Rachel Theis

Module 11 Assignment

CSD 402

3.2.2925

**JavaFX**

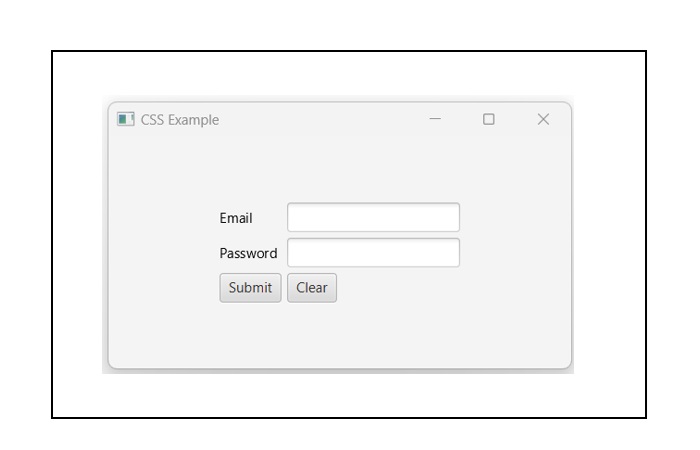
JavaFX is a library of graphical user interface (GUI) tools and components that can be used to develop rich desktop and web applications (GeeksForGeeks, 2021). JavaFX is compatible with the most common platforms, including Mac, Windows, Linux, iOS, and Android (GeeksForGeeks, 2021). JavaFX was introduced to supersede Java Swing and is considered to be lightweight and hardware-simulated (GeeksForGeeks, 2021). JavaFX is an open-source framework and is available as a public API. JavaFX is considered to be a modern and advanced choice for GUI development and includes the following features that make it a good choice for many mobile and desktop applications (GeeksForGeeks, 2021):

* Java Library
* Platform Independent
* FXML
* Scene Builder
* Hardware Accelerated Graphics Pipeline
* WebView
* Built in UI Code
* CSS styling
* Rich set of APIs
* High-performance Media Engine

Two components of JavaFX are GridPane and Accordion. GridPane allows developers to assign positions on a page to elements called children. Accordion allows for dropdown menu functionality within a GUI.

**GridPane**

GridPane is a component of the JavaFX framework. GridPane is a layout framework that arranges elements, or children, in a grid (Jenkov, 2020). GridPane uses columns and rows to place children. All cells within the same row or column must be the same height or width, but rows and columns can vary in size (Jenkov, 2020). This differentiates GridPane from TilePane (in TilePane, all cells must be uniform) (Jenkov, 2020). Below is an example of how GridPane appears in a graphical interface (TutorialsPoint, n.d.-b):



(TutorialsPoint, n.d.-b)

**Syntax of GridPane**

The syntax to create a GridPane is: GridPane gp = new GridPane();. This is created through a constructor. To add children to the GridPane developers can follow the below syntax options:

GridPane.setRowIndex(button, 0);

GridPane.setColumnIndex(button, 1);

//This option sets one constraint at a time; here the button is being placed in the first row and second column

Lablel label = new label ();

GridPane.setConstrains(label, 2, 0);

//This option sets the constrains at once

Gridpane.GetChildren().addAll(button, label);

(oracle, 2015)

Here is another option for adding buttons, here the setting constrains and adding children steps are combined into one.

GridPane gridpane = new GridPane();

gridpane.add(new Button (), 1, 0);

gridpane.add(new Label (), 2, 0);

(oracle, 2015)

**Properties of GridPane**

GridPane has four direct properties: ObjectPropert<Pos>, BooleanProperty, DoubleProperty (hgap), and DoubleProperty (vgap) (Victor, 2019).

* ObjectProperty<Pos>: This property aligns an element within the grid
* BooleanProperty: This property is only used for debugging purposes and controls whether lines are displayed for rows and columns
* DoubleProperty (hgap): This property specifies the width of the horizontal gaps between columns.
* DoubleProperty(vgap): This property specifies the height of the vertical gaps between rows.

GridPane also inherits certain properties from other JavaFX classes: javafx.scene.layout.Region, javafx.scene.Parent, javafx.scene.Node

**Methods of GridPane**

GridPane has numerous methods that add functionality to GridPane. Below are some of the most commonly used GridPane methods (oracle, 2015).

