

A decorative graphic on the left side of the slide, consisting of a 5x5 grid of squares in various shades of gray and blue, creating a subtle gradient effect.

JavaFX

Version 1.1

JavaFX

- GUI-Technologien für Java

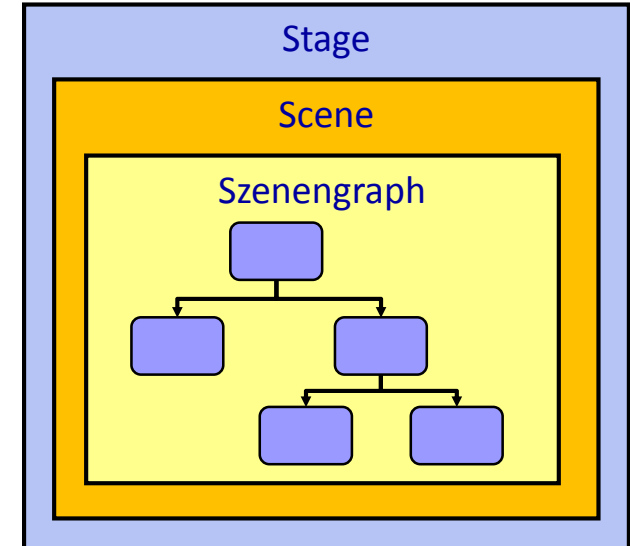
- AWT: Wenige, native Steuerelemente; Bestandteil des JRE.
- Swing: Umfassende GUI-Bibliothek; Keine nativen Steuerelemente; Bestandteil des JRE.
- SWT: Umfassende GUI-Bibliothek mit nativen Steuerelementen; Basis des Eclipse-Frameworks.
- JavaFX

- JavaFX

- Framework für *Rich Internet Applications (RIAs)*
 - Desktop-Anwendungen
 - Web-Anwendung mit Java-Browser-Plugin
- Programmierung in Java seit JavaFX 2.0 (2011)
- Benutzeroberfläche kann auch in XML-Sprache *FXML* beschrieben werden.

