Advanced JavaFX

Name Dr. Abdallah Karakra |Comp 2311 | Masri521

21/01/2023

BIRZEIT UNIVERSITY

CHAPTER 31

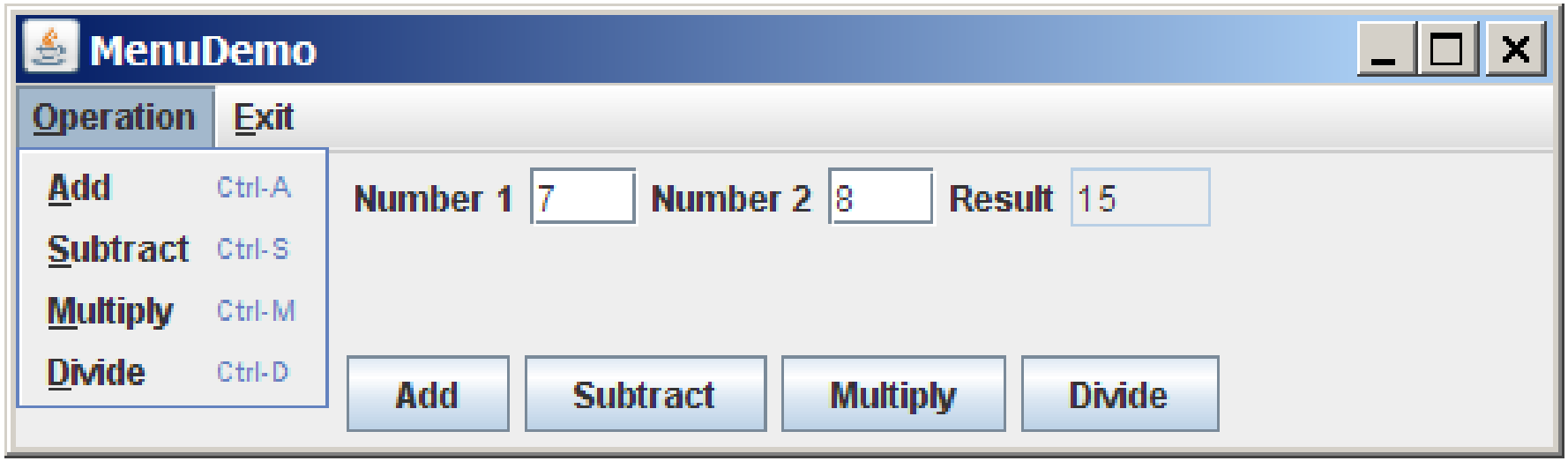
Advanced JavaFX

Menu

Menus make selection easier and are widely used in window applications. JavaFX provides five classes that implement menus: MenuBar, Menu, MenuItem, CheckMenuItem, and RadioMenuItem.

MenuBar is a top-level menu component used to hold the menus. A menu consists of menu items that the user can select (or toggle on or off). A menu item can be an instance of MenuItem, CheckMenuItem, or RadioButtonMenuItem. Menu items can be associated with nodes and keyboard accelerators.

Creating Menus

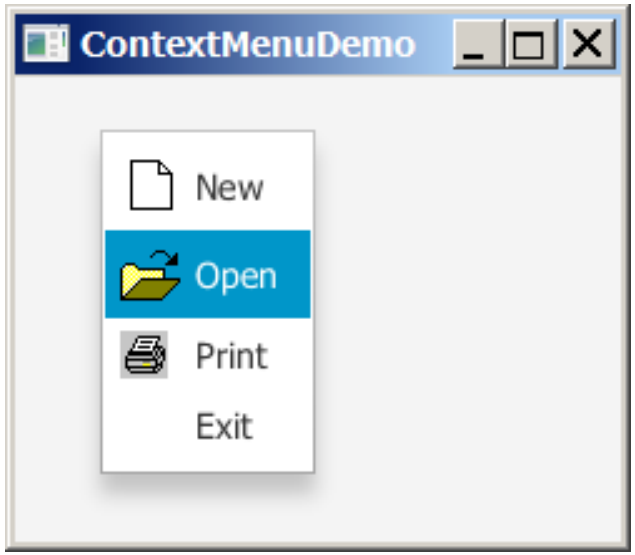


Context Menu

A context menu, also known as a popup menu, is like a regular menu, but does not have a menu bar and can float anywhere on the screen. Creating a context menu is similar to creating a regular menu.

