JavaFX TitledPane and Accordion

JavaFX is a versatile framework for creating graphical user interfaces using Java coding. It offers a variety of layout options for creating modern-looking desktop applications. This often involves grouping similar elements together in containers. The TitledPane and Accordion classes are two examples that can be used to create interactive aspects when designing user interfaces. These components are beneficial when designing interfaces that need to present a lot of content while preserving screen space. This paper will explain the basic properties and common uses of TitledPane and Accordion and explore how they might be used in a sample application, such as a restaurant menu.

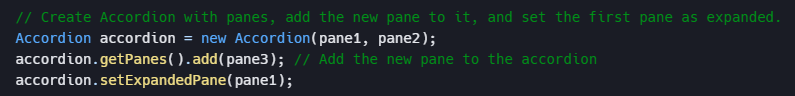
A TitledPane is a user interface container that includes a title bar and a content area. When collapsed, only the title is visible. When the pane is expanded, the content inside becomes visible as well. This makes TitledPane ideal for hiding content that should only be shown when the user explicitly requests it. Any kind of content—such as forms, text, images, or lists—can be placed inside a TitledPane, allowing developers to manage screen space efficiently.

The TitledPane is created with the TitledPane() constructor. If the title and node content aren’t used as parameters, an empty TitledPane is created and can be filled with .setText() and .setContent() later. For example, if building a restaurant menu, you might use separate titled panes for different categories like in the following code::

A screen shot of a computer code

AI-generated content may be incorrect.

The Accordion control works closely with TitledPane—in fact, it is composed of multiple titled panes grouped together. The key difference is that in an Accordion, only one pane can be expanded at a time. This ensures the user is only viewing one section's content, preventing visual clutter and making the interface more navigable.

To create an Accordion, you instantiate it and add multiple TitledPane objects to it using its getPanes() method. For instance, when designing a restaurant menu application, the Accordion could display different meal categories. When the user expands one pane (e.g., Sides), any other open pane (e.g., Drinks) will automatically collapse. Like the TitledPane() constructor, the Accordion can be initialized with its contents, or they can be added separately. Here is an example of creating an Accordion with panes, adding another and setting the first panel expanded: 

In the previous example the setExpandedPane method sets a selected pane as expanded on launch. This is one of many different methods that can be used to customize the design and interaction with other parts of the program. Let’s say the menu application wants to allow users to submit orders and then display them. This can be achieved with the following code:

A screen shot of a computer code

AI-generated content may be incorrect.

Note the VBox had already been initialized as an instance variable so it can be accessed by multiple methods. Separately in the code there is implementation to add orders to it. Here we see it placed in a ScrollPane, which then in turn is placed within a Titled Pane. The Scrollpane allows for scrolling through multiple orders if needed. This example also allows us to see TitledPane’s Boolean properties Collapsible, Expanded, and Animated in use. Unless specified they will be true but can be turned off. In this case we wanted the pane to be expanded but not collapsible or animated.

Another important aspect to note is that in addition to creating these objects, we must also set them within the layout. The outer most layer is the Stage, which then presents the scene. In this program all the different parts are placed in a VBox which is then used to set the scene, along with an optional size setting.

A computer screen shot of text

AI-generated content may be incorrect.

This was just a brief overview of implementing the TitledPane and Accordion nodes in a JavaFx application. There are many ways to tweak settings and interactions between nodes, allowing an immense amount of customization in the design process. Below is the result of the example program after the user has input a few orders

A screenshot of a computer

AI-generated content may be incorrect.

**References**

Semantic Portal. (n.d.). *Java basics – Graphics JavaFX UI controls: TitledPanel and Accordion*. Retrieved July 19, 2025, from <http://semantic-portal.net/java-basics-graphics-javafx-ui-controls-titled-panel-and-accordion>

Oracle. (n.d.). *TitledPane (JavaFX 8)*. Oracle Help Center. Retrieved July 19, 2025, from <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TitledPane.html>

Oracle. (n.d.). *Accordion (JavaFX 8)*. Oracle Help Center. Retrieved July 19, 2025, from <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Accordion.html>

Oracle. (n.d.). *Using JavaFX UI controls: Accordion and TitledPane*. Oracle Help Center. Retrieved July 19, 2025, from <https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/accordion-titledpane.htm>