

# КОМΠΟΝΟΒΚΑ

[https://fxdocs.github.io/docs/html5/#\\_layout](https://fxdocs.github.io/docs/html5/#_layout)

# Структура приложения

- Каждое JavaFX приложение состоит из иерархии нескольких основных компонентов:
  - Stage (окна или подмости)
  - Scene (сцены)
  - Node (узлы)

# Stage

Stage -> Window->Object

Stage()

Stage(StageStyle s)

В JavaFX application создается первичное окно, которое передается в метод start()

По умолчанию окно создается с заголовком, содержащим значок, текст и кнопки



# Управление

<https://openjfx.io/javadoc/18/javafx.graphics/javafx/stage/Stage.html>

По умолчанию открывается документация для 21

# Stage

- Методы `getXXX()` `setXXX()` для управление размером, положением, прозрачностью
- Закрытие окна программным путем `close()`

# Стили

StageStyle.DECORATED стиль по умолчанию

StageStyle.UNDECORATED окно белого цвета без рамки и заголовка

StageStyle.TRANSPARENT прозрачное окно без рамки и заголовка

StageStyle.UTILITY аналогично StageStyle.DECORATED но без внутренней рамки

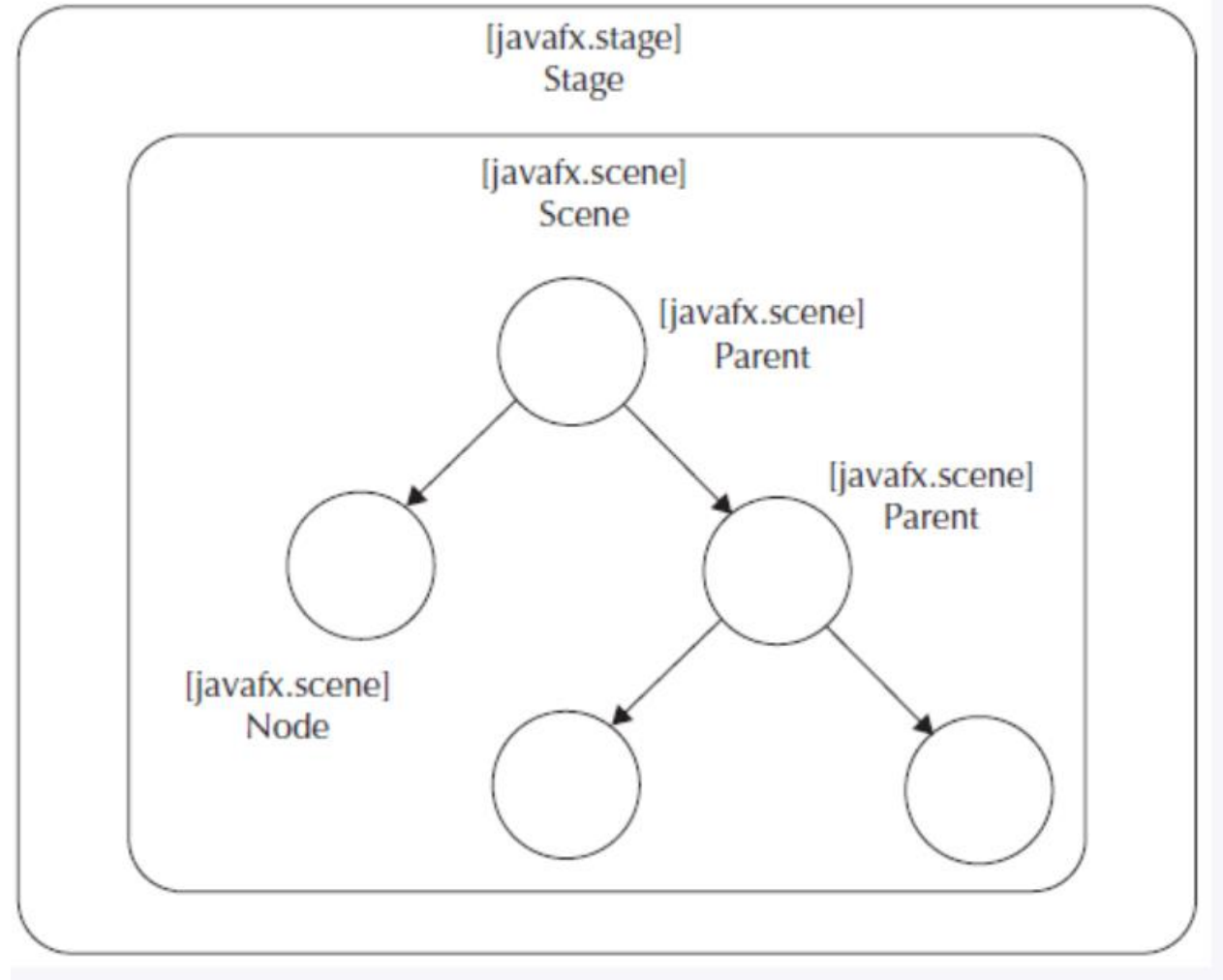
```
Stage w = new Stage(StageStyle.DECORATED);
```

```
Stage w = new Stage();  
w.initStyle(StageStyle.UTILITY);
```

```
public class AppStub extends Application{
    public static void main(String[] args) {
        launch(args);
    }
    @Override
    public void start(Stage stage) {
        stage.setTitle("JavaFX Application");
        stage.setWidth(500);
        stage.setHeight(300);
        try {
            stage.getIcons().add(new Image (getClass().getResourceAsStream(". . .")));
        } catch(Exception e){
            System.out.println("Image nkt found");
        }
        stage.show();
    }
}
```

