

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.text.*;
import java.sql.*;

public class CustomerUpdateFrame extends JFrame
{
    public CustomerUpdateFrame()
    {
        setTitle("Customer Update Information");
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension d = tk.getScreenSize();
        int width = 900, height = 500;
        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();
        CustomerUpdatePanel panel = new CustomerUpdatePanel();
        contentPane.add(panel);
    }
    /* Test routine
    public static void main(String[] args){
        JFrame frame = new CustomerUpdateFrame();
        frame.show();
    }
    */
    public void mainDriver()
    {
        JFrame cFrame = new CustomerUpdateFrame();
        cFrame.show();
    }
}

class CustomerUpdatePanel extends JPanel implements ActionListener,
    DocumentListener, KeyListener{

    private JButton addButton, updateButton, deleteButton, exitButton,
        firstButton, prevButton, nextButton, lastButton;
    private JLabel c_numberLabel, lastNameLabel, firstNameLabel, middleNameLabel,
        addressLabel, cityLabel, stateLabel, zipLabel,
customerSinceMoLabel,
        customerSinceDayLabel, customerSinceYearLabel, customerTypeLabel,
        serviceTypeLabel, lastReadingMoLabel, lastReadingDayLabel,
        lastReadingYearLabel, lastReadingLabel, currentReadingMoLabel,
        currentReadingDayLabel, currentReadingYearLabel,
currentReadingLabel,
        outstandingBalanceLabel, currentBalanceLabel, lastPaymentMoLabel,
        lastPaymentDayLabel, lastPaymentYearLabel;

```

```

    private JTextField c_numberField, lastNameField, firstNameField,
middleNameField,
        addressField, cityField, stateField, zipField,
customerSinceMoField,
        customerSinceDayField, customerSinceYearField, customerTypeField,
serviceTypeField, lastReadingMoField, lastReadingDayField,
        lastReadingYearField, lastReadingField, currentReadingMoField,
currentReadingDayField, currentReadingYearField,
currentReadingField,
        outstandingBalanceField, currentBalanceField, lastPaymentMoField,
lastPaymentDayField, lastPaymentYearField;

public static boolean addFlag = false;
public static NumberFormat currency = NumberFormat.getCurrencyInstance();
CustomerUpdate currentCustomerUpdate = null;

public CustomerUpdatePanel(){

    c_numberLabel = new JLabel("Number: ");
    c_numberField = new JTextField("", 7);
    lastNameLabel = new JLabel("Last Name: ");
    lastNameField = new JTextField("", 15);
    firstNameLabel = new JLabel("First Name: ");
    firstNameField = new JTextField("", 15);
    middleNameLabel = new JLabel("Middle Name: ");
    middleNameField = new JTextField("", 15);
    addressLabel = new JLabel("Address: ");
    addressField = new JTextField("", 26);
    cityLabel = new JLabel("City: ");
    cityField = new JTextField("", 15);
    stateLabel = new JLabel("State: ");
    stateField = new JTextField("", 5);
    zipLabel = new JLabel("Zip Code: ");
    zipField = new JTextField("", 7);
    customerSinceMoLabel = new JLabel("Customer Since Month: ");
    customerSinceMoField = new JTextField("", 4);
    customerSinceDayLabel = new JLabel("Day: ");
    customerSinceDayField = new JTextField("", 4);
    customerSinceYearLabel = new JLabel("Year: ");
    customerSinceYearField = new JTextField("", 8);
    customerTypeLabel = new JLabel("Customer Type: ");
    customerTypeField = new JTextField("", 4);
    serviceTypeLabel = new JLabel("Service Type: ");
    serviceTypeField = new JTextField("", 4);
    lastReadingMoLabel = new JLabel("Last Reading Month: ");
    lastReadingMoField = new JTextField("", 4);
    lastReadingDayLabel = new JLabel("Day: ");
    lastReadingDayField = new JTextField("", 4);
    lastReadingYearLabel = new JLabel("Year: ");
    lastReadingYearField = new JTextField("", 8);
    lastReadingLabel = new JLabel("Last Reading: ");
    lastReadingField = new JTextField("", 10);
    currentReadingMoLabel = new JLabel("Current Reading Month: ");
    currentReadingMoField = new JTextField("", 4);
    currentReadingDayLabel = new JLabel("Day: ");
    currentReadingDayField = new JTextField("", 4);
    currentReadingYearLabel = new JLabel("Year: ");

```

