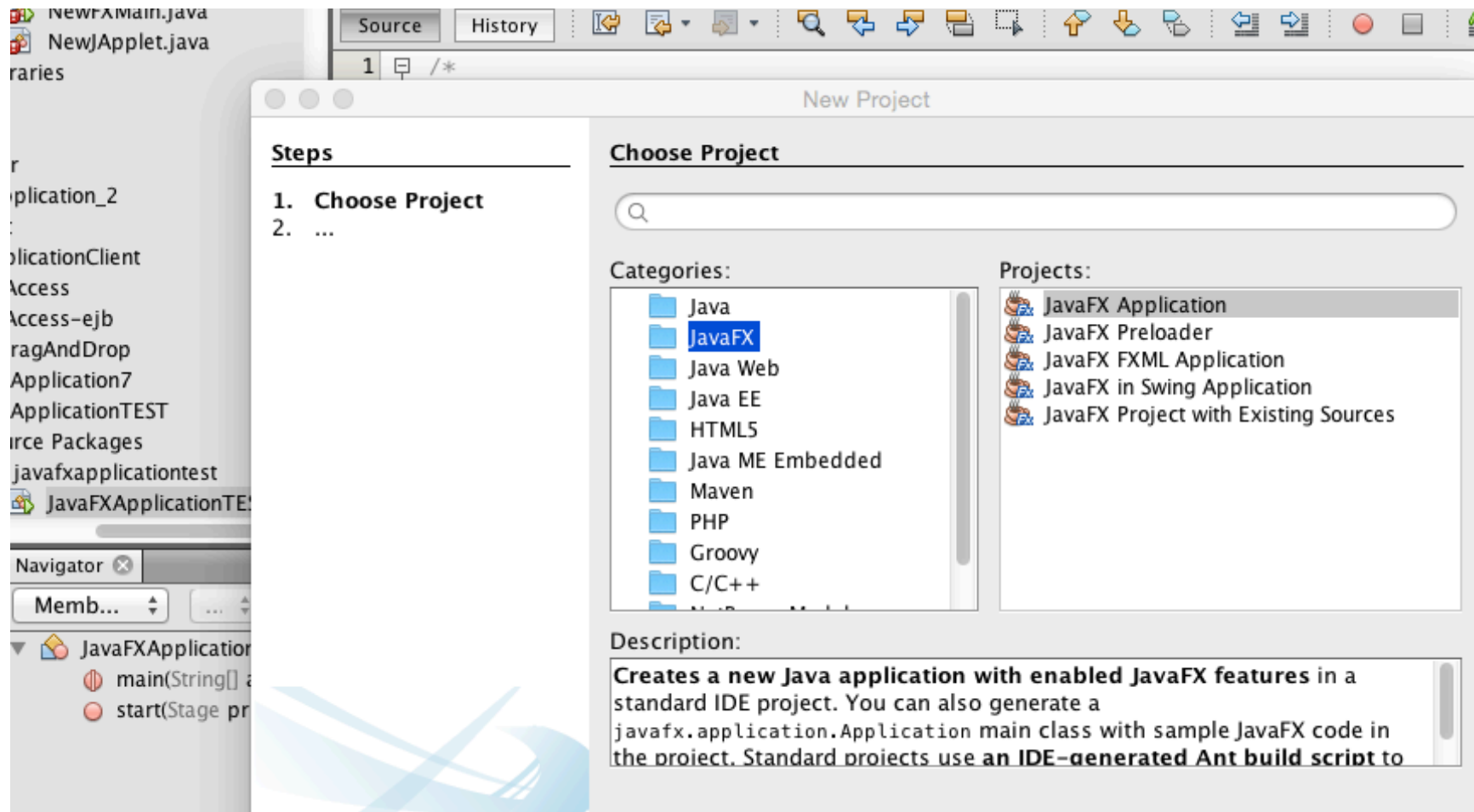


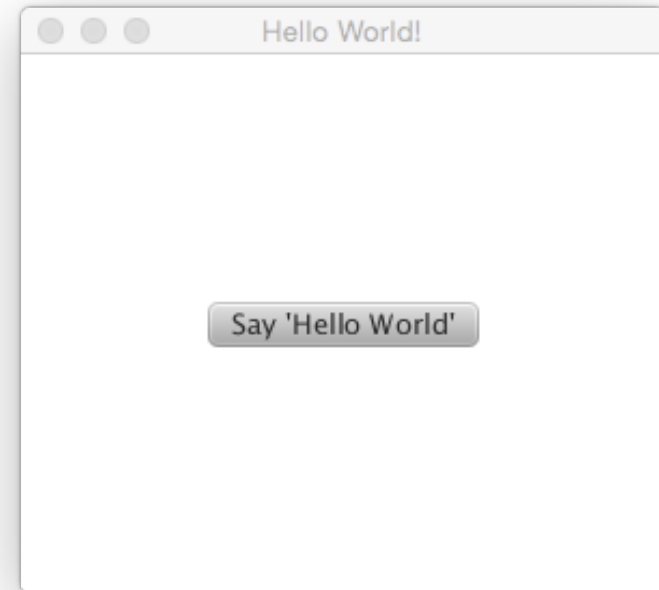
Grafica e non solo: Java FX

Creazione di una Applicazione JavaFX



Java FX

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

Java FX

```
public class JavaFXApplicationTEST 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();
```

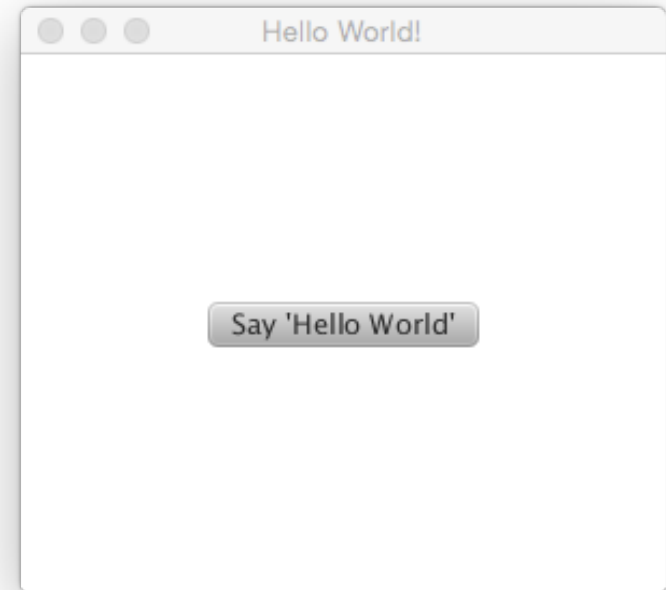
```
    }
```

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

```
        launch(args);
```

```
    }
```

