Costruttori

Definizione dei costruttori

- Se per una classe A non scrivo nessun costruttore, il sistema automaticamente crea il costruttore A();

- Se invece definisco almeno un costruttore non void, ad es. A(int s), il sistema non crea il costruttore A();

Definizione dei costruttori

- Se B è figlia di A, il costruttore di B come prima cosa invoca A(), a meno che la prima istruzione non sia una super.

Invocazione dei costruttori

```
public class A {
  public A() {
    System.out.println("Creo A");
  }
}
public class B extends A {
  public B() {
    System.out.println("Creo B");
  }
  public B(int k) {
    System.out.println("Creo B_int”);
  }
}
```

Output:
Creo A
Creo B_int

```
public static void main(String [] a) {
   B b=new B(1);
 }
```

Invocazione dei costruttori

```
public class A {
  public A(int k) {
    System.out.println("Creo A");
  }
}
public class B extends A {
  public B() {
    System.out.println("Creo B");
  }
  public B(int k) {
    System.out.println("Creo B_int”);
  }
}
```

Output:
ERRORE !

Perchè ?

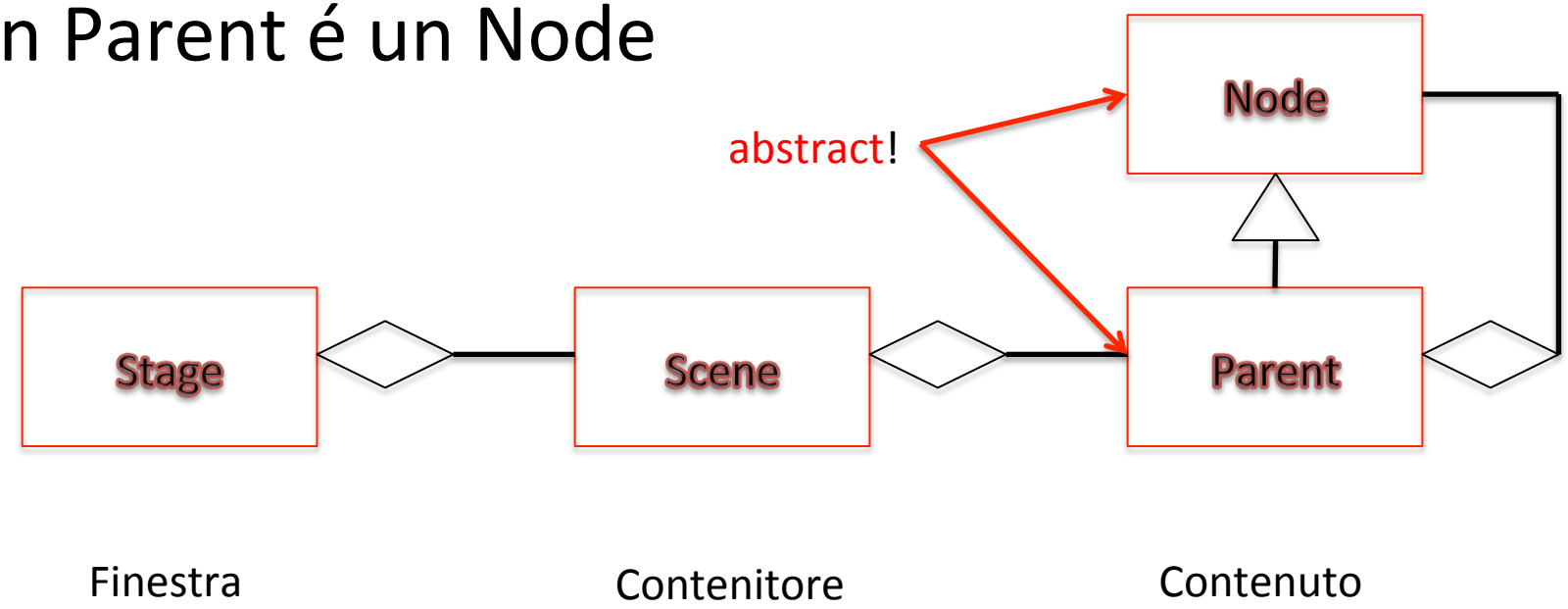
```
public static void main(String [] a) {
   B b=new B(1);
 }
```

