

```

/*      Program illustrates
        1) Date package
        2) Gregorian calendar package
*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.event.*;
import java.text.*;
import java.util.*;

public class DME7Frame extends JFrame implements ActionListener,
    ListSelectionListener
{
    int currentYear, currentMonth, currentDay, month;
    int yearValue = 0;
    GregorianCalendar dateTodayGreg;

    private JMenuItem actionExitMenuItem, actionConfirmMenuItem,
actionValidateMenuItem,
        actionCalculateMenuItem, aboutMenuItem;

    private JButton confirmButton, validateButton, calculateButton, exitButton;
    private JCheckBox typeCommercial, typeResidential, typeGeneral;
    private JRadioButton year2002, year2003, year2004, yearHidden,
        serviceType1, serviceType2, serviceTypeHidden;
    private JList monthList;

    private static String[] monthName = {"January", "February", "March", "April",
        "May", "June", "July", "August", "September", "October",
        "November", "December"};

    private JLabel customerNumberLabel, customerNameLabel, customerTypeLabel,
        serviceTypeLabel, dayLabel, startMeterReadingLabel,
        endMeterReadingLabel, serviceReadingSpacerLabel;

    private JTextField customerNumberField, customerNameField, customerTypeField,
        serviceTypeField, monthField, dayField, yearField,
startMeterReadingField,
        endMeterReadingField, monthNameField, controlField ;

    private ImageIcon IconImage2002Off, IconImage2003Off, IconImage2004Off,
        IconImage2002On, IconImage2003On, IconImage2004On;

    private String clientName, serviceName;
    private int serviceMaxDay;;

    DME5BCustomer DMECustomer = new DME5BCustomer();

    public DME7Frame()
    {
        dateTodayGreg = new GregorianCalendar();
        Date dateToday = new Date();
        DateFormat stdDateFormat = DateFormat.getDateInstance(DateFormat.FULL);
        String dateTodayString = stdDateFormat.format(dateToday);
        Date meterReadingDate = null;

```

```

String meterReadingDateString = null;
currentYear = dateTodayGreg.get(Calendar.YEAR);
currentMonth = dateTodayGreg.get(Calendar.MONTH) + 1;
currentDay = dateTodayGreg.get(Calendar.DAY_OF_MONTH);

setTitle("Denton Municipal Electric (Date: " + dateTodayString + ")");
Toolkit tk = Toolkit.getDefaultToolkit();
ImageIcon LogoIconImage = new ImageIcon("cherry.GIF");
Image LogoIcon = LogoIconImage.getImage();
setIconImage(LogoIcon);
ImageIcon ValidIconImage = new ImageIcon("validation.GIF");
Image ValidIcon = ValidIconImage.getImage();

Dimension d = tk.getScreenSize();
int width = 500, height = 460;
setBounds((d.width - width)/2, (d.height - height)/2, width, height);
setResizable(true);
addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        setVisible(false);
        System.exit(0);
    }
});

Container contentPane = getContentPane();

JMenu fileMenu = new JMenu("File");
actionExitMenuItem = new JMenuItem("Exit", 'x');
fileMenu.setMnemonic('f');
fileMenu.add(actionExitMenuItem);

JMenu actionMenu = new JMenu("Action");
actionConfirmMenuItem = new JMenuItem("Confirm Customer", 'c');
actionValidateMenuItem = new JMenuItem("Validate New Data", 'v');
actionCalculateMenuItem = new JMenuItem("Calculate KWH", 'k');
actionMenu.setMnemonic('a');
actionMenu.add(actionConfirmMenuItem);
actionMenu.add(actionValidateMenuItem);
actionMenu.addSeparator();
actionMenu.add(actionCalculateMenuItem);

aboutMenuItem = new JMenuItem("About", 't');

actionExitMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_E,
    Event.ALT_MASK));

actionConfirmMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C,
    Event.ALT_MASK));

actionCalculateMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_K,
    Event.ALT_MASK));

JMenuBar menuBar = new JMenuBar();
menuBar.add(fileMenu);
menuBar.add(actionMenu);