```
currentReadingYearField = new JTextField("", 8);
currentReadingLabel = new JLabel("Current Reading: ");
currentReadingField = new JTextField("", 10);
outstandingBalanceLabel = new JLabel("Outstanding Balance: ");
outstandingBalanceField = new JTextField("", 10);
currentBalanceLabel = new JLabel("Current Balance: ");
currentBalanceField = new JTextField("", 10);
lastPaymentMoLabel = new JLabel("Last Payment Month: ");
lastPaymentMoField = new JTextField("", 4);
lastPaymentDayLabel = new JLabel("Day: ");
lastPaymentDayField = new JTextField("", 4);
lastPaymentYearLabel = new JLabel("Year: ");
lastPaymentYearField = new JTextField("", 8);
```

```
JPanel updatePanel = new JPanel();
addButton = new JButton("Add");
updateButton = new JButton("Update");
deleteButton = new JButton("Delete");
exitButton = new JButton("Exit");
updatePanel.add(addButton);
updatePanel.add(updateButton);
updatePanel.add(deleteButton);
updatePanel.add(exitButton);
```

```
JPanel navigationPanel = new JPanel();
firstButton = new JButton("First");
prevButton = new JButton("Prev");
nextButton = new JButton("Next");
lastButton = new JButton("Last");
navigationPanel.add(firstButton);
navigationPanel.add(prevButton);
navigationPanel.add(nextButton);
navigationPanel.add(lastButton);
```

```
addButton.addActionListener(this);
updateButton.addActionListener(this);
deleteButton.addActionListener(this);
exitButton.addActionListener(this);
firstButton.addActionListener(this);
prevButton.addActionListener(this);
nextButton.addActionListener(this);
lastButton.addActionListener(this);
```

```
c_numberField.addKeyListener(this);
lastNameField.addKeyListener(this);
firstNameField.addKeyListener(this);
middleNameField.addKeyListener(this);
addressField.addKeyListener(this);
cityField.addKeyListener(this);
stateField.addKeyListener(this);
zipField.addKeyListener(this);
customerSinceMoField.addKeyListener(this);
customerSinceDayField.addKeyListener(this);
customerSinceYearField.addKeyListener(this);
customerTypeField.addKeyListener(this);
serviceTypeField.addKeyListener(this);
lastReadingMoField.addKeyListener(this);
```

```

lastReadingDayField.addKeyListener(this);
lastReadingYearField.addKeyListener(this);
lastReadingField.addKeyListener(this);
currentReadingMoField.addKeyListener(this);
currentReadingDayField.addKeyListener(this);
currentReadingYearField.addKeyListener(this);
currentReadingField.addKeyListener(this);
    outstandingBalanceField.addKeyListener(this);
    currentBalanceField.addKeyListener(this);
lastPaymentMoField.addKeyListener(this);
lastPaymentDayField.addKeyListener(this);
lastPaymentYearField.addKeyListener(this);

