**Mini project for phone book using java**

Source code:

package phonebook2;

import java.awt.Color;

import java.awt.Component;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.awt.event.KeyEvent;

import java.awt.print.PrinterException;

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.MessageFormat;

import java.util.Vector;

import javax.swing.JCheckBox;

import javax.swing.JOptionPane;

import javax.swing.JSlider;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.UIManager;

import javax.swing.table.AbstractTableModel;

import javax.swing.table.DefaultTableCellRenderer;

import javax.swing.table.TableColumn;

public class Form extends javax.swing.JFrame {

private Connection con = null;

private PersonModel model = null;

private String database = "notebook";

private String tableName = database + ".table2";

private String mysql\_password = "";

public Form() {

initComponents();

Dimension d = Toolkit.getDefaultToolkit().getScreenSize();

int x = d.width/2 - this.getWidth()/2,

y = d.height/2 - this.getHeight()/2;

setLocation(x, y);

scrollPane.getViewport().setBackground(Color.WHITE);

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

mainPanel = new javax.swing.JPanel();

settingPanel = new javax.swing.JPanel();

cb\_horizLine = new javax.swing.JCheckBox();

cb\_verticLine = new javax.swing.JCheckBox();

slider = new javax.swing.JSlider();

cb\_selRow = new javax.swing.JCheckBox();

cb\_selCol = new javax.swing.JCheckBox();

cb\_shwDlg = new javax.swing.JCheckBox();

infoPanel = new javax.swing.JPanel();

label1 = new javax.swing.JLabel();

Name = new javax.swing.JTextField();

City = new javax.swing.JTextField();

Phone = new javax.swing.JTextField();

label2 = new javax.swing.JLabel();

label3 = new javax.swing.JLabel();

Insert = new javax.swing.JButton();

Reset = new javax.swing.JButton();

searchPanel = new javax.swing.JPanel();

label4 = new javax.swing.JLabel();

fieldName = new javax.swing.JComboBox();

value = new javax.swing.JTextField();

tablePanel = new javax.swing.JPanel();

scrollPane = new javax.swing.JScrollPane();

dataTable = new javax.swing.JTable();

Backup = new javax.swing.JButton();

DeleteAll = new javax.swing.JButton();

Print = new javax.swing.JButton();

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

setTitle("phonebook2");

setResizable(false);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowOpened(java.awt.event.WindowEvent evt) {

formWindowOpened(evt);

}

});

mainPanel.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Arial", 0, 12))); // NOI18N

settingPanel.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Setting", javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 0, 12))); // NOI18N

cb\_horizLine.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

cb\_horizLine.setSelected(true);

cb\_horizLine.setText("show horiz-line");

cb\_horizLine.setNextFocusableComponent(cb\_verticLine);

cb\_horizLine.addActionListener(new java.awt.event.ActionListener() {

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

cb\_horizLineActionPerformed(evt);

}

});

cb\_verticLine.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

cb\_verticLine.setSelected(true);

cb\_verticLine.setText("show vertic-line");

cb\_verticLine.setNextFocusableComponent(cb\_selRow);

cb\_verticLine.addActionListener(new java.awt.event.ActionListener() {

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

cb\_verticLineActionPerformed(evt);

}

});

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

slider.setOrientation(javax.swing.JSlider.VERTICAL);

slider.setValue(20);

slider.setNextFocusableComponent(Name);

slider.addChangeListener(new javax.swing.event.ChangeListener() {

public void stateChanged(javax.swing.event.ChangeEvent evt) {

sliderStateChanged(evt);

}

});

cb\_selRow.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

cb\_selRow.setSelected(true);

cb\_selRow.setText("select row");

cb\_selRow.setNextFocusableComponent(cb\_selCol);

cb\_selRow.addActionListener(new java.awt.event.ActionListener() {

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

cb\_selRowActionPerformed(evt);

}

});

cb\_selCol.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

cb\_selCol.setText("select column");

cb\_selCol.setNextFocusableComponent(slider);

cb\_selCol.addActionListener(new java.awt.event.ActionListener() {

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

cb\_selColActionPerformed(evt);