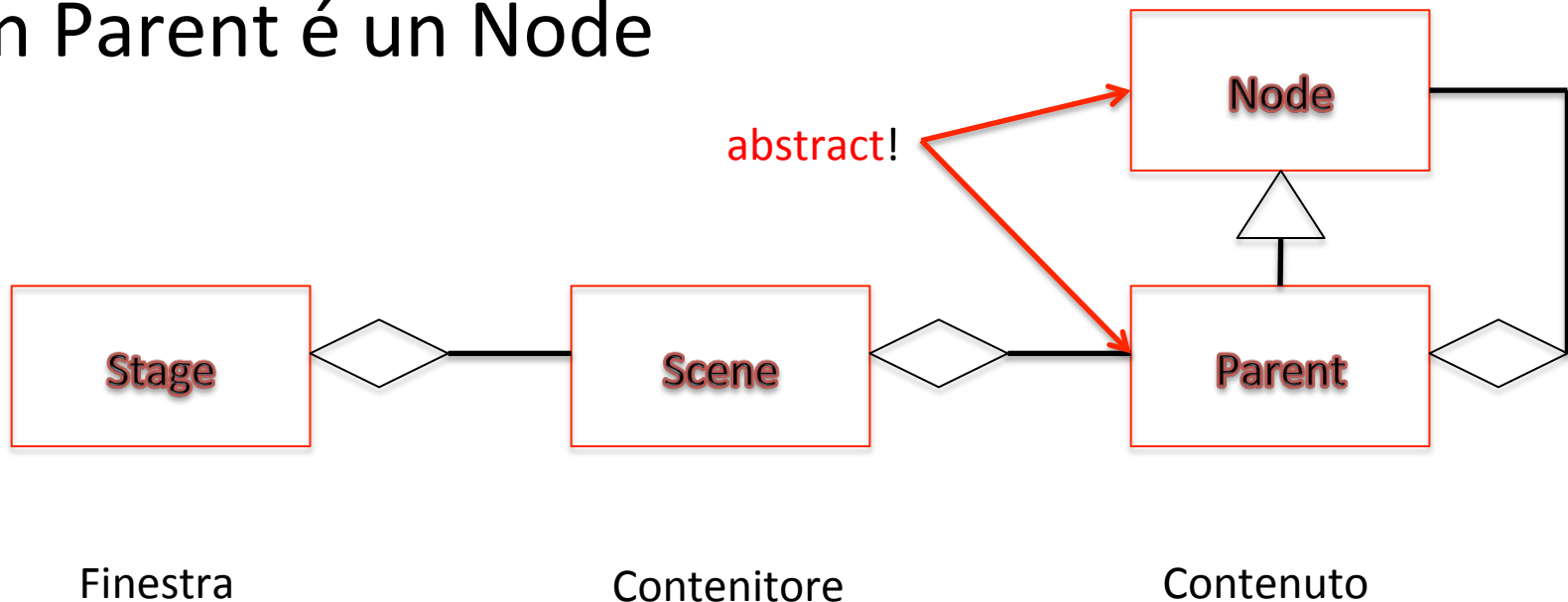
```
}
```



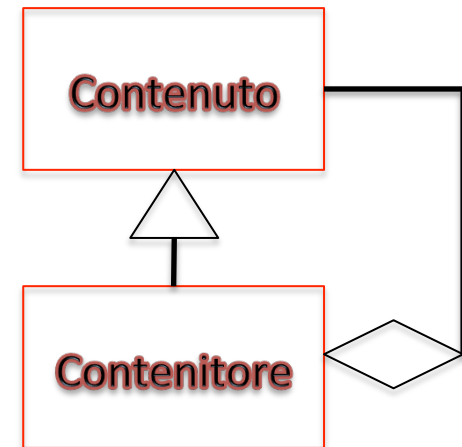
Stage/Scene/Parent/Node

Finestra == Stage

- Uno Stage contiene una Scene
- Una Scene ha un Parent
- Un Parent é un Node



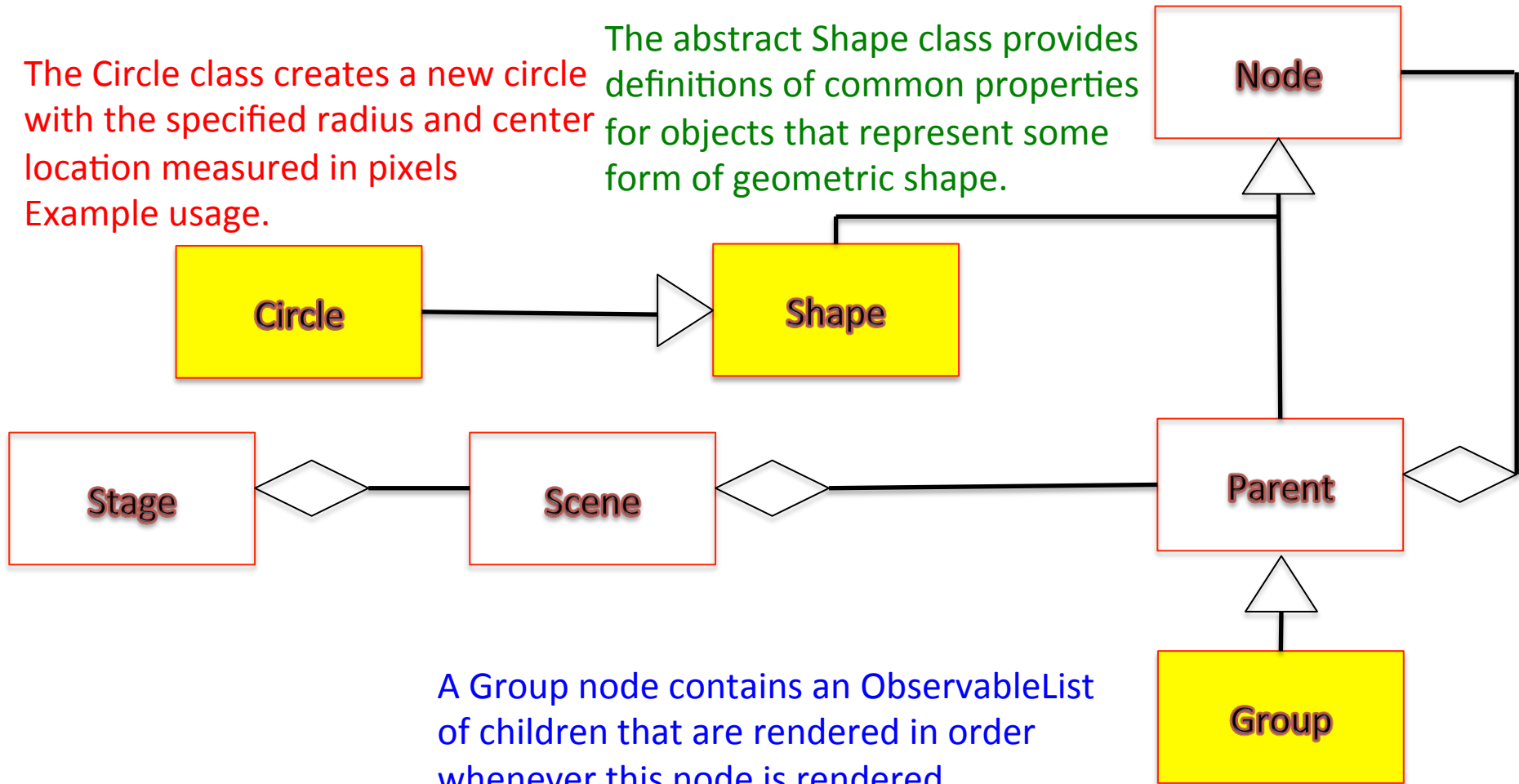
Combinazione speciale di is-a e has-a



Group – Shape - Circle

The Circle class creates a new circle with the specified radius and center location measured in pixels
Example usage.

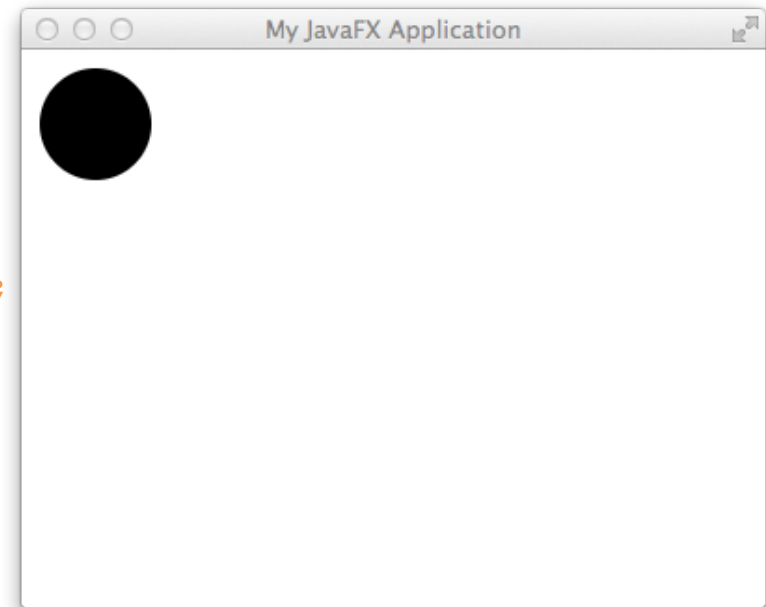
The abstract Shape class provides definitions of common properties for objects that represent some form of geometric shape.



A Group node contains an ObservableList of children that are rendered in order whenever this node is rendered.

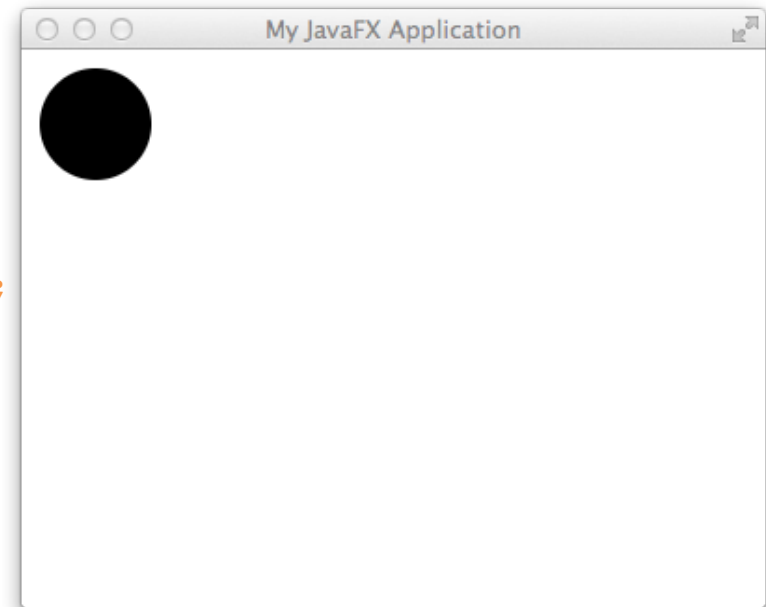
Applicazione minima

```
package it.unitn.disi.javafxapplication;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
public class MinimalApp extends Application {
    public void start(Stage stage) {
        Node circ = new Circle(40, 40, 30);
        Parent root = new Group(circ);
        Scene scene = new Scene(root, 400, 300);
        stage.setTitle("My JavaFX Application");
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String[] args) {
        Application.launch(args);
    } }
```