```

```

    menuBar.add(aboutMenuItem);
    setJMenuBar(menuBar);

customerNumberLabel = new JLabel("Customer Number: ");
customerNameLabel = new JLabel("Customer Name: ");

JPanel customerLabelPanel = new JPanel();
customerLabelPanel.setLayout(new BorderLayout());
customerLabelPanel.add(customerNumberLabel, BorderLayout.NORTH);
customerLabelPanel.add(customerNameLabel, BorderLayout.SOUTH);

customerNumberField = new JTextField(6);
customerNameField = new JTextField("none selected", 30);
JPanel customerDataPanel = new JPanel();
customerDataPanel.setLayout(new BorderLayout());
customerDataPanel.add(customerNumberField, BorderLayout.NORTH);
customerDataPanel.add(customerNameField, BorderLayout.CENTER);

customerTypeLabel = new JLabel("Customer Type: ");
typeCommercial = new JCheckBox("Commercial");
typeResidential = new JCheckBox("Residential");
typeGeneral = new JCheckBox("General");
JPanel checkCustomerTypePanel = new JPanel();
checkCustomerTypePanel.setLayout(new BorderLayout());
checkCustomerTypePanel.add(typeCommercial, BorderLayout.NORTH);
checkCustomerTypePanel.add(typeResidential, BorderLayout.CENTER);
checkCustomerTypePanel.add(typeGeneral, BorderLayout.SOUTH);

JPanel customerTypeDataPanel = new JPanel();
customerTypeDataPanel.setLayout(new BorderLayout());
customerTypeDataPanel.add(customerTypeLabel, BorderLayout.NORTH);
customerTypeDataPanel.add(checkCustomerTypePanel, BorderLayout.SOUTH);

serviceTypeLabel = new JLabel ("Service Type: ");
serviceType1 = new JRadioButton ("Type 1");
serviceType2 = new JRadioButton ("Type 2");
serviceTypeHidden = new JRadioButton("", true);
ButtonGroup serviceTypeGroup = new ButtonGroup();
serviceTypeGroup.add(serviceType1);
serviceTypeGroup.add(serviceType2);
JPanel serviceTypePanel = new JPanel();
serviceTypePanel.setLayout(new BorderLayout());
serviceTypePanel.add(serviceType1, BorderLayout.NORTH);
serviceTypePanel.add(serviceType2, BorderLayout.CENTER);

    JPanel meterReadingLabelPanel = new JPanel();
startMeterReadingLabel = new JLabel("Starting Meter Reading: ");
endMeterReadingLabel = new JLabel("Ending Meter Reading: ");
meterReadingLabelPanel.setLayout(new BorderLayout());
meterReadingLabelPanel.add(startMeterReadingLabel, BorderLayout.NORTH);
meterReadingLabelPanel.add(endMeterReadingLabel, BorderLayout.SOUTH);

    JPanel meterReadingDataPanel = new JPanel();
startMeterReadingField = new JTextField(5);
endMeterReadingField = new JTextField(5);
meterReadingDataPanel.setLayout(new BorderLayout());
meterReadingDataPanel.add(startMeterReadingField, BorderLayout.NORTH);