- First, you create an instance of ContextMenu,
- then you can add MenuItem, CheckMenuItem, and RadioMenuItem to the context menu.

Creating Context Menus

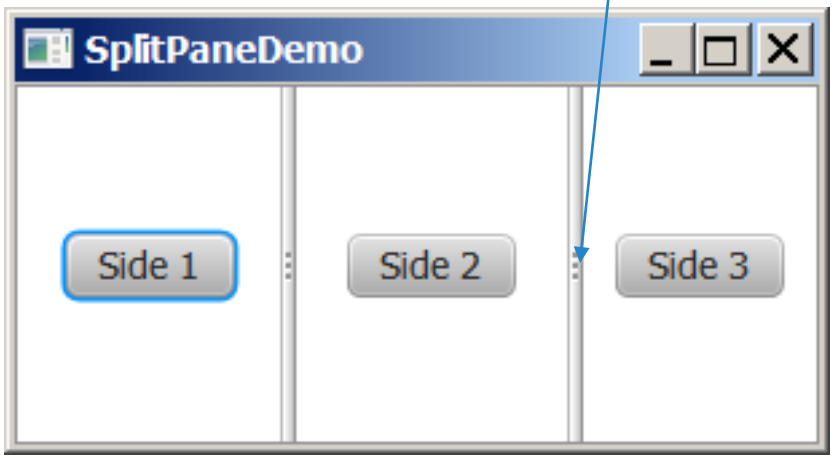


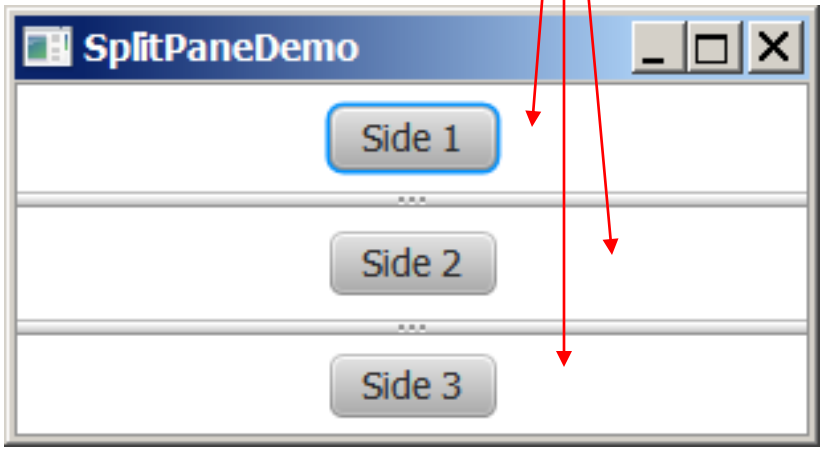

```
ContextMenu contextMenu = new ContextMenu();
MenuItem menuItemNew = new MenuItem("New",
  new ImageView("image/new.gif"));
MenuItem menuItemOpen = new MenuItem("Open",
  new ImageView("image/open.gif"));
MenuItem menuItemPrint = new MenuItem("Print",
  new ImageView("image/print.gif"));
MenuItem menuItemExit = new MenuItem("Exit");
contextMenu.getItems().addAll(menuItemNew, menuItemOpen,
  menuItemPrint, menuItemExit);

Pane pane = new Pane();
Scene scene = new Scene(pane, 300, 250);
primaryStage.setTitle("ContextMenuDemo"); // Set the window title
primaryStage.setScene(scene); // Place the scene in the window
primaryStage.show(); // Display the window
```

SplitPane

The SplitPane class can be used to display multiple panes and allow the user to adjust the size of the panes.