# Scene

- `javafx.scene.Parent`
- `javafx.scene.Node`



- <https://openjfx.io/javadoc/18/javafx.graphics/javafx/scene/package-summary.html>

# Scene

Scene(Parent root)

Scene(Parent root, double width, double height)

Scene(Parent root, Paint fill)

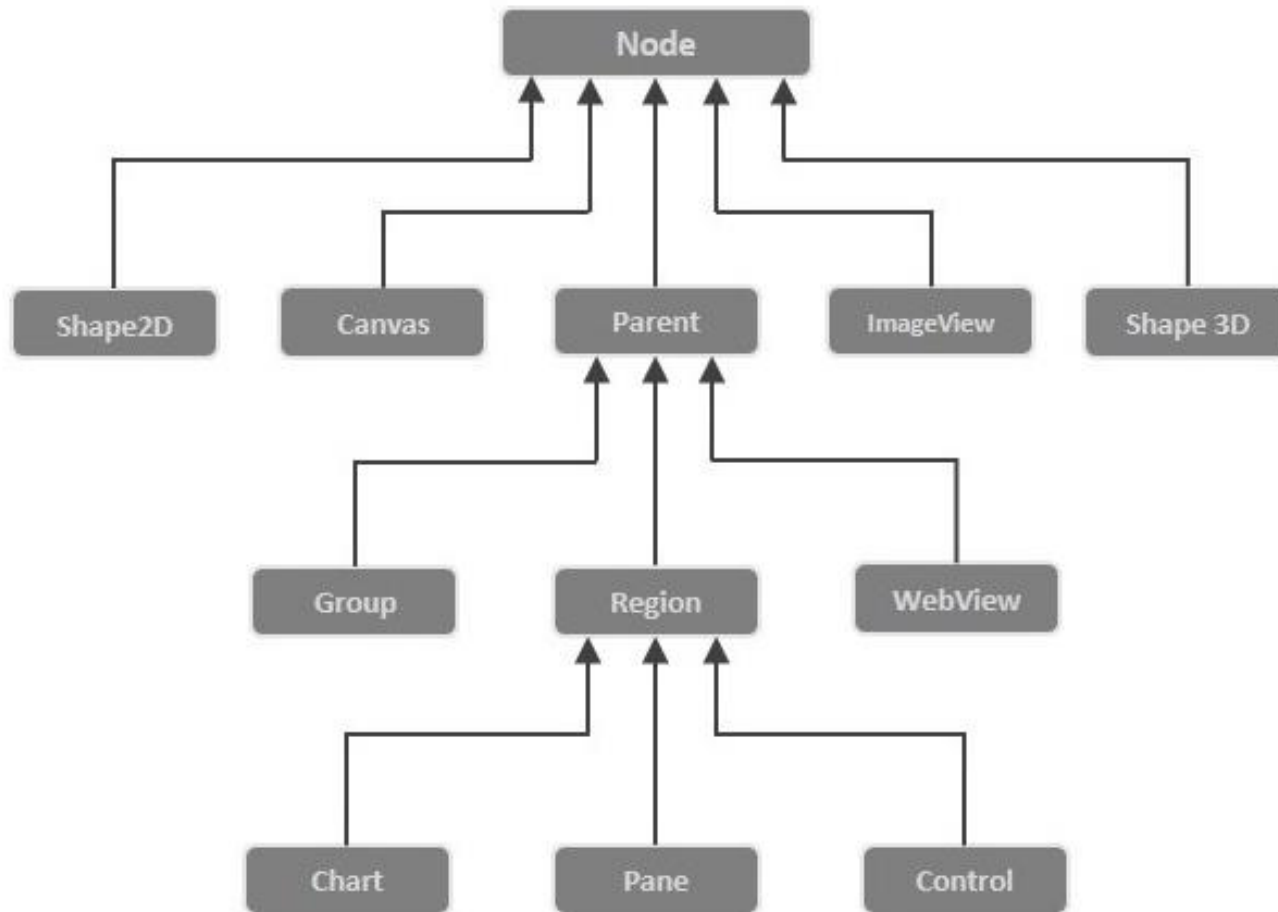
Scene(Parent root, double width, double height, Paint fill)

