//Dodie Matthew

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class NoteTaker extends JFrame

{

// Constants for set up of the note taking area

public static final int WIDTH = 600;

public static final int HEIGHT = 300;

public static final int LINES = 13;

public static final int CHAR\_PER\_LINE = 45;

// Objects in GUI

private JTextArea theText; // Text area to take notes

private JMenuBar mBar; // Horizontal menu bar

private JPanel textPanel; // Scrolling text area panel

private JMenu notesMenu; // Vertical menu for notes

// THESE ITEMS ARE NOT YET USED

// YOU WILL BE CREATING THEM IN THIS LAB

private JMenu viewMenu; // Vertical menu for views

private JMenu lafMenu; // Vertical menu look and feel

private JMenu sbMenu; // Vertical menu for scroll bar

private JScrollPane scrolledText; // Scroll bars

// Default notes

private String note1 = "No Note 1.";

private String note2 = "No Note 2.";

/\*\*

Constructor

\*/

public NoteTaker()

{

// Create a closeable JFrame with a specific size

super("Note Taker");

setSize(WIDTH, HEIGHT);

setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);

// Get contentPane and set layout of the window

Container contentPane = getContentPane();

contentPane.setLayout(new BorderLayout());

// Creates the vertical menus

createNotes();

createViews();

// Creates horizontal menu bar and

// adds vertical menus to it

mBar = new JMenuBar();

mBar.add(notesMenu);

mBar.add(viewMenu);

// ADD THE viewMenu TO THE MENU BAR HERE

setJMenuBar(mBar);

// Creates a panel to take notes on

textPanel = new JPanel();

textPanel.setBackground(Color.blue);

theText = new JTextArea(LINES, CHAR\_PER\_LINE);

theText.setBackground(Color.white);

// CREATE A JScrollPane OBJECT HERE CALLED

// scrolledText AND PASS IN theText, THEN

// CHANGE THE LINE BELOW BY PASSING IN scrolledText

scrolledText = new JScrollPane(theText);

textPanel.add(scrolledText);

contentPane.add(textPanel, BorderLayout.CENTER);

}

/\*\*

Creates vertical menu associated with Notes

menu item on menu bar.

\*/

public void createNotes()

{

notesMenu = new JMenu("Notes");

JMenuItem item;

item = new JMenuItem("Save Note 1");

item.addActionListener(new MenuListener());

notesMenu.add(item);

item = new JMenuItem("Save Note 2");

item.addActionListener(new MenuListener());

notesMenu.add(item);

item = new JMenuItem("Open Note 1");

item.addActionListener(new MenuListener());

notesMenu.add(item);

item = new JMenuItem("Open Note 2");

item.addActionListener(new MenuListener());

notesMenu.add(item);

item = new JMenuItem("Clear");

item.addActionListener(new MenuListener());

notesMenu.add(item);

item = new JMenuItem("Exit");

item.addActionListener(new MenuListener());

notesMenu.add(item);

}

/\*\*

Creates vertical menu associated with Views

menu item on the menu bar.

\*/

public void createViews()

{

//Creat Jmenu for views tab

viewMenu = new JMenu("Views");

createLookAndFeel();

lafMenu.addActionListener(new MenuListener());

viewMenu.add(lafMenu);

createScrollBars();

sbMenu.addActionListener(new MenuListener());

viewMenu.add(sbMenu);

}

/\*\*

Creates the look and feel submenu.

\*/

public void createLookAndFeel()

{

//Creating Jmenu and JmenuItem

lafMenu = new JMenu("Look and Feel");

JMenuItem item;

item = new JMenuItem("Metal");

item.addActionListener(new MenuListener());

lafMenu.add(item);

item = new JMenuItem("Motif");

item.addActionListener(new MenuListener());

lafMenu.add(item);

item = new JMenuItem("Windows");

item.addActionListener(new MenuListener());

lafMenu.add(item);

}

/\*\*

Creates the scroll bars submenu.

\*/

public void createScrollBars()

{

//Create JmenuItem and Jmenu

sbMenu = new JMenu("Scroll Bars");

JMenuItem item;

item = new JMenuItem("Never");

item.addActionListener(new MenuListener());

sbMenu.add(item);

item = new JMenuItem("Always");

item.addActionListener(new MenuListener());

sbMenu.add(item);

item = new JMenuItem("As Needed");

item.addActionListener(new MenuListener());

sbMenu.add(item);

}

/\*\*

Private inner class that handles the Menu object's

action events.

\*/

private class MenuListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

String actionCommand = e.getActionCommand();

if (actionCommand.equals("Save Note 1"))

note1 = theText.getText();

else if (actionCommand.equals("Save Note 2"))

note2 = theText.getText();

else if (actionCommand.equals("Clear"))

theText.setText("");

else if (actionCommand.equals("Open Note 1"))

theText.setText(note1);

else if (actionCommand.equals("Open Note 2"))

theText.setText(note2);

else if (actionCommand.equals("Exit"))

System.exit(0);

//Adding action Listeners and error messages to the else if statements to catch error

else if (actionCommand.equals("Metal"))

{

try {

//set cross platform Java L&F("Metal")

UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");

}

catch (Exception except){

System.out.println("Error");

}

}

else if (actionCommand.equals("Motif"))

{

try {

//set cross platform Java L&F("Motif")

UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");

}

catch (Exception except){

System.out.println("Error");

}

}

else if (actionCommand.equals("Windows"))

{

try {

//set cross platform Java L&F("Windows")

UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

}

catch (Exception except){

System.out.println("Error");

}

}

else if (actionCommand.equals("Never")) {

scrolledText.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER);

scrolledText.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_NEVER);

SwingUtilities.updateComponentTreeUI(getContentPane());

}

else if (actionCommand.equals("Always")) {

scrolledText.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

scrolledText.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

SwingUtilities.updateComponentTreeUI(getContentPane());

}

else if (actionCommand.equals("As Needed")) {

scrolledText.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED);

scrolledText.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED);

SwingUtilities.updateComponentTreeUI(getContentPane());

}

// ADD 6 BRANCHES TO THE ELSE-IF STRUCTURE

// TO ALLOW ACTION TO BE PERFORMED FOR EACH

// MENU ITEM YOU HAVE CREATED

else

theText.setText("Error in memo interface");

}

}

/\*\*

The main method creates an instance of the

NoteTaker class which causes it to display

its window.

\*/

public static void main(String[] args)

{

NoteTaker gui = new NoteTaker();

gui.setVisible(true);

}

}