```

```

meterReadingDataPanel.add(endMeterReadingField, BorderLayout.SOUTH);

JPanel serviceReadingLabelPanel = new JPanel();
serviceReadingLabelPanel.setLayout(new BorderLayout());
serviceReadingLabelPanel.add(serviceTypeLabel, BorderLayout.NORTH);
serviceReadingSpacerLabel = new JLabel("----- ");
serviceReadingLabelPanel.add(serviceReadingSpacerLabel,
BorderLayout.CENTER);
serviceReadingLabelPanel.add(meterReadingLabelPanel, BorderLayout.SOUTH);

JPanel serviceReadingDataPanel = new JPanel();
serviceReadingDataPanel.setLayout(new BorderLayout());
serviceReadingDataPanel.add(serviceTypePanel, BorderLayout.NORTH);
serviceReadingDataPanel.add(meterReadingDataPanel, BorderLayout.SOUTH);

JPanel dayPanel = new JPanel();
dayPanel.setLayout(new BorderLayout());
dayLabel = new JLabel("Day: ");
dayField = new JTextField(2);
dayPanel.add(dayLabel, BorderLayout.NORTH);
dayPanel.add(dayField, BorderLayout.SOUTH);

JPanel monthListPanel = new JPanel();
monthList = new JList(monthName);
monthList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
monthList.setVisibleRowCount(4);
JScrollPane monthListScroll = new JScrollPane(monthList);
Border monthTitleBorder = BorderFactory.createTitledBorder("Month");
monthListScroll.setBorder(monthTitleBorder);

IconImage2002Off = new ImageIcon("2002-Off.GIF");
IconImage2003Off = new ImageIcon("2003-Off.GIF");
IconImage2004Off = new ImageIcon("2004-Off.GIF");
IconImage2002On = new ImageIcon("2002-On.GIF");
IconImage2003On = new ImageIcon("2003-On.GIF");
IconImage2004On = new ImageIcon("2004-On.GIF");
year2002 = new JRadioButton(" 2002", IconImage2002Off);
year2003 = new JRadioButton(" 2003", IconImage2003Off);
year2004 = new JRadioButton(" 2004", IconImage2004Off);
yearHidden = new JRadioButton("", true);
ButtonGroup yearGroup = new ButtonGroup();
yearGroup.add(year2002);
yearGroup.add(year2003);
yearGroup.add(year2004);
JPanel radioButtonPanel = new JPanel();
radioButtonPanel.setLayout(new BorderLayout());
radioButtonPanel.add(year2002, BorderLayout.NORTH);
radioButtonPanel.add(year2003, BorderLayout.CENTER);
radioButtonPanel.add(year2004, BorderLayout.SOUTH);
Border yearTitleBorder = BorderFactory.createTitledBorder("Year");
radioButtonPanel.setBorder(yearTitleBorder);

JPanel datePanel = new JPanel();
datePanel.setLayout(new BorderLayout());
datePanel.add(dayPanel, BorderLayout.NORTH);
datePanel.add(monthListScroll, BorderLayout.SOUTH);

```

```
controlField = new JTextField
    ("Enter the customer number and click the 'Confirm Customer'
Button..");
```

```
JPanel controlPanel = new JPanel();
confirmButton = new JButton("Confirm Customer");
validateButton = new JButton("Validate New Data");
calculateButton = new JButton("Calculate KWH");
exitButton = new JButton("Exit");
controlPanel.add(confirmButton);
controlPanel.add(validateButton);
controlPanel.add(calculateButton);
controlPanel.add(exitButton);
```

```
actionExitMenuItem.addActionListener(this);
actionConfirmMenuItem.addActionListener(this);
actionValidateMenuItem.addActionListener(this);
actionCalculateMenuItem.addActionListener(this);
aboutMenuItem.addActionListener(this);
```

```
typeCommercial.addActionListener(this);
typeResidential.addActionListener(this);
typeGeneral.addActionListener(this);
year2002.addActionListener(this);
year2003.addActionListener(this);
year2004.addActionListener(this);
confirmButton.addActionListener(this);
validateButton.addActionListener(this);
calculateButton.addActionListener(this);
exitButton.addActionListener(this);
monthList.addListSelectionListener(this);
```

```
customerNameField.setEditable(false);
startMeterReadingField.setEditable(false);
controlField.setEditable(false);
```

```
JPanel mainPanel = new JPanel();
mainPanel.setLayout(new GridBagLayout());
GridBagConstraints c = new GridBagConstraints();
c.weightx = 10;
c.weighty = 10;
c.ipadx = 1;
```

```
c.anchor = GridBagConstraints.EAST;
c = getConstraints(c, 1, 1, 1, 1);
mainPanel.add(customerLabelPanel, c);
c = getConstraints(c, 1, 3, 1, 1);
mainPanel.add(serviceReadingLabelPanel, c);
```

```
c.anchor = GridBagConstraints.WEST;
```

```
c = getConstraints(c, 2, 1, 2, 1);
mainPanel.add(customerDataPanel, c);
c = getConstraints(c, 2, 2, 1, 1);
mainPanel.add(customerTypeDataPanel, c);
c = getConstraints(c, 2, 3, 1, 1);
mainPanel.add(serviceReadingDataPanel, c);
```

```

        c = getConstraints(c, 3, 2, 1, 1);
        mainPanel.add(datePanel, c);
        c = getConstraints(c, 3, 3, 1, 1);
        mainPanel.add(radioButtonPanel, c);

        c.anchor = GridBagConstraints.SOUTH;
        c = getConstraints(c, 1, 14, 4, 1);
        mainPanel.add(controlField, c);
        c = getConstraints(c, 1, 15, 4, 1);
        mainPanel.add(controlPanel, c);

        contentPane.add(mainPanel);
    }

private GridBagConstraints getConstraints(GridBagConstraints c,
    int x, int y, int width, int height)
{
    c.gridx = x;
    c.gridy = y;
    c.gridwidth = width;
    c.gridheight = height;
    return c;
}

public void actionPerformed(ActionEvent e){
    String commercialSelected, residentialSelected, generalSelected;
    controlField.setText("Enter the customer number and click the 'Confirm
Customer' Button.");

    Object source = e.getSource();

    if (source instanceof JMenuItem)
    {
        JMenuItem item = (JMenuItem) source;
        String text = item.getText();
        if(text.equals("Exit"))
        {
            setVisible(false);
            System.exit(0);
        }
        else if (text.equals("Confirm Customer"))
        {
            confirmButton.doClick();
        }
        else if (text.equals("Validate New Data"))
        {
            validateButton.doClick();
        }
        else if (text.equals("Calculate KWH"))
        {
            calculateButton.doClick();
        }
        else if (text.equals("About"))
        {
            String messageText =