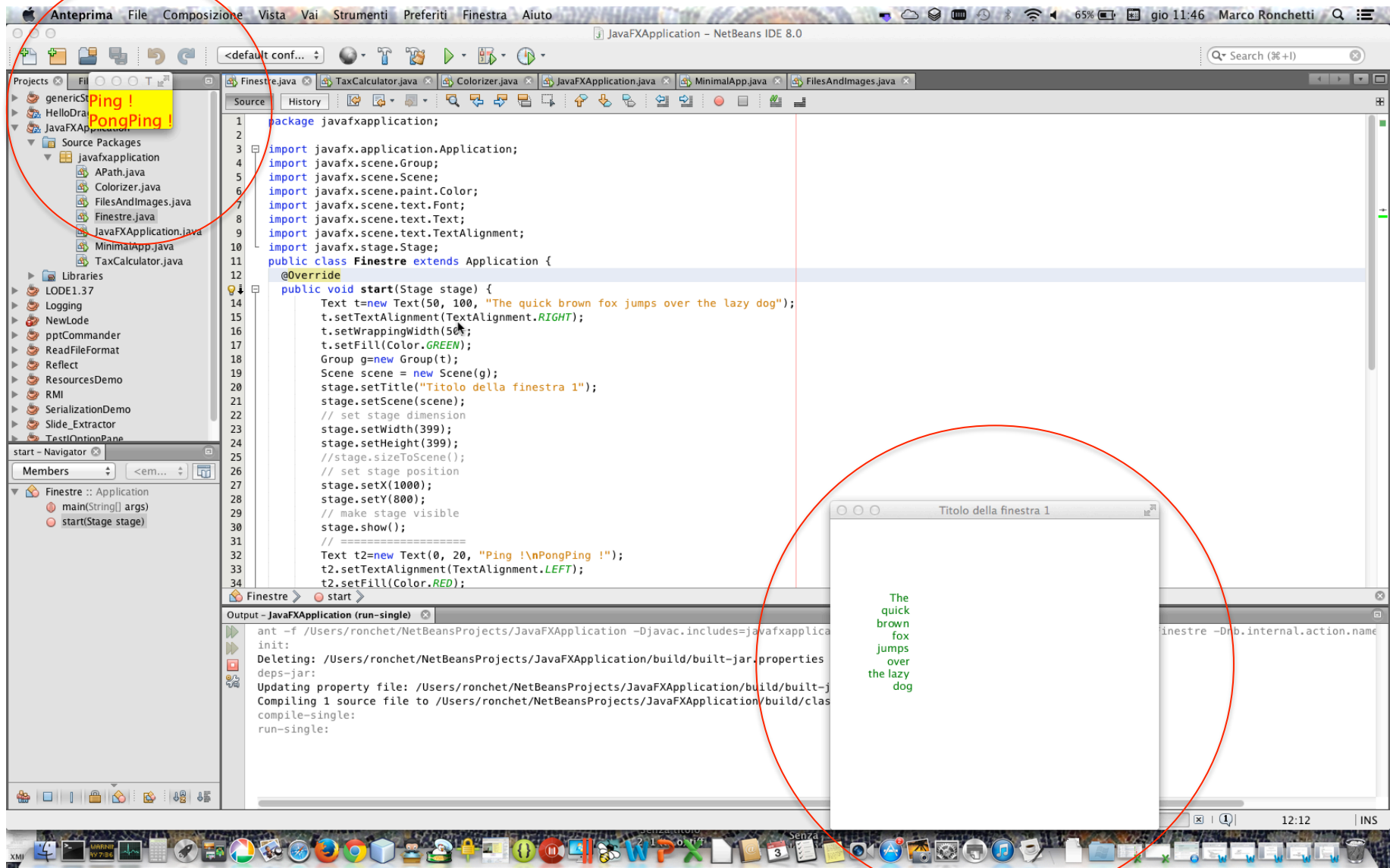


Applicazione minima

```
package it.unitn.disi.javafxapplication;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
public class MinimalApp extends Application {
    public void start(Stage stage) {
        Circle circ = new Circle(40, 40, 30);
        Group root = new Group(circ);
        Scene scene = new Scene(root, 400, 300);
        stage.setTitle("My JavaFX Application");
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String[] args) {
        Application.launch(args);
    } }
```

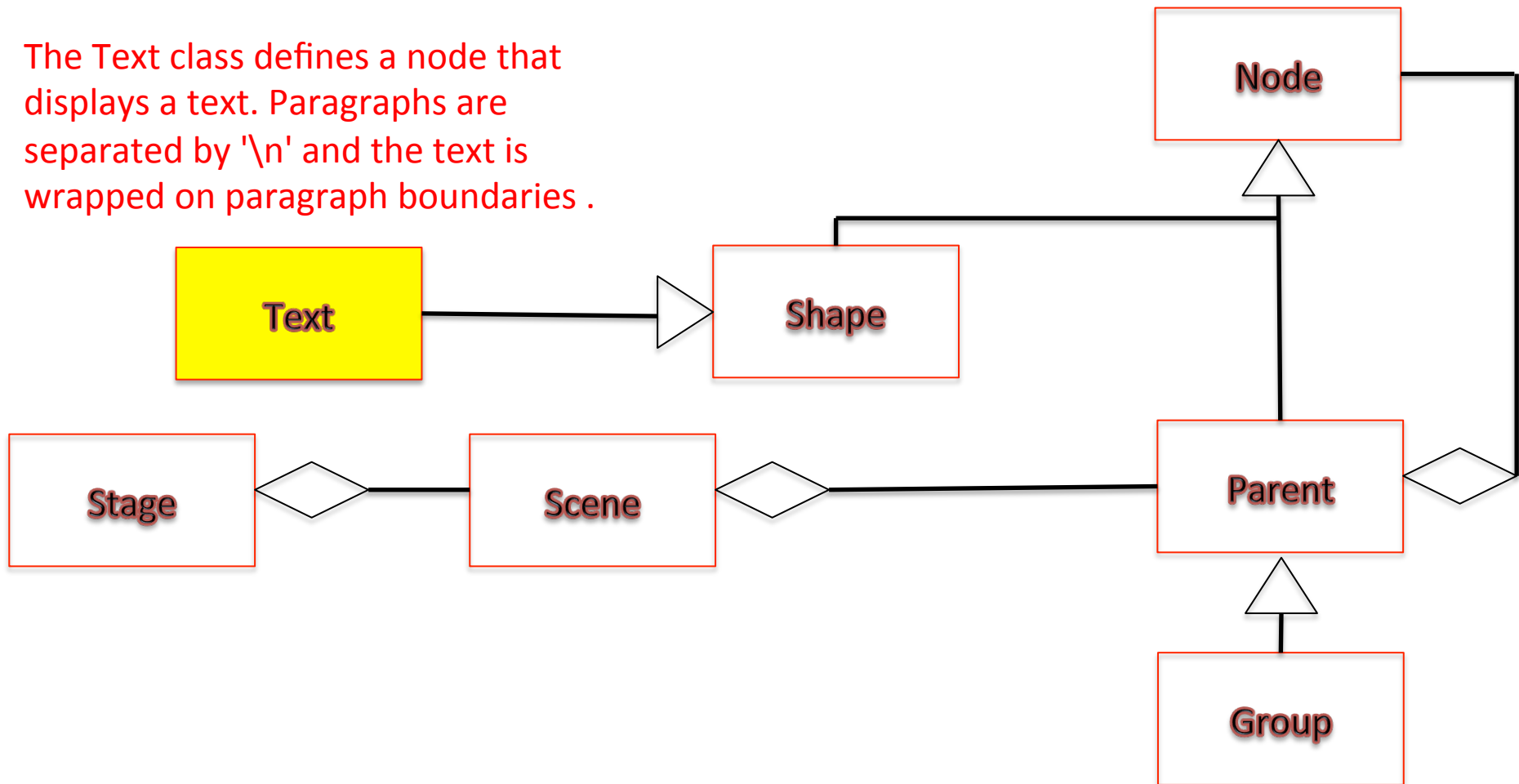


Finestre multiple



Group – Shape - Circle

The Text class defines a node that displays a text. Paragraphs are separated by '\n' and the text is wrapped on paragraph boundaries .



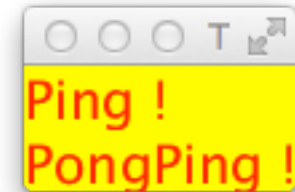
Finestre multiple: Prima finestra

```
public class Finestre extends Application {  
    public void start(Stage stage) {  
        Text t=new Text(50, 100, "The quick brown fox jumps over  
            the lazy dog");  
        t.setTextAlignment(TextAlignment.RIGHT);  
        t.setWrappingWidth(50);  
        t.setFill(Paint.valueOf("GREEN"));  
        Group g=new Group(t);  
        Scene scene = new Scene(g);  
        stage.setTitle("Titolo della finestra 1");  
        stage.setScene(scene);  
        // set stage dimension  
        stage.setWidth(399);  
        stage.setHeight(399);  
        // set stage position  
        stage.setX(1000);  
        stage.setY(800);  
        // make stage visible  
        stage.show();  
    }  
}
```



Finestre multiple: Seconda finestra

```
Text t2=new Text(0, 20, "Ping !\nPongPing !");
t2.setTextAlignment(TextAlignment.LEFT);
t2.setFill(Paint.valueOf("RED"));
t2.setFont(new Font(20));
Group g2=new Group(t2);
Scene scene2 = new Scene(g2);
scene2.setFill(Paint.valueOf("YELLOW"));
Stage stage2=new Stage();
stage2.setTitle("Titolo della finestra 2");
stage2.setScene(scene2);
stage2.setX(100);
stage2.setY(80);
stage2.sizeToScene();
stage2.show();
}
public static void main(String[] args) {
    launch(args);
} }
```



Attenzione agli import!

```
t2.setFill(  
    Paint.valueOf("RED"));  
t2.setFont(new Font(20));
```

💡 Add import for java.awt.Font

💡 Add import for javafx.scene.text.Font

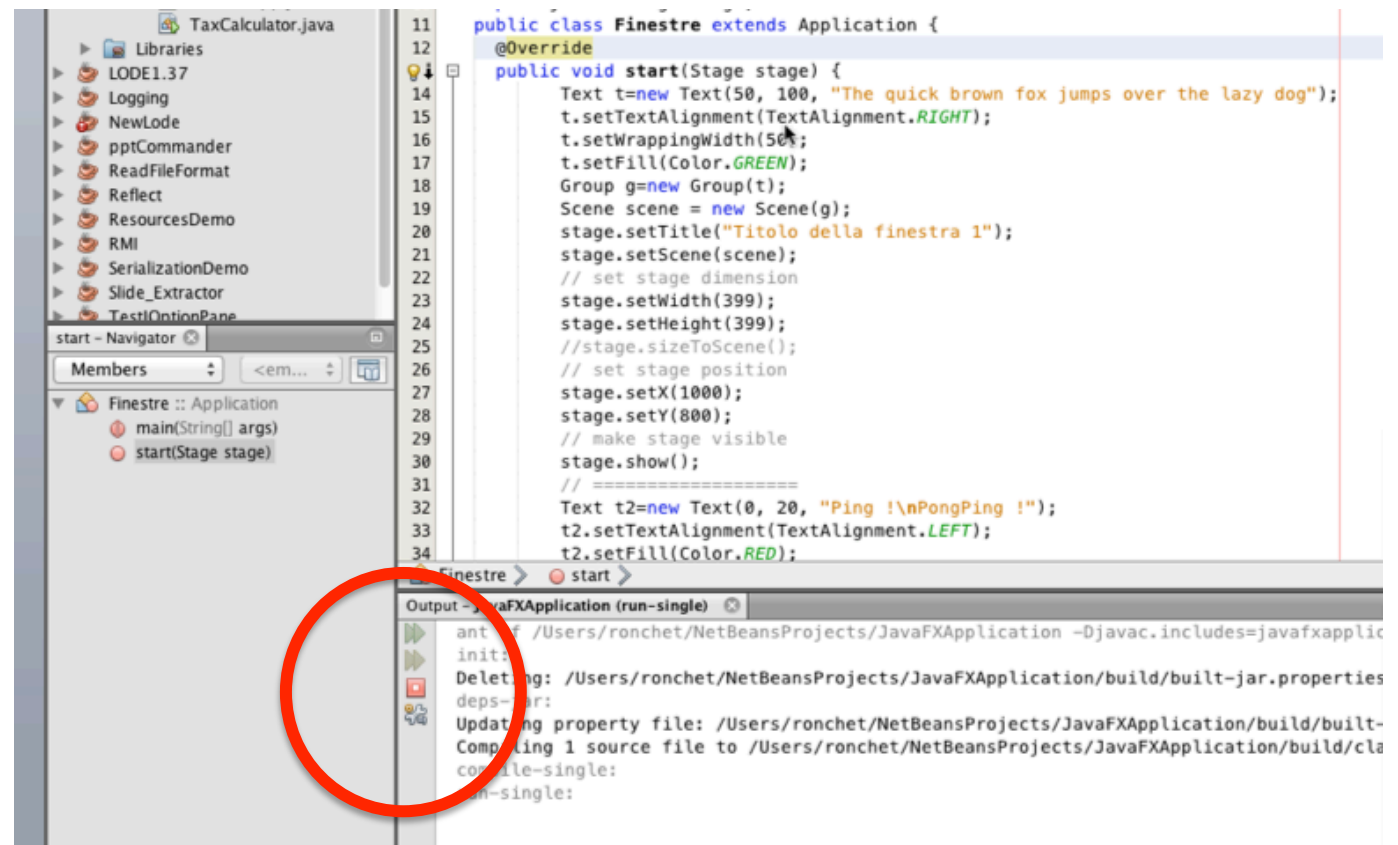
💡 Create class "Font" with constructor "Font(int)" in package javafx_001