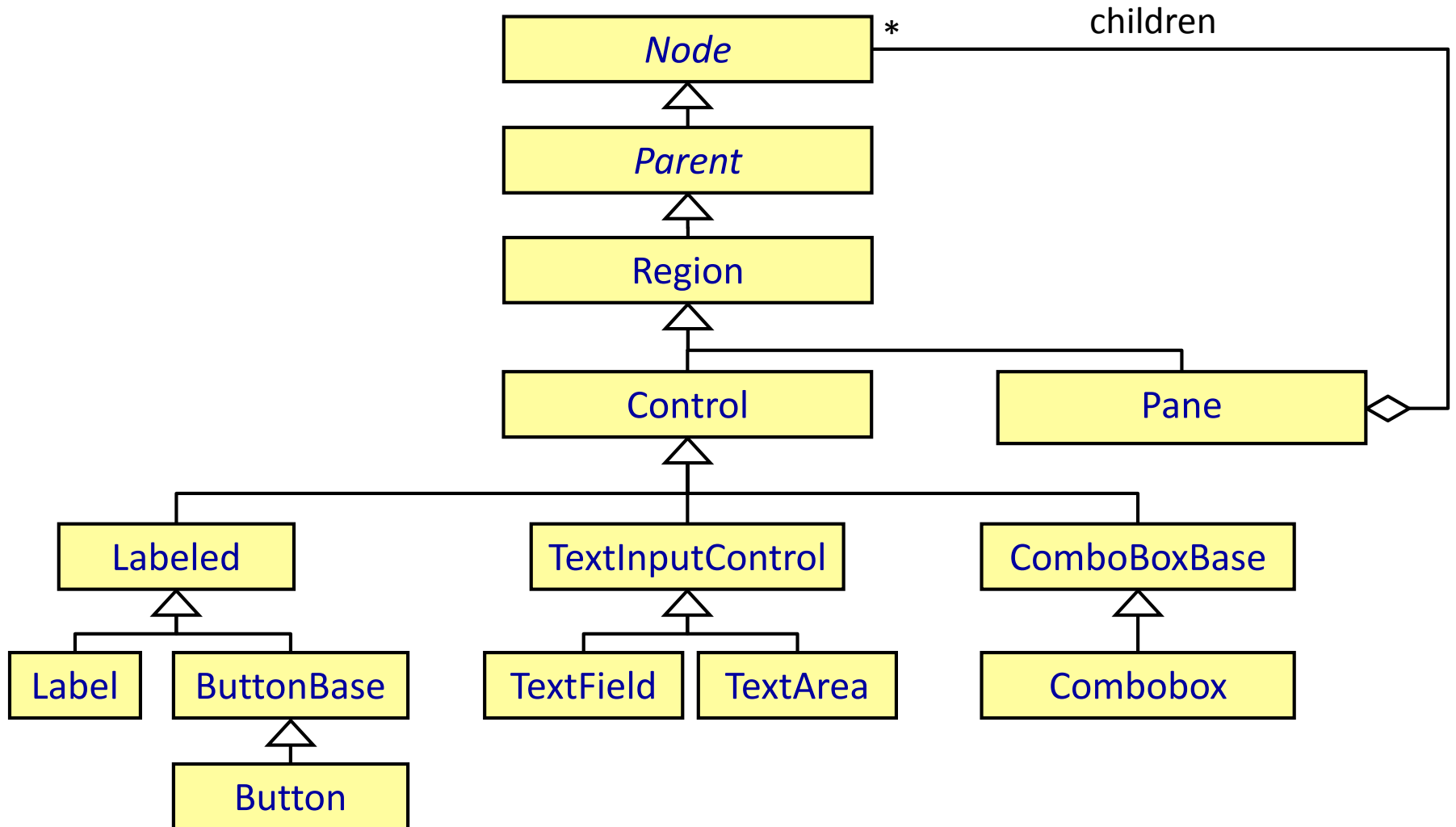
Struktur einer JavaFX-Anwendung

```
public class MyApp extends Application {  
    @Override  
    public void start(Stage stage) {  
        // build scene graph  
        Scene scene = new Scene(sceneGraphRoot,  
                                width, height);  
  
        stage.setScene(scene);  
        stage.show();  
    }  
  
    public static void main(String[] args) {  
        Application.launch(args);  
    }  
}
```