* clearConstrains(Nodechild): Removes parent constraints from child node (oracle, 2015)
* computeMinWidth(double height): Computes the minimum width of the region (oracle, 2015)
* computeMinHeight(double width): Computes the minimum height of the region (oracle, 2015)
* getAlignment(): Returns the alignment property’s value (oracle, 2015)
* setAlignment(Posvalue): Sets the alignment property’s value (oracle, 2015)
* GetMargin(Nodechild): Returns margin constraints (oracle, 2015)
* computePrefWidth(double height): Computes the preferred width of the region (oracle, 2015)
* computePrefHeight(double width): Computes the preferred height of the region (oracle, 2015)

**Accordion**

Accordion is a component of the JavaFX GUI development framework . Accordion is a type of control container that allows for internal sections and subsections to be expanded or collapsed. Accordion is a group of TitlePanes, and only one TitledPane can be opened at one time. Below is an example of how an accordion can look:

A white rectangular object with two white stripes

AI-generated content may be incorrect.

(TutorialsPoint, n.d.-a)

**Syntax of Accordion:**

The syntax to create a new accordion is: Accordion accordion = new Accordion(); (Jenkov, 2019).

To add TitledPane objects to an accordion, the following syntax is used (Jenkov, 2019):

Accordion accordion = new Accordion();

TitledPane pane1 = new TitledPane("Example A" , new Label("Expand to see example A"));

TitledPane pane2 = new TitledPane("Example B" , new Label("Expand to see example B"));

TitledPane pane3 = new TitledPane("Example B", new Label("Expand to see example C"));

accordion.getPanes().add(pane1);

accordion.getPanes().add(pane2);

accordion.getPanes().add(pane3);

(Jenkov, 2019)

**Properties of Accordion:**

Accordion is associated directly with one method, ObjectProperty<TiltedPane>, which is the expanded TitledPane within the accordion (oracle, 2015). Accordion also inherits properties (contextMenu, skin, tooltip) from javafx.scene.control.Control, javafx.scene.layout.Region, javafx.scene.Parent, javafx.scene.Node (oracle, 2015).

**Methods of Accordion:**

Accordion is directly associated with five methods (oracle, 2015):

* protected Skin<?>: createDefaultSkin(); creates a new instance of default skin for the specific control (oracle, 2015)
* ObjectProperty<TitledPane>:expandedPaneProperty(); this is the expanded TitledPane in the accordion (oracle, 2015)
* TitledPane: getExpandedPane(); gets the expanded TitledPane in the Accordion (oracle, 2015)
* ObservableList<TitledPane>: getPanes(); gets the list of TitledPane in the accordion (oracle, 2015)
* void: setExpandedPane(TitledPane value); voids the expanded TitledPane that is currently visible (oracle, 2015)

Accordion also inherits methods from the following classes: javafx.scene.control.Control, class javafx.scene.layout.Region, class javafx.scene.Parent, javafx.scene.Node (oracle, 2015).

**References**

GeeksForGeeks. (2021, October 5). *JavaFX Tutorial*. GeeksforGeeks. https://www.geeksforgeeks.org/javafx-tutorial/

Jenkov, J. (2019, April 27). *JavaFX Accordion*. Tutorials.jenkov.com. https://jenkov.com/tutorials/javafx/accordion.html

Jenkov, J. (2020, April 29). *JavaFX GridPane*. Jenkov.com. https://jenkov.com/tutorials/javafx/gridpane.html

oracle. (n.d.). *GridPane (JavaFX 8)*. Docs.oracle.com. https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html

oracle. (2015, February 10). *Accordion (JavaFX 8)*. Oracle.com. https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Accordion.html

TutorialsPoint. (n.d.-a). *How to create accordion in JavaFX?* Www.tutorialspoint.com. https://www.tutorialspoint.com/how-to-create-accordion-in-javafx

TutorialsPoint. (n.d.-b). *JavaFX - Layout GridPane - Tutorialspoint*. Www.tutorialspoint.com. https://www.tutorialspoint.com/javafx/layout\_gridpane.htm

Victor, P. (2019, December 13). *JavaFX GridPane*. EDUCBA. https://www.educba.com/javafx-gridpane/