c_numberField.getDocument().addDocumentListener(this);
lastNameField.getDocument().addDocumentListener(this);
firstNameField.getDocument().addDocumentListener(this);
middleNameField.getDocument().addDocumentListener(this);
addressField.getDocument().addDocumentListener(this);
cityField.getDocument().addDocumentListener(this);
stateField.getDocument().addDocumentListener(this);
zipField.getDocument().addDocumentListener(this);
customerSinceMoField.getDocument().addDocumentListener(this);
customerSinceDayField.getDocument().addDocumentListener(this);
customerSinceYearField.getDocument().addDocumentListener(this);
customerTypeField.getDocument().addDocumentListener(this);
serviceTypeField.getDocument().addDocumentListener(this);
lastReadingMoField.getDocument().addDocumentListener(this);
lastReadingDayField.getDocument().addDocumentListener(this);
lastReadingYearField.getDocument().addDocumentListener(this);
lastReadingField.getDocument().addDocumentListener(this);
currentReadingMoField.getDocument().addDocumentListener(this);
currentReadingDayField.getDocument().addDocumentListener(this);
currentReadingYearField.getDocument().addDocumentListener(this);
currentReadingField.getDocument().addDocumentListener(this);
    outstandingBalanceField.getDocument().addDocumentListener(this);
    currentBalanceField.getDocument().addDocumentListener(this);
lastPaymentMoField.getDocument().addDocumentListener(this);
lastPaymentDayField.getDocument().addDocumentListener(this);
lastPaymentYearField.getDocument().addDocumentListener(this);

setLayout(new GridBagLayout());
GridBagConstraints c = new GridBagConstraints();
c.weightx = 100;
c.weighty = 100;
c.ipadx = 5;

c.anchor = GridBagConstraints.EAST;
c = getConstraints(c, 1, 1, 1, 1);
add(c_numberLabel, c);
c = getConstraints(c, 1, 2, 1, 1);
add(lastNameLabel, c);
c = getConstraints(c, 1, 3, 1, 1);
add(firstNameLabel, c);
c = getConstraints(c, 1, 4, 1, 1);
add(middleNameLabel, c);
c = getConstraints(c, 1, 5, 1, 1);
add(addressLabel, c);

```

```

c = getConstraints(c, 1, 6, 1, 1);
add(cityLabel, c);
c = getConstraints(c, 1, 7, 1, 1);
add(stateLabel, c);
c = getConstraints(c, 1, 8, 1, 1);
add(zipLabel, c);
c = getConstraints(c, 1, 9, 1, 1);
add(customerSinceMoLabel, c);
c = getConstraints(c, 1, 10, 1, 1);
add(customerSinceDayLabel, c);
c = getConstraints(c, 1, 11, 1, 1);
add(customerSinceYearLabel, c);
c = getConstraints(c, 1, 12, 1, 1);
add(customerTypeLabel, c);
c = getConstraints(c, 1, 13, 1, 1);
add(serviceTypeLabel, c);

c.anchor = GridBagConstraints.WEST;
c = getConstraints(c, 2, 1, 1, 1);
add(c_numberField, c);
c = getConstraints(c, 2, 2, 1, 1);
add(lastNameField, c);
c = getConstraints(c, 2, 3, 1, 1);
add(firstNameField, c);
c = getConstraints(c, 2, 4, 1, 1);
add(middleNameField, c);
c = getConstraints(c, 2, 5, 1, 1);
add(addressField, c);
c = getConstraints(c, 2, 6, 1, 1);
add(cityField, c);
c = getConstraints(c, 2, 7, 1, 1);
add(stateField, c);
c = getConstraints(c, 2, 8, 1, 1);
add(zipField, c);
c = getConstraints(c, 2, 9, 1, 1);
add(customerSinceMoField, c);
c = getConstraints(c, 2, 10, 1, 1);
add(customerSinceDayField, c);
c = getConstraints(c, 2, 11, 1, 1);
add(customerSinceYearField, c);
c = getConstraints(c, 2, 12, 1, 1);
add(customerTypeField, c);
c = getConstraints(c, 2, 13, 1, 1);
add(serviceTypeField, c);

c.anchor = GridBagConstraints.EAST;
c = getConstraints(c, 3, 1, 1, 1);
add(lastReadingMoLabel, c);
c = getConstraints(c, 3, 2, 1, 1);
add(lastReadingDayLabel, c);
c = getConstraints(c, 3, 3, 1, 1);
add(lastReadingYearLabel, c);
c = getConstraints(c, 3, 4, 1, 1);
add(lastReadingLabel, c);
c = getConstraints(c, 3, 5, 1, 1);
add(currentReadingMoLabel, c);
c = getConstraints(c, 3, 6, 1, 1);

```