Grafica e non solo: Java FX

Stage/Scene/Parent/Node

Finestra == Stage

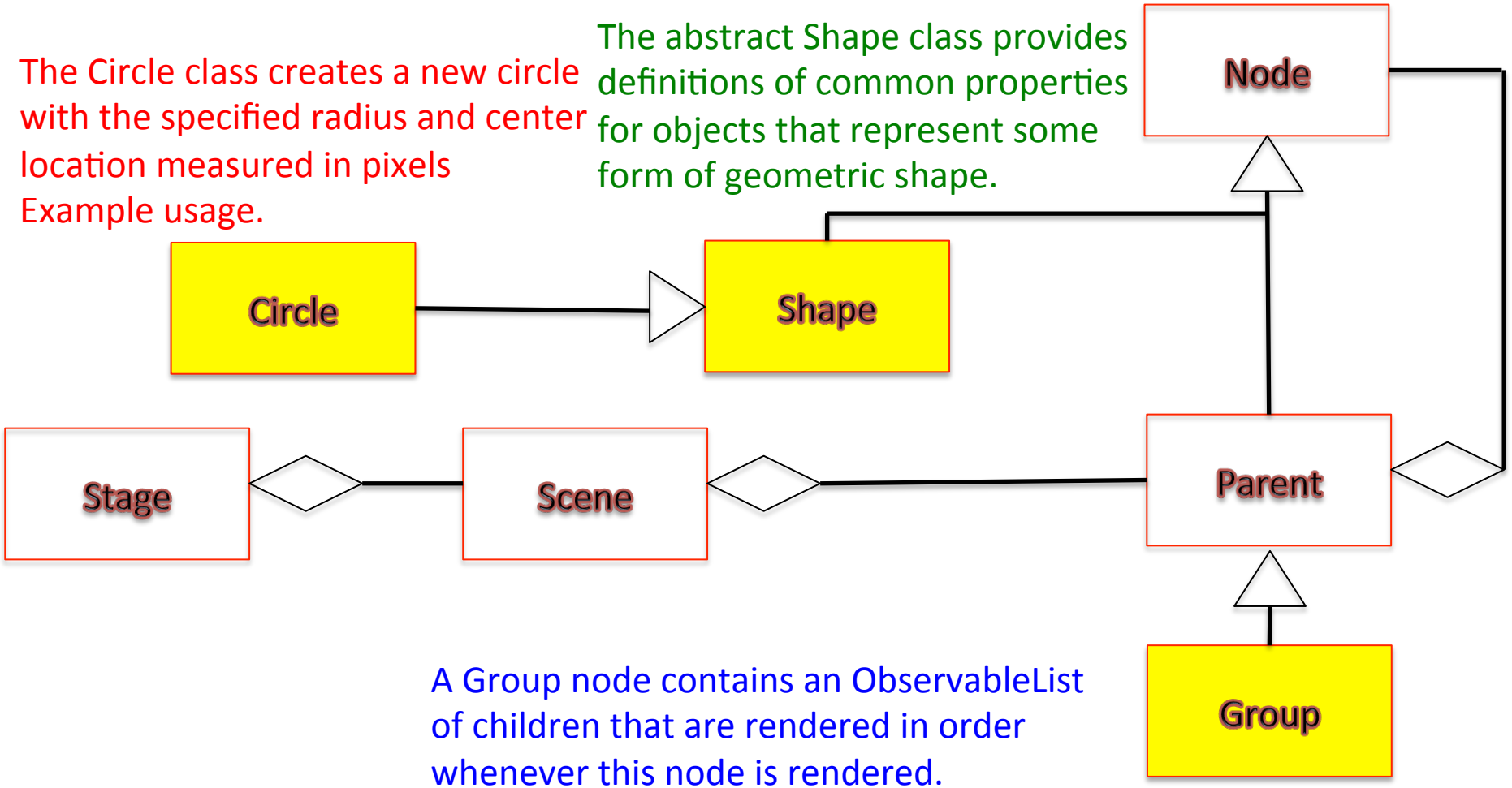
- Uno Stage contiene una Scene
- Una Scene ha un Parent
- Un Parent é un Node