}

});

cb\_shwDlg.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

cb\_shwDlg.setSelected(true);

cb\_shwDlg.setText("show deleting dialog");

cb\_shwDlg.setNextFocusableComponent(slider);

javax.swing.GroupLayout settingPanelLayout = new javax.swing.GroupLayout(settingPanel);

settingPanel.setLayout(settingPanelLayout);

settingPanelLayout.setHorizontalGroup(

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

.addGroup(settingPanelLayout.createSequentialGroup()

.addContainerGap()

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

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

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

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

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

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

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

.addComponent(slider, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

settingPanelLayout.setVerticalGroup(

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

.addGroup(settingPanelLayout.createSequentialGroup()

.addContainerGap()

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

.addComponent(slider, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addGroup(settingPanelLayout.createSequentialGroup()

.addComponent(cb\_horizLine, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(cb\_verticLine, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(cb\_selRow, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(cb\_selCol, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(cb\_shwDlg, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE)))

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

);

infoPanel.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "insert", javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 0, 12))); // NOI18N

label1.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

label1.setText("name :");

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

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

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

label2.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

label2.setText("phone :");

label3.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

label3.setText("city :");

Insert.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

Insert.setText("insert");

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

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

InsertActionPerformed(evt);

}

});

Reset.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

Reset.setText("reset");

Reset.setNextFocusableComponent(fieldName);

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

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

ResetActionPerformed(evt);

}

});

javax.swing.GroupLayout infoPanelLayout = new javax.swing.GroupLayout(infoPanel);

infoPanel.setLayout(infoPanelLayout);

infoPanelLayout.setHorizontalGroup(

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

.addGroup(infoPanelLayout.createSequentialGroup()

.addContainerGap()

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

.addGroup(infoPanelLayout.createSequentialGroup()

.addComponent(label1, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(Name))

.addGroup(infoPanelLayout.createSequentialGroup()

.addComponent(label3, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(City))

.addGroup(infoPanelLayout.createSequentialGroup()

.addComponent(label2, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

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

.addGroup(infoPanelLayout.createSequentialGroup()

.addComponent(Insert, javax.swing.GroupLayout.PREFERRED\_SIZE, 65, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(Reset, javax.swing.GroupLayout.PREFERRED\_SIZE, 65, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(Phone))))

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

);

infoPanelLayout.setVerticalGroup(

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

.addGroup(infoPanelLayout.createSequentialGroup()

.addContainerGap()

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

.addComponent(label1)

.addComponent(Name, javax.swing.GroupLayout.PREFERRED\_SIZE, 24, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

.addComponent(label3)

.addComponent(City, javax.swing.GroupLayout.PREFERRED\_SIZE, 24, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

.addComponent(label2)

.addComponent(Phone, javax.swing.GroupLayout.PREFERRED\_SIZE, 24, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

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

.addComponent(Insert, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

);

searchPanel.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Search", javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 0, 12))); // NOI18N

label4.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

label4.setText("field :");

fieldName.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

fieldName.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "name", "city", "phone" }));

fieldName.setNextFocusableComponent(value);

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

value.setNextFocusableComponent(cb\_horizLine);

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

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

valueKeyReleased(evt);

}

});

javax.swing.GroupLayout searchPanelLayout = new javax.swing.GroupLayout(searchPanel);

searchPanel.setLayout(searchPanelLayout);

searchPanelLayout.setHorizontalGroup(

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

.addGroup(searchPanelLayout.createSequentialGroup()

.addContainerGap()

.addComponent(label4, javax.swing.GroupLayout.PREFERRED\_SIZE, 40, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

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

.addComponent(value)

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

.addContainerGap())

);

searchPanelLayout.setVerticalGroup(

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

.addGroup(searchPanelLayout.createSequentialGroup()

.addContainerGap()

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

.addComponent(label4)

.addComponent(fieldName, javax.swing.GroupLayout.PREFERRED\_SIZE, 24, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

.addComponent(value, javax.swing.GroupLayout.PREFERRED\_SIZE, 24, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