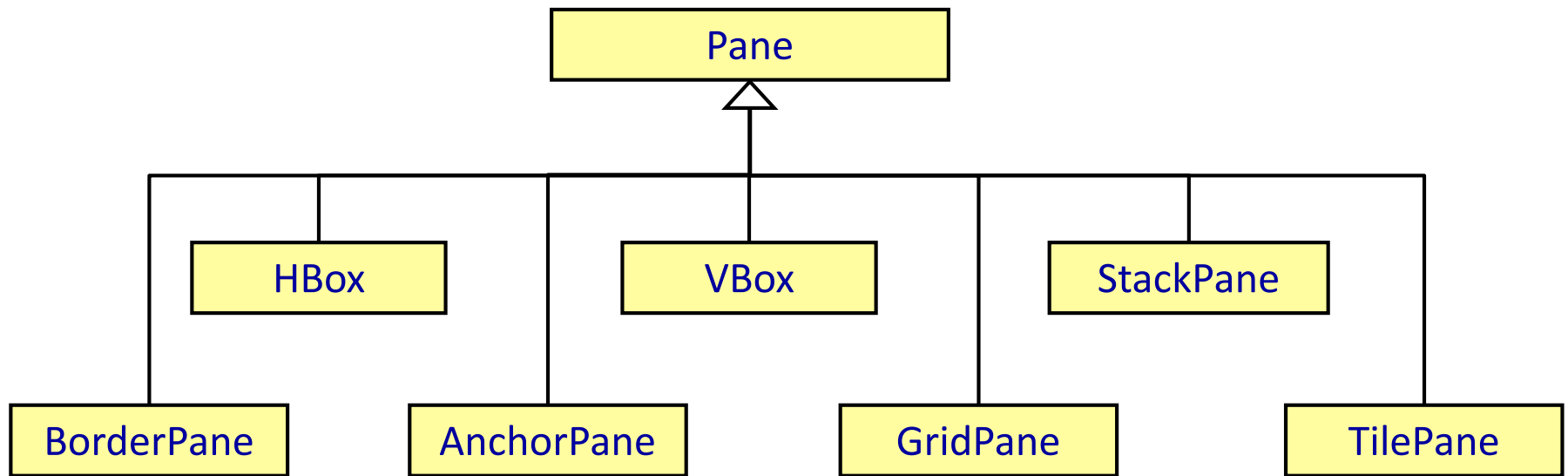


- *Stage* ist der Behälter auf oberster Ebene (das Fenster).
- *Scene* ist der Behälter für den Szenengraph.
- Die Knoten des *Szenengraphen* repräsentieren die Elemente der Benutzeroberfläche (Behälter und Steuerelemente).

Steuerelemente



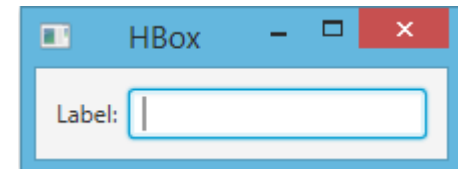
Layout-Container (Panels)



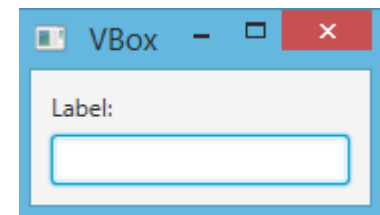
HBox/VBox

Kindelemente werden horizontal (HBox) bzw. vertikal (VBox) angeordnet.

```
HBox hbox = new HBox(5);  
hbox.setPadding(new Insets(10));  
hbox.setAlignment(Pos.BASELINE_CENTER);  
  
Label label = new Label("Label:");  
TextField text = new TextField();  
HBox.setHgrow(text, Priority.ALWAYS);  
  
hbox.getChildren().addAll(label, text);
```



```
VBox vbox = new VBox(5);  
vbox.setPadding(new Insets(10));  
vbox.setAlignment(Pos.BOTTOM_LEFT);  
  
Label label = new Label("Label:");  
TextField text = new TextField();  
  
vbox.getChildren().addAll(label, text);
```



GridPane

Kindelemente werden in einem Gitter mit unterschiedlich hohen Zeilen und Spalten angeordnet

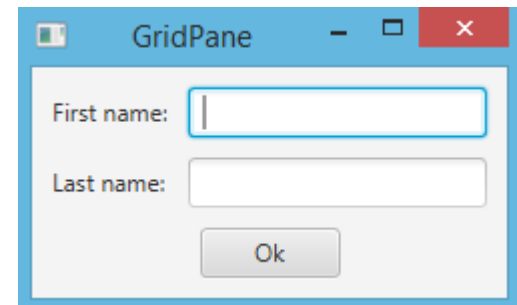
```
GridPane grid = new GridPane();
grid.setPadding(new Insets(10));
grid.setHgap(10);
grid.setVgap(10);

TextField text1 = new TextField();
GridPane.setHgrow(text1, Priority.ALWAYS);
TextField text2 = new TextField();
GridPane.setHgrow(text2, Priority.ALWAYS);

Button button = new Button("Ok");
button.setPrefWidth(70);
GridPane.setHalignment(button, HPos.CENTER);

grid.add(new Label("First name:"), 0, 0);
grid.add(text1, 1, 0);
grid.add(new Label("Last name:"), 0, 1);
grid.add(text2, 1, 1);

grid.add(button, 0, 2, 2, 1); // column span
```



