

# Objektovo orientované programovanie

Učiteľ:  
**Ing. Jozef Wagner PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPG

## Teória 21

1. Ovládacie prvky
2. Komponenty pre rozmiestňovanie
3. Scene Graph
4. Z-Order

# Ovládacie prvky

Ovládacie prvky – **controls** – slúžia pre interakciu s používateľom

- **Label** - Textový popis bez interakcie.
- **Button** - Klikateľné tlačidlo.
- **CheckBox** - Prepínač (true / false).
- **RadioButton** - Prepínač v skupine (len jedna možnosť).
- **Slider** - Posuvník s hodnotou (horizontálny alebo vertikálny).
- **Separator** - Oddelovač (horizontálny alebo vertikálny).
- **TextField** - Jednoriadkové textové pole. (TextArea - Viacriadkové textové pole)
- **PasswordField** - Textové pole so skrytým obsahom.

Untitled

Label: Lorem ipsum dolor

Button:

CheckBox: ☒ Lorem ipsum dolor

RadioButton: ☒ Lorem ipsum dolor

Slider:

Separator:

TextField:

PasswordField:

```
Button button = new Button("Klikni ma");  
  
button.setOnAction(event -> {  
    System.out.println("Button bol kliknutý");  
});
```

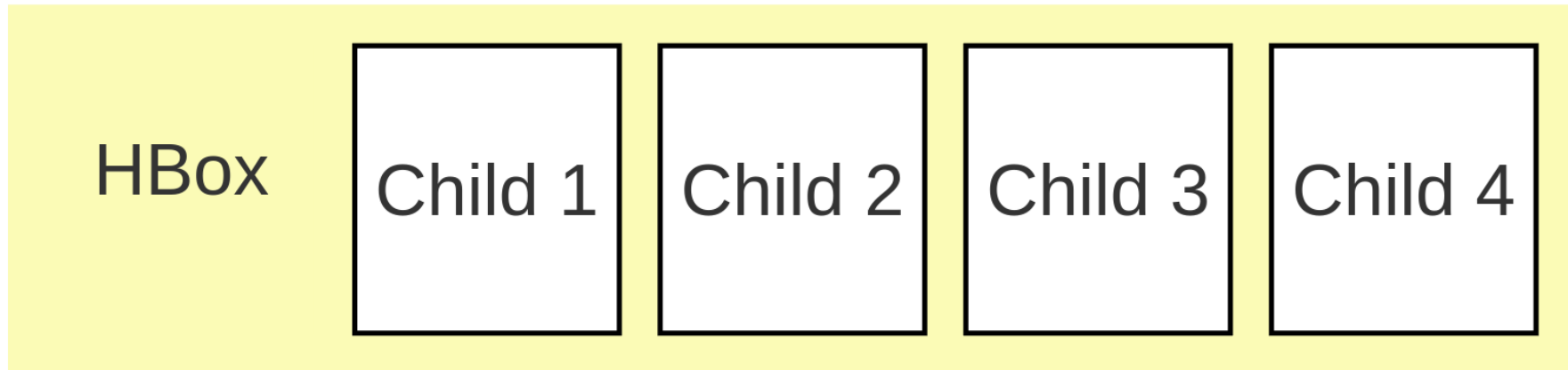
# Komponenty pre rozmiestňovanie

Tieto tzv. **layout komponenty**, alebo kontajnery – **containers** – nám definujú, ako budú jednotlivé prvky v okne rozložené.

- **HBox** - rozmiestni prvky horizontálne v jednom riadku.
- **VBox** - rozmiestni prvky vertikálne v jednom stĺpci pod sebou.
- **BorderPane** - vytvára hlavičku a pätu okna a taktiež okraje.
- **GridPane** - Tabuľkové rozloženie (riadky a stĺpce).
- **StackPane** - Prvky sú nad sebou (vrstvy).
- **FlowPane** - Tokové rozloženie – zalamuje prvky podľa šírky.
- **AnchorPane** - Ukotvenie prvkov k okrajom kontajnera.
- **TilePane** - Prvky majú rovnakú veľkosť v mriežke.
- **Pane** - Bez automatického rozloženia, prvky majú presnú polohu a veľkosť