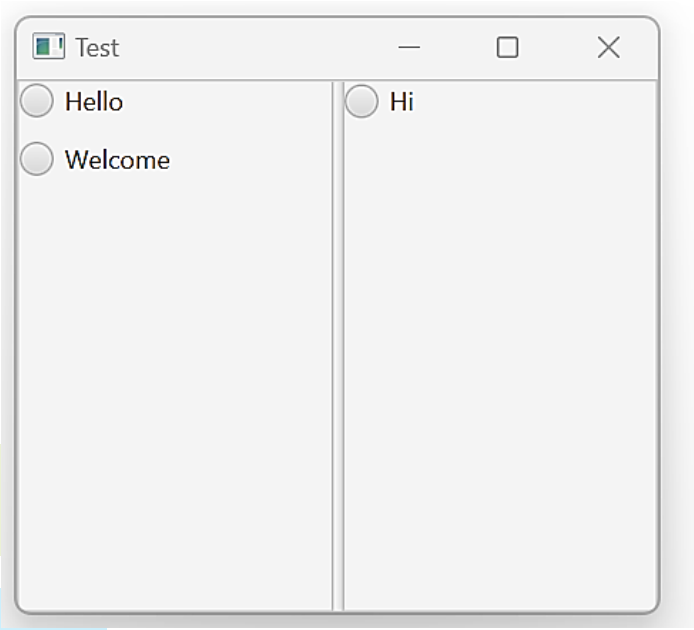





```
SplitPane content = new SplitPane();
content.setOrientation(Orientation.VERTICAL);
```

```
SplitPane content = new SplitPane();
content.setOrientation(Orientation.HORIZONTAL);
```

SplitPane



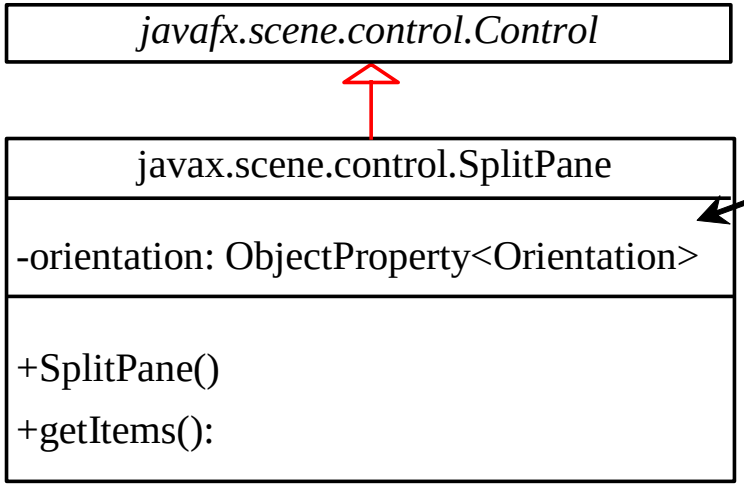

```
SplitPane content = new SplitPane();
content.setOrientation(Orientation.HORIZONTAL);

VBox vBox = new VBox(10);
VBox vBox1 = new VBox(10);
RadioButton rbHello = new RadioButton("Hello");
RadioButton rbWelcome = new RadioButton("Welcome");
RadioButton rbHi = new RadioButton("Hi");

vBox.getChildren().addAll(rbHello, rbWelcome);
vBox1.getChildren().addAll(rbHi);
content.getItems().addAll(vBox,vBox1);

primarystage.setScene(new Scene (content,300, 250));
primarystage.setTitle("Test");
primarystage.show();
```

Using SplitPane



The `getter` and `setter` methods for property values and a getter for property itself are provided in the class, but omitted in the UML diagram for brevity.

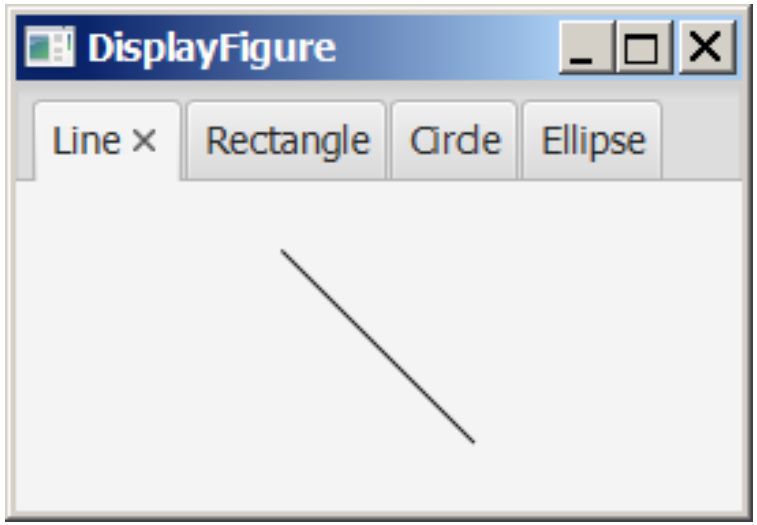
Specifies the orientation of the pane.