Group – Shape - Circle

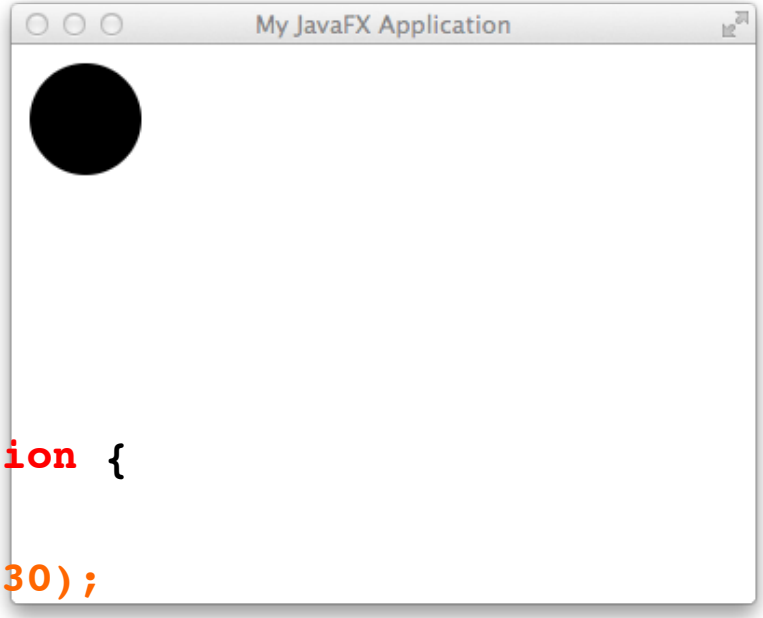
The Circle class creates a new circle with the specified radius and center location measured in pixels
Example usage.

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



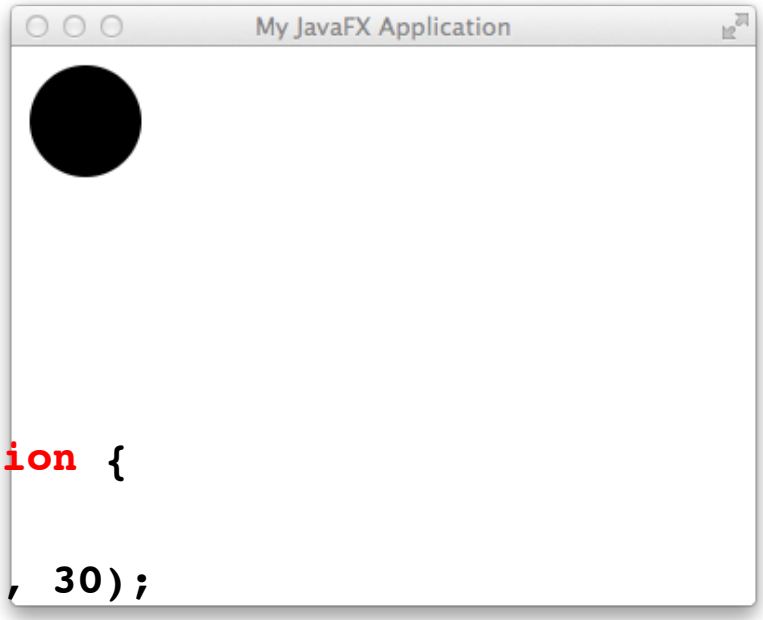
A Group node contains an ObservableList of children that are rendered in order whenever this node is rendered.