```

```

        "This application is designed to
illustrate several GUI\n"+
        "features that include: labels,
textfields, radio buttons\n"+
        "(both grouped and ungrouped),
check boxes, menu items, \n"+
        "and gridbaglayouts with panels
within panels. If you should\n"+
        "\nthink of other improvements,
let me know. /n/nThank you." +
        "\nThe 'Management.'";
messageText,
JOptionPane.showMessageDialog(this,
"Denton Municipal",
JOptionPane.INFORMATION_MESSAGE);
    }
}

if (source == confirmButton)
{
    controlField.setText("Confirm Customer Button selected");
    String customerNumber = customerNumberField.getText();
    boolean valid = DMECustomer.setCustomerNumber(customerNumber);
    if (valid)
    {
        customerNameField.setText(DMECustomer.getCustomerName());
        if (DMECustomer.getServiceType() == 1)
            serviceType1.setSelected(true);
        else
            serviceType1.setSelected(false);
        if (DMECustomer.getServiceType() == 2)
            serviceType2.setSelected(true);
        else
            serviceType2.setSelected(false);

        monthList.setSelectedIndex(currentMonth - 1);
        if (currentYear == 2002)
        {
            controlField.setText("Year set to 2002");
            yearValue = 2002;
            year2002.setSelectedIcon(IconImage2002On);
            year2002.setSelected(true);
        }
        else
        {
            year2002.setSelectedIcon(IconImage2002Off);
            year2002.setSelected(false);
        }
    }
    if (currentYear == 2003)
    {
        controlField.setText("Year set to 2003");
        yearValue = 2003;
        year2003.setSelectedIcon(IconImage2003On);
        year2003.setSelected(true);
    }
    else
    {

