**Implement the following projects which focus on different aspects of Java programming, including:**

Object-Oriented Programming principles

Exception handling

File operations

Data structures

Input validation

User interface design

**5. Temperature Converter**

Build a program that converts temperatures between Celsius and Fahrenheit with GUI using object oriented programming in Java.

Key points:

- Use JavaFX or Swing

- Handle decimal points

- Implement input validation

import javax.swing.\*;

import javax.swing.border.\*;

import java.awt.\*;

import java.text.DecimalFormat;

// Custom exception for temperature validation

class InvalidTemperatureException extends Exception {

public InvalidTemperatureException(String message) {

super(message);

}

}

// Temperature conversion logic class

class TemperatureConverter {

private static final double ABSOLUTE\_ZERO\_C = -273.15;

private static final double ABSOLUTE\_ZERO\_F = -459.67;

public static double celsiusToFahrenheit(double celsius) throws InvalidTemperatureException {

validateCelsius(celsius);

return (celsius \* 9/5) + 32;

}

public static double fahrenheitToCelsius(double fahrenheit) throws InvalidTemperatureException {

validateFahrenheit(fahrenheit);

return (fahrenheit - 32) \* 5/9;

}

private static void validateCelsius(double celsius) throws InvalidTemperatureException {

if (celsius < ABSOLUTE\_ZERO\_C) {

throw new InvalidTemperatureException(

"Temperature cannot be below absolute zero (-273.15°C)");

}

}

private static void validateFahrenheit(double fahrenheit) throws InvalidTemperatureException {

if (fahrenheit < ABSOLUTE\_ZERO\_F) {

throw new InvalidTemperatureException(

"Temperature cannot be below absolute zero (-459.67°F)");

}

}

}

// Main GUI class

public class TemperatureConverterGUI extends JFrame {

// Initialize final fields at declaration

private final JTextField inputField = new JTextField(10);

private final JTextField resultField = new JTextField(10);

private final JComboBox<String> fromUnitCombo = new JComboBox<>(new String[]{"Celsius", "Fahrenheit"});

private final JComboBox<String> toUnitCombo = new JComboBox<>(new String[]{"Fahrenheit", "Celsius"});

private final JLabel errorLabel = new JLabel(" ");

private final DecimalFormat decimalFormat = new DecimalFormat("#.##");

public TemperatureConverterGUI() {

// Set up the main frame

setTitle("Temperature Converter");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

// Create main panel with padding

JPanel mainPanel = new JPanel();

mainPanel.setLayout(new BorderLayout(10, 10));

mainPanel.setBorder(new EmptyBorder(20, 20, 20, 20));

// Create input panel

JPanel inputPanel = createInputPanel();

JPanel buttonPanel = createButtonPanel();

JPanel resultPanel = createResultPanel();

// Add panels to main panel

mainPanel.add(inputPanel, BorderLayout.NORTH);

mainPanel.add(buttonPanel, BorderLayout.CENTER);

mainPanel.add(resultPanel, BorderLayout.SOUTH);

// Add main panel to frame

add(mainPanel);

// Set up window properties

pack();

setLocationRelativeTo(null);

// Set error label properties

errorLabel.setForeground(Color.RED);

// Make result field non-editable

resultField.setEditable(false);

// Add unit combo box listener

fromUnitCombo.addActionListener(e -> {

toUnitCombo.setSelectedIndex(fromUnitCombo.getSelectedIndex() == 0 ? 0 : 1);

});

}

private JPanel createInputPanel() {

JPanel panel = new JPanel(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.insets = new Insets(5, 5, 5, 5);

// Add components to panel

gbc.gridx = 0; gbc.gridy = 0;

panel.add(new JLabel("Enter Temperature:"), gbc);

gbc.gridx = 1;

panel.add(inputField, gbc);

gbc.gridx = 2;

panel.add(fromUnitCombo, gbc);

return panel;

}

private JPanel createButtonPanel() {

JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER));

// Convert button

JButton convertButton = new JButton("Convert");

convertButton.addActionListener(e -> performConversion());

// Clear button

JButton clearButton = new JButton("Clear");

clearButton.addActionListener(e -> clearFields());

// Add buttons to panel

panel.add(convertButton);

panel.add(clearButton);

return panel;

}

private JPanel createResultPanel() {

JPanel panel = new JPanel(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.insets = new Insets(5, 5, 5, 5);

// Add components to panel

gbc.gridx = 0; gbc.gridy = 0;

panel.add(new JLabel("Result:"), gbc);

gbc.gridx = 1;

panel.add(resultField, gbc);

gbc.gridx = 2;

panel.add(toUnitCombo, gbc);

gbc.gridx = 0; gbc.gridy = 1;

gbc.gridwidth = 3;

panel.add(errorLabel, gbc);

return panel;

}

// Rest of the methods remain the same

private void performConversion() {

try {

// Clear previous error message

errorLabel.setText(" ");

// Validate input

String input = inputField.getText().trim();

if (input.isEmpty()) {

throw new InvalidTemperatureException("Please enter a temperature value");

}

// Parse input

double temperature = Double.parseDouble(input);

double result;

// Perform conversion based on selected units

if (fromUnitCombo.getSelectedItem().equals("Celsius")) {

result = TemperatureConverter.celsiusToFahrenheit(temperature);

resultField.setText(decimalFormat.format(result) + " °F");

} else {

result = TemperatureConverter.fahrenheitToCelsius(temperature);

resultField.setText(decimalFormat.format(result) + " °C");

}

} catch (NumberFormatException e) {

errorLabel.setText("Please enter a valid number");

resultField.setText("");

} catch (InvalidTemperatureException e) {

errorLabel.setText(e.getMessage());

resultField.setText("");

}

}

private void clearFields() {

inputField.setText("");

resultField.setText("");

errorLabel.setText(" ");

}

public static void main(String[] args) {

try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

} catch (Exception e) {

e.printStackTrace();

}

SwingUtilities.invokeLater(() -> {

new TemperatureConverterGUI().setVisible(true);

});

}

}

o/p

