**Ender’s Edition:**

1. Design your biodata by using various AWT components.

import java.awt.\*;

import java.awt.event.\*;

public class P1 extends Frame {

    // Components

    private Label nameLabel, ageLabel, genderLabel, addressLabel;

    private TextField nameTextField, ageTextField, addressTextField;

    private Choice genderChoice;

    private Button submitButton;

    public P1() {

        setLayout(new GridLayout(5, 2));

        nameLabel = new Label("Name:");

        ageLabel = new Label("Age:");

        genderLabel = new Label("Gender:");

        addressLabel = new Label("Address:");

        nameTextField = new TextField();

        ageTextField = new TextField();

        addressTextField = new TextField();

        genderChoice = new Choice();

        genderChoice.add("Male");

        genderChoice.add("Female");

        genderChoice.add("Other");

        submitButton = new Button("Submit");

        add(nameLabel);

        add(nameTextField);

        add(ageLabel);

        add(ageTextField);

        add(genderLabel);

        add(genderChoice);

        add(addressLabel);

        add(addressTextField);

        add(new Label());

        add(submitButton);

        submitButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                System.out.println("Biodata Submitted!");

            }

        });

        setTitle("Biodata Form");

        setSize(300, 200);

        setVisible(true);

        addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent windowEvent) {

                System.exit(0);

            }

        });

    }

    public static void main(String[] args) {

        new P1();

    }

}

1. Design an applet/Application using List components to add names of 10 different cities.

import java.awt.\*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

public class P2 extends Frame {

P2(){

    setLayout(new FlowLayout());

List l = new List();

l.add("Mumbai");

 l.add("Hyderabad");

 l.add("Pune");

 l.add("Panjim");

l.add("Chennai");

 l.add("Ahmedabad");

 l.add("New York");

l.add("Tokyo");

l.add("Kyoto");

setSize(600,600);

setLayout(new FlowLayout());

setTitle("List of Cities");

 setVisible(true);

 add(l);

 addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent windowEvent) {

                System.exit(0);

            }

        });

setTitle("City List App");

        setSize(300, 200);

        setVisible(true);

}

public static void main(String[] args) {

P2 p = new P2();

 } }

1. WAP to use Border Layout .

import java.awt.\*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

public class Border extends Frame

{ Border()

{ setSize(500,500);

 setTitle("BORDERLAYOUT");

setLayout(new BorderLayout());

 add(new Label("CENTER",Label.CENTER), BorderLayout.CENTER);

add(new Label("NORTH",Label.CENTER), BorderLayout.NORTH);

add(new Label("SOUTH",Label.CENTER), BorderLayout.SOUTH);

add(new Label("WEST",Label.CENTER), BorderLayout.WEST);

add(new Label("EAST",Label.CENTER), BorderLayout.EAST);

setVisible(true);

addWindowListener(new WindowAdapter(){

    public void windowClosing(WindowEvent windowEvent){

    System.exit(0);

    }

    });

setTitle("Border");

        setSize(300, 200);

        setVisible(true);

}

public static void main(String[] args)

{

Border p = new Border();

} }

1. WAP which creates Menu of different colors and disable menu item for Black color.

import java.awt.\*;

public class menu extends Frame

 { menu(){ setSize(500,500);

setTitle("MENU");

setLayout(new BorderLayout());

setVisible(true);

Menu m = new Menu("Colors:");

 MenuBar m1 = new MenuBar();

MenuItem menuItem1 = new MenuItem("RED");

MenuItem menuItem2 = new MenuItem("GREEN");

MenuItem menuItem3 = new MenuItem("BLACK");

m.add(menuItem1);

m.add(menuItem2);

 m.add(menuItem3);

menuItem3.setEnabled(false);

m1.add(m);

setMenuBar(m1);

}

public static void main(String[] args) {

menu p = new menu();

} }

1. WAP to develop a frame to select the different states of India using JComboBox

import java.awt.\*;

 import javax.swing.\*;

public class combobox extends JFrame

 { combobox()

{ Container contentPane = getContentPane();

 setSize(500,500); setTitle("JComboBox");

 contentPane.setLayout(new FlowLayout());

 JComboBox states = new JComboBox();

states.addItem("Maharashtra");

states.addItem("Goa");

 states.addItem("Himachal Pradesh");

 contentPane.add(states);

setVisible(true);

 }

public static void main(String[] args)

