Maddison Kiefer

Dr. Schwartz

Advanced Java Programming

11/06/2023

**Project 3-1 Creating a User Interface I**

**Source Code:**

//@author Maddison Kiefer

// Needed imports

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.time.format.DateTimeFormatter;

import java.time.LocalDateTime;

import java.io.\*;

import java.util.Random;

// Class definition for the user interface

public class UserInterface implements ActionListener {

// Declaration of member variables

JMenuItem menuItem1, menuItem2, menuItem3, menuItem4;

JMenuBar menuBar;

JTextField textField;

JMenu menu;

JFrame frame;

// Constructor for UserInterface

public UserInterface() {

// Create a new JFrame

frame = new JFrame();

// Set the layout to null for manual component placement

frame.setLayout(null);

// Create a menu bar

menuBar = new JMenuBar();

// Create a text field

textField = new JTextField();

// Create a new menu that is called 'Options'

menu = new JMenu("Options");

// Create the menu items

menuItem1 = new JMenuItem("Date & Time");

menuItem2 = new JMenuItem("Write Into File");

menuItem3 = new JMenuItem("Change Frame Color");

menuItem4 = new JMenuItem("Exit");

// Add the menu items to the menu

menu.add(menuItem1);

menu.add(menuItem2);

menu.add(menuItem3);

menu.add(menuItem4);

// Add the menu to the menu bar

menuBar.add(menu);

// Set the menu bar for the frame

frame.setJMenuBar(menuBar);

// Add action listeners for the menu items

menuItem1.addActionListener(this);

menuItem2.addActionListener(this);

menuItem3.addActionListener(this);

menuItem4.addActionListener(this);

// Set the bounds of the text field

textField.setBounds(150, 50, 150, 30);

// Add the text field to the frame

frame.add(textField);

// Set the frame to be visible

frame.setVisible(true);

// Set the size of the frame

frame.setSize(400, 200);

// Set the default close operation for the frame

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

// Action Listener methods for the menu items

@Override

public void actionPerformed(ActionEvent e) {

// Declaring a random object for generating random values

Random random = new Random();

// Check which menu item was clicked

if (e.getSource() == menuItem1) {

// Get the current date and time, format it, and display it in the text field

DateTimeFormatter dateTime = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");

LocalDateTime current = LocalDateTime.now();

textField.setText(dateTime.format(current) + " ");

}

if (e.getSource() == menuItem2) {

// Get text from the text field and write it to a file called log.txt

String string1 = textField.getText();

try {

FileWriter fileWrite = new FileWriter("log.txt");

fileWrite.write(string1);

fileWrite.close();

}catch (IOException ex) {

textField.setText("Exception is " + ex);

}

}

if (e.getSource() == menuItem3) {

// Generate a random hue value within the green range

float randomHue = random.nextFloat() \* 60 + 60;

// Convert to HSB color

Color randomGreenColor = Color.getHSBColor(randomHue / 360, 1, 1);

// Change the frame's content pane color to the random green hue

frame.getContentPane().setBackground(randomGreenColor);

}

if (e.getSource() == menuItem4) {

// Exits the application

System.exit(0);

}

}

// Main method to start the application

public static void main(String args[]) {

// Creates a new instance of UserInterface

UserInterface userInterface = new UserInterface();

}

}

**Executing the Application:**

Showing the User Interface



Showing the ‘Options’ menu



Displaying the current date and time



Printing the text field to a log.txt file



Changing the color of the frame background to a random color hue of the color green