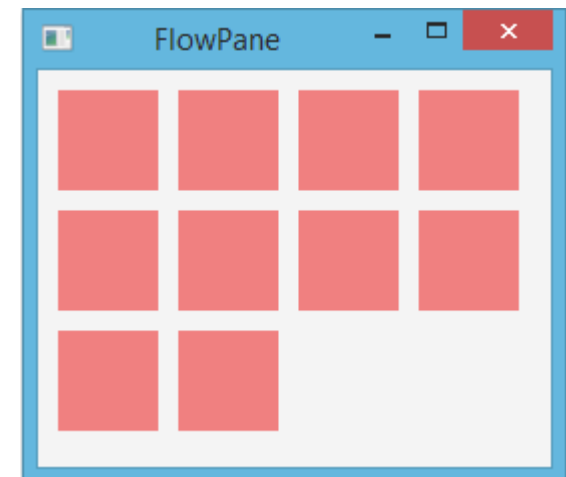
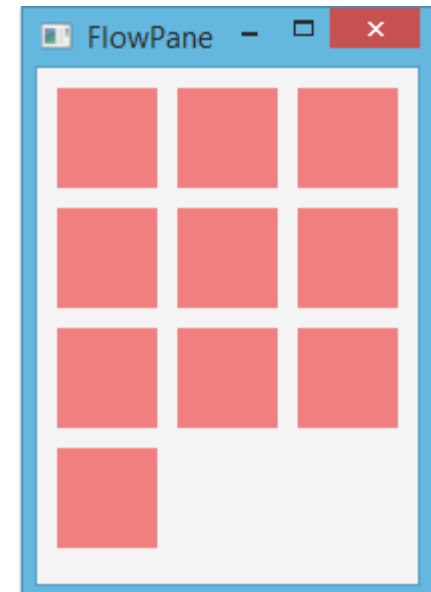
FlowPane/TilePane

Kindelemente werden zeilen-/spaltenweise angeordnet.

- *FlowPane*: Zellen können unterschiedlich groß sein.
- *TilePane*: Zellen sind gleich groß. Ausrichtung innerhalb der Zelle kann definiert werden (*setAlignment*).

```
FlowPane flowPane = new FlowPane();
flowPane.setPadding(new Insets(10));
flowPane.setHgap(10);
flowPane.setVgap(10);

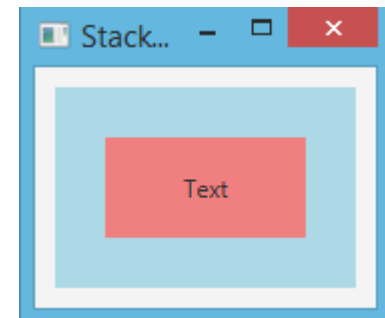
for (int i=0; i<10; i++)
    flowPane.getChildren().add(
        new Rectangle(50, 50, Color.LIGHTCORAL));
```



StackPane

Kindelemente werden übereinander angeordnet.

```
StackPane stackPane = new StackPane();  
stackPane.setPadding(new Insets(10));  
  
Rectangle rect1 = new Rectangle(150, 100,  
                                Color.LIGHTBLUE);  
Rectangle rect2 = new Rectangle(100, 50,  
                                Color.LIGHTCORAL);  
Label label = new Label("Text");  
stackPane.getChildren().addAll(rect1, rect2, label);
```