# Scene

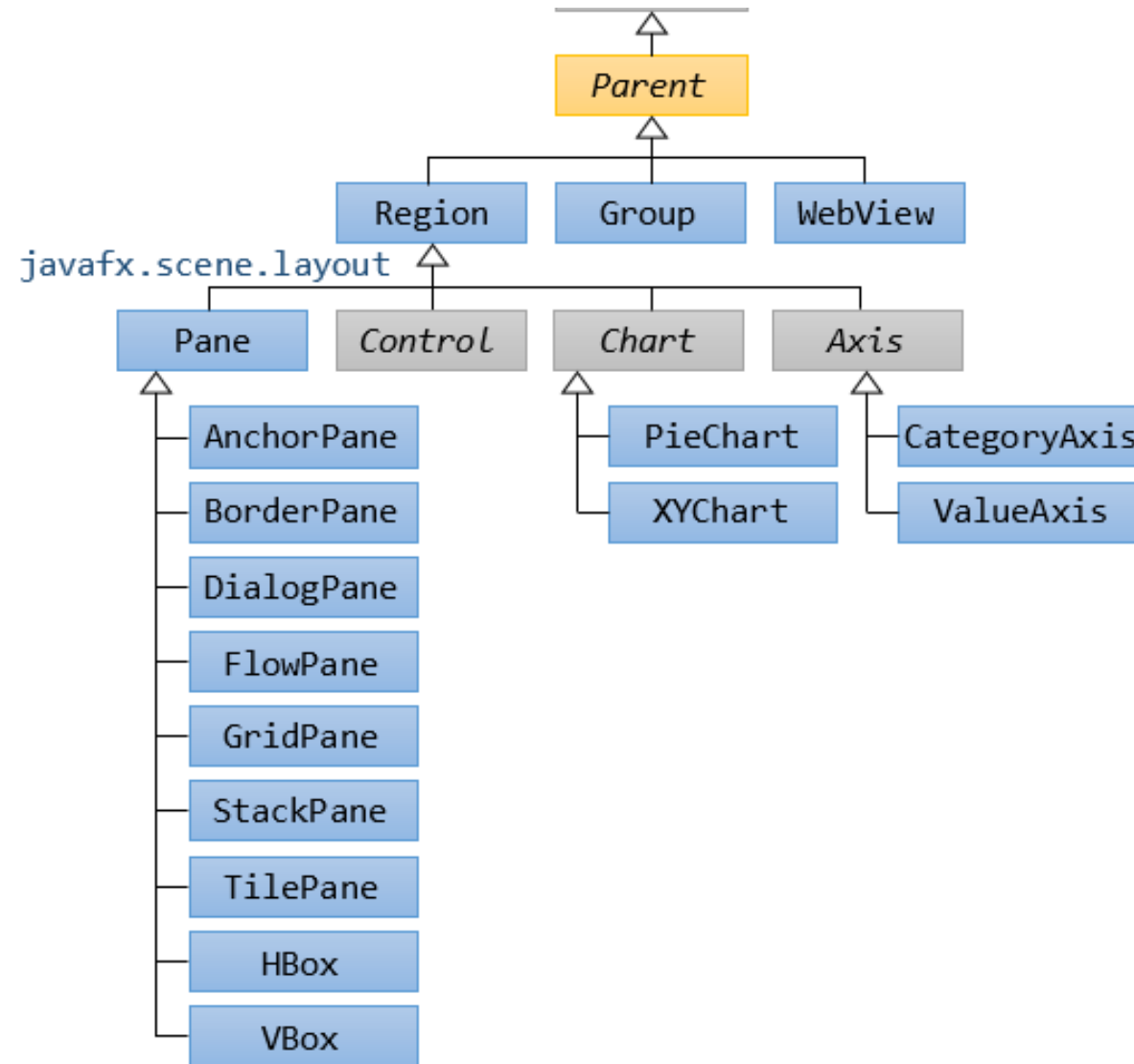
Цвет фона

`setFill()` `getFill()`

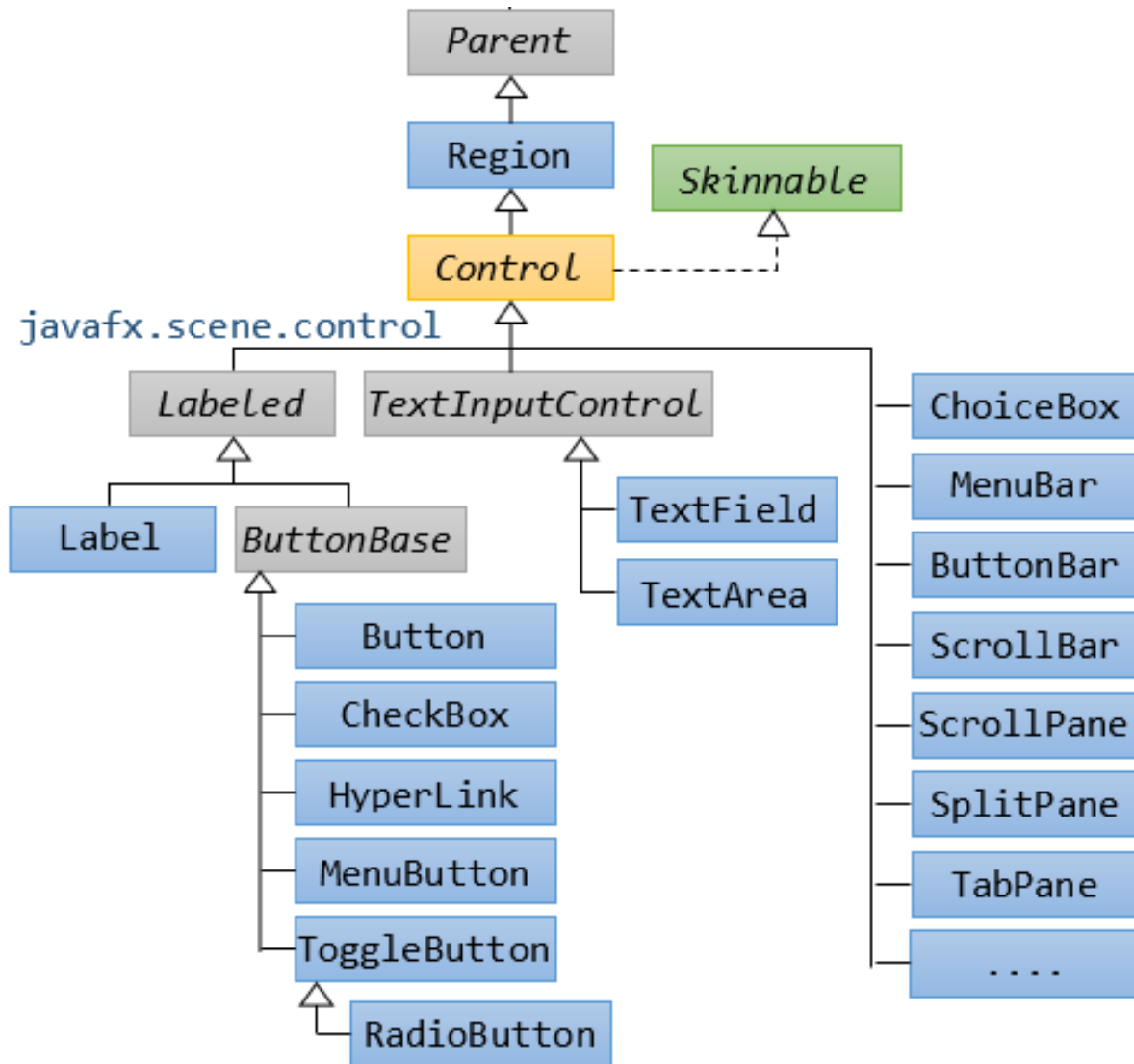
# Узлы сцены



# Parents



# Controls



# Node

- узел в графе сцены, наследникам предоставляются поля для настройки размеров
- `getLayoutX()`
- `getLayoutY()`
- `relocate(double x, double y)`
- `setLayoutX(double value)`
- `setLayoutY(double value)`
- координаты и размеры при компоновке, в зависимости от иерархии графа узлов и их настроек могут отличаться от ожидаемых
- Node не может содержать дочерних элементов

# Group и Pane

- Классы без компоновки
- Конструкторы – пустой, со списком/коллекцией узлов
