

Presented by: **Maryam Siddiqui**

**www.linkedin.com/in/maryam-siddiqui-ghani**

Task: **Temperature Conversion**

**Project Report: Temperature Converter GUI**

**Introduction**

The "Temperature Converter GUI" is a Java application that provides a user-friendly graphical interface to convert temperatures between Celsius, Fahrenheit, and Kelvin scales. This project report presents an in-depth analysis of the features, functions, and design aspects of the GUI temperature converter application.

**Features**

Certainly, here are the features of the "Temperature Converter GUI" project:

* User-Friendly Interface: The graphical user interface (GUI) is designed to be intuitive and easy to use, enabling users to quickly and accurately perform temperature conversions.
* Multiple Temperature Scales: The application supports conversions between three common temperature scales: Celsius, Fahrenheit, and Kelvin. Users can choose both the source and target scales.
* Real-Time Conversion: As users input temperature values and select scales, the application performs the conversion calculations and displays the result in real time.
* Clear and Reset: The "Clear" button allows users to reset both input and output fields, making it convenient to perform multiple conversions without restarting the application.
* Graceful Exit: The "Exit" button provides a way for users to close the application gracefully when they are done using it.
* Aesthetic Design: The GUI incorporates a visually appealing design, featuring a calming background color and well-chosen fonts for improved readability.
* Interactive Buttons: The "Convert," "Clear," and "Exit" buttons provide interactive functionality, enhancing user engagement and control over the application.
* Responsive Feedback: User actions, such as button clicks, result in immediate feedback, ensuring a smooth and responsive user experience.
* Conversion Accuracy: The application employs accurate conversion formulas for temperature scales, ensuring precise and reliable results.
* Flexible User Input: Users can enter temperature values using text fields, offering the flexibility to input a wide range of numeric values.
* Choice of Source and Target Scales: Combo boxes allow users to easily select the source and target temperature scales from dropdown menus.
* Real-Time Display: The converted temperature value is displayed in the output text field as soon as the conversion is performed.
* Ease of Use: The application requires minimal technical knowledge, making it accessible to users of varying skill levels.
* Educational Tool: The project can serve as an educational tool for learning about temperature conversion and GUI development in Java.
* Robust Handling: The application incorporates event handling and error-checking mechanisms to ensure smooth operation and handle potential user input errors.
* Dynamic Labels: Labels dynamically change to provide context and instructions based on user actions and selections.
* Efficient Code Structure: The code is organized into well-defined functions, enhancing readability, maintainability, and scalability.
* Modular Design: The application is modular in design, making it easier to modify and extend with additional features or improvements.
* Themed Interface: The application follows a consistent theme with a visually appealing color scheme and layout.
* Cross-Platform Compatibility: The application can be run on various operating systems that support Java, ensuring broad accessibility.
* Overall, the "Temperature Converter GUI" project offers a comprehensive and user-centric solution for temperature conversion, combining functionality, aesthetics, and ease of use to create an effective and engaging tool.

**User Interface Design**

The application features a visually appealing and intuitive user interface, designed using the Swing framework. The interface includes the following components:

* **Labels:** Clearly labeled fields and buttons provide user guidance and enhance usability.
* **Text Fields:** Two text fields allow users to input temperature values and view the converted result.
* **Combo Boxes:** Dropdown menus let users select the original and target temperature scales (Celsius, Fahrenheit, or Kelvin).
* **Buttons:** The "Convert," "Clear," and "Exit" buttons offer interactive functionality to perform conversions, reset fields, and exit the application.

**Conversion Logic**

The core functionality of the application is its temperature conversion logic. The program enables users to convert temperatures between different scales by performing the necessary calculations based on user input.

* The program supports conversions between Celsius, Fahrenheit, and Kelvin scales using appropriate conversion formulas.
* Users can select the original and target scales using the combo boxes, and the application computes and displays the converted value in real-time.

**User Interaction**

The application offers various interactive features to enhance the user experience:

* **Convert Button:** Initiates the temperature conversion process when clicked. The converted value is displayed in the target text field.
* **Clear Button:** Resets both input and output fields to their default states, allowing users to perform new conversions easily.
* **Exit Button:** Closes the application gracefully when clicked, ensuring a seamless user experience.

**Aesthetic Appeal:**

The GUI design incorporates pleasing aesthetics and visual elements:

* **Background Color:** A calming background color (blue-green) enhances the visual appeal and maintains a consistent theme throughout the interface.
* **Fonts:** The choice of fonts, sizes, and styles for labels and text fields ensures readability and a professional look.

**Functions and Implementation**

**Temperature Conversion Logic:**

The `jButton1ActionPerformed` function is the event handler for the "Convert" button. It implements the conversion logic based on the selected scales:

* It retrieves the selected scales from the combo boxes.
* It performs the appropriate temperature conversion calculations based on the selected scales.
* The converted value is displayed in the target text field.

**Clear Functionality**

The `jButton2ActionPerformed` function is the event handler for the "Clear" button. When clicked, this function resets both input and output fields, allowing users to perform new conversions without any confusion.

**Exit Functionality**

The `jButton3ActionPerformed` function is the event handler for the "Exit" button. Clicking this button gracefully closes the application.

**Conclusion**

The "Temperature Converter GUI" project demonstrates the power of Java's Swing framework to create user-friendly and functional graphical interfaces. By offering a clear and intuitive design, interactive features, and accurate temperature conversion, this application provides a valuable tool for users to quickly and efficiently convert temperatures across different scales. The project report has highlighted the application's features, functions, and design principles, showcasing its effectiveness in addressing the task of temperature conversion while prioritizing user experience and aesthetic appeal.

**Acknowledgments**

Maryam Siddiqui as part of an internship task successfully completed this project for Info Aidtech Grateful acknowledgment is extended to mentors and guides who provided valuable support throughout the development process.

**Contact Information**

For inquiries or feedback related to this project, please contact:

Maryam Siddiqui

Note: This project report template serves as a starting point and can be customized further to match your specific project's details and requirements.

Feel free to expand and tailor this project report template according to your project's specifics.

**Source code:**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

package temprature;

/\*\*

\*

\* @author Arooj

\*/

public class Interface extends javax.swing.JFrame {

/\*\*

\* Creates new form Interface

\*/

public Interface() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jComboBox1 = new javax.swing.JComboBox<>();

jComboBox2 = new javax.swing.JComboBox<>();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Temperature Conversion");

setResizable(false);

jPanel1.setBackground(new java.awt.Color(0, 102, 102));

jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setText("Temperature Conversion");

jLabel2.setForeground(new java.awt.Color(255, 255, 255));

jLabel2.setText("Created By Maryam Siddiqui");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap(51, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 316, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addComponent(jLabel2)

.addContainerGap())))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(16, 16, 16)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel2))

);

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Celsius", "Fahrenheit", "Kelvin" }));

jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Celsius", "Fahrenheit", "Kelvin", " " }));

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jTextField2.setEditable(false);

jButton1.setText("Convert");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Clear");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Exit");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jButton3, javax.swing.GroupLayout.PREFERRED\_SIZE, 91, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jComboBox2, 0, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jComboBox1, 0, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(69, 69, 69)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26, Short.MAX\_VALUE)

.addComponent(jButton2))

.addComponent(jTextField1)

.addComponent(jTextField2))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jButton3, javax.swing.GroupLayout.DEFAULT\_SIZE, 49, Short.MAX\_VALUE)

.addGap(16, 16, 16))

);

pack();

}// </editor-fold>

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jTextField1.setText("");

jTextField2.setText("");

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

System.exit(0);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String box1 = (String)jComboBox1.getSelectedItem();

String box2 = (String)jComboBox2.getSelectedItem();

if(box1.equals("Celsius")&& box2.equals("Fahrenheit"))

{

double c =Double.parseDouble(jTextField1.getText());

double f =(double) (c\* 1.8 + 32);

jTextField2.setText(String.valueOf(f));

}

if(box1.equals("Celsius")&& box2.equals("Kelvin"))

{

double c =Double.parseDouble(jTextField1.getText());

double k =(double) (c + 273.15);

jTextField2.setText(String.valueOf(k));

}

if(box1.equals("Celsius")&& box2.equals("Celsius"))

{

double c =Double.parseDouble(jTextField1.getText());

jTextField2.setText(String.valueOf(c));

}

if(box1.equals("Fahrenheit") && box2.equals("Celsius"))

{

double f = Double.parseDouble(jTextField1.getText());

double c = (double)((f - 32)\*5/9);

jTextField2.setText(String.valueOf(c));

}

else if(box1.equals("Fahrenheit") && box2.equals("Kelvin"))

{

double f = Double.parseDouble(jTextField1.getText());

double k = (double)((f - 32)\*5/9 + 273.15);

jTextField2.setText(String.valueOf(k));

}

else if(box1.equals("Fahrenheit") && box2.equals("Fahrenheit"))

{

double f = Double.parseDouble(jTextField1.getText());

jTextField2.setText(String.valueOf(f));

}

//------------------------------------------------------------------------------

if(box1.equals("Kelvin") && box2.equals("Celsius"))

{

double k = Double.parseDouble(jTextField1.getText());

double c = (double)(0 - 273.15);

jTextField2.setText(String.valueOf(c));

}

else if(box1.equals("Kelvin") && box2.equals("Fahrenheit"))

{

double k = Double.parseDouble(jTextField1.getText());

double f = (double)(0 - 273.15) \* 9/5 +32;

jTextField2.setText(String.valueOf(f));

}

else if(box1.equals("Kelvin") && box2.equals("Kelvin"))

{

double k = Double.parseDouble(jTextField1.getText());

jTextField2.setText(String.valueOf(k));

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Interface().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JComboBox<String> jComboBox1;

private javax.swing.JComboBox<String> jComboBox2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JPanel jPanel1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

// End of variables declaration

}

**Output:**