Applicazione minima



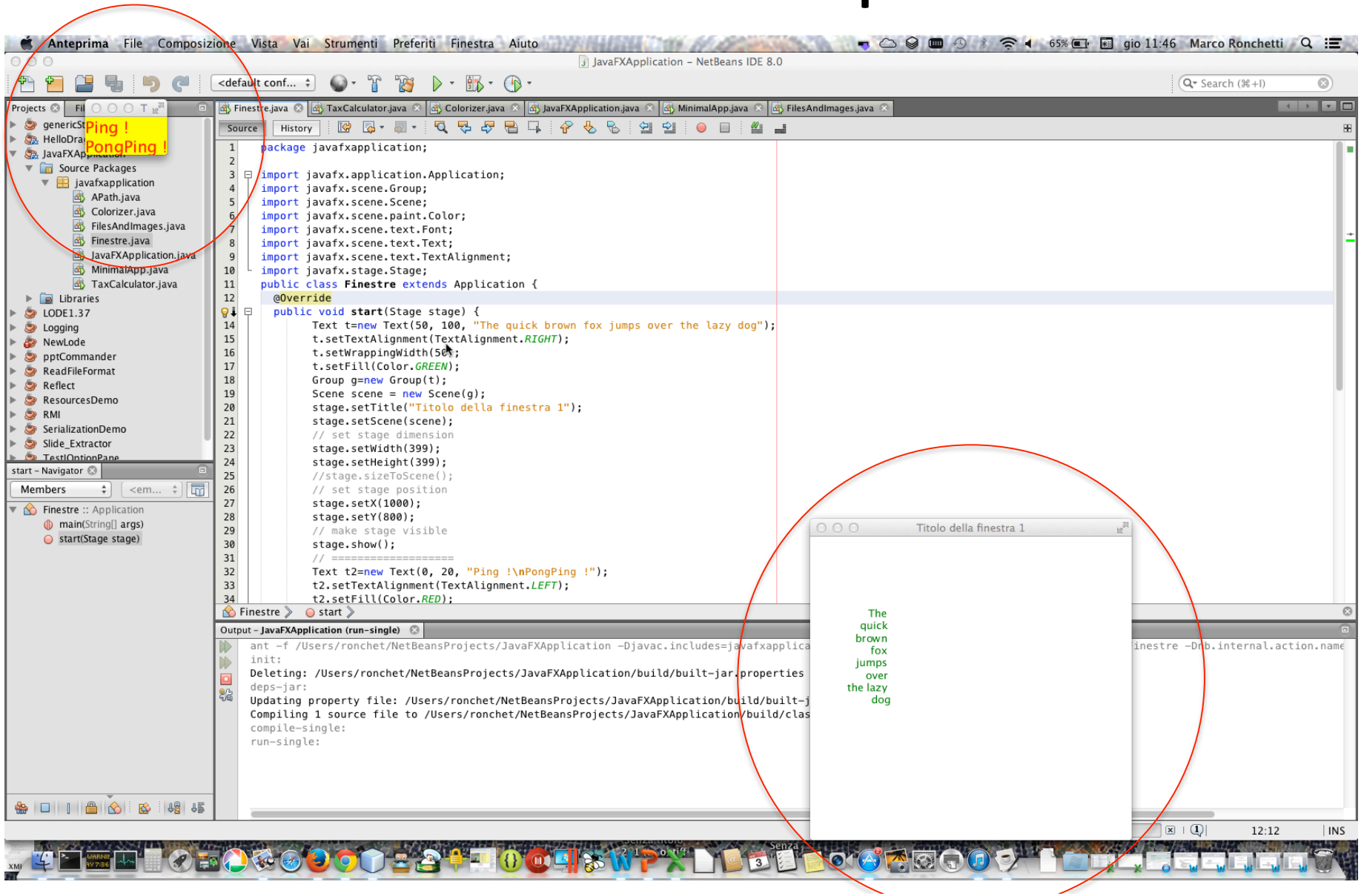

```
package it.unitn.disi.javafxapplication;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
public class MinimalApp extends Application {
     public void start(Stage stage) {
         Node circ = 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



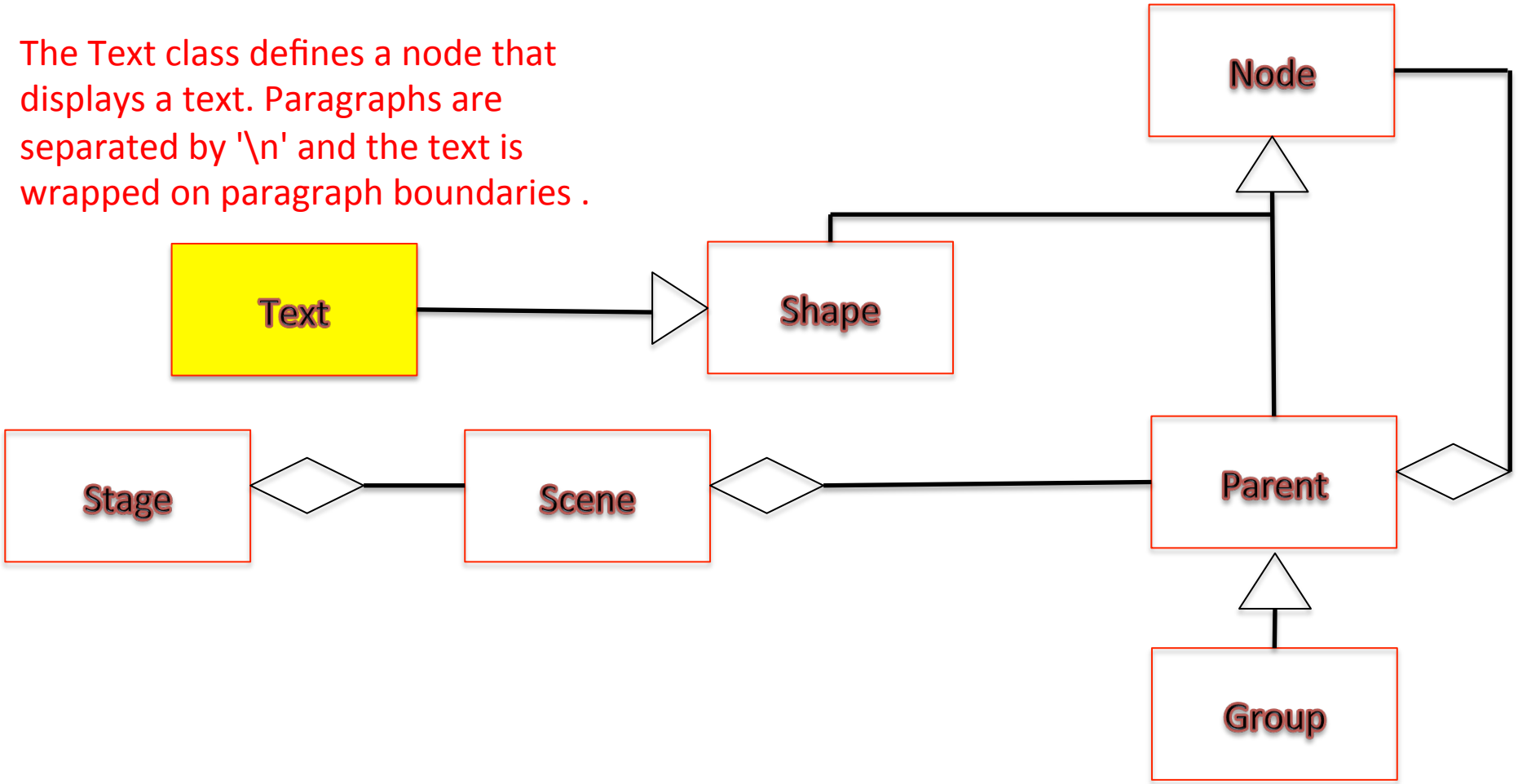

```
package it.unitn.disi.javafxapplication;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
public class MinimalApp extends Application {
     public void start(Stage stage) {
         Circle circ = 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(Color.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(Color.RED);
        t2.setFont(new Font(20));
         Group g2=new Group(t2);
         Scene scene2 = new Scene(g2);
         scene2.setFill(Color.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);
} }
```

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…)