- Добавление/удаление узлов через список узлов `getChildren()`
- Требуют указания местоположения
  - `relocate(double x, double y)`
  - `setLayoutX(double value)`
  - `setLayoutY(double value)`

# Group

```
Group root = new Group();
    Button button1 = new Button(«Кнопка 1");
    Button button2 = new Button(«Кнопка 2");
    button1.relocate(100.0, 50.0);
    button2.relocate(100.0, 100.0);
    root.getChildren().addAll(button1, button2);
Scene scene = new Scene(root, 400, 300, Color.KHAKI);
stage.setTitle("Класс Group");
stage.setScene(scene);
stage.show();
```

# Pane

```
Button button1 = new Button(«Кнопка 1»);  
Button button2 = new Button(«Кнопка 2»);  
    button1.relocate(100.0, 50.0);  
    button2.relocate(100.0, 100.0);  
Pane root = new Pane (button1, button2);  
Scene scene = new Scene(root, 400, 300, Color.KHAKI);  
    stage.setTitle("Класс Pane");  
    stage.setScene(scene);  
    stage.show();
```

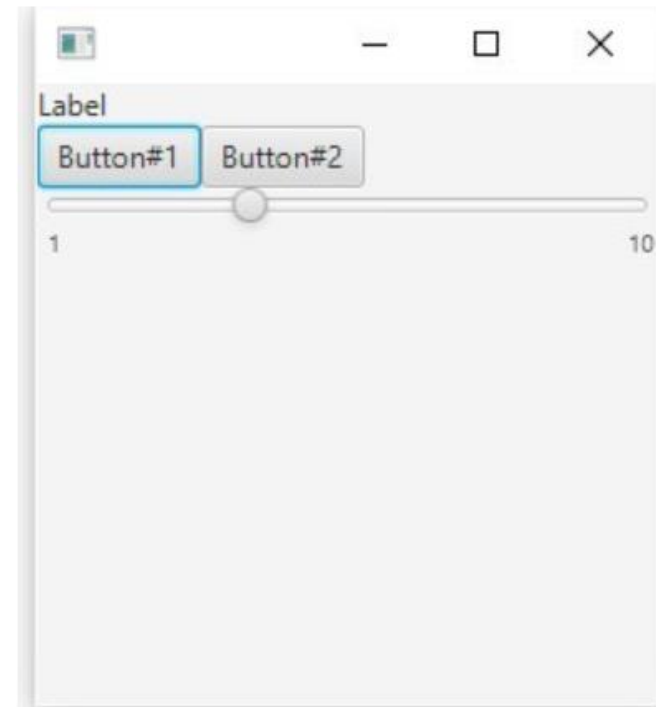
# Менеджеры HBox и VBox

- Выравнивание по горизонтали и вертикали

```
HBox root = new HBox(15.0); //spacing расстояние между узлами  
root.setAlignment(Pos.CENTER); //выравнивание  
Button button1 = new Button(«Кнопка 1»);  
Button button2 = new Button(«Кнопка 2»);  
root.getChildren().addAll(button1, button2);
```

