DeJanae Faison

5.11.2025

CSD402 M11 Assignment

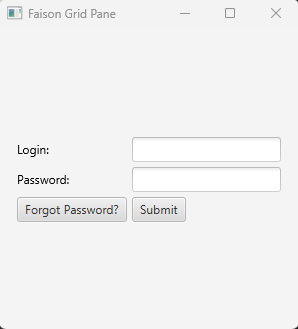
# JavaFX GridPane

JavaFX is Java’s graphic user interface library to develop applications across multiple platforms. This library is dense with tools to create flexible user interfaces that supports CSS styling, audio formats, videos, images and even 3D graphics. JavaFX’s GridPane is a layout in which all the UI nodes (or elements) are aligned in a table format in rows and columns.

GridPane uses the Pane as a matrix or a map and an indicator on where elements can be placed. Although uses a row and column format, it does not display grids on the pane but as stated previously provides the program with a map to know where on pane the element to be placed. For example:

|  |  |  |
| --- | --- | --- |
| [0,0] | [1,0] | [2,0] |
| [0,1] | [1,1] | [2,1] |
| [0,2] | [1,2] | [2,2] |

With TutorialsPoint as a base, we will be creating this login form using a GridPane:



1. Create a public class with the desired name that extends the Application class.



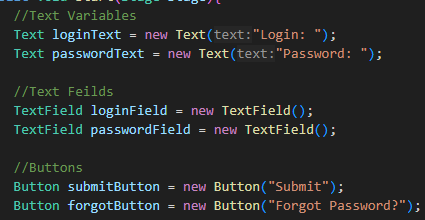
1. Create two methods under the class we created
   1. A start method



* 1. A main method for launching the actual program



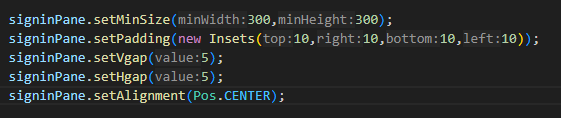
1. Next, we are going to create the elements that will go into our grid pane:
   1. Text = will act as our label for input fields
   2. TextField = the input fields that well be next to their appropriate labels
   3. Button = action buttons



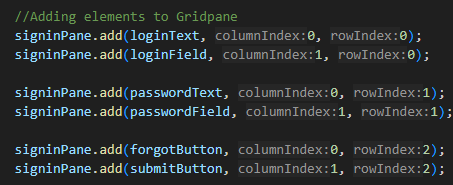
1. Then we are going to create our GridPane



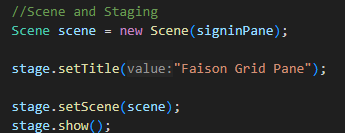
1. After that we will be applying some style to our pane:
   1. ‘.setMinSize(width,height)’ will control the smallest our window can be
   2. ‘.setPadding()’ is the spacing around the elements/around the pane
   3. ‘.setVgap()’ sets the vertical spacing between each column
   4. ‘.setHgap()’ sets the horizontal spacing between each row
   5. ‘.setAllignment()’ determines the element’s position within it’s grid placement



1. Next we will be adding our elements to the GridPane
   1. After calling out gridpane variable, we add ‘.add’ then the name of the variable we want to add then the column and row number we want it placed in



1. Lastly we set our scene and stage our pane:



Important GridPane Methods:

|  |  |
| --- | --- |
| **Method** | **Function** |
| .add(nodeName, columnNumber, rowNumber); | Appends element to Grid pane with the name of the element and coordinates. |
| .setRowIndex(nodeName, rowNumber);  .setColumnIndex(nodeName, columnNumber); | Places the node at the desired location. |
| .setMinSize(width, height); | Set the smallest size the window can be, similarly with setMaxSize sets the largest the window can be |

# JavaFX Accordion

Another UI element in JavaFX’s user interface library is Accordion; the Accordian is known more familiarly as a dropdown menu. It serves as a container for other sections of the application that can be folded and expanded when desired. This creates a clean layout and does not overwhelm the user with all the submenus open and present.

An Accordian is made from a collection of different Tilted Panes; a Tilted Pane is the container for the nodes and is a child of the Accordian:

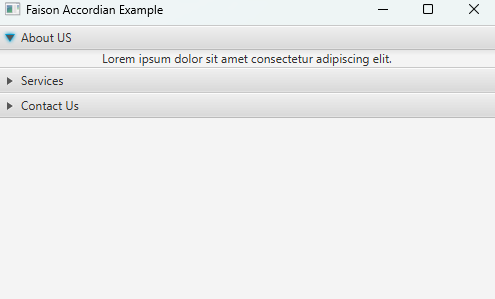
→Stage

→→Accordian

→→→Tilted Pane

→→→→Pane Contents

How to make an accordion with this desired output, with TutorialsPoint as a base:



1. Create a public class with the desired name (AccordianApp) that extends Application; this means the class will inherit attributes from the Application class (this will automatically import as well).