# HBox

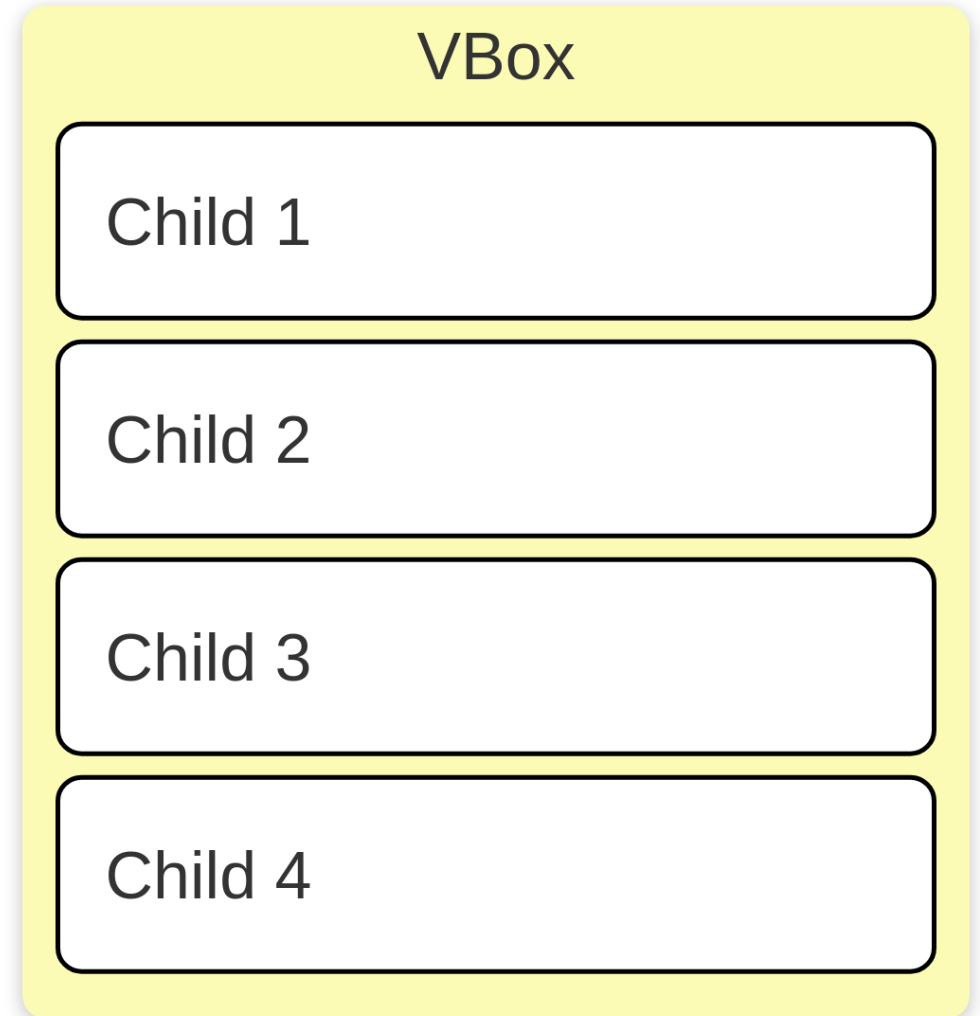
Tento kontajner rozmiestni prvky horizontálne v jednom riadku.



