# TASK DESCRIPTION

This is a GUI program for fonts testing with various font styles, font sizes including bold and italic version of fonts. Whole program is completed by following steps:

1. All the components (labels, combo box, checkbox) needed is defined first.
2. Then values and properties are set to these components.
3. Layout for top section is designed with specific properties.
4. Layout for middle section is designed with specific properties.
5. Layout for bottom section is designed with specific properties.
6. Then, event handler is defined to make changes to the main component according to different conditions and passed on to different components.
7. All the layout is then added to main pane which is added to scene.
8. This scene is added to stage to display the application.

# TASK OUTPUT

|  |  |
| --- | --- |
| **Test Data** | **Screenshot** |
| Bold: no  Italic: no  FontName: verdana  FontSize: 24 |  |
| Bold: yes  Italic: no  FontName: verdana  FontSize: 24 |  |
| Bold: yes  Italic: yes  FontName: verdana  FontSize: 24 |  |
| Bold: no  Italic: yes  FontName: calibri  FontSize: 24 |  |
| Bold: no  Italic: no  FontName: calibri  FontSize: 42 |  |

# TASK CODE

**import** javafx.geometry.Pos;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Button;

**import** javafx.scene.control.CheckBox;

**import** javafx.scene.control.ComboBox;

**import** javafx.scene.control.Label;

**import** javafx.scene.image.ImageView;

**import** javafx.scene.layout.Border;

**import** javafx.scene.layout.BorderPane;

**import** javafx.scene.layout.HBox;

**import** javafx.scene.layout.Pane;

**import** javafx.scene.text.Font;

**import** javafx.scene.text.FontPosture;

**import** javafx.scene.text.FontWeight;

**import** javafx.scene.text.Text;

**import** javafx.stage.Stage;

**import** javafx.geometry.Insets;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Observable;

**import** javax.swing.plaf.basic.BasicTreeUI.CellEditorHandler;

**import** javax.swing.text.StyledEditorKit.FontFamilyAction;

**import** javafx.application.Application;

**import** javafx.collections.FXCollections;

**import** javafx.collections.ObservableList;

**import** javafx.event.ActionEvent;

**import** javafx.event.EventHandler;

**public** **class** FontTester **extends** Application {

**public** **void** start(Stage stage)

{

Text text = **new** Text(50, 50, "Programming is Fun !"); // declaring text with specified properties

ComboBox<String> comboBoxName = **new** ComboBox<>(); // declaring combobox to store font names

ComboBox<String> comboBoxSize = **new** ComboBox<>(); // declaring combobox to store font size values

CheckBox checkBold = **new** CheckBox("Bold"); // declaring checkbox for bold fonts

CheckBox checkItalic = **new** CheckBox("Italic"); // declaring checkbox for italic fonts

Label labelName = **new** Label("Font Name"); // declaring label for font name

Label labelSize = **new** Label("Font Size"); // declaring label for font size

String[] fontSize = {"8", "12", "16", "24", "30", "36", "42", "48", "56"}; // declaring an array of all the font sizes

ObservableList<String> sizeItems = FXCollections.*observableArrayList*(fontSize); // declaring list to store font size array

comboBoxSize.getItems().addAll(sizeItems); //add font size values to combobox

comboBoxSize.setValue(sizeItems.get(3)); // setting initial value for font size

// declaring arraylist of font names

String[] fontsList = {"Arial", "Times New Roman", "Calibri", "Verdana", "Helvetica", "Sans Serif", "Arial Narrow", "Consolas", "Adobe Hebrew", "Georgia"};

ObservableList<String> fontItems = FXCollections.*observableArrayList*(fontsList); // declaring list to store font namea array

comboBoxName.setItems(fontItems); // adding font name array to combobox

comboBoxName.setValue(fontItems.get(3)); // setting initial value for font name

// setting initial value for text including font family, font size and font style

text.setFont(Font.*font*(comboBoxName.getValue(), FontWeight.***NORMAL***, FontPosture.***REGULAR***, Integer.*parseInt*(comboBoxSize.getValue())));

// designing layout for top section

HBox hBoxFontSize = **new** HBox(); // declaring HBox to store various UI objects

hBoxFontSize.getChildren().addAll(labelName, comboBoxName, labelSize, comboBoxSize); // adding labels and combobox to hbox

hBoxFontSize.setAlignment(Pos.***CENTER***); // setting hbox position

hBoxFontSize.setPadding(**new** Insets(5,5,5,5)); // setting hbox padding

labelName.setPadding(**new** Insets(5,5,5,5)); // setting label padding

comboBoxName.setPadding(**new** Insets(5,5,5,5)); // setting fontname combobox padding

labelSize.setPadding(**new** Insets(5,5,5,5)); // setting label padding

comboBoxSize.setPadding(**new** Insets(5,5,5,5)); // setting fontsize combobox padding

// designing layout for center section

HBox hBoxText = **new** HBox(); // declaring HBox to store text

hBoxText.getChildren().add(text); // adding text to hbox

hBoxText.setAlignment(Pos.***CENTER***); // setting hbox position

hBoxText.setPadding(**new** Insets(15,15,15,15)); // setting hbox padding

// designing for bottom section

HBox hBoxFontStyle = **new** HBox(); // declaring HBox to store checkboxes

hBoxFontStyle.getChildren().addAll(checkBold, checkItalic); // adding two checkboxes to hbox

hBoxFontStyle.setAlignment(Pos.***CENTER***); // setting hbox position

checkBold.setPadding(**new** Insets(5,5,5,5)); // setting checkbox bold padding

checkItalic.setPadding(**new** Insets(5,5,5,5)); // setting checkbox italic padding

// creating event handler to handle events by selecting check boxes and combobox values

EventHandler<ActionEvent> styleHandler = e ->

{

// using if else to check what are the selected values and then apply it to text to make changes

**if** (checkBold.isSelected() && checkItalic.isSelected())

{

text.setFont(Font.*font*(comboBoxName.getValue(), FontWeight.***BOLD***, FontPosture.***ITALIC***, Integer.*parseInt*(comboBoxSize.getValue())));

}

**else** **if** (checkBold.isSelected())

{

text.setFont(Font.*font*(comboBoxName.getValue(), FontWeight.***BOLD***, FontPosture.***REGULAR***, Integer.*parseInt*(comboBoxSize.getValue())));

}

**else** **if** (checkItalic.isSelected())

{

text.setFont(Font.*font*(comboBoxName.getValue(), FontWeight.***NORMAL***, FontPosture.***ITALIC***, Integer.*parseInt*(comboBoxSize.getValue())));

}

**else**

{

text.setFont(Font.*font*(comboBoxName.getValue(), FontWeight.***NORMAL***, FontPosture.***REGULAR***, Integer.*parseInt*(comboBoxSize.getValue())));

}

};

checkBold.setOnAction(styleHandler); // passing event handler to bold check box

checkItalic.setOnAction(styleHandler); // passing event handler to italic check box

comboBoxName.setOnAction(styleHandler); // passing event handler to font name combobox

comboBoxSize.setOnAction(styleHandler); // passing event handler to font size combobox

// creating border pane to store three Hbox and setting some default properties to it

BorderPane pane1 = **new** BorderPane();

pane1.setPadding(**new** Insets(5,5,5,5)); // sets paddings

pane1.setCenter(hBoxText); // sets pane to center

pane1.setTop(hBoxFontSize); // sets pane to top

pane1.setBottom(hBoxFontStyle); // sets pane to bottom

// creating scene to store border pane

Scene scene = **new** Scene(pane1, 700, 250);

stage.setTitle("Font Tester"); // sets title of the scene

stage.setScene(scene); // passing scene to main stage

stage.show(); // shows stage

}

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

{

// to launch the application

*launch*(args);

}

}