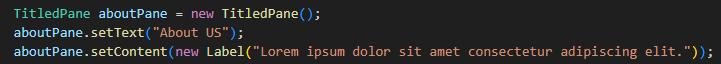
1. Create two methods under our AccordianApp class:
   1. A start method that will hold the contents of our accordion:



* 1. A main method that will launch the application.



1. Next is to initialize the Tilted Panes that will go within our according.
   1. The ‘. setText (““);’ will add the string within the quotes as the header of our drop-down menu.
   2. ‘. setContent (new Label (“”));’ will create a new label as the contents under our header. A label is used so it can fit within the desired space.
   3. This step is repeated for as many drop downs as needed.



1. Then the Accordion is made:



1. After the accordion is made, the panes are appended with the following functions:
   1. ‘.getPanes()’ is a method will fetch the tilted panes and their contents
   2. ‘.addAll(paneNames)’ will append all the panes to the accordion we made in step four.



1. Next, we create a VBox which is a class that lays out the contents within it in vertical columns. Place the accordion we made as an argument; making it a child of the vertical columns:



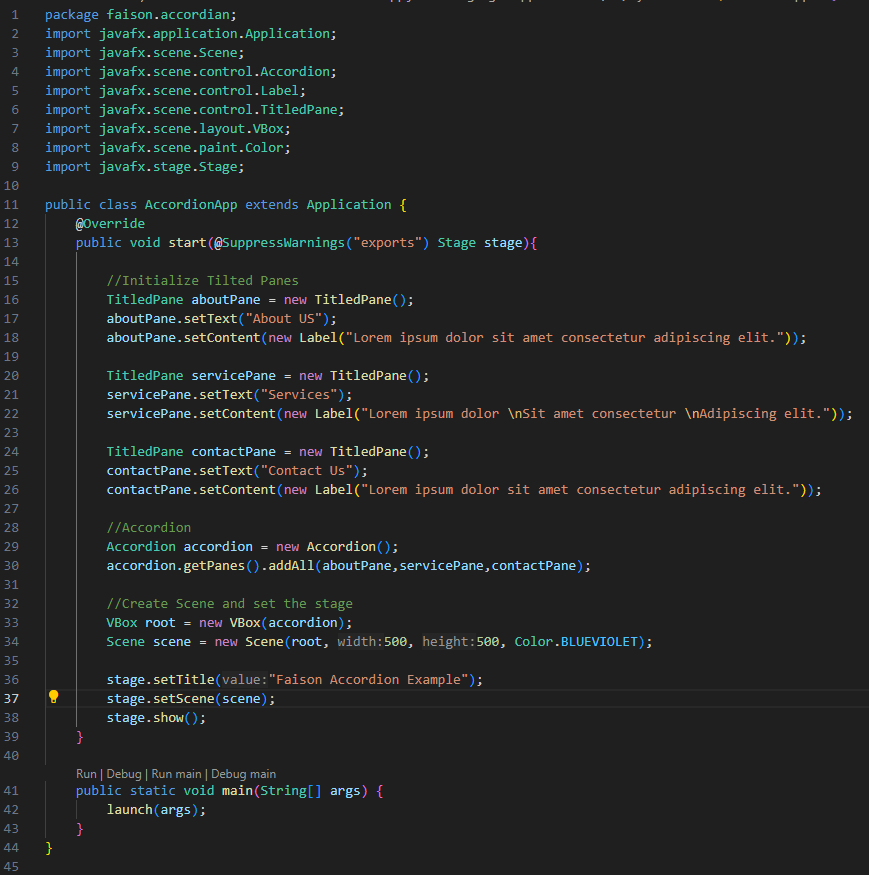
1. Then we set the Scene, we make a new scene (in short it is a container for everything).
   1. We make the Vbox we just created as a child of the scene
   2. Define the width and height



1. Finally, we add our scene to the stage parameter we made in our start method.
   1. ‘.setTitle(“”)’ adds a string title for the overall window
   2. ‘.setScene(scene)’ appends our scene
   3. ‘stage.show();’ shows the actual window we created



Overall the code should look like this:



*Resources*:

JavaFX Tutorial - Tutorialspoint. (n.d.). [Www.tutorialspoint.com](http://Www.tutorialspoint.com). <https://www.tutorialspoint.com/javafx/index.htm>

JavaFX - Layout GridPane - Tutorialspoint. (n.d.). [Www.tutorialspoint.com](http://Www.tutorialspoint.com). <https://www.tutorialspoint.com/javafx/layout_gridpane.htm>

JavaFX - Accordion. (2024). Tutorialspoint.com. <https://www.tutorialspoint.com/javafx/javafx_accordion.htm>

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

Jenkov, J. (n.d.). JavaFX Accordion. Tutorials.jenkov.com. <https://jenkov.com/tutorials/javafx/accordion.html>

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

JavaFX | VBox Class. (2018, September 6). GeeksforGeeks. <https://www.geeksforgeeks.org/javafx-vbox-class/>

Jenkov, J. (n.d.). JavaFX TitledPane. Tutorials.jenkov.com. <https://jenkov.com/tutorials/javafx/titledpane.html>