```

add(currentReadingDayLabel, c);
c = getConstraints(c, 3, 7, 1, 1);
add(currentReadingYearLabel, c);
c = getConstraints(c, 3, 8, 1, 1);
add(currentReadingLabel, c);
c = getConstraints(c, 3, 9, 1, 1);
add(outstandingBalanceLabel, c);
c = getConstraints(c, 3, 10, 1, 1);
add(currentBalanceLabel, c);
c = getConstraints(c, 3, 11, 1, 1);
add(lastPaymentMoLabel, c);
c = getConstraints(c, 3, 12, 1, 1);
add(lastPaymentDayLabel, c);
c = getConstraints(c, 3, 13, 1, 1);
add(lastPaymentYearLabel, c);

c.anchor = GridBagConstraints.WEST;
c = getConstraints(c, 4, 1, 1, 1);
add(lastReadingMoField, c);
c = getConstraints(c, 4, 2, 1, 1);
add(lastReadingDayField, c);
c = getConstraints(c, 4, 3, 1, 1);
add(lastReadingYearField, c);
c = getConstraints(c, 4, 4, 1, 1);
add(lastReadingField, c);
c = getConstraints(c, 4, 5, 1, 1);
add(currentReadingMoField, c);
c = getConstraints(c, 4, 6, 1, 1);
add(currentReadingDayField, c);
c = getConstraints(c, 4, 7, 1, 1);
add(currentReadingYearField, c);
c = getConstraints(c, 4, 8, 1, 1);
add(currentReadingField, c);
c = getConstraints(c, 4, 9, 1, 1);
add(outstandingBalanceField, c);
c = getConstraints(c, 4, 10, 1, 1);
add(currentBalanceField, c);
c = getConstraints(c, 4, 11, 1, 1);
add(lastPaymentMoField, c);
c = getConstraints(c, 4, 12, 1, 1);
add(lastPaymentDayField, c);
c = getConstraints(c, 4, 13, 1, 1);
add(lastPaymentYearField, c);

c.anchor = GridBagConstraints.CENTER;
c = getConstraints(c, 1, 14, 4, 1);
add(updatePanel, c);
c = getConstraints(c, 1, 15, 4, 1);
add(navigationPanel, c);

try
{
    CustomerUpdatedDB.connect();
    CustomerUpdatedDB.open();
    currentCustomerUpdate = CustomerUpdatedDB.moveFirst();
}

```

```

        catch (ClassNotFoundException e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
            setVisible(false);
//            System.exit(1);
        }
        catch (SQLException e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
        performCustomerUpdateDisplay();
        enableButtons(true);
    }

    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;
    }

    private void performCustomerUpdateDisplay(){
        c_numberField.setText(currentCustomerUpdate.getCustomerNumber());
        lastNameField.setText(currentCustomerUpdate.getLastName());
        firstNameField.setText(currentCustomerUpdate.getFirstName());
        middleNameField.setText(currentCustomerUpdate.getMiddleName());
        addressField.setText(currentCustomerUpdate.getAddress());
        cityField.setText(currentCustomerUpdate.getCity());
        stateField.setText(currentCustomerUpdate.getState());
        zipField.setText(currentCustomerUpdate.getZip());

        customerSinceMoField.setText(Integer.toString(currentCustomerUpdate.getCustomerS
inceMo()));

        customerSinceDayField.setText(Integer.toString(currentCustomerUpdate.getCustomer
SinceDay()));