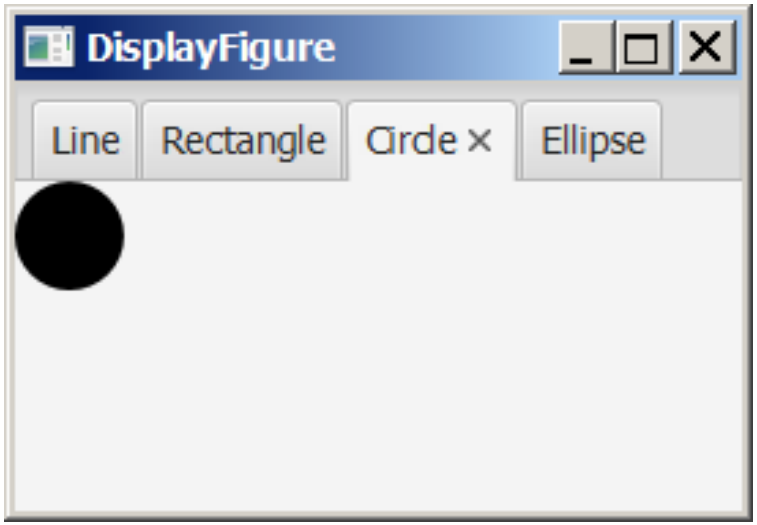
Constructs a default split pane with horizontal orientation.

Returns a list of items in the pane.

TabPane

The TabPane class can be used to display multiple panes with tabs.



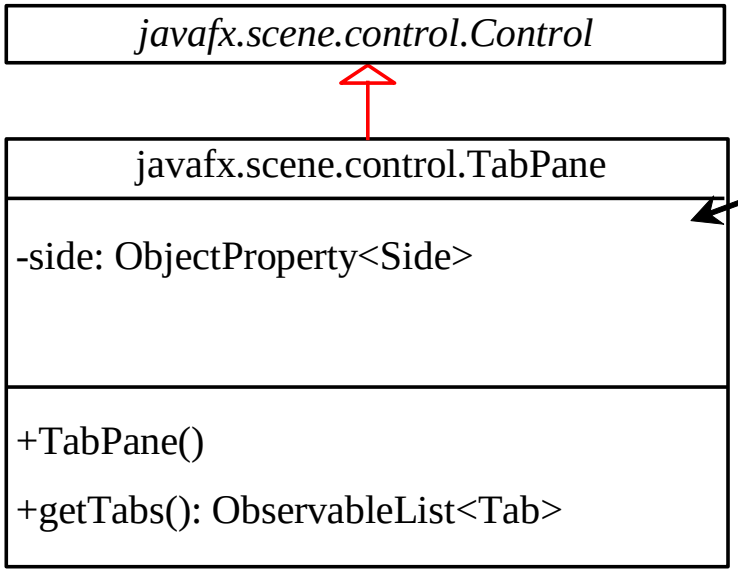


TabPane

```
TabPane tabPane = new TabPane();
Tab tab1 = new Tab("Line");
StackPane pane1 = new StackPane();
pane1.getChildren().add(new Line(10, 10, 80, 80));
tab1.setContent(pane1);
Tab tab2 = new Tab("Rectangle");
tab2.setContent(new Rectangle(10, 10, 200, 200));
Tab tab3 = new Tab("Circle");
tab3.setContent(new Circle(50, 50, 20));
Tab tab4 = new Tab("Ellipse");
tab4.setContent(new Ellipse(10, 10, 100, 80));
tabPane.getTabs().addAll(tab1, tab2, tab3, tab4);

Scene scene = new Scene(tabPane, 300, 250);
primaryStage.setTitle("DisplayFigure"); // Set the window title
primaryStage.setScene(scene); // Place the scene in the window
primaryStage.show(); // Display the window
```