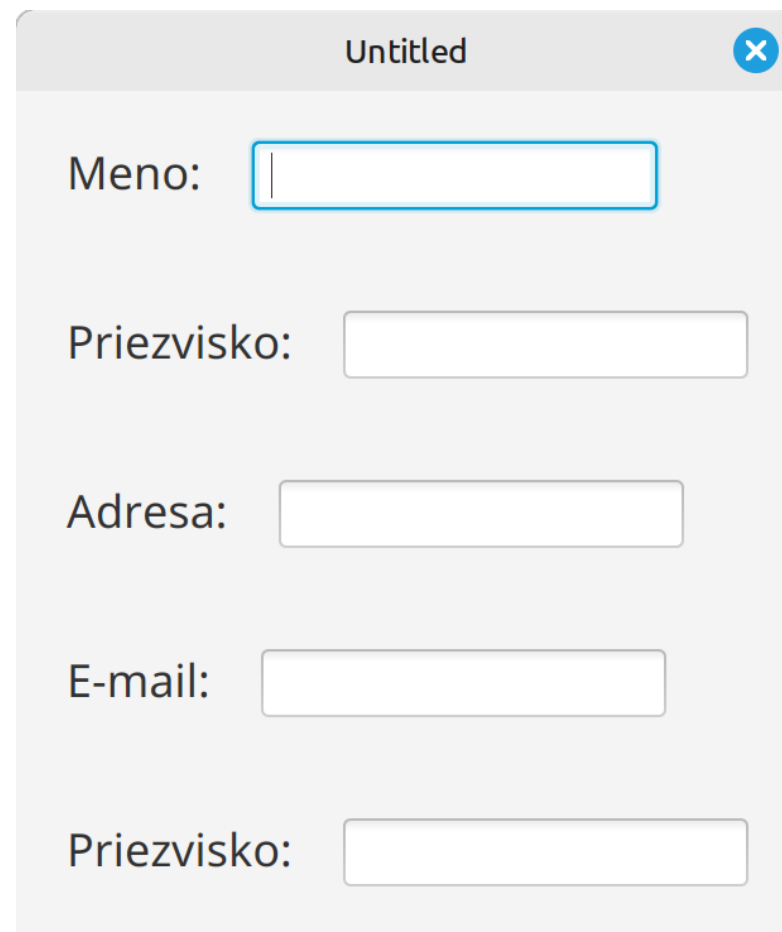
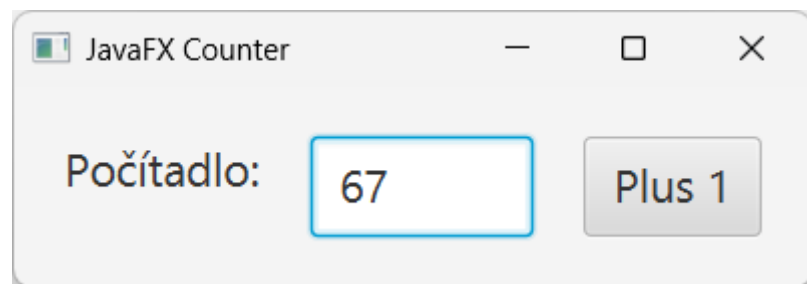
```
HBox hbox = new HBox(20);  
hbox.setAlignment(Pos.CENTER);  
hbox.getChildren().addAll(  
    new Label("Lorem ipsum"),  
    new Button("Close"));
```

# VBox

VBox je kontajner, ktorý  
rozmiestni prvky vertikálne pod  
sebou



# HBox a VBox





# BorderPane

Ide o zložitejší kontajner, ktorý vytvára priestor pre hlavičku a pätu okna a taktiež umožňuje vložiť prvky na ľavý a pravý okraj.

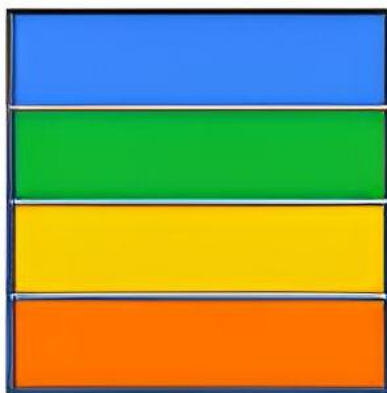


# BorderPane

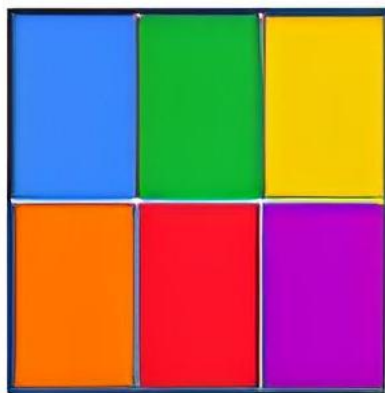
```
HBox menu = new HBox(  
    new Button("Súbor"),  
    new Button("Upraviť"),  
    new Button("Pomoc")  
);
```

```
BorderPane root = new BorderPane();  
root.setTop(menu); // horná časť  
root.setBottom(new Label("BOTTOM - pätička")); // spodná časť  
root.setLeft(new Button("LEFT")); // ľavá časť  
root.setRight(new Button("RIGHT")); // pravá časť  
root.setCenter(new Button("CENTER - hlavný obsah")); // stred
```

