CSCI 2120:
Software Design & Development II

UNIT4: UI management

GUI framework
JavaFX: Layout Panes (Containers)

Overview

1. Introduction
2. Creating a Layout
3. Creating Nodes
4. Instantiating the Respective Class
5. Setting the Properties of the Layout
6. Adding the Shape Object to the Group
7. Layout Panes

Introduction

- After constructing all the required nodes in a scene, we will generally arrange them in order.

- This arrangement of the components within the container is called the Layout of the container. We can also say that we followed a layout as it includes placing all the components at a particular position within the container.

- JavaFX provides several predefined layouts such as **HBox, VBox, Border Pane, Stack Pane, Text Flow, Anchor Pane, Title Pane, Grid Pane, Flow Panel, etc.**

- Each of the above mentioned layout is represented by a class and all these classes belongs to the package j**avafx.layout.**

- The class named **Pane** is the base class of all the layouts in JavaFX.

Creating a Layout

To create a layout, you need to −

- Create node.

- Instantiate the respective class of the required layout.

- Set the properties of the layout.

- Add all the created nodes to the layout.

Creating Nodes

First of all, create the required nodes of the JavaFX application by instantiating their respective classes.

For example, if you want to have a text field and two buttons namely, play and stop in a HBox layout - you will have to initially create those nodes as shown in the following code block −

```
//Creating a text field
TextField textField = new TextField();

//Creating the play button
Button playButton = new Button("Play");

//Creating the stop button
  Button stopButton = new Button("stop");
```

Instantiating the Respective Class

After creating the nodes (and completing all the operations on them), instantiate the class of the required layout.

For Example, if you want to create a Hbox layout, you need to instantiate this class as follows.

```
HBox hbox = new HBox();
```

Setting the Properties of the Layout

After instantiating the class, you need to set the properties of the layout using their respective setter methods.

For example − If you want to set space between the created nodes in the HBox layout, then you need to set value to the property named spacing. This can be done by using the setter method setSpacing() as shown below −

```
hbox.setSpacing(10);
```

Adding the Shape Object to the Group

Finally, you need to add the object of the shape to the group by passing it as a parameter of the constructor as shown below.

```
//Creating a Group object
Group root = new Group(line);
```

Layout Panes

Following are the various Layout panes (classes) provided by JavaFX. These classes exist in the package **javafx.scene.layout.**

S.No	Shape & Description
1	**HBox** The HBox layout arranges all the nodes in our application in a single horizontal row. The class named HBox of the package javafx.scene.layout represents the text horizontal box layout.
2	**VBox** The VBox layout arranges all the nodes in our application in a single vertical column. The class named VBox of the package javafx.scene.layout represents the text Vertical box layout.
3	**BorderPane** The Border Pane layout arranges the nodes in our application in top, left, right, bottom and center positions. The class named BorderPane of the package javafx.scene.layout represents the border pane layout.
4	**StackPane** The stack pane layout arranges the nodes in our application on top of another just like in a stack. The node added first is placed at the bottom of the stack and the next node is placed on top of it. The class named StackPane of the package javafx.scene.layout represents the stack pane layout.
5	**TextFlow** The Text Flow layout arranges multiple text nodes in a single flow. The class named TextFlow of the package javafx.scene.layout represents the text flow layout.

Layout Panes

Following are the various Layout panes (classes) provided by JavaFX. These classes exist in the package **javafx.scene.layout.**

S.No	Shape & Description
6	**AnchorPane** The Anchor pane layout anchors the nodes in our application at a particular distance from the pane. The class named AnchorPane of the package javafx.scene.layout represents the Anchor Pane layout.
7	**TilePane** The Tile Pane layout adds all the nodes of our application in the form of uniformly sized tiles. The class named TilePane of the package javafx.scene.layout represents the TilePane layout.
8	**GridPane** The Grid Pane layout arranges the nodes in our application as a grid of rows and columns. This layout comes handy while creating forms using JavaFX. The class named GridPane of the package javafx.scene.layout represents the GridPane layout.
9	**FlowPane** The flow pane layout wraps all the nodes in a flow. A horizontal flow pane wraps the elements of the pane at its height, while a vertical flow pane wraps the elements at its width. The class named FlowPane of the package javafx.scene.layout represents the Flow Pane layout.

END