The TabPane Class



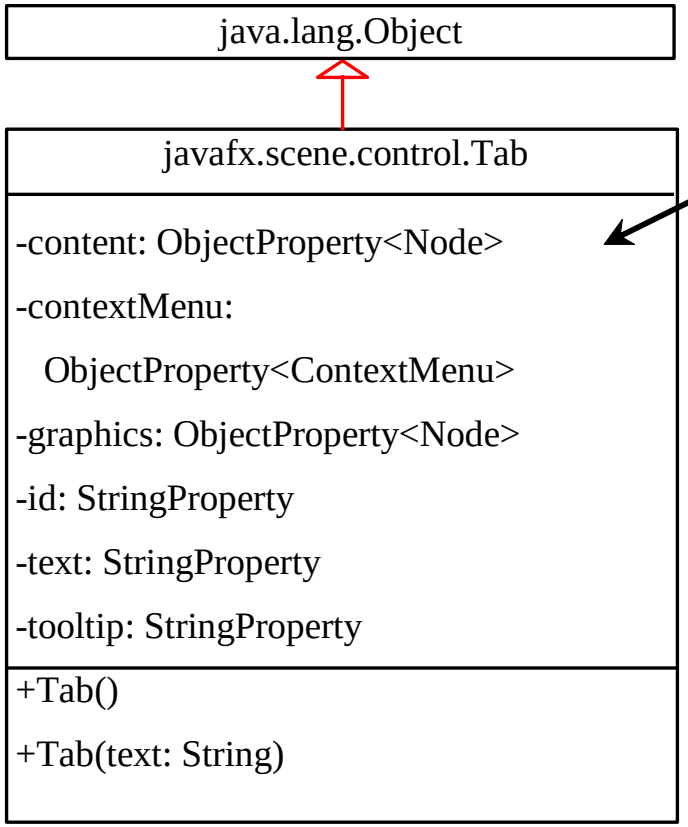
The `getter` and `setter` methods for property values and a getter for property itself are provided in the class, but omitted in the UML diagram for brevity.

The position of the tab in the tab pane. Possible values are: Side.TOP, Side.BOTTOM, Side.LEFT, and Side.RIGHT (default: Side.TOP).

Creates a default tab pane.

Returns a list of tabs in this tab pane.

The Tab Class



The `getter` and `setter` methods for property values and a getter for property itself are provided in the class, but omitted in the UML diagram for brevity.

The content associated with the tab.

The context menu associated with the tab.

The graphics in the tab.

The id for the tab.

The text shown in the tab.

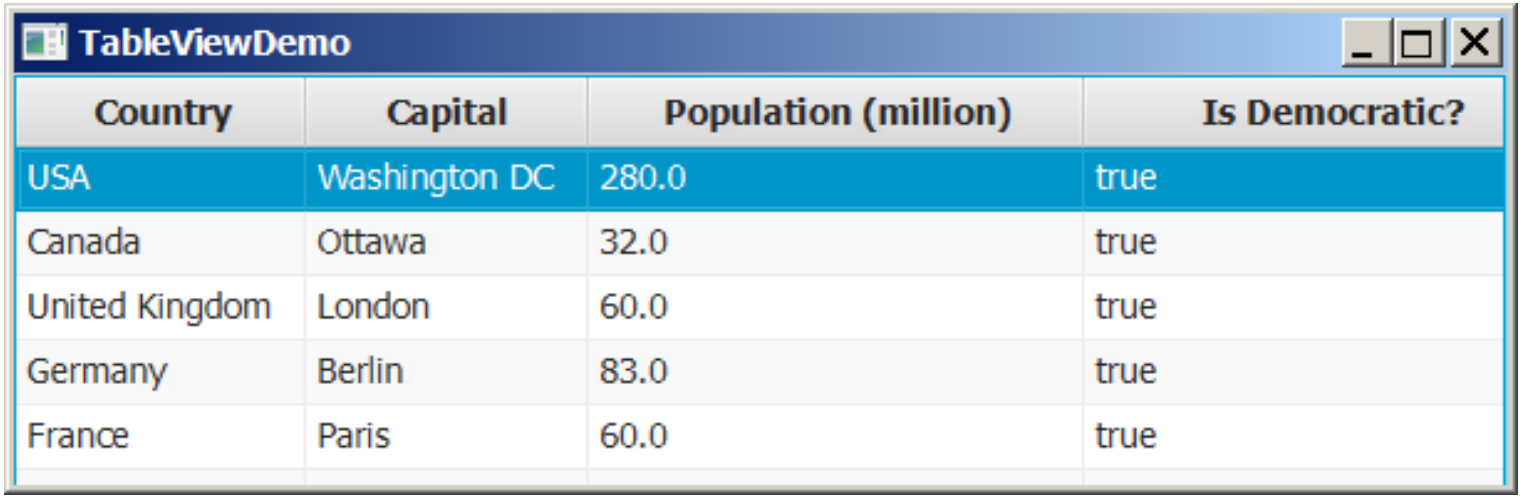
The tooltip associated with the tab.

TableView

You can display tables using the **TableView** class.