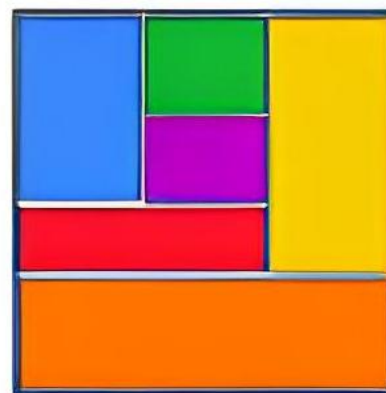
# Komponenty pre rozmiestňovanie



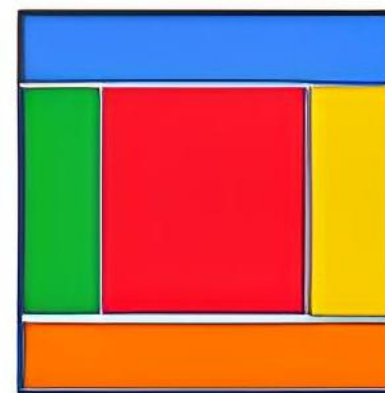
VBox



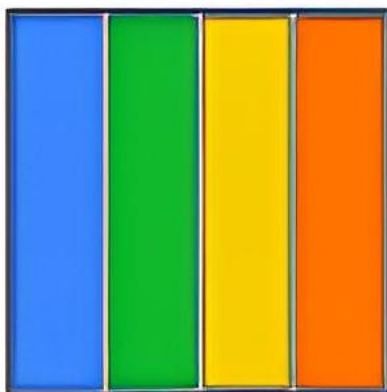
TilePane



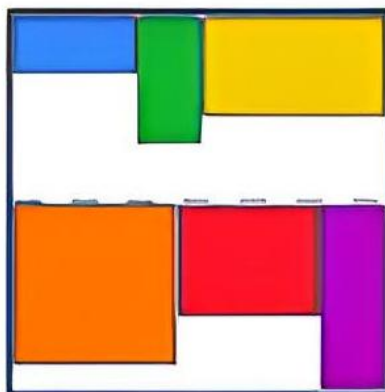
GridPane



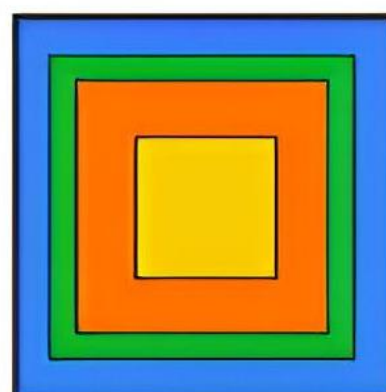
BorderPane



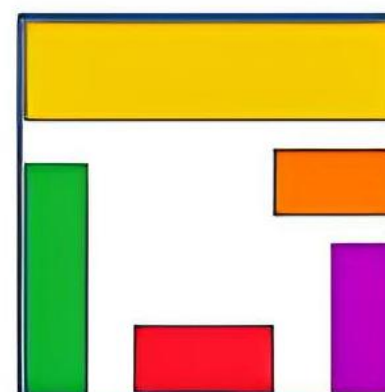
HBox



FlowPane



StackPane



AnchorPane

# Scene Graph

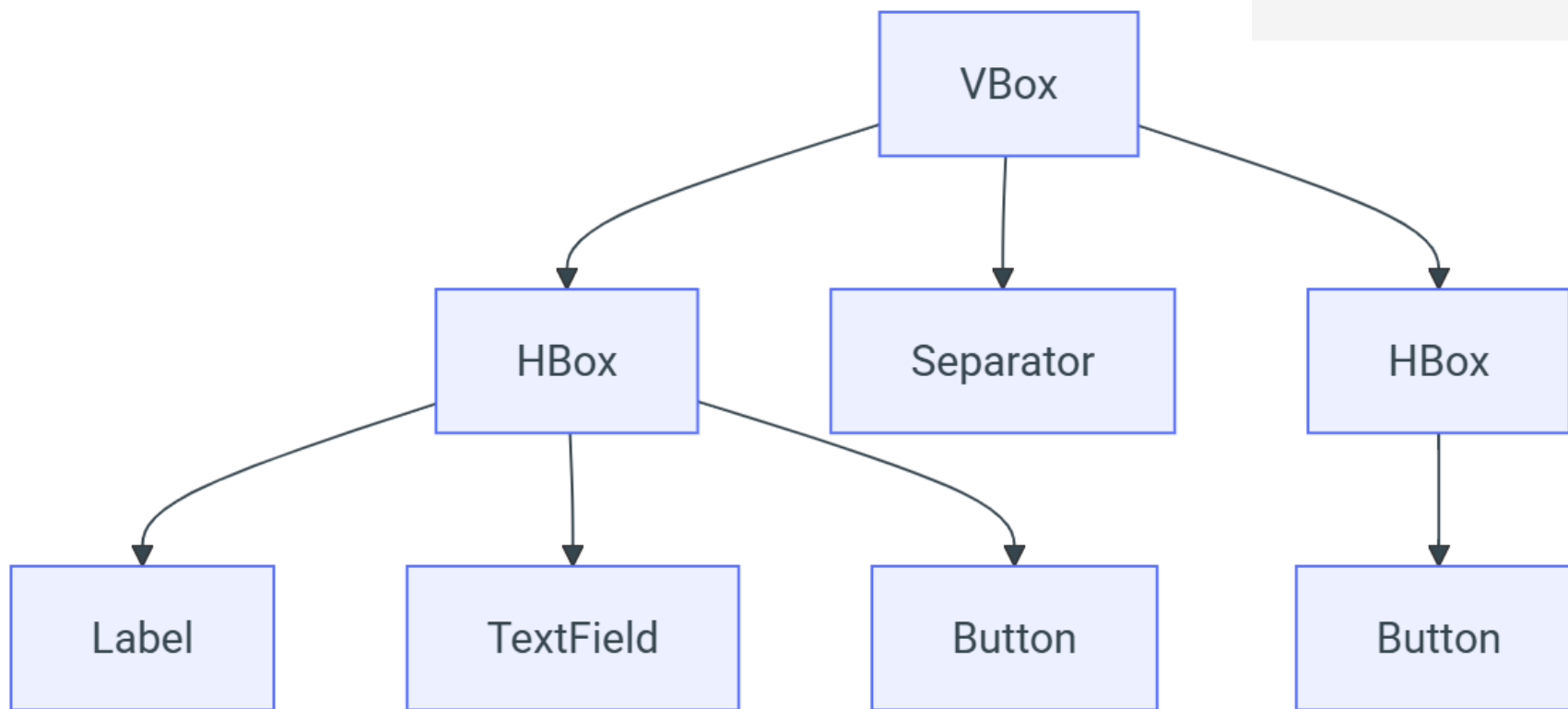
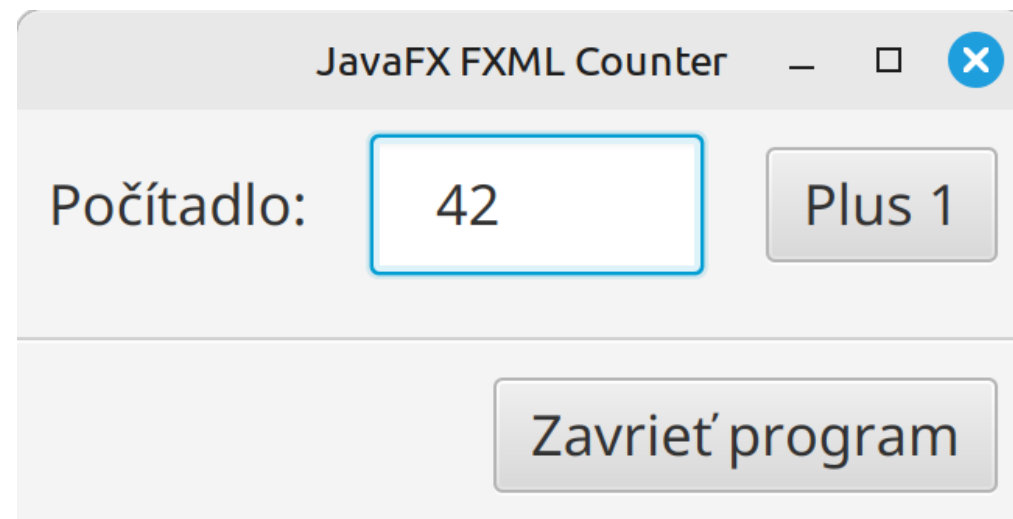
Komponenty sa vkladajú do scény (**Scene**), ktorá sa pripojí na okno aplikácie (**Stage**).

Z komponentov sa vytvorí stromová štruktúra s jedným koreňovým komponentom.

Táto stromová štruktúra sa volá **scene graph**.

- Do scény sa potom *priamo vloží iba koreňový komponent*.
- Ostatné komponenty sú priamo alebo nepriamo vnorené v tomto koreňovom uzle.

