wap in java to draw a line , oval, rectangle and text using graphics method

import javax.swing.\*;

import java.awt.\*;

public class DrawShapes extends JPanel {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

// Draw a rectangle

g.setColor(Color.RED); // Set color for the rectangle

g.drawRect(50, 50, 200, 100); // Draw a rectangle (x, y, width, height)

// Draw an oval

g.setColor(Color.BLUE); // Set color for the oval

g.drawOval(300, 50, 200, 100); // Draw an oval inside a bounding rectangle

// Draw a line

g.setColor(Color.GREEN); // Set color for the line

g.drawLine(50, 200, 500, 200); // Draw a line (x1, y1, x2, y2)

// Draw text

g.setColor(Color.BLACK); // Set color for the text

g.drawString("This is some text!", 50, 250); // Draw text (text, x, y)

}

public static void main(String[] args) {

JFrame frame = new JFrame();

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

frame.setSize(600, 400);

frame.add(new DrawShapes());

frame.setVisible(true);

}

}

// Wap in java to create a menu using the frame

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class MenuExample extends JFrame implements ActionListener {

// Constructor to set up the frame and menu

public MenuExample() {

// Create a JFrame

setTitle("Menu Example");

setSize(400, 300);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

// Create the menu bar

JMenuBar menuBar = new JMenuBar();

// Create "File" menu and add menu items

JMenu fileMenu = new JMenu("File");

JMenuItem newItem = new JMenuItem("New");

JMenuItem openItem = new JMenuItem("Open");

JMenuItem exitItem = new JMenuItem("Exit");

// Add action listeners for the menu items

newItem.addActionListener(this);

openItem.addActionListener(this);

exitItem.addActionListener(this);

// Add menu items to the "File" menu

fileMenu.add(newItem);

fileMenu.add(openItem);

fileMenu.addSeparator(); // Add a separator line

fileMenu.add(exitItem);

// Create "Edit" menu and add menu items

JMenu editMenu = new JMenu("Edit");

JMenuItem cutItem = new JMenuItem("Cut");

JMenuItem copyItem = new JMenuItem("Copy");

JMenuItem pasteItem = new JMenuItem("Paste");

// Add action listeners for the edit items

cutItem.addActionListener(this);

copyItem.addActionListener(this);

pasteItem.addActionListener(this);

// Add menu items to the "Edit" menu

editMenu.add(cutItem);

editMenu.add(copyItem);

editMenu.add(pasteItem);

// Create "Help" menu

JMenu helpMenu = new JMenu("Help");

JMenuItem aboutItem = new JMenuItem("About");

// Add action listener for the "About" menu item

aboutItem.addActionListener(this);

// Add "About" item to the "Help" menu

helpMenu.add(aboutItem);

// Add menus to the menu bar

menuBar.add(fileMenu);

menuBar.add(editMenu);

menuBar.add(helpMenu);

// Set the menu bar for the frame

setJMenuBar(menuBar);

// Make the frame visible

setVisible(true);

}

// Handling actions for menu items

@Override

public void actionPerformed(ActionEvent e) {

String command = e.getActionCommand();

switch (command) {

case "New":

JOptionPane.showMessageDialog(this, "New File Created");

break;

case "Open":

JOptionPane.showMessageDialog(this, "File Opened");

break;

case "Exit":

System.exit(0);

break;

case "Cut":

JOptionPane.showMessageDialog(this, "Cut Action Performed");

break;

case "Copy":

JOptionPane.showMessageDialog(this, "Copy Action Performed");

break;

case "Paste":

JOptionPane.showMessageDialog(this, "Paste Action Performed");

break;

case "About":

JOptionPane.showMessageDialog(this, "This is a simple menu example");

break;

}

}

// Main method to run the program

public static void main(String[] args) {

new MenuExample(); // Create and display the menu

}

}