);

tablePanel.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Display", javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 0, 12))); // NOI18N

dataTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

}

));

dataTable.setAutoResizeMode(javax.swing.JTable.AUTO\_RESIZE\_OFF);

dataTable.setGridColor(new java.awt.Color(204, 204, 204));

dataTable.setRowHeight(20);

dataTable.setSelectionBackground(new java.awt.Color(102, 182, 250));

dataTable.setSelectionMode(javax.swing.ListSelectionModel.SINGLE\_SELECTION);

dataTable.getTableHeader().setReorderingAllowed(false);

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

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

dataTableKeyReleased(evt);

}

});

scrollPane.setViewportView(dataTable);

Backup.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

Backup.setText("backup");

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

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

BackupActionPerformed(evt);

}

});

DeleteAll.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

DeleteAll.setText("delete all");

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

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

DeleteAllActionPerformed(evt);

}

});

Print.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

Print.setText("print");

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

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

PrintActionPerformed(evt);

}

});

javax.swing.GroupLayout tablePanelLayout = new javax.swing.GroupLayout(tablePanel);

tablePanel.setLayout(tablePanelLayout);

tablePanelLayout.setHorizontalGroup(

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

.addGroup(tablePanelLayout.createSequentialGroup()

.addContainerGap()

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

.addComponent(scrollPane, javax.swing.GroupLayout.PREFERRED\_SIZE, 280, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(tablePanelLayout.createSequentialGroup()

.addComponent(Backup, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(DeleteAll, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(Print, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE)))

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

);

tablePanelLayout.setVerticalGroup(

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

.addGroup(tablePanelLayout.createSequentialGroup()

.addComponent(scrollPane, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

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

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

.addComponent(Backup, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(DeleteAll, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(Print, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)))

);

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);

mainPanel.setLayout(mainPanelLayout);

mainPanelLayout.setHorizontalGroup(

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

.addGroup(mainPanelLayout.createSequentialGroup()

.addContainerGap()

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

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

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

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

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

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

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

);

mainPanelLayout.setVerticalGroup(

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

.addGroup(mainPanelLayout.createSequentialGroup()

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

.addGroup(mainPanelLayout.createSequentialGroup()

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

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

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

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

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

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

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

);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

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

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

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

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void ResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_ResetActionPerformed

for(Component comp : this.infoPanel.getComponents())

if(comp instanceof JTextField)

((JTextField)comp).setText("");

}//GEN-LAST:event\_ResetActionPerformed

private void formWindowOpened(java.awt.event.WindowEvent evt) {//GEN-FIRST:event\_formWindowOpened

try{

// found jdbc driver for mysql

Class.forName("com.mysql.jdbc.Driver");

// create connection to mysql

String url = "jdbc:mysql://localhost/",

uname = "root",

upass = mysql\_password;

con = DriverManager.getConnection(url, uname, upass);

createDatabase();

createTable();

Vector data = getData("");

showData(data);

//set database

con.setCatalog(database);

}catch(Exception e){

JOptionPane.showMessageDialog(this, e.getMessage() + "\nCan not connected to mysql!");

this.dispose();

}

}//GEN-LAST:event\_formWindowOpened

private void InsertActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_InsertActionPerformed

try{

if(isEmpty())

throw new Exception("please enter all data!");

// if phone is frequency dont insert

if(isExitsPhone())

throw new Exception("phone number is frequency!");

boolean ok = insert();

if(ok){

Vector data = getData("");

showData(data);

} else

throw new Exception("an error occure at inserting!");

}catch(Exception e){

JOptionPane.showMessageDialog(this, e.getMessage());

}

}//GEN-LAST:event\_InsertActionPerformed

private void dataTableKeyReleased(java.awt.event.KeyEvent evt) {//GEN-FIRST:event\_dataTableKeyReleased

if(evt.getKeyCode() != KeyEvent.VK\_DELETE)

return;

JTable table = (JTable)evt.getSource();

int row = table.getSelectedRow();

if(row < 0)

return;