{

combobox p = new combobox();

 } }

1. Develop a program to demonstrate the use of tree component in swing.

import javax.swing.\*;

import javax.swing.tree.DefaultMutableTreeNode;

public class JTreeExample {

    public static void main(String[] args) {

        DefaultMutableTreeNode root = new DefaultMutableTreeNode("Color");

        DefaultMutableTreeNode blackNode = new DefaultMutableTreeNode("Black");

        root.add(blackNode);

        DefaultMutableTreeNode blueNode = new DefaultMutableTreeNode("Blue");

        root.add(blueNode);

        JTree tree = new JTree(root);

        JScrollPane treeView = new JScrollPane(tree);

        JFrame frame = new JFrame("JTree Example");

        frame.add(treeView);

        frame.setSize(300, 300);

        frame.setVisible(true);

    }

}

1. Develop a program to demonstrate the use of JTable.

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class JTableExample {

    public static void main(String[] args) {

        Object[][] data = {

            {"101", "Amit", "670000"},

            {"102", "Jai", "780000"},

            {"101", "Sachin", "700000"}

        };

        String[] columnNames = {"ID", "Name", "Salary"};

        JTable table = new JTable(data, columnNames);

        JScrollPane tableScrollPane = new JScrollPane(table);

        JFrame frame = new JFrame("JTable Example");

        frame.add(tableScrollPane);

        frame.setSize(300, 300);

        frame.setVisible(true);

    }

}

1. WAP to demonstrate various mouse events using MouseListener and MouseMotionListener interface

import java.awt.\*;

import java.awt.event.\*;

public class MouseEventDemo extends Frame implements MouseListener, MouseMotionListener {

    private Label statusLabel;

    public MouseEventDemo() {

        setLayout(new FlowLayout());

        statusLabel = new Label("Move the mouse over the frame.");

        add(statusLabel);

        addMouseListener(this);

        addMouseMotionListener(this);

        setTitle("Mouse Event Demo");

        setSize(300, 200);

        setVisible(true);

        addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent windowEvent) {

                System.exit(0);

            }

        });

    }

public static void main(String[] args) {

        new MouseEventDemo();

    }

    @Override

    public void mouseClicked(MouseEvent e) {

        statusLabel.setText("Mouse Clicked at (" + e.getX() + ", " + e.getY() + ")");

    }

    @Override

    public void mousePressed(MouseEvent e) {

        statusLabel.setText("Mouse Pressed at (" + e.getX() + ", " + e.getY() + ")");

    }

    @Override

    public void mouseReleased(MouseEvent e) {

        statusLabel.setText("Mouse Released at (" + e.getX() + ", " + e.getY() + ")");

    }

    @Override

    public void mouseEntered(MouseEvent e) {

        statusLabel.setText("Mouse Entered at (" + e.getX() + ", " + e.getY() + ")");

    }

    @Override

    public void mouseExited(MouseEvent e) {

        statusLabel.setText("Mouse Exited at (" + e.getX() + ", " + e.getY() + ")");

    }

    @Override

    public void mouseDragged(MouseEvent e) {

        statusLabel.setText("Mouse Dragged at (" + e.getX() + ", " + e.getY() + ")");

    }

    @Override

    public void mouseMoved(MouseEvent e) {

        statusLabel.setText("Mouse Moved at (" + e.getX() + ", " + e.getY() + ")");

    }

}

1. WAP to demonstrate the use of JTextfield and JPasswordField using Listener interface

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class frame extends JFrame implements ActionListener {

    private JTextField usernameField;

    private JPasswordField passwordField;

    public frame() {

        setLayout(new FlowLayout());

        usernameField = new JTextField(15);

        passwordField = new JPasswordField(15);

        JButton loginButton = new JButton("Login");

        loginButton.addActionListener(this);

        add(new JLabel("Username:"));

        add(usernameField);

        add(new JLabel("Password:"));

        add(passwordField);

        add(loginButton);

        setTitle("Login Demo");

        setSize(250, 150);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLocationRelativeTo(null);  // Center the frame

        setVisible(true);

    }

    public static void main(String[] args) {

        new frame();

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        String username = usernameField.getText();

        String password = new String(passwordField.getPassword());

        System.out.println("Username: " + username);

        System.out.println("Password: " + password);