        customerSinceYearField.setText(Integer.toString(currentCustomerUpdate.getCustome
rSinceYear()));
        customerTypeField.setText(currentCustomerUpdate.getCustomerType());

        serviceTypeField.setText(Integer.toString(currentCustomerUpdate.getServiceType()
));

        lastReadingMoField.setText(Integer.toString(currentCustomerUpdate.getLastReading
Mo()));

        lastReadingDayField.setText(Integer.toString(currentCustomerUpdate.getLastReadin
gDay()));

        lastReadingYearField.setText(Integer.toString(currentCustomerUpdate.getLastReadi
ngYear()));

        lastReadingField.setText(Integer.toString(currentCustomerUpdate.getLastReading()
));

```

```

currentReadingMoField.setText(Integer.toString(currentCustomerUpdate.getCurrentReadingMo()));

currentReadingDayField.setText(Integer.toString(currentCustomerUpdate.getCurrentReadingDay()));

currentReadingYearField.setText(Integer.toString(currentCustomerUpdate.getCurrentReadingYear()));

currentReadingField.setText(Integer.toString(currentCustomerUpdate.getCurrentReading()));

outstandingBalanceField.setText(Double.toString(currentCustomerUpdate.getOutstandingBalance()));

currentBalanceField.setText(Double.toString(currentCustomerUpdate.getCurrentBalance()));

lastPaymentMoField.setText(Integer.toString(currentCustomerUpdate.getLastPaymentMo()));

lastPaymentDayField.setText(Integer.toString(currentCustomerUpdate.getLastPaymentDay()));

lastPaymentYearField.setText(Integer.toString(currentCustomerUpdate.getLastPaymentYear()));
}

private void enableButtons(boolean flag1){
    boolean flag2 = false;
    if (flag1 == false)
        flag2 = true;
    updateButton.setEnabled(flag2);
    addButton.setEnabled(flag1);
    deleteButton.setEnabled(flag1);
    firstButton.setEnabled(flag1);
    nextButton.setEnabled(flag1);
    prevButton.setEnabled(flag1);
    lastButton.setEnabled(flag1);
}

private double parseCurrency(String currencyString){
    if (currencyString.charAt(0) == '$')
        currencyString = currencyString.substring(1);
    return Double.parseDouble(currencyString);
}

public void actionPerformed(ActionEvent e){
    try{
        Object source = e.getSource();
        if (source == exitButton){
            CustomerUpdateDB.close();
            setVisible(false);
            System.exit(0);
        }
        else if (source == firstButton){

```

```

        currentCustomerUpdate = CustomerUpdateDB.moveFirst();
        performCustomerUpdateDisplay();
        enableButtons(true);
    }
    else if (source == prevButton){
        currentCustomerUpdate = CustomerUpdateDB.movePrevious();
        performCustomerUpdateDisplay();
        enableButtons(true);
    }
    else if (source == nextButton){
        currentCustomerUpdate = CustomerUpdateDB.moveNext();
        performCustomerUpdateDisplay();
        enableButtons(true);
    }
    else if (source == lastButton){
        currentCustomerUpdate = CustomerUpdateDB.moveLast();
        performCustomerUpdateDisplay();
        enableButtons(true);
    }
    else if (source == addButton){
        c_numberField.requestFocus();
        enableButtons(false);
        c_numberField.setText("");
        lastNameField.setText("");
        firstNameField.setText("");
        middleNameField.setText("");
        addressField.setText("");
        cityField.setText("");
        stateField.setText("");
        zipField.setText("");
        customerSinceMoField.setText("");
        customerSinceDayField.setText("");
        customerSinceYearField.setText("");
        customerTypeField.setText("");
        serviceTypeField.setText("");
        lastReadingMoField.setText("");
        lastReadingDayField.setText("");
        lastReadingYearField.setText("");
        lastReadingField.setText("");
        currentReadingMoField.setText("");
        currentReadingDayField.setText("");
        currentReadingYearField.setText("");
        currentReadingField.setText("");
        outstandingBalanceField.setText("");
        currentBalanceField.setText("");
        lastPaymentMoField.setText("");
        lastPaymentDayField.setText("");
        lastPaymentYearField.setText("");
        addFlag = true;
    }
    else if (source == updateButton){
        int customerSinceMo =
Integer.parseInt(customerSinceMoField.getText());
        int customerSinceDay =
Integer.parseInt(customerSinceDayField.getText());
        int customerSinceYear =
Integer.parseInt(customerSinceYearField.getText());

```

