UNIT-6 [Around 5 marks]

GUI with JavaFX

[Imp]

JavaFX is a Java library used to develop Desktop applications as well as Rich Internet Applications (RIA). The applications built in JavaFX, can run on multiple platforms including Web, Mobile and Desktops. JavaFX is intended to replace swing in Java applications as a GUI framework. It provides more functionalities than swing. It supports various operating systems including Windows, Linux and Mac OS.

⊛ JavaFX vs. Swing: [Imp]

Swing	JavaFX
1. Swing is the standard toolkit for Java developer in creating GUI.	1. JavaFX provides platform support for creating desktop applications.
2. Swing has a more sophisticated set of GUI components.	2. JavaFX has decent but lesser number of GUI components.
3. Swing does not have support for customization using CSS and XML.	3. JavaFX has a support for customization using CSS and XML.
4. With Swing, it is very difficult to create beautiful 3-D applications.	4. With JavaFX one can also create beautiful 3-D applications.
5. Swing has a UI component library and act as a legacy.	5. JavaFX has several components built over Swing.

⊛ JavaFX Layouts: [Short concepts Imp]

JavaFX has following built-in layout panes: FlowPane, HBox, VBox, BorderPane, GridPane.

FlowPane: FlowPane layout organizes the nodes in a flow that are wrapped at the flowpane's boundry. The horizontal FlowPane arranges the nodes in a row and wrap them according to the flowpane's width. The vertical flowpane arranges the nodes in a column and wrap them according to the flowpane's height.

alignment, columnHalignment, hgap, orientation, vgap, rowValignment etc. are various properties of FlowPane class. FlowPane(), FlowPane (Double Hgap, Double Vgap), FlowPane(Node... Children), FlowPane (Orientation orientation) etc. are some constructors of FlowPane.

BorderPane: BorderPane arranges the nodes at the left, right, centre, top, and bottom of the screen. This class provides various methods like setRight(), setLeft(), setCenter(), setBottom(), and setTop() which are used to ~~the~~ set the position for the specified nodes.

Bottom, Centre, Left, Right, and Top are the properties of BorderPane class. BorderPane(), BorderPane(Node Center), BorderPane(Node Center, Node top, Node right, Node bottom, Node left) are its constructors.

GridPane: GridPane layout allows us to add the multiple nodes in multiple rows and columns. It is seen as a flexible grid of rows and columns where nodes can be placed in any cell of the grid.

alignment, gridLinesVisible, hgap, vgap are its properties. This class contains only one constructor Public GridPane(), which creates a gridpane with 0 hgap/vgap.

HBox: HBox layout pane arranges the nodes in a single row. alignment, fillHeight, spacing are its properties. new HBox(), new HBox (Double spacing) are its constructors.

VBox: VBox Layout pane arranges the nodes in a single column. alignment, fill Width, Spacing are its properties. VBox(), VBox(Double Spacing), VBox (Double spacing, Node? children), VBox (Node? children) are its constructors.

⊛ JavaFX UI Controls: [Imp]

The UI elements are the one which are actually shown to the user for interaction or information exchange. Layout defines the organization of the UI elements on the screen. Behaviour is the reaction of the UI element when some event is occured on it.

The package javafx.scene.control provides all the necessary classes for the UI components like Button, Label etc. Every class represents a specific UI control and defines some methods for their styling.

Label: Label is a component that is used to define a simple text on the screen. Typically, a label is placed with the node, it describes. javafx.scence.control.Label class represents label control. Label(), Label(String text), Label (String text, Node graphics), are its constructors.

TextField: Text Field is basically used to get input from the user in the form of text. javafx.scene.control.TextField represents TextField. It provides various methods to deal with textfields in JavaFx.

Button: Button is a component that controls the function of the application. Button class is used to create a labelled button. javafx.scence.control.Button class represents Button. An event is generated whenever the button gets clicked.

RadioButton: The Radio Button is used to provide various options to the user. The user can only choose one option among all. A radio button is either selected or deselected. It can be used in a scenario of multiple choice questions like in quiz where one option needs to be choosen.

CheckBox: Check Box is used to get the kind of information from the user which contains various choices. User marks the checkbox either on (true) or off (false). It can be used in a scenario where the user is prompted to select more than one option.

HyperLink: Hyperlink are used to refer any of the webpage through our application. It is represented by the class javafx.scene.control.Hyperlink. It is similar to anchor links in HTML.

Menu: JavaFx provides a Menu class to implement menus. Menu is the main component of any application. javafx.scene.control.Menu class provides all the methods to deal with menus.

Tooltip: JavaFx ToolTip is used to provide hint to the user about any component. It is mainly used to provide hints about the textfields or password fields being used in the application.

FileChooser: JavaFx FileChooser enables users to browse the files from the file system. The FileChooser class provides two types of methods: showOpenDialog() and showSaveDialog(). javafx.stage.FileChooser class represents FileChooser.

⊛ Steps of creating GUI using javaFx: [Imp]

Step1: Extend javafx.application.Application and override start().

Step2: Create any UI control component using javafx GUI, according to our need.

Step3: Create any layout and add UI component to it.

Step4: Create a Scene by instantiating javafx.scene.Scene class.

Step5: Prepare the stage with javafx.stage.Stage class.

Step6: Create an event for the UI component if any and call setOnAction() method for handling event, in which we can define any method which contains code for how event is handeled.

Step7: Create the main method to launch the application.

❤️

If my notes really helped you,then you can support me on esewa for my hardwork.

Esewa ID: 9806470952