    }

}

1. WAP to demonstrate the use of WindowAdapter class

import java.awt.\*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

public class WindowAdapterDemo extends Frame {

    public WindowAdapterDemo() {

        setTitle("WindowAdapter Demo");

        setSize(300, 200);

        // Add a WindowAdapter to handle window closing event

        addWindowListener(new MyWindowAdapter());

        setVisible(true);

    }

    public static void main(String[] args) {

        new WindowAdapterDemo();

    }

    // Custom WindowAdapter class

    class MyWindowAdapter extends WindowAdapter {

        @Override

        public void windowClosing(WindowEvent e) {

            System.out.println("Window is closing");

            System.exit(0);

        }

        @Override

        public void windowOpened(WindowEvent e) {

            System.out.println("Window is opened");

        }

        // You can override other methods as needed

    }

}

1. WAP to demonstrate the use of InetAddress class and its factory methods

import java.net.InetAddress;

import java.net.UnknownHostException;

public class InetAddressDemo {

    public static void main(String[] args) {

        try {

            // Get the local host address

            InetAddress localHost = InetAddress.getLocalHost();

            System.out.println("Local Host Address: " + localHost);

            // Get the IP address of a specific host by name

            String hostName = "www.google.com";

            InetAddress googleAddress = InetAddress.getByName(hostName);

            System.out.println("Google IP Address: " + googleAddress);

            // Get all IP addresses associated with a host

            InetAddress[] allAddresses = InetAddress.getAllByName(hostName);

            System.out.println("All IP Addresses for " + hostName + ":");

            for (InetAddress address : allAddresses) {

                System.out.println(address);

            }

        } catch (UnknownHostException e) {

            e.printStackTrace();

        }

    }

}

1. WAP to demonstrate the use of URL and URLConnection class and its methods

import java.io.IOException;

import java.net.URL;

import java.util.Scanner;

public class URLConnectionDemo {

    public static void main(String[] args) {

        try {

            // Create a URL object

            URL url = new URL("https://www.example.com");

            // Open a connection to the URL

            Scanner scanner = new Scanner(url.openStream());

            // Retrieve information from the connection

            System.out.println("Content Type: " + url.openConnection().getContentType());

            System.out.println("Content Length: " + url.openConnection().getContentLength());

            System.out.println("Last Modified: " + url.openConnection().getLastModified());

            // Read and print content using Scanner

            System.out.println("\nContent:");

            while (scanner.hasNextLine()) {

                System.out.println(scanner.nextLine());

            }

            // Close the scanner

            scanner.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

1. WAP to insert and retrieve the data from database using JDBC

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class JDBCConnectionDemo {

    static final String JDBC\_URL = "jdbc:mysql://localhost:3306/productdb";

    static final String USER = "root";

    static final String PASSWORD = "";

    public static void main(String[] args) {

        Connection connection = null;

        Connection conn = null;

    try {

        // Create a connection to the MySQL database

        conn = DriverManager.getConnection("jdbc:mysql://localhost/electricity\_billing", "root", "");

            // Open a connection

            connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

            System.out.println("Connected to the database");

            // Perform database operations here...

            // Add fruits to the fruit table

            addFruits(connection);

        } catch (SQLException e) {

            e.printStackTrace();

        } catch (NumberFormatException e) {

            e.printStackTrace();

        } finally {

            // Close the connection in a finally block to ensure it always gets closed

            if (connection != null) {

                try {

                    connection.close();

                    System.out.println("Connection closed");

                } catch (SQLException e) {

                    e.printStackTrace();

                }

            }

        }

    }

    private static void addFruits(Connection connection) throws SQLException {

        // SQL statement to insert data into the fruit table

        String insertFruitSQL = "INSERT INTO fruit (fruit\_name, fruit\_qt) VALUES (?, ?)";

        // Sample data

        String[] fruitNames = {"Apple", "Banana", "Orange"};

        int[] fruitQuantities = {10, 20, 15};

        try (PreparedStatement preparedStatement = connection.prepareStatement(insertFruitSQL)) {

            for (int i = 0; i < fruitNames.length; i++) {

                // Set values for the parameters

                preparedStatement.setString(1, fruitNames[i]);

                preparedStatement.setInt(2, fruitQuantities[i]);

                // Execute the insert statement

                preparedStatement.executeUpdate();

            }

            System.out.println("Fruits added to the database");

        }

    }

}

1. WAP servlet to send username and password using HTML forms and authenticate the user

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated