**NOW FOR JavaFX**

**//MenuDemo.java**

**import javafx.application.\*;**

**import javafx.scene.\*;**

**import javafx.stage.\*;**

**import javafx.scene.layout.\*;**

**import javafx.scene.control.\*;**

**import javafx.event.\*;**

**import javafx.geometry.\*;**

**public class MenuDemo extends Application {**

**Label response;**

**public static void main(String[] args) {**

**// Start the JavaFX application by calling launch().**

**launch(args);**

**}**

**// Override the start() method.**

**public void start(Stage myStage) {**

**// Give the stage a title.**

**myStage.setTitle("Demonstrate Menus");**

**// Use a BorderPane for the root node.**

**BorderPane rootNode = new BorderPane();**

**// Create a scene.**

**Scene myScene = new Scene(rootNode, 300, 300);**

**// Set the scene on the stage.**

**myStage.setScene(myScene);**

**// Create a label that will report the selection.**

**response = new Label("Menu Demo");**

**// Create the menu bar.**

**MenuBar mb = new MenuBar();**

**// Create the File menu.**

**Menu fileMenu = new Menu("File");**

**MenuItem open = new MenuItem("Open");**

**MenuItem close = new MenuItem("Close");**

**MenuItem save = new MenuItem("Save");**

**MenuItem exit = new MenuItem("Exit");**

**fileMenu.getItems().addAll(open, close, save,**

**new SeparatorMenuItem(), exit);**

**// Add File menu to the menu bar.**

**mb.getMenus().add(fileMenu);**

**// Create the Options menu.**

**Menu optionsMenu = new Menu("Options");**

**// Create the Input Devices submenu.**

**Menu inDevicesMenu = new Menu("Input Devices");**

**MenuItem keyboard = new MenuItem("Keyboard");**

**MenuItem mouse = new MenuItem("Mouse");**

**MenuItem touchscreen = new MenuItem("Touchscreen");**

**inDevicesMenu.getItems().addAll(keyboard, mouse, touchscreen);**

**optionsMenu.getItems().add(inDevicesMenu);**

**// Create the Clock Style submenu.**

**Menu clockMenu = new Menu("Clock Style");**

**MenuItem analog = new MenuItem("Analog");**

**MenuItem digital = new MenuItem("Digital");**

**clockMenu.getItems().addAll(analog, digital);**

**optionsMenu.getItems().add(clockMenu);**

**// Add a separator.**

**optionsMenu.getItems().add(new SeparatorMenuItem());**

**// Create the Reset menu item.**

**MenuItem reset = new MenuItem("Reset");**

**optionsMenu.getItems().add(reset);**

**// Add Options menu to the menu bar.**

**mb.getMenus().add(optionsMenu);**

**// Create the Help menu.**

**Menu helpMenu = new Menu("Help");**

**MenuItem about = new MenuItem("About");**

**helpMenu.getItems().add(about);**

**// Add Help menu to the menu bar.**

**mb.getMenus().add(helpMenu);**

**// Create one event handler that will handle all menu action events.**

**EventHandler<ActionEvent> MEHandler =**

**new EventHandler<ActionEvent>() {**

**public void handle(ActionEvent ae) {**

**String name = ((MenuItem)ae.getTarget()).getText();**

**// If Exit is chosen, the program is terminated.**

**if(name.equals("Exit")) Platform.exit();**

**response.setText( name + " selected");**

**}**

**};**

**// Set action event handlers for the menu items.**

**open.setOnAction(MEHandler);**

**close.setOnAction(MEHandler);**

**save.setOnAction(MEHandler);**

**exit.setOnAction(MEHandler);**

**keyboard.setOnAction(MEHandler);**

**mouse.setOnAction(MEHandler);**

**touchscreen.setOnAction(MEHandler);**

**analog.setOnAction(MEHandler);**

**digital.setOnAction(MEHandler);**

**reset.setOnAction(MEHandler);**

**about.setOnAction(MEHandler);**

**// Add the menu bar to the top of the border pane and**

**// the response label to the center position.**

**rootNode.setTop(mb);**

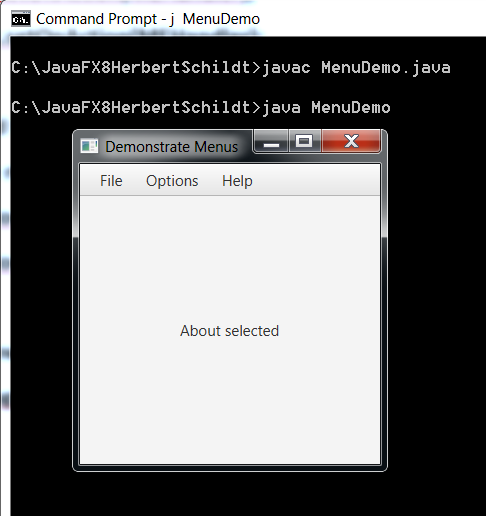
**rootNode.setCenter(response);**

**// Show the stage and its scene.**

**myStage.show();**

**}**

**}**



**// try clicking on all menu items and look at the center of the screen.**

**// you now prove that all menu items are readily accessible!!!!**

**// you now have a terrific template.**

**// DatetoStringExample1.java**

**import java.text.DateFormat;**

**import java.text.SimpleDateFormat;**

**import java.util.Date;**

**import java.util.Calendar;**

**public class DateToStringExample1 {**

**public static void main(String args[]){**

**Date date = Calendar.getInstance().getTime();**

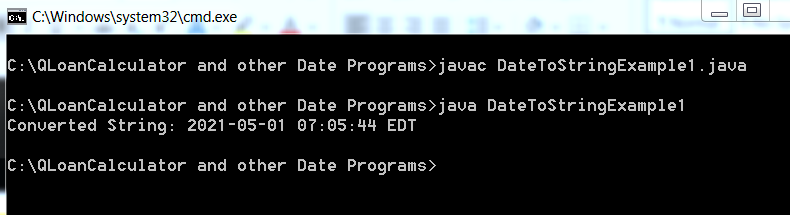
**DateFormat dateFormat = new SimpleDateFormat("yyyy-mm-dd hh:mm:ss z");**

**String strDate = dateFormat.format(date);**

**System.out.println("Converted String: " + strDate);**

**}**

**}**



**// DatetoStringExample2.java**

**import java.text.ParseException;**

**import java.text.SimpleDateFormat;**

**import java.util.Date;**

**import java.util.Locale;**

**public class DateToStringExample2 {**

**public static void main(String[] args) {**

**Date date = new Date();**

**SimpleDateFormat formatter = new SimpleDateFormat("MM/dd/yyyy");**

**String strDate = formatter.format(date);**

**System.out.println("Date Format with MM/dd/yyyy : "+strDate);**

**formatter = new SimpleDateFormat("dd-M-yyyy hh:mm:ss");**

**strDate = formatter.format(date);**

**System.out.println("Date Format with dd-M-yyyy hh:mm:ss : "+strDate);**

**formatter = new SimpleDateFormat("dd MMMM yyyy");**

**strDate = formatter.format(date);**

**System.out.println("Date Format with dd MMMM yyyy : "+strDate);**

**formatter = new SimpleDateFormat("dd MMMM yyyy zzzz");**

**strDate = formatter.format(date);**

**System.out.println("Date Format with dd MMMM yyyy zzzz : "+strDate);**

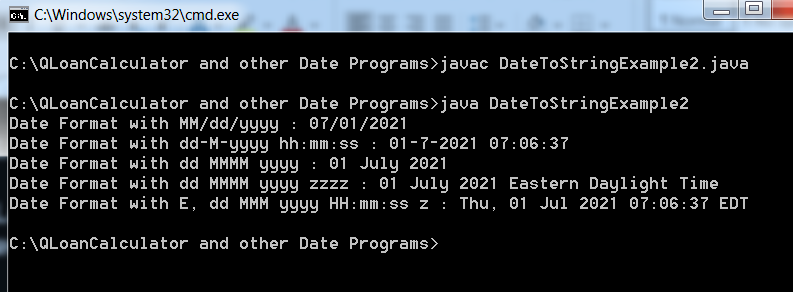
**formatter = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z");**

**strDate = formatter.format(date);**

**System.out.println("Date Format with E, dd MMM yyyy HH:mm:ss z : "+strDate);**

**}**

**}**



C’est Finis !!!!!!