

5.6: Menus – Basics – Menu – Menu bars – MenuItem.

JavaFX Menus, MenuItem and MenuBar:

Menu is a popup menu that contains several menu items that are displayed when the user clicks a menu. The user can select a menu item after which the menu goes into a hidden state.

MenuBar is usually placed at the top of the screen which contains several menus. JavaFX MenuBar is typically an implementation of a menu bar.

Constructor of the MenuBar class are:

1. **MenuBar()**: creates a new empty menubar.
2. **MenuBar(Menu... m)**: creates a new menubar with the given set of menu.

Constructor of the Menu class are:

1. **Menu()**: creates an empty menu
2. **Menu(String s)**: creates a menu with a string as its label
3. **Menu(String s, Node n)**: Constructs a Menu and sets the display text with the specified text and sets the graphic Node to the given node.
4. **Menu(String s, Node n, MenuItem... i)**: Constructs a Menu and sets the display text with the specified text, the graphic Node to the given node, and inserts the given items into the items list.

Commonly used methods:

Method	Explanation
getItems()	returns the items of the menu
hide()	hide the menu
show()	show the menu
getMenus()	The menus to show within this MenuBar.
isUseSystemMenuBar()	Gets the value of the property useSystemMenuBar
setUseSystemMenuBar(boolean v)	Sets the value of the property useSystemMenuBar.
setOnHidden(EventHandler v)	Sets the value of the property onHidden.

setOnHiding(EventHandler v)	Sets the value of the property onHiding.
setOnShowing(EventHandler v)	Sets the value of the property onShowing.
setOnShown(EventHandler v)	Sets the value of the property onShown.

JavaFX Menu

- ✓ In the JavaFX application, in order to create a menu, menu items, and menu bar, Menu, MenuItem, and MenuBar class is used. The menu allows us to choose options among available choices in the application.
- ✓ All methods needed for this purpose are present in the javafx.scene.control.Menu class.

Example: Java program to create a menu bar and add menu to it and also add menuitems to the menu

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.Menu;
import javafx.scene.control.MenuBar;
import javafx.scene.control.MenuItem;
import javafx.scene.layout.VBox;
public class MenuUI extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception
    {
        Menu newmenu = new Menu("File");
        Menu newmenu2 = new Menu("Edit");

        MenuItem m1 = new MenuItem("Open");
        MenuItem m2 = new MenuItem("Save");
        MenuItem m3 = new MenuItem("Exit");
        MenuItem m4 = new MenuItem("Cut");
        MenuItem m5 = new MenuItem("Copy");
        MenuItem m6 = new MenuItem("Paste");
        newmenu.getItems().add(m1);
        newmenu.getItems().add(m2);
        newmenu.getItems().add(m3);
        newmenu2.getItems().add(m4);
        newmenu2.getItems().add(m5);
        newmenu2.getItems().add(m6);
    }
}
```

```
MenuBar newmb = new MenuBar();
newmb.getMenus().add(newmenu);
newmb.getMenus().add(newmenu2);
VBox box = new VBox(newmb);
Scene scene = new Scene(box,400,200);
primaryStage.setScene(scene);
primaryStage.setTitle("JavaFX Menu Example");
primaryStage.show();
}
public static void main(String[] args)
{
    Application.launch(args);
}
}
```

Output:

