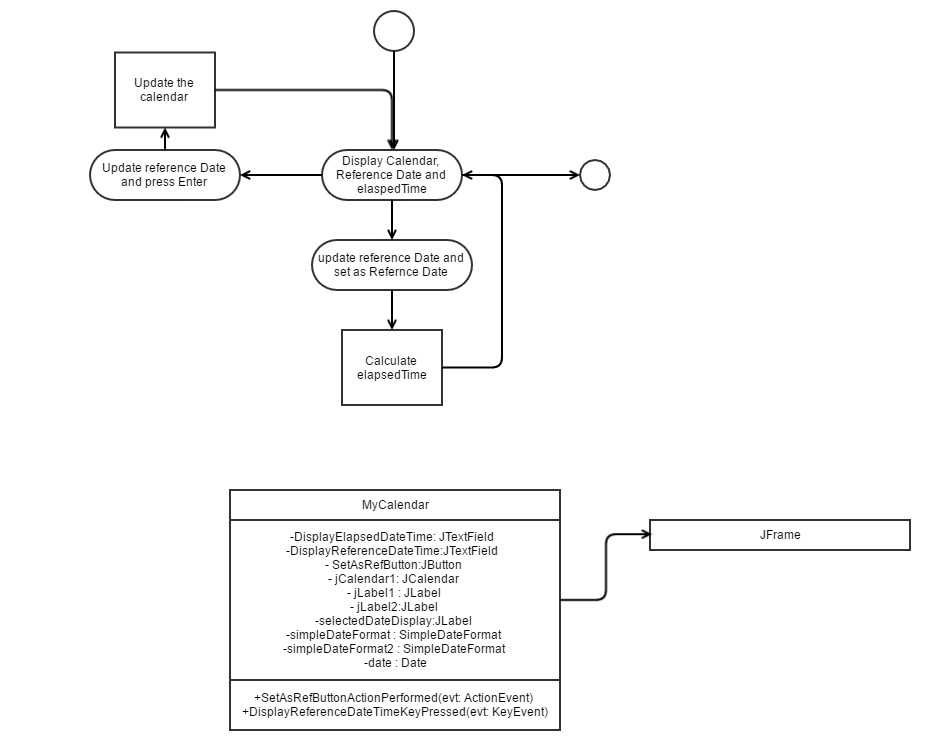
Assignment 2

This is a simple calendar application.

Design



Implementation

import java.awt.Color;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JOptionPane;

public class MyCalendar extends javax.swing.JFrame {

private SimpleDateFormat simpleDateFormat = new SimpleDateFormat("MMMM dd, yyyy: (hh.mm.ss)");

private SimpleDateFormat simpleDateFormat2 = new SimpleDateFormat("MMMM dd, yyyy");

private Date date = new Date(System.currentTimeMillis());

/\*\* Creates new form MyCalendar \*/

public MyCalendar() {

initComponents();

//get current date and display on the Text

DisplayReferenceDateTime.setText(simpleDateFormat.format(date));

jCalendar1.remove(0);

jCalendar1.setDate(date);

selectedDateDisplay.setText(simpleDateFormat2.format(date));

}

/\*\* 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() {

jCalendar1 = new com.toedter.calendar.JCalendar();

SetAsRefButton = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

DisplayReferenceDateTime = new javax.swing.JTextField();

DisplayElapsedDateTime = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

selectedDateDisplay = new javax.swing.JLabel();

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

jCalendar1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

jCalendar1.setDecorationBackgroundVisible(false);

jCalendar1.setWeekOfYearVisible(false);

SetAsRefButton.setText("Set as Ref");

SetAsRefButton.setToolTipText("Click to set the current Displayed Date as Reference Date");

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

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

SetAsRefButtonActionPerformed(evt);

}

});

jLabel1.setText("Date");

DisplayReferenceDateTime.setToolTipText("This is the display Date.");

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

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

DisplayReferenceDateTimeActionPerformed(evt);

}

});

DisplayReferenceDateTime.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyPressed(java.awt.event.KeyEvent evt) {

DisplayReferenceDateTimeKeyPressed(evt);

}

});

DisplayElapsedDateTime.setHorizontalAlignment(javax.swing.JTextField.CENTER);

DisplayElapsedDateTime.setText("00 00 00: (00.00.00)");

DisplayElapsedDateTime.setEnabled(false);

jLabel2.setText("Time from Reference date");

selectedDateDisplay.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

selectedDateDisplay.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

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

.addContainerGap()

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

.addComponent(selectedDateDisplay, javax.swing.GroupLayout.DEFAULT\_SIZE, 337, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addComponent(jLabel1)

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

.addComponent(DisplayReferenceDateTime, javax.swing.GroupLayout.DEFAULT\_SIZE, 221, Short.MAX\_VALUE)

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

.addComponent(SetAsRefButton))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(18, 18, 18)

.addComponent(DisplayElapsedDateTime, javax.swing.GroupLayout.DEFAULT\_SIZE, 194, Short.MAX\_VALUE))

.addComponent(jCalendar1, javax.swing.GroupLayout.DEFAULT\_SIZE, 337, Short.MAX\_VALUE))

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(jLabel1)

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

.addComponent(SetAsRefButton))

.addGap(4, 4, 4)

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

.addComponent(jLabel2)

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

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

.addComponent(selectedDateDisplay)

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

.addComponent(jCalendar1, javax.swing.GroupLayout.DEFAULT\_SIZE, 222, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

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

//Modify the Current Date and time to your prefered Date and time.

String userDateTime = DisplayReferenceDateTime.getText();

Date convertedUserDateTime = null;

try {

convertedUserDateTime = simpleDateFormat.parse(userDateTime);

if(date.before(convertedUserDateTime)){

long dateTimeLong = convertedUserDateTime.getTime() - date.getTime();

long secondsInMilli = 1000;

long minutesInMilli = secondsInMilli \* 60;

long hoursInMilli = minutesInMilli \* 60;

long elapsedHours = (dateTimeLong / hoursInMilli);

dateTimeLong = dateTimeLong % hoursInMilli;

long elapsedMinutes = (dateTimeLong / minutesInMilli);

dateTimeLong = dateTimeLong % minutesInMilli;

long elapsedSeconds = (dateTimeLong / secondsInMilli);

String dateTime = elapsedHours+":"+elapsedMinutes+":"+elapsedSeconds;

DisplayElapsedDateTime.setText(dateTime);

}

else

{

long dateTimeLong = date.getTime() - convertedUserDateTime.getTime();

int secondsInMilli = 1000;

int minutesInMilli = secondsInMilli \* 60;

int hoursInMilli = minutesInMilli \* 60;

int elapsedHours = (int) (dateTimeLong / hoursInMilli);

dateTimeLong = dateTimeLong % hoursInMilli;

int elapsedMinutes = (int) (dateTimeLong / minutesInMilli);

dateTimeLong = dateTimeLong % minutesInMilli;

int elapsedSeconds = (int) (dateTimeLong / secondsInMilli);

String dateTime = (elapsedHours-12)+":"+elapsedMinutes+":"+elapsedSeconds;

dateTime = elapsedHours-12 == 0 ? dateTime : "- "+dateTime;

DisplayElapsedDateTime.setText(dateTime);

}

// JOptionPane.showMessageDialog(this, simpleDateFormat.format(convertedUserDateTime));

DisplayReferenceDateTime.setText(simpleDateFormat.format(convertedUserDateTime));

jCalendar1.setDate(simpleDateFormat.parse(DisplayReferenceDateTime.getText()));

selectedDateDisplay.setText(simpleDateFormat2.format(simpleDateFormat2.parse(DisplayReferenceDateTime.getText().split("\\:")[0])));

} catch (ParseException ex) {

JOptionPane.showMessageDialog(this, "Invalid date and time format. \n Correct Date Time format is M d, yyyy (hh:mm:ss)");

}

}

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

}

private void DisplayReferenceDateTimeKeyPressed(java.awt.event.KeyEvent evt) {

if(evt.getKeyCode() == 10){

try{

jCalendar1.setDate(simpleDateFormat.parse(DisplayReferenceDateTime.getText()));

selectedDateDisplay.setText(simpleDateFormat2.format(simpleDateFormat2.parse(DisplayReferenceDateTime.getText().split("\\:")[0])));

jCalendar1.getDayChooser().setBackground(Color.red);

}catch(Exception ex){

System.out.print(ex.getMessage());

}

}

}

/\*\*

\* @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(MyCalendar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MyCalendar.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 MyCalendar().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JTextField DisplayElapsedDateTime;

private javax.swing.JTextField DisplayReferenceDateTime;

private javax.swing.JButton SetAsRefButton;

private com.toedter.calendar.JCalendar jCalendar1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel selectedDateDisplay;

// End of variables declaration

}

Output (Screenshot)