💡 Create class "Font" in javafx_001.JavaFX_001

```
Paint.valueOf(  
    "YELLOW"));  
Stage stage2 = new Stage();
```

Terminazione

- Quando termina il processo?
(Un Processo é un Programma in esecuzione)

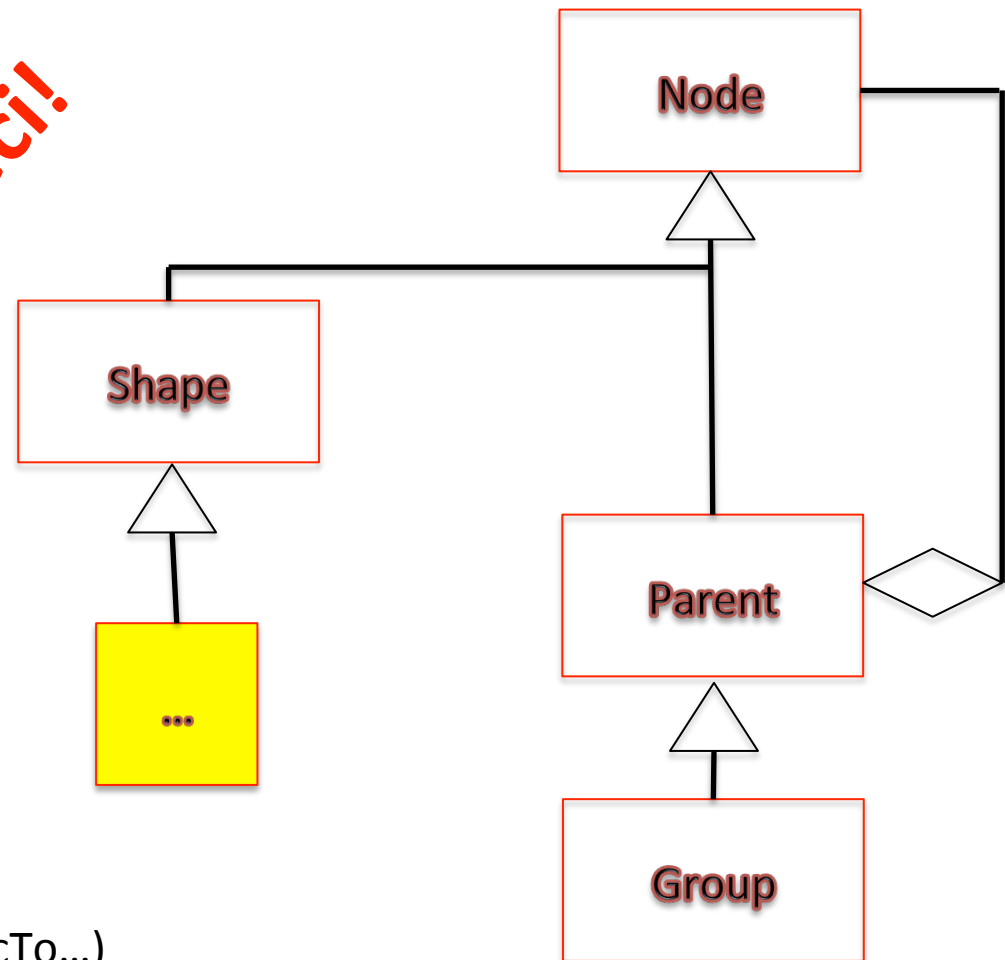


Shape hierarchy

Shape

- Line
- Polyline
- Polygon
- Rectangle
- Arc
- Circle
- Ellipse
- QuadCurve
- CubicCurve
- Text
- SVGPath
- Path composto di PathElement (ArcTo...)

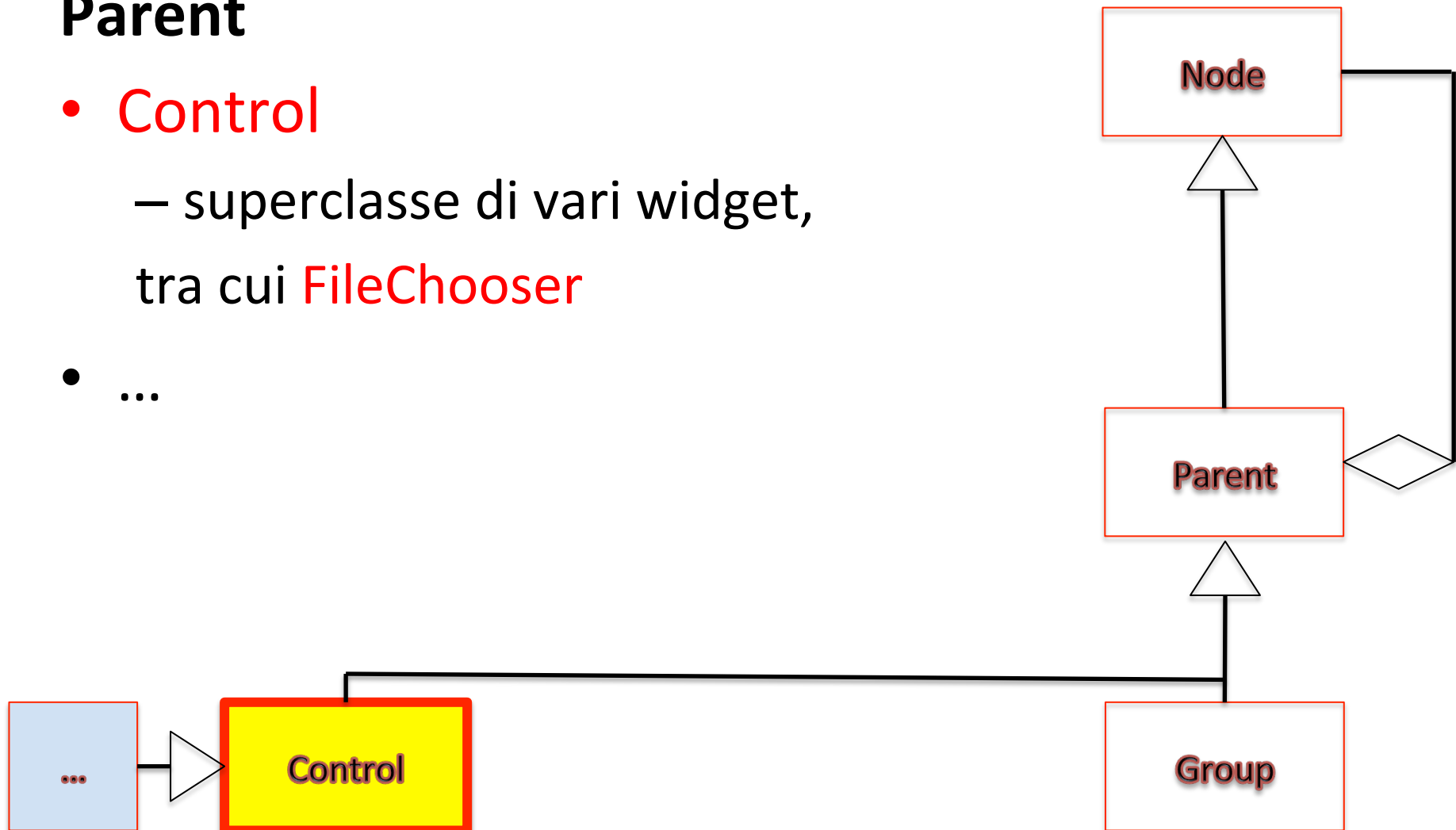
Giocateci!



Parent hierarchy (partial)

Parent

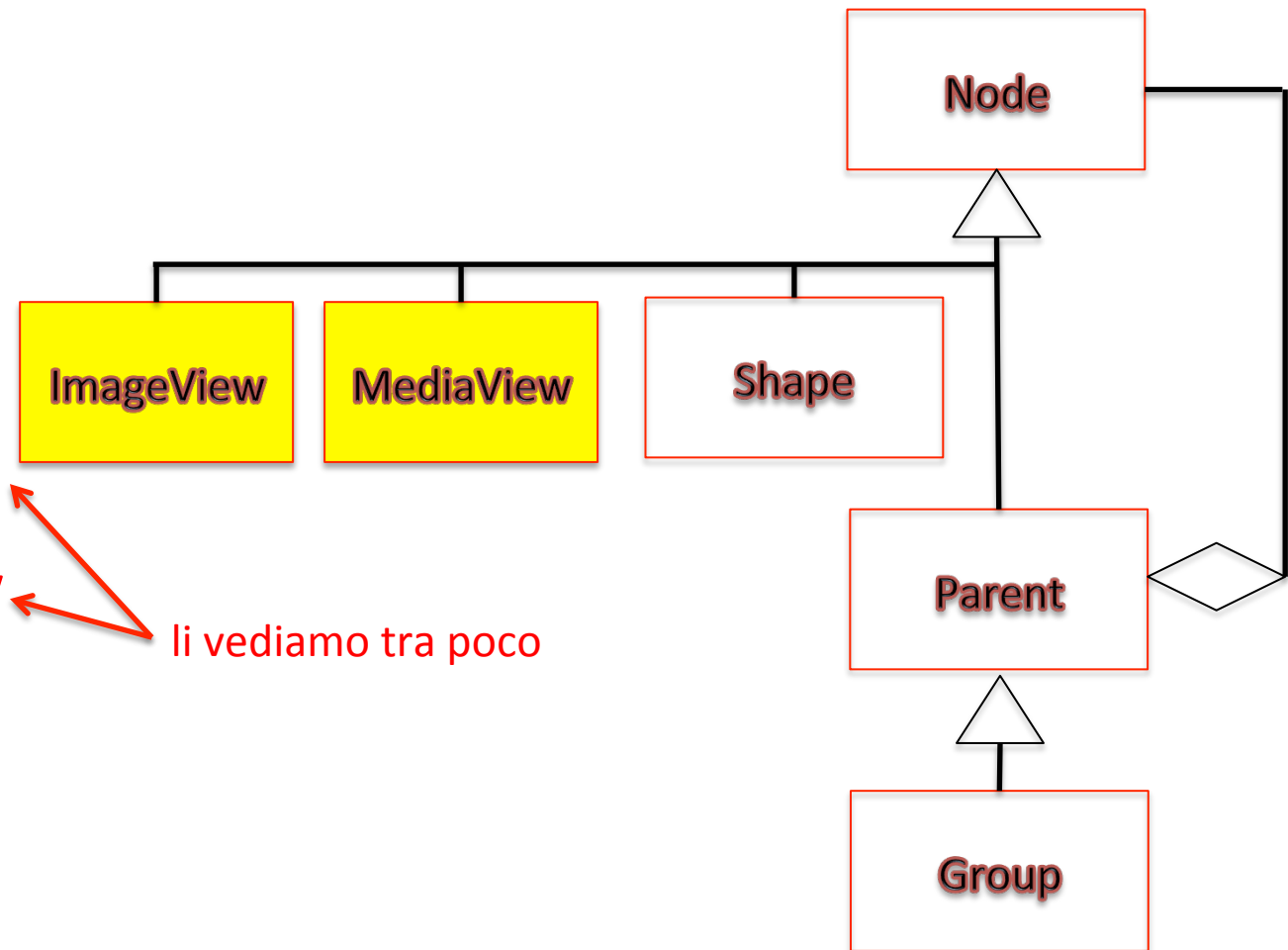
- **Control**
 - superclasse di vari widget, tra cui **FileChooser**
- ...



Node hierarchy (partial)

Node

- Parent
- Shape
- **ImageView**
- **MediaView**



li vediamo tra poco

ImageView - Image

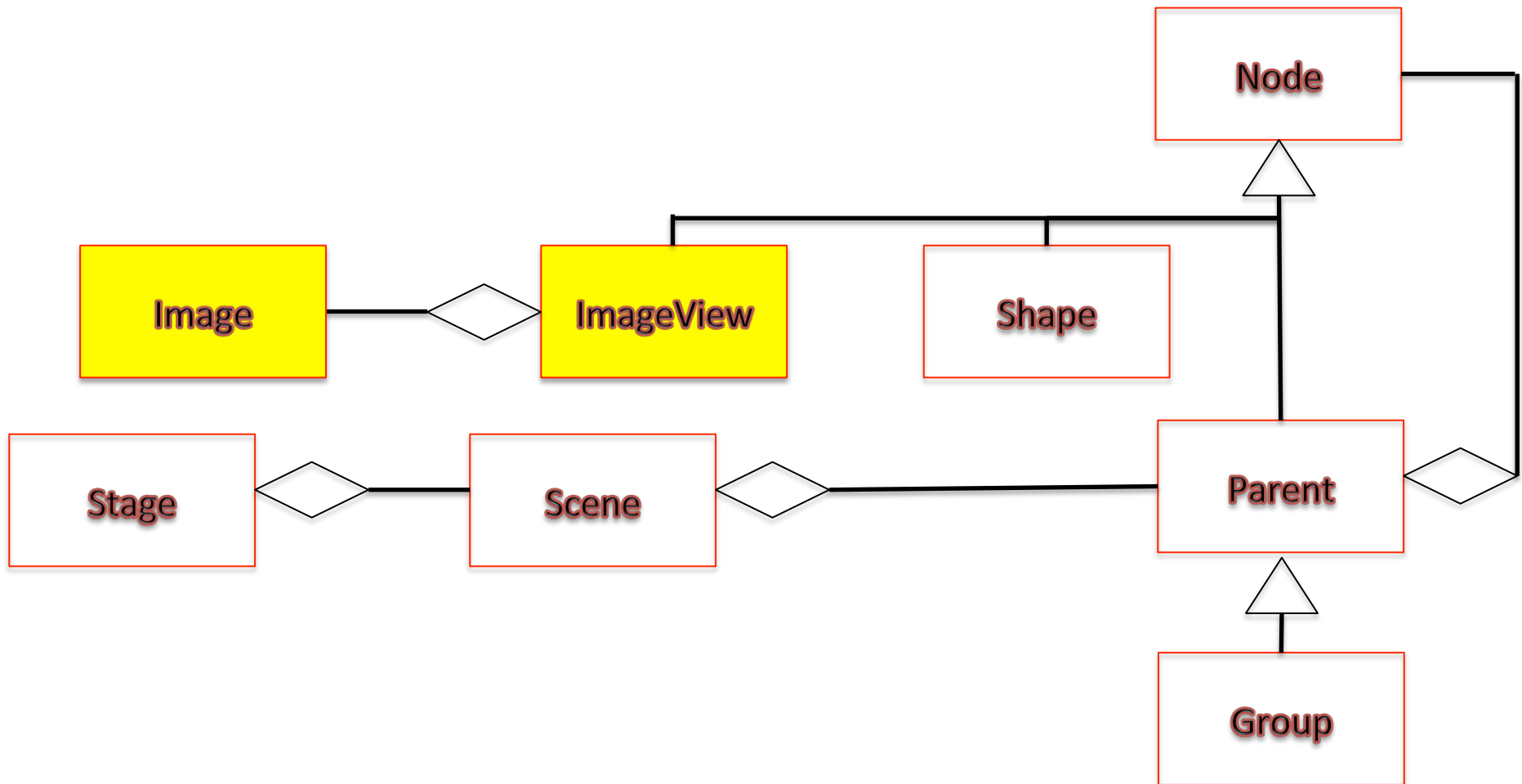
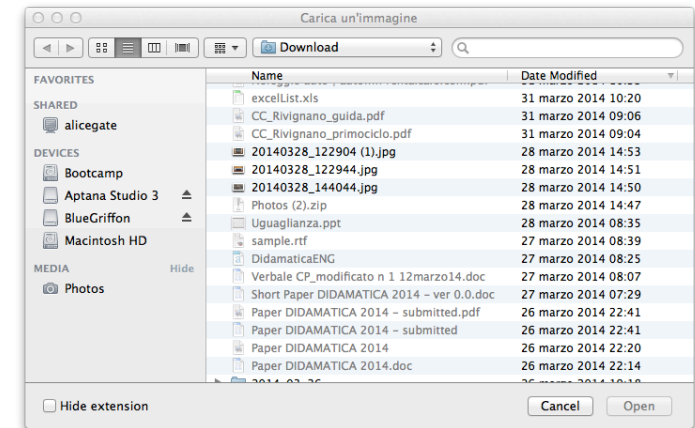


Image and File



```
public class FilesAndImages extends Application {  
    public void start(Stage stage) {  
        FileChooser fileChooser = new FileChooser();  
        fileChooser.setTitle("Carica un'immagine");  
        fileChooser.getExtensionFilters().addAll(  
            new FileChooser.ExtensionFilter("JPG", "*.jpg"),  
            new FileChooser.ExtensionFilter("PNG", "*.png")  
        );  
        String url = System.getProperty("user.home");  
        File f=new File(url);  
        fileChooser.setInitialDirectory(f); // bugged on MacOSX  
        File file = fileChooser.showOpenDialog(stage);  
        if (file == null) {  
            System.out.println("No file chosen");  
            System.exit(1);  
        }  
    }  
}
```

Image and File

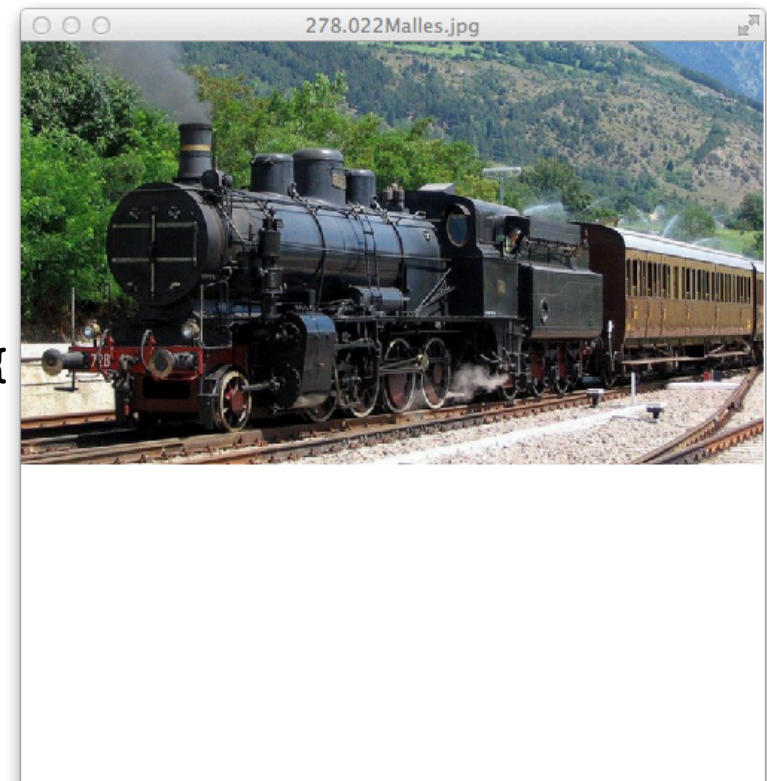
```
Image image = new Image("file://" +  
    file.getAbsolutePath(), 500, 500, true, true);  
ImageView iw = new ImageView(image);  
Group root = new Group(iw);  
Scene scene = new Scene(root, 500,500);  
stage.setTitle(file.getName());  
stage.setScene(scene);  
stage.sizeToScene();  
stage.show();
```

```
}
```

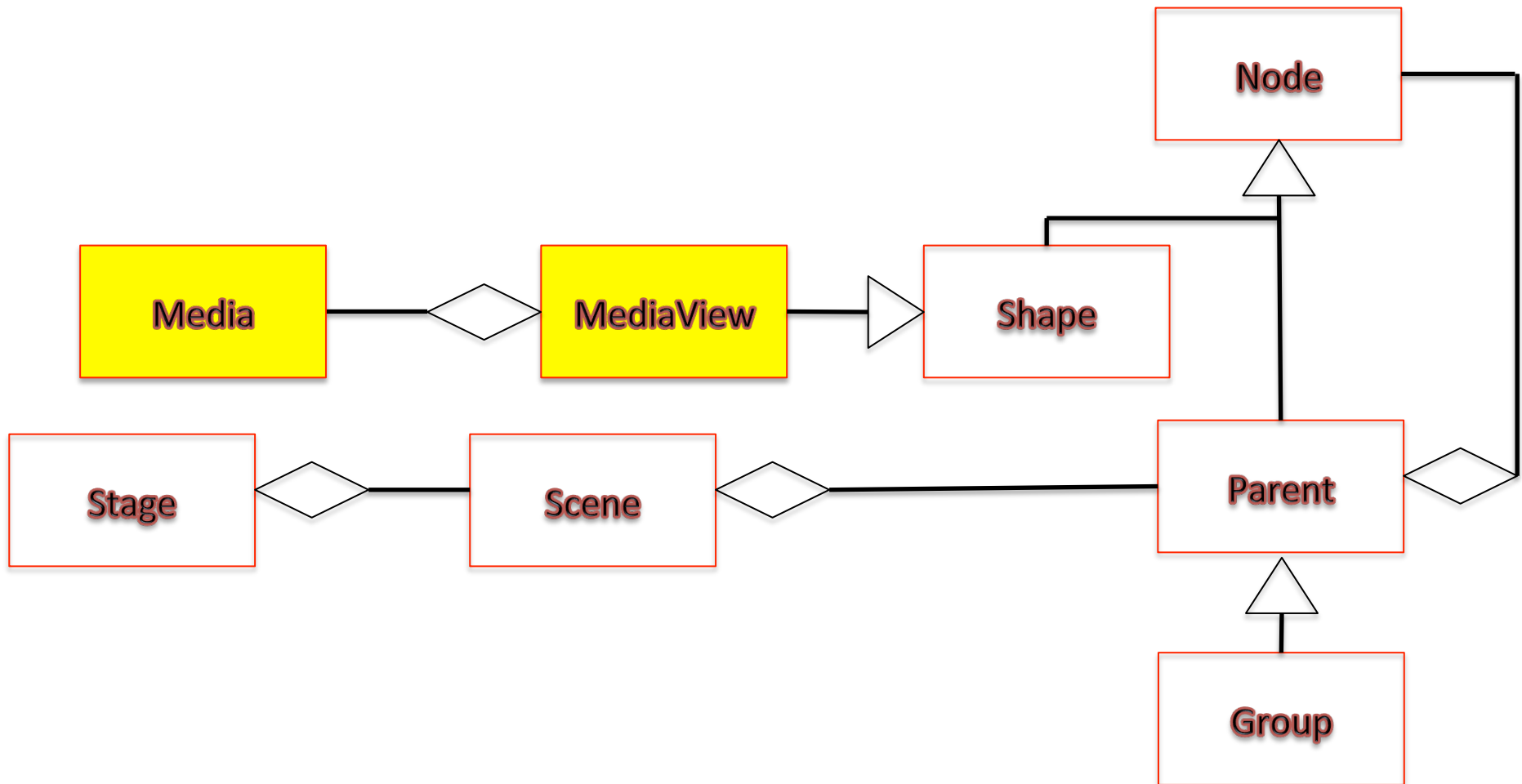
```
public static void main(String[] args) {  
    Application.launch(args);
```

```
}
```

```
}
```

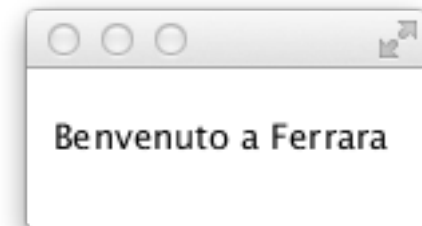


MediaView - Media



MediaView

```
public class Sounds extends Application{  
    public void start(Stage stage) {  
        Media media = new Media("http://www.ferraraterraeacqua.it/  
it/audioguide/audioguide-di-ferrara-citta-del-rinascimento/  
01_benvenuto-a-ferrara.mp3");  
        MediaPlayer mediaPlayer = new MediaPlayer(media);  
        mediaPlayer.setAutoPlay(true);  
        // create mediaView and add media player to the viewer  
        MediaView mediaView = new MediaView(mediaPlayer);  
        Group root = new Group(mediaView);  
        root.getChildren().add(  
            new Text(10, 30, "Benvenuto a Ferrara"));  
        Scene scene = new Scene(root, 150, 60);  
        stage.setScene(scene);  
        stage.sizeToScene();  
        stage.show();  
    }  
    public static void main(String[] args) {  
        Application.launch(args);  
    }  
}
```

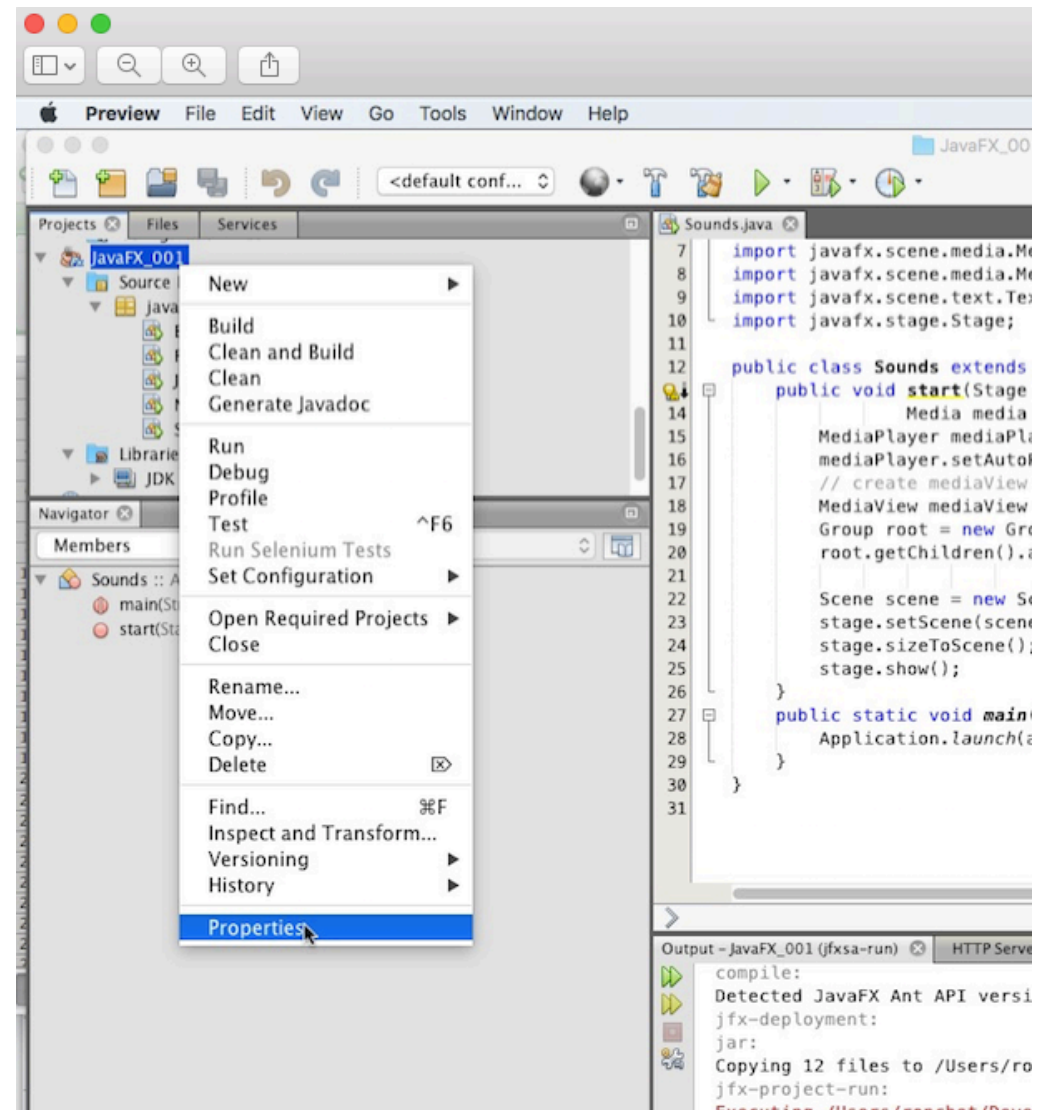


<http://docs.oracle.com/javafx/2/media/overview.htm>

Crea un programma distribuibile - 1

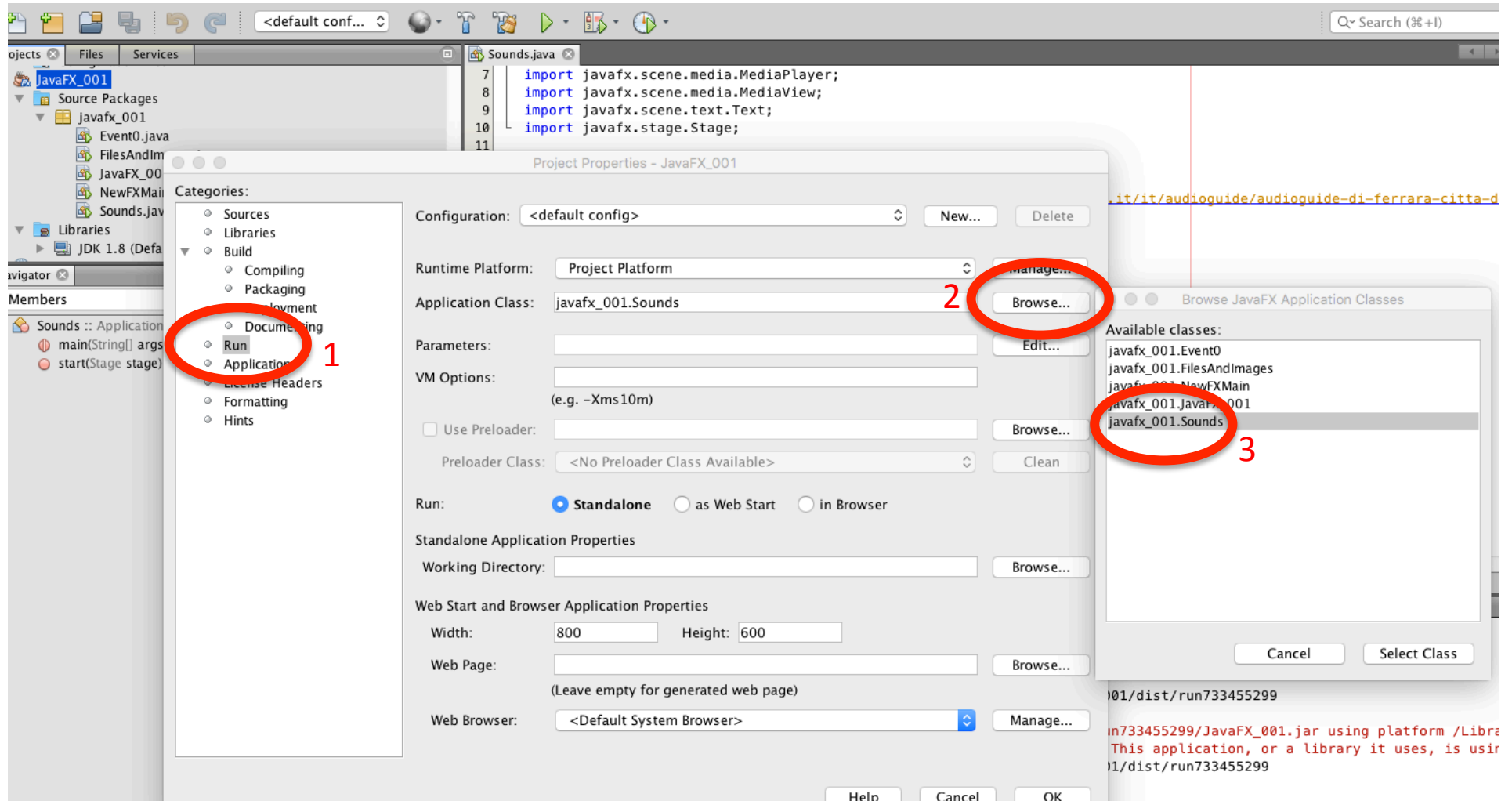
Go to Project Properties

(right click on the project)



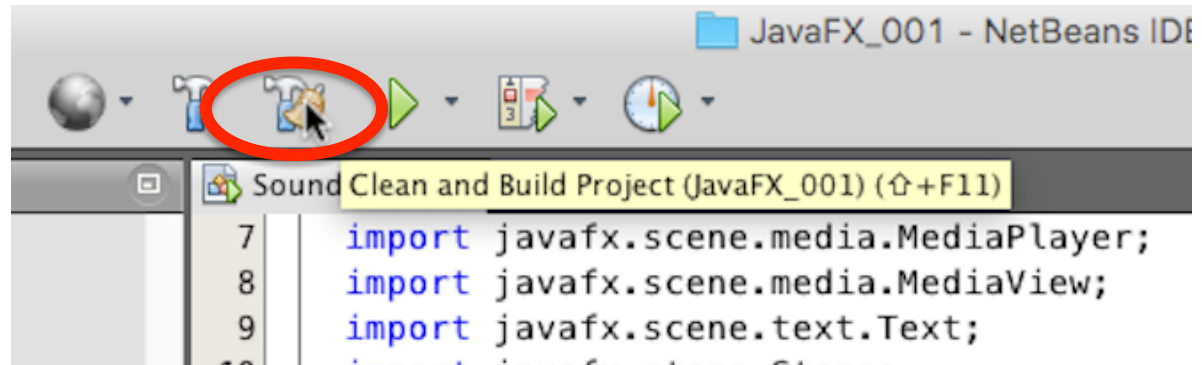
Crea un programma distribuibile – 2

Scegli “run” e seleziona la classe con il main che ti interessa



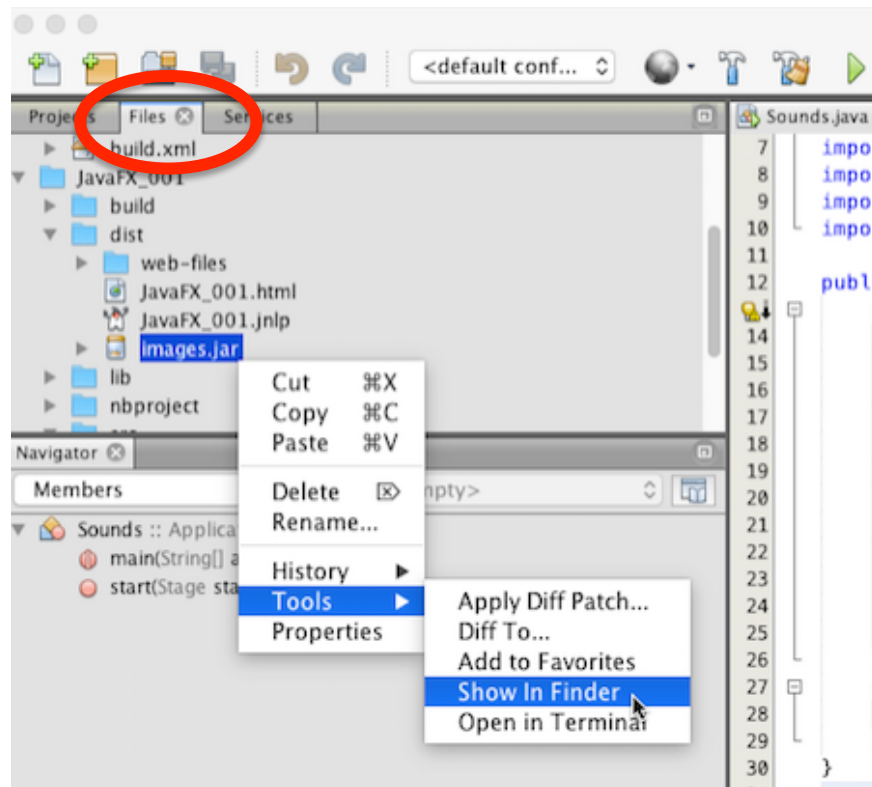
Crea un programma distribuibile - 3

Clean and build project



Go to the Files Tab,
Select the .jar file,
Right click on it,
Choose Tools -> Show in Finder.

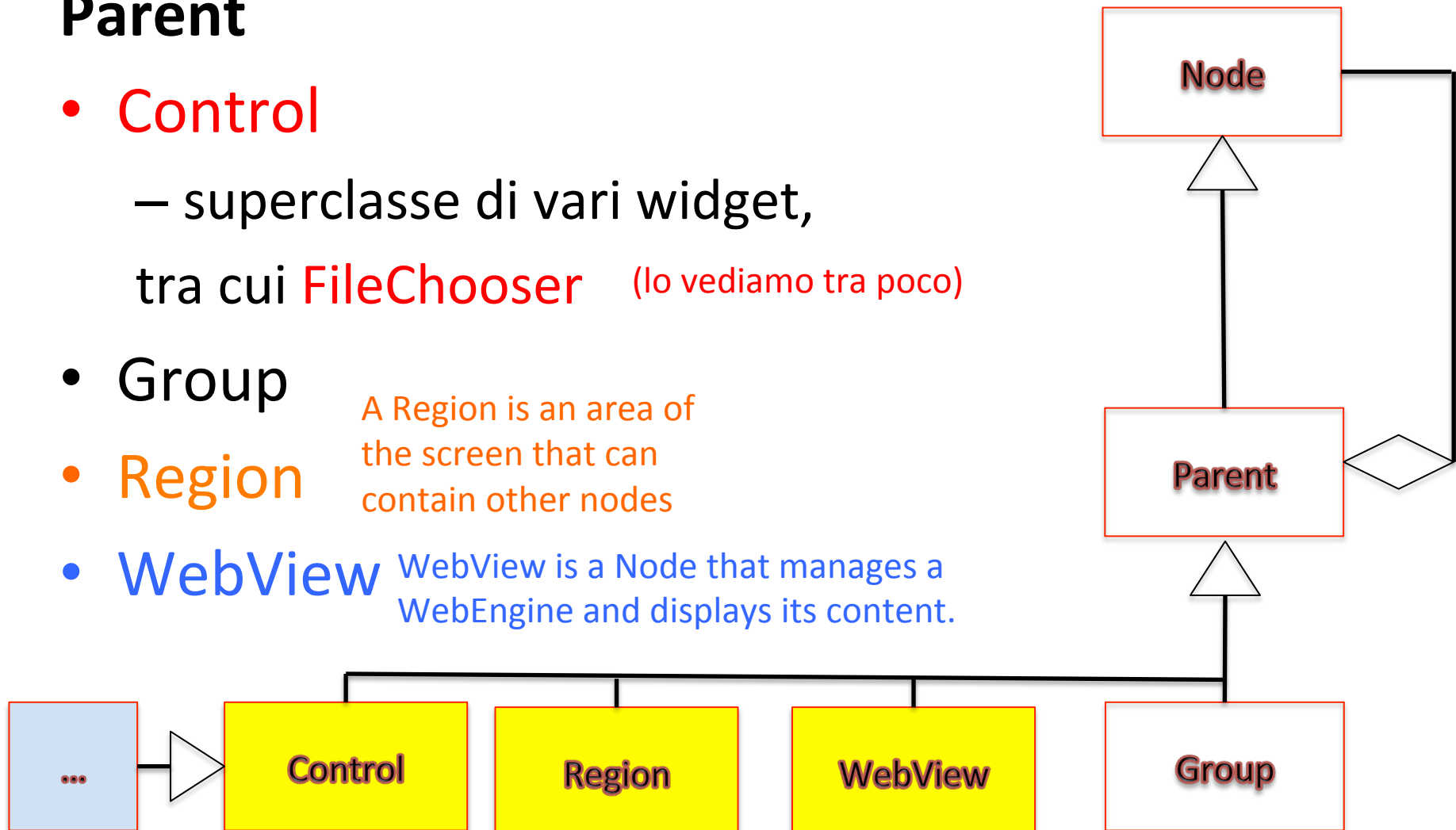
Double clicking on the file
You will start the process.