```

```

        year2003.setSelectedIcon(IconImage2003Off);
        year2003.setSelected(false);
    }
    if (currentYear == 2004)
    {
        controlField.setText("Year set to 2004");
        yearValue = 2004;
        year2004.setSelectedIcon(IconImage2004On);
        year2004.setSelected(true);
    }
    else
    {
        year2004.setSelectedIcon(IconImage2004Off);
        year2004.setSelected(false);
    }

    if (DMECustomer.getCustomerType().equalsIgnoreCase("R"))
        typeResidential.setSelected(true);
    else
        typeResidential.setSelected(false);
    if (DMECustomer.getCustomerType().equalsIgnoreCase("C"))
        typeCommercial.setSelected(true);
    else
        typeCommercial.setSelected(false);
    if (DMECustomer.getCustomerType().equalsIgnoreCase("G"))
        typeGeneral.setSelected(true);
    else
        typeGeneral.setSelected(false);

    dayField.setText(Integer.toString(currentDay));

startMeterReadingField.setText(Integer.toString(DMECustomer.getLastReading()));
    }
    else
    {
        customerNumberField.setText("");
        customerNumberField.requestFocus();
    }
}

    if (source == calculateButton)
    {
        String endMeterReading = endMeterReadingField.getText();
        boolean valid = DMECustomer.setEndMeterReading(endMeterReading);
        if (valid)
        {
            int KWHUsed =
Integer.parseInt(endMeterReadingField.getText()) -
                DMECustomer.getLastReading();
            controlField.setText("KWH Used is: "+ KWHUsed);
        }
        else
        {
            endMeterReadingField.setText("");
            endMeterReadingField.requestFocus();
        }
    }
}

```

```

if (source == exitButton)
    setVisible(false);
//    System.exit(0);

if (source == validateButton)
{
    boolean valid = false;
    controlField.setText("Validate Data Button selected");
    String meterReadingDay = dayField.getText();
    valid = DMECustomer.setMeterReadingDay(meterReadingDay);
    if (!valid)
    {
        dayField.setText("");
        dayField.requestFocus();
    }
    valid = false;
    String endMeterReading = endMeterReadingField.getText();
    valid = DMECustomer.setEndMeterReading(endMeterReading);
    if (!valid)
    {
        endMeterReadingField.setText("");
        endMeterReadingField.requestFocus();
    }
    GregorianCalendar meterReadingDateGreg = new
    GregorianCalendar(yearValue, month,
        Integer.parseInt(dayField.getText()));
    if (!meterReadingDateGreg.before(dateTodayGreg))
        controlField.setText("Date is in the future ...
Invalid");
}

if (source == year2002)
{
    controlField.setText("Year set to 2002");
    yearValue = 2002;
    year2002.setSelectedIcon(IconImage2002On);
    year2002.setSelected(true);
}

if (source == year2003)
{
    controlField.setText("Year set to 2003");
    yearValue = 2003;
    year2003.setSelectedIcon(IconImage2003On);
    year2003.setSelected(true);
}

if (source == year2004)
{
    controlField.setText("Year set to 2004");
    yearValue = 2004;
    year2004.setSelectedIcon(IconImage2004On);
    year2004.setSelected(true);
}

if (source == typeCommercial)
{

```

```

        controlField.setText("Service Type 'Commerical' selected");
        commercialSelected = "yes";
    }
    else
        commercialSelected = "no";
    if (source == typeResidential)
    {
        controlField.setText("Service Type 'Residential' selected");
        residentialSelected = "yes";
    }
    else
        residentialSelected = "no";
    if (source == typeGeneral)
    {
        controlField.setText("Service Type 'General' selected");
        generalSelected = "yes";
    }
    else
        generalSelected = "no";
}

```

```

public void valueChanged(ListSelectionEvent e)
{
    Object o = e.getSource();
    if ((o == monthList)&&(monthList.hasFocus()))
    {
        month = monthList.getSelectedIndex();
        serviceMaxDay = DateItem.selectDay(month +1);
    }
}

```

```

public void keyPressed(KeyEvent e)
{
    int keyNumber = e.getKeyCode();
    if (keyNumber == KeyEvent.VK_ESCAPE)
    {
    }
}

```

```

public void keyReleased(KeyEvent e)
{
}

```

```

public void keyTyped(KeyEvent e)
{
}

```

```

public static void mainDriver()
{
    JFrame frame = new DME7Frame();
    frame.show();
}

```

```

public static void main(String[] args)
{
    JFrame frame = new DME7Frame();
    frame.show();
}

```