```

        int serviceType =
Integer.parseInt(serviceTypeField.getText());
        int lastReadingMo =
Integer.parseInt(lastReadingMoField.getText());
        int lastReadingDay =
Integer.parseInt(lastReadingDayField.getText());
        int lastReadingYear =
Integer.parseInt(lastReadingYearField.getText());
        int lastReading =
Integer.parseInt(lastReadingField.getText());
        int currentReadingMo =
Integer.parseInt(currentReadingMoField.getText());
        int currentReadingDay =
Integer.parseInt(currentReadingDayField.getText());
        int currentReadingYear =
Integer.parseInt(currentReadingYearField.getText());
        int currentReading =
Integer.parseInt(currentReadingField.getText());
        double outstandingBalance =
parseFloat(outstandingBalanceField.getText());
        double currentBalance =
parseFloat(currentBalanceField.getText());
        int lastPaymentMo =
Integer.parseInt(lastPaymentMoField.getText());
        int lastPaymentDay =
Integer.parseInt(lastPaymentDayField.getText());
        int lastPaymentYear =
Integer.parseInt(lastPaymentYearField.getText());
        CustomerUpdate customer = new CustomerUpdate(
            c_numberField.getText(),
            lastNameField.getText(),
            firstNameField.getText(),
            middleNameField.getText(),
            addressField.getText(),
            cityField.getText(),
            stateField.getText(),
            zipField.getText(),
            customerSinceMo,
            customerSinceDay,
            customerSinceYear,
            customerTypeField.getText(),
            serviceType,
            lastReadingMo,
            lastReadingDay,
            lastReadingYear,
            lastReading,
            currentReadingMo,
            currentReadingDay,
            currentReadingYear,
            currentReading,
            outstandingBalance,
            currentBalance,
            lastPaymentMo,
            lastPaymentDay,
            lastPaymentYear);
    if (addFlag == false){
        CustomerUpdatedB.updateRecord(customer);

```

```

        currentCustomerUpdate = customer;
    }
    if (addFlag == true){
        CustomerUpdatedDB.addRecord(customer);
        currentCustomerUpdate = CustomerUpdatedDB.moveFirst();
        addFlag = false;
    }
    currentCustomerUpdate = customer;
    performCustomerUpdateDisplay();
    enableButtons(true);
}
else if(source == deleteButton){
CustomerUpdatedDB.deleteRecord(currentCustomerUpdate.getCustomerNumber());
    firstButton.requestFocus();
    firstButton.doClick();
    performCustomerUpdateDisplay();
    enableButtons(true);
}
}
catch (NumberFormatException nfe){
    JOptionPane.showMessageDialog(this, nfe.getMessage());
}
catch (SQLException sqle){
    JOptionPane.showMessageDialog(this, sqle.getMessage());
}
}

public void keyPressed(KeyEvent e){
    int keyNumber = e.getKeyCode();
    if (keyNumber == KeyEvent.VK_ESCAPE)
    {
        performCustomerUpdateDisplay();
        enableButtons(true);
        c_numberField.requestFocus();
    }
}
public void keyReleased(KeyEvent e){
}
public void keyTyped(KeyEvent e){
}

public void insertUpdate(DocumentEvent e){
    enableButtons(false);
}
public void removeUpdate(DocumentEvent e){
    enableButtons(false);
}
public void changedUpdate(DocumentEvent e){
}
}

```