Parent hierarchy

Parent

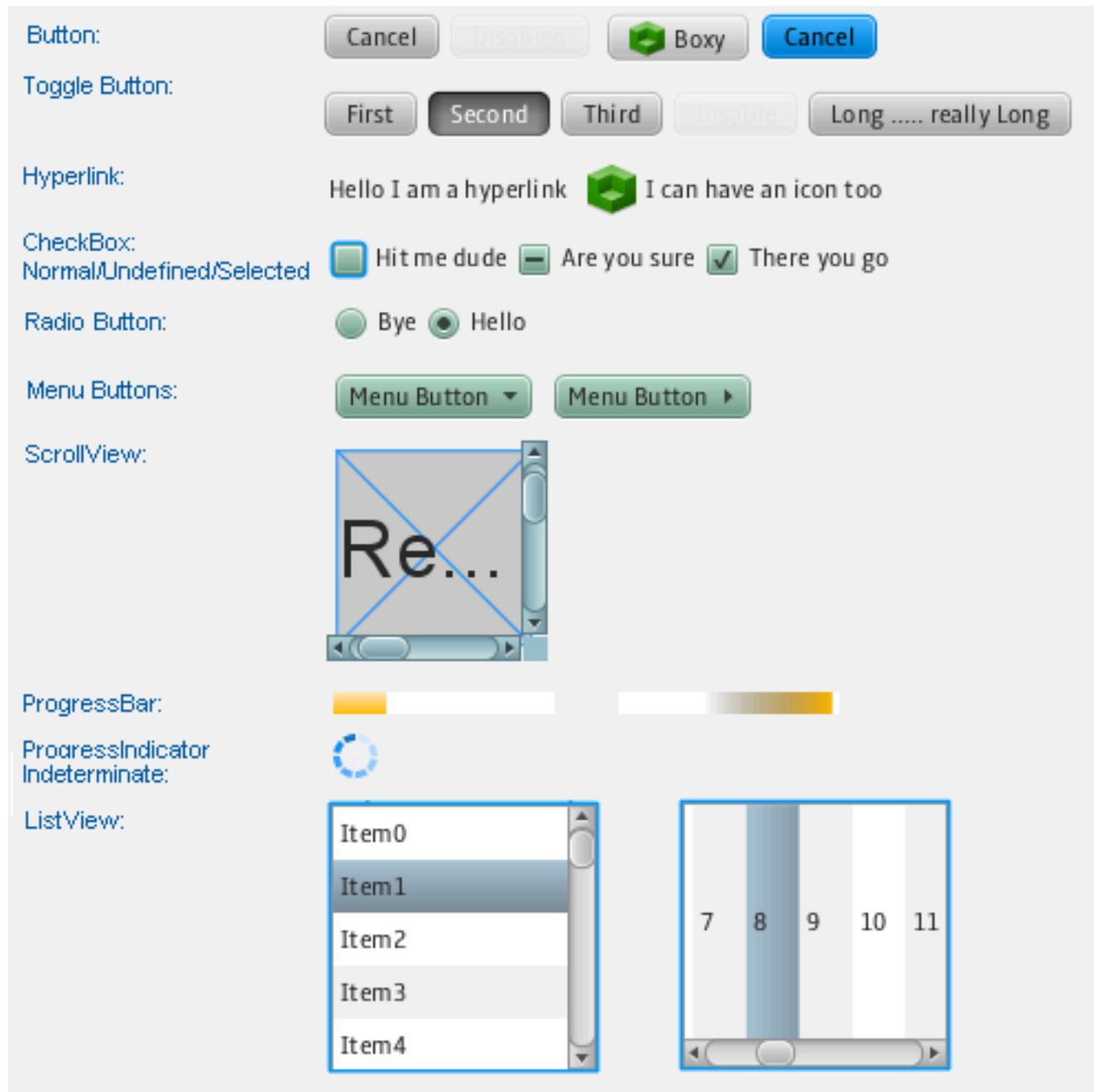
- **Control**
 - superclasse di vari widget,
tra cui **FileChooser** (lo vediamo tra poco)
- **Group**
- **Region** A Region is an area of the screen that can contain other nodes
- **WebView** WebView is a Node that manages a WebEngine and displays its content.



JavaFX UI Controls



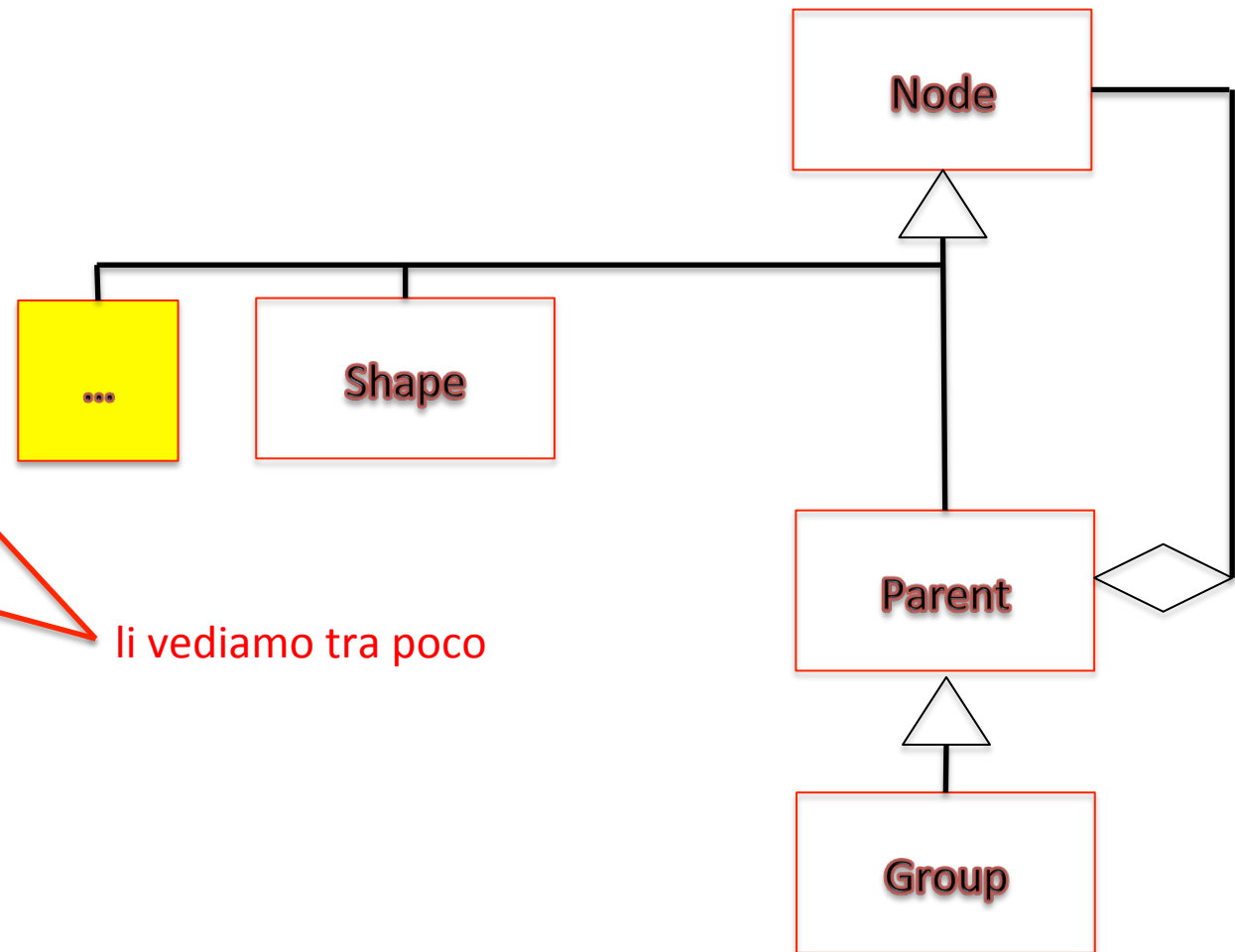
Controls



Node hierarchy

Node

- Parent
- Shape
- **ImageView**
- **MediaView**
- Canvas



li vediamo tra poco

Canvas

Una "tela del pittore" con un metodo per ottenere il suo **GraphicsContext** che ha varie primitive per disegnarci sopra:

- fillArc()
- fillRect()
- drawImage()
- ...

<http://docs.oracle.com/javafx/2/canvas/jfxpub-canvas.htm>