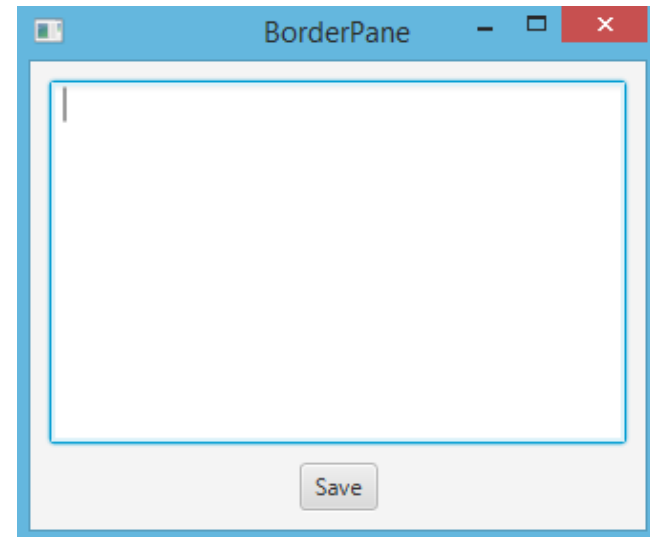


BorderPane

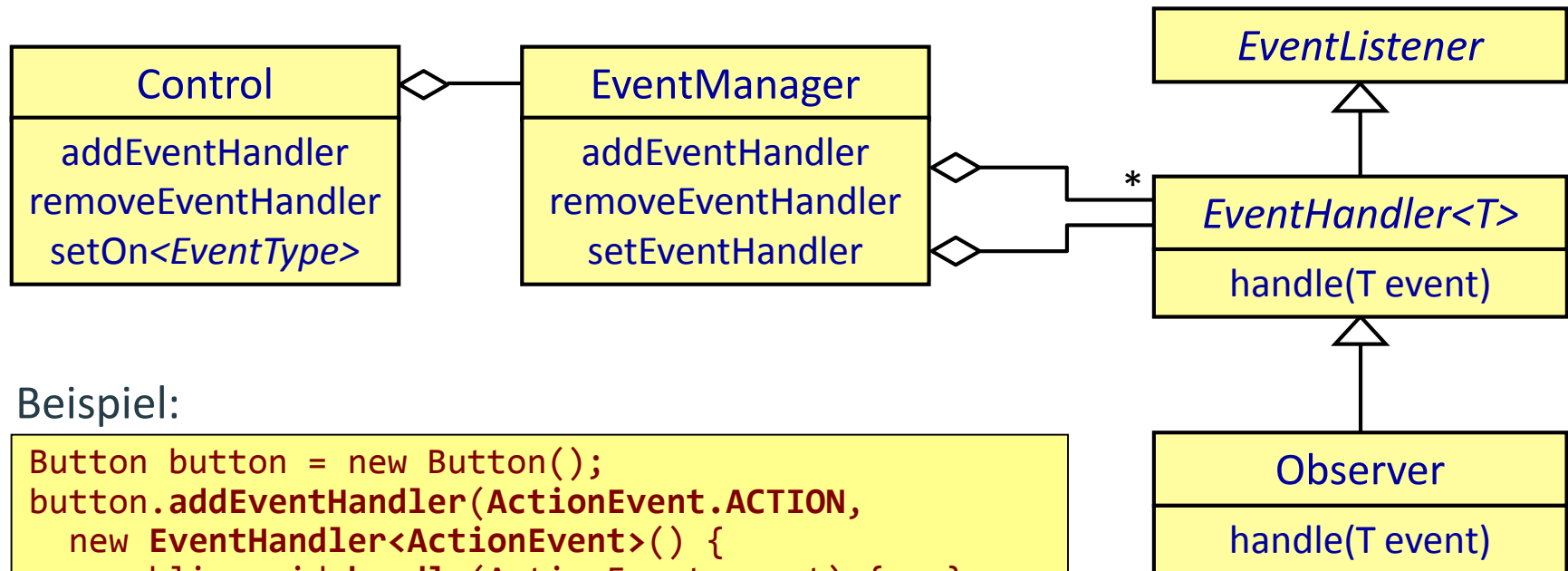
Kindelemente können oben, unten, links, rechts und im Zentrum des Behälters platziert werden.

- Elemente im oberen und unteren Bereich bekommen minimale Höhe und maximale Breite.
- Elemente im linken und rechten Bereich bekommen maximale Höhe und minimale Breite.
- Element im Zentrum bekommt den restlichen Platz.

```
BorderPane borderPane = new BorderPane();  
borderPane.setPadding(new Insets(10));  
  
TextArea text = new TextArea();  
Button button = new Button("Save");  
BorderPane.setMargin(button,  
                    new Insets(10, 0, 0, 0));  
BorderPane.setAlignment(button, Pos.CENTER);  
  
borderPane.setCenter(text);  
borderPane.setBottom(button);
```



Ereignisbehandlung (vereinfacht)



Beispiel:

```
Button button = new Button();
button.addEventHandler(ActionEvent.ACTION,
    new EventHandler<ActionEvent>() {
        public void handle(ActionEvent event) { ... }
    });

button.setOnAction(
    new EventHandler<ActionEvent>() {
        public void handle(ActionEvent event) { ... }
    });
```