# Scene Graph



```
// Hlavný kontajner
VBox root = new VBox(10);

// Horný riadok s počítadlom
HBox counterRow = new HBox(20);
Label label = new Label("Počítadlo:");
TextField counterField = new TextField("0");
Button btnPlus = new Button("Plus 1");
counterRow.getChildren().addAll(label, counterField, btnPlus);

// Oddelovač
Separator separator = new Separator();
// Spodný riadok - tlačidlo Zavrieť
HBox bottomRow = new HBox();
Button btnClose = new Button("Zavrieť program");
bottomRow.getChildren().add(btnClose);

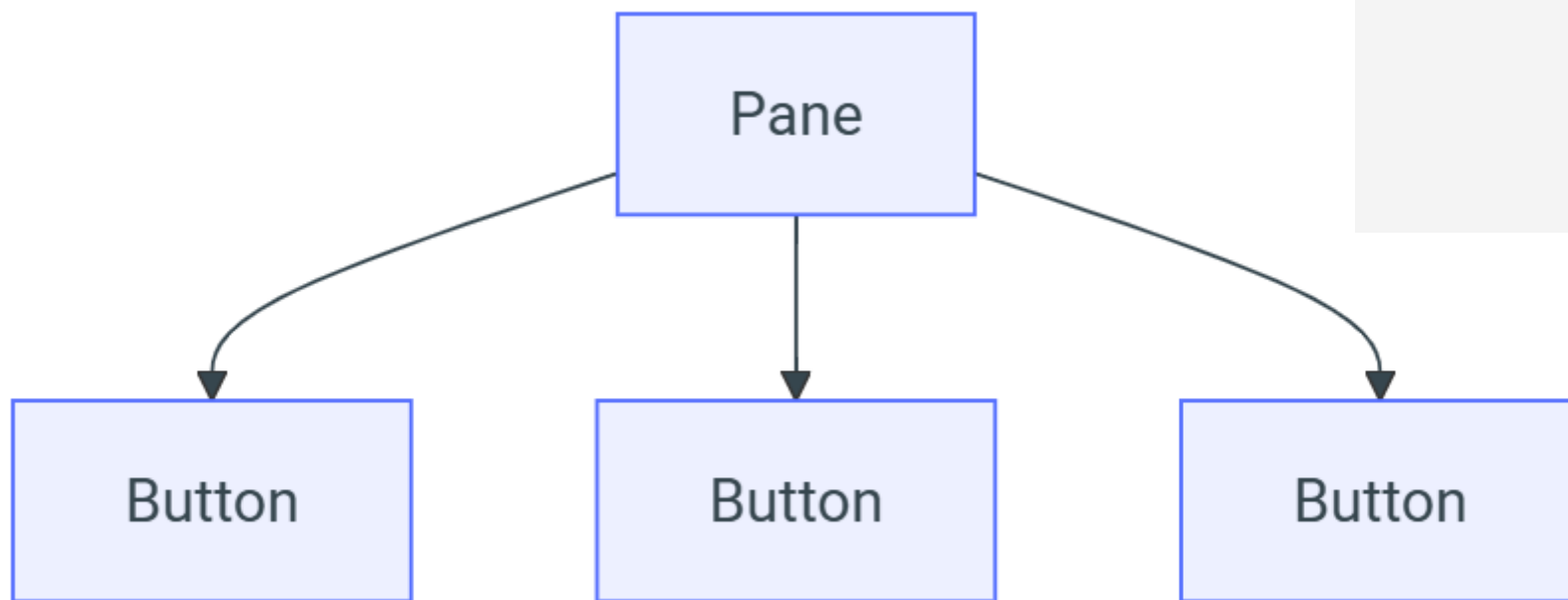
// Všetko dáme dokopy
root.getChildren().addAll(counterRow, separator, bottomRow);
Scene scene = new Scene(root, 380, 220);
```

# Z-Order

Z-ordering znamená **poradie vykresľovania grafických prvkov** - čo je navrchu a čo naspodku

Štandardne používame 2 osy, X a Y. Pozícia 0,0 je v ľavom hornom rohu. Prekrytie je určene treťou osou, Z, ktorá ide kolmo od obrazovky.

# Z-Order





# Z-Order

JavaFX určuje Z-order poradie na základe **pozície prvku v strome uzlov - scene graphu**

- Ako prvý sa vykresľuje koreňový uzol a potom sa vykresľujú jeho deti.
- Deti sa vykresľujú v poradí, v akom sú do kontajnera umiestnené.

Teda **posledný komponent bude úplne navrchu**.

Zmenu poradia vieme vykonať premiestnením komponentov v rámci scene graphu, či už na iné miesto v strome alebo na inú pozíciu v rámci zoznamu detských komponentov