Bellevue University

**Module 7**

Donnell Perkins

Intermediate Java Programming – CSD405

Instructor: Darrell Payne

Due: 07/02/2023

Intro

For this assignment we are to research JavaFX Layout topics. I chose to talk about the Accordion and the GridPane layouts. These two layouts appealed to me because I thought they would be great tools when creating webpages and making sure that the user is getting the specific information that they need and keeping the website neat. And because an accordion can contain many kinds of elements I will discuss and give an example of an accordion with a drop down that contains a GridPane.

JavaFX Accordion & JavaFX GridPane

I would say that the Accordion is a great tool to add to your website when you have a few options that you would like to share with the user but once the option is selected the information and the window is now no longer needed and can be minimized to referenced later. You can almost think of it as a container because it basically is. Each Accordion contains a few title panes, with is also a layout itself. Each pane can hold a few elements to display to the user. And only one pane can be displayed at one time per accordion. To use the Accordion class in your code you will need to import the [***javafx.scene.control.Accordion***](https://www.tutorialspoint.com/how-to-create-accordion-in-javafx) class at the top of you file.

To create an accordion and add each drop down you create the number of panes, set the contents, and use methods getPanes() to get the created panes and addAll() methods puts them all together in the accordion.

GridPanes are helpful for an application especially when you are adding many elements to a scene. You wouldn’t want them scattered all over the webpage. Gridpanes are there to help give you a structure to work with.  [(Tutorials Point, 2023)](https://www.tutorialspoint.com/javafx/layout_gridpane.htm)All of the notes that are added to it are arranged in a way that they form a grid of rows and columns. The gridPane class has a couple properties that are frequently used.

Alignment – Represents the alignment of the pane and you can set the value of this property using the setAlignment() method.

hgap – The property type is Double, and it represents the horizontal gap between columns.

vgap – The property type is Double as well and presents the vertical gap between rows.

I did not us this in my example but if you would like to display the grid lines you can use the property gridLinesVisible.

gridLinesVisible – This is a Boolean property and when set true the lines are displayed otherwise you don’t see the lines.

To use GridPanes you must import the ***javafx.scene.layout.GridPane.***

The Code

In my first drop down example I created a space for the user to select a character using radio buttons. I added the radio buttons to a Vbox so that they displayed vertically. To use the VBox imported ***javafx.scene.layout.VBox*** class. In my example we used the setPadding to indent the radio buttons so that they are not along the edge of the window. In the geometry class we can use Inset() to help achieve that. And we must return all the nodes to the VBox by calling the getChildren() method. And for this drop down we are using the TitledPane to give the pane a title and display its contents on initialize.

In my next drop down, I created some label and field examples and using a GridPane, which we discussed above, to display them. Set the minimum size for the grid to ensure all the data you need is displayed. To create the labels, we must include the ***javafx.scene.text.Text***. And I can add the text fields using ***javafx.scene.control.TextField.*** Then set the vertical and horizonal gaps for the columns between the labels and fields. Calling the setAlignment() method for the grid and I chose to use the position in the geometry position class and positioned the grid to the left of the window. Then arranging each node in the grid so that the labels and fields display the way that I want them to. For this drop down we just created the TitledPane and then set the drop down label and contents separately.

Now that both drop downs have been created, we can put them together to form the accordion. We create an accordion object and use the getPanes() methods to add in each pane/dropdown that I want to be a part of this accordion. Then we use a Vbox to display the dropdowns vertically.

# Works Cited

GeekForGeeks. (2023, 07 02). *JavaFX How to set padding between nodes of a gridpane*. Retrieved from Geek For Geeks: https://www.geeksforgeeks.org/javafx-how-to-set-padding-between-nodes-of-a-gridpane/

GeeksForGeeks. (2023, 07 02). *JavaFX Vbox Class*. Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/javafx-vbox-class/

Tutorial Point. (2023, 07 02). *How to create Accordion in Javafx*. Retrieved from TutorialsPoint: https://www.tutorialspoint.com/how-to-create-accordion-in-javafx

Tutorials Point. (2023, 07 02). *Layout GridPane*. Retrieved from Tutorialpoint.com: https://www.tutorialspoint.com/javafx/layout\_gridpane.htm