Object phone = table.getValueAt(row, 2);

if(cb\_shwDlg.isSelected()){

int option = JOptionPane.showConfirmDialog(this, "Do you want to delete this item?",

"delete", JOptionPane.YES\_NO\_CANCEL\_OPTION);

if(option != JOptionPane.YES\_OPTION)

return;

}

try{

boolean ok = delete(phone);

if(ok){

Vector data = getData("");

showData(data);

} else

throw new Exception("an error occure at deleting!");

}catch(Exception e){

JOptionPane.showMessageDialog(this, e.getMessage());

}

}//GEN-LAST:event\_dataTableKeyReleased

private void valueKeyReleased(java.awt.event.KeyEvent evt) {//GEN-FIRST:event\_valueKeyReleased

String text = value.getText().trim();

try{

String where = "";

if(!text.equals(""))

where = " WHERE " + fieldName.getSelectedItem() + " LIKE '" + text + "%'";

Vector data = getData(where);

showData(data);

}catch(Exception e){

}

}//GEN-LAST:event\_valueKeyReleased

private void cb\_horizLineActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_cb\_horizLineActionPerformed

boolean show = ((JCheckBox)evt.getSource()).isSelected();

dataTable.setShowHorizontalLines(show);

}//GEN-LAST:event\_cb\_horizLineActionPerformed

private void cb\_verticLineActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_cb\_verticLineActionPerformed

boolean show = ((JCheckBox)evt.getSource()).isSelected();

dataTable.setShowVerticalLines(show);

}//GEN-LAST:event\_cb\_verticLineActionPerformed

private void cb\_selRowActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_cb\_selRowActionPerformed

boolean show = ((JCheckBox)evt.getSource()).isSelected();

dataTable.setRowSelectionAllowed(show);

}//GEN-LAST:event\_cb\_selRowActionPerformed

private void cb\_selColActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_cb\_selColActionPerformed

boolean show = ((JCheckBox)evt.getSource()).isSelected();

dataTable.setColumnSelectionAllowed(show);

}//GEN-LAST:event\_cb\_selColActionPerformed

private void sliderStateChanged(javax.swing.event.ChangeEvent evt) {//GEN-FIRST:event\_sliderStateChanged

int height = ((JSlider)evt.getSource()).getValue() + 1;

dataTable.setRowHeight(height);

}//GEN-LAST:event\_sliderStateChanged

private void PrintActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_PrintActionPerformed

MessageFormat headerFmt = new MessageFormat("Notebook"),

footerFmt = new MessageFormat("Page {0}");

JTable.PrintMode printMode = JTable.PrintMode.FIT\_WIDTH;

try {

boolean status = dataTable.print(printMode, headerFmt, footerFmt);

if (status) {

JOptionPane.showMessageDialog(dataTable.getParent(), "printing is Complete",

"printing Result", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(dataTable.getParent(), "printing Cancelled",

"printing Result",JOptionPane.INFORMATION\_MESSAGE);

}

} catch (PrinterException pe) {

String errorMessage = MessageFormat.format("printing Failed", new Object[] {pe.getMessage()});

JOptionPane.showMessageDialog(dataTable.getParent(), errorMessage,

"printing Result", JOptionPane.ERROR\_MESSAGE);

} catch (SecurityException se) {

String errorMessage = MessageFormat.format("printing Failed", new Object[] {se.getMessage()});

JOptionPane.showMessageDialog(dataTable.getParent(), errorMessage,

"TableDemo.printingResult", JOptionPane.ERROR\_MESSAGE);

}

}//GEN-LAST:event\_PrintActionPerformed

private void DeleteAllActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_DeleteAllActionPerformed

try{

int row = deleteall();

Vector data = getData("");

showData(data);

throw new Exception(row + " row(s) deleted!");

}catch(Exception e){

JOptionPane.showMessageDialog(this, e.getMessage());

}

}//GEN-LAST:event\_DeleteAllActionPerformed

private void BackupActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_BackupActionPerformed

try{

BufferedWriter writer = new BufferedWriter(new FileWriter("backup.sql"));

Vector allData = getData("");

for(int row = 0; row < allData.size(); ++row){

Person data = (Person)allData.elementAt(row);

String sql = "insert into " + tableName + " values ('" + data.getName() + "' , '" +

data.getCity() + "' , '" + data.getPhone() + "');";

writer.write(sql);

writer.newLine();

}

writer.close();

JOptionPane.showMessageDialog(this, "successfull backupping!" , "backup", JOptionPane.INFORMATION\_MESSAGE);

}catch(Exception e){

JOptionPane.showMessageDialog(this, "an error occure at backupping!" , "formWindowOpened", JOptionPane.ERROR\_MESSAGE);

}

}//GEN-LAST:event\_BackupActionPerformed

private boolean isEmpty(){

Component[] comps = infoPanel.getComponents();

for(int i = 0; i < comps.length; ++i){

Component comp = comps[i];

if(comp instanceof JTextField){

JTextField field = (JTextField)comp;

if(field.getText().equals(""))

return true;

}

}

return false;

}

private void createDatabase() throws Exception{

Statement stm = this.con.createStatement();

stm.execute("CREATE DATABASE IF NOT EXISTS " + database +

" DEFAULT CHARACTER SET utf8 COLLATE utf8\_persian\_ci");

}

private void createTable() throws Exception{

Statement stm = this.con.createStatement();

String query = "CREATE TABLE IF NOT EXISTS " + tableName +

"(name VARCHAR( 15 ) CHARACTER SET utf8 COLLATE utf8\_persian\_ci NOT NULL ," +

" city VARCHAR( 15 ) CHARACTER SET utf8 COLLATE utf8\_persian\_ci NOT NULL ," +

" phone VARCHAR( 15 ) NOT NULL, " +

"PRIMARY KEY (phone) )";

stm.execute(query);

}

private boolean insert()throws Exception{

String query = "INSERT INTO " + tableName + " VALUES(?, ?, ?)";

PreparedStatement pstm = this.con.prepareStatement(query);

pstm.setObject(1, Name.getText().trim());

pstm.setObject(2, City.getText().trim());

pstm.setObject(3, Phone.getText().trim());

int result = pstm.executeUpdate();

return (result > 0);

}

private boolean update(Object[] data)throws Exception{

String query = "UPDATE " + tableName + " SET name=?, city=? WHERE phone=?";

PreparedStatement pstm = con.prepareStatement(query);

pstm.setObject(1, data[0]);

pstm.setObject(2, data[1]);

pstm.setObject(3, data[2]);

int result = pstm.executeUpdate();

return (result > 0);

}

private boolean isExitsPhone() throws Exception{

String query = "SELECT COUNT(\*) FROM " + tableName + " WHERE phone=?";

PreparedStatement pstm = this.con.prepareStatement(query);

pstm.setObject(1, Phone.getText());

ResultSet result = pstm.executeQuery();

result.next();

return (result.getInt(1) > 0);

}

private boolean delete(Object data)throws Exception{

String query = "DELETE FROM " + tableName + " WHERE phone=?";

PreparedStatement pstm = this.con.prepareStatement(query);

pstm.setObject(1, data);

int result = pstm.executeUpdate();

return (result > 0);

}

private int deleteall()throws Exception{

Statement stm = con.createStatement();

int result = stm.executeUpdate("DELETE FROM " + tableName );

return result;

}

private Vector getData(String where) throws Exception{

Vector info = new Vector();

Statement stm = con.createStatement();

ResultSet result = stm.executeQuery("SELECT \* FROM " + tableName + where);

while(result.next()){

String name = result.getObject("name").toString(),

city = result.getObject("city").toString(),

phone = result.getObject("phone").toString();

info.addElement(new Person(name, city, phone));

}

return info;

}

private void showData(Vector data) throws Exception{

model = new PersonModel(data);

dataTable.setModel(model);

int[] width = {

80, 80, 120

};

int count = dataTable.getColumnModel().getColumnCount();

for(int i = 0; i < count; ++i){

dataTable.getColumnModel().getColumn(i).setMinWidth(width[i]);

dataTable.getColumnModel().getColumn(i