TableView, TableColumn, and TableCell are used to display and manipulate a table. TableView displays a table. TableColumn defines the columns in a table. TableCell represents a cell in the table. Creating a TableView is a multistep process.

- First, you need to create an instance of TableView and associate data with the TableView.
- Second, you need to create columns using the TableColumn class and set a column cell value factory to specify how to populate all cells within a single TableColumn



Country	Capital	Population (million)	Is Democratic?
USA	Washington DC	280.0	true
Canada	Ottawa	32.0	true
United Kingdom	London	60.0	true
Germany	Berlin	83.0	true
France	Paris	60.0	true

The TableView Class

javafx.scene.control.Control

javafx.scene.control.TableView<S>

-editable: BooleanProperty

-items:
ObjectProperty<ObservableList<S>>

-placeholder: ObjectProperty<Node>

-selectionModel: ObjectProperty<
TableViewSelectionModel<S>>

+TableView()

+TableView(items: ObservableList<S>)

The `getter` and `setter` methods for property values and a getter for property itself are provided in the class, but omitted in the UML diagram for brevity.

Specifies whether this TableView is editable. For a cell to be editable, TableView, TableColumn, and TableCell for the cell should all be true.

The data model for the TableViee.

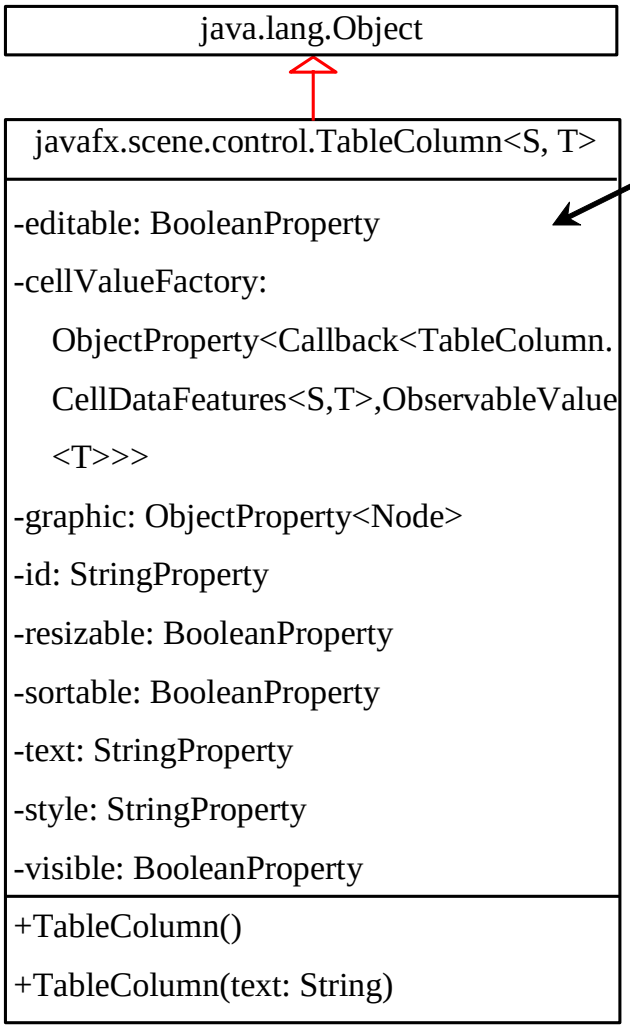
This Node is shown when table has no contents.

Specifies single or multiple selections.

Creates a default TableView with no content.

Creates a default TableView with the specified content.

The TableColumn Class



The `getter` and `setter` methods for property values and a getter for property itself are provided in the class, but omitted in the UML diagram for brevity.

Specifies whether this TableColumn allows editing.

The cell value factory to specify how to populate all cells within a single column.

The graphic for this TableColumn.

The id for this TableColumn.

Indicates whether the column is resizable.

Indicates whether the column is sortable.

Text in the table column header.

Specify the CSS style for the column.

Specify whether the column is visible (default: true).

Creates a default TableColumn.

Creates a TableView with the specified header text.

Add New Row

You can display tables using the **TableView** class.

TableViewDemo

Country	Capital	Population...	Is Democr...	Location	
				latitude	longitude
USA	Washington DC	280.0	true		
Canada	Ottawa	32.0	true		
United Kingdom	London	60.0	true		
Germany	Berlin	83.0	true		
France	Paris	60.0	true		

GUI development and JavaFX Basics

BIRZEIT UNIVERSITY