# Менеджеры HBox и VBox

```
VBox root = new VBox();  
HBox hBox = new HBox();  
hBox.getChildren().addAll(new Button("Button#1"), new Button("Button#2"));  
Slider slider = new Slider(1.0, 10.0, 4.0);  
slider.setShowTickLabels(true);  
root.getChildren().add(new Label("Label"));  
root.getChildren().addAll(hBox, slider);
```



# FlowPane

- Размещение по порядку с горизонтальной или вертикальной ориентацией

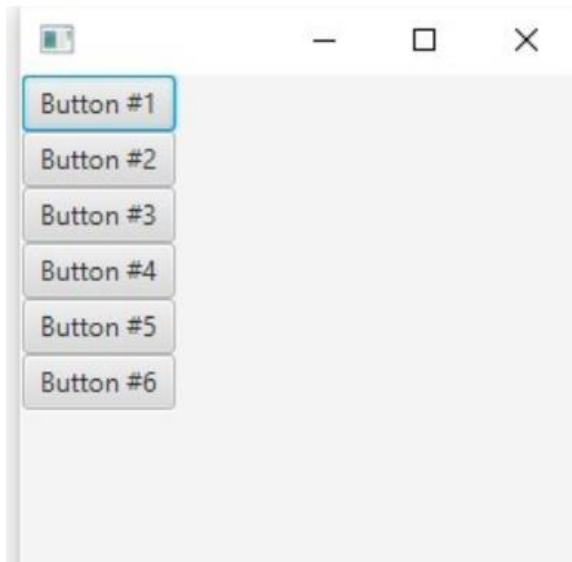
```
FlowPane root = new FlowPane();  
root.getChildren().add(new Button("Button #1"));  
root.getChildren().add(new Button("Button #2"));  
root.getChildren().add(new Button("Button #3"));  
root.getChildren().add(new Button("Button #4"));  
root.getChildren().add(new Button("Button #5"));  
root.getChildren().add(new Button("Button #6"));
```

# Orientation.HORIZONTAL



# Orientation.VERTICAL

```
root.setOrientation(Orientation.VERTICAL);
```



```
root.setVgap(8);  
root.setHgap(4);
```

# BorderPane

```
BorderPane root = new BorderPane();  
root.setLeft(new Button("Left"));  
root.setTop(new Button("Top"));  
root.setRight(new Button("Right"));  
root.setBottom(new Button("Bottom"));  
root.setCenter(new Button("Center"));
```

# StackPane

```
StackPane root = new StackPane();  
TextArea textArea = new TextArea("TextArea in StackPane");  
StackPane.setMargin(textArea, new Insets(10.0, 0.0, 30.0, 50.0));  
Button button = new Button("Button in StackPane");  
StackPane.setAlignment(button, Pos.CENTER_RIGHT);  
root.getChildren().addAll(textArea, button);
```

# TilePane

```
TilePane root = new TilePane(5.0,10.0);  
root.getChildren().add(new Button("Button #1"));  
root.getChildren().add(new Button("Button #2"));  
root.getChildren().add(new Button("«Large Button #3"));  
root.getChildren().add(new Button("Button #4"));  
root.getChildren().add(new Button("Button #5"));  
root.getChildren().add(new Button("Button #6"));
```

# GridPane

```
GridPane root = new GridPane();  
// Для отображения сетки  
root.setGridLinesVisible(true);  
root.add(new Label("0x0"), 0, 0);  
root.add(new Label("0x1"), 0, 1);  
root.add(new Label("1x1"), 1, 1);  
root.add(new Label("1x2"), 1, 2);  
root.add(new Label("5x5"), 5, 5);
```

# GridPane

// дополним

```
ColumnConstraints columnConstraints = new ColumnConstraints();  
columnConstraints.setPrefWidth(130.0); ColumnConstraints  
columnConstraints1 = new ColumnConstraints();  
columnConstraints1.setPercentWidth(20);  
root.getColumnConstraints().addAll(columnConstraints, columnConstraints1);  
//
```

# AnchorPane

```
AnchorPane root = new AnchorPane();  
Button button = new Button("Button in AnchorPane");  
root.getChildren().add(button);  
AnchorPane.setRightAnchor(button, 10.0);  
AnchorPane.setBottomAnchor(button, 10.0);
```

# Еще есть

- TabPane
- TitledPane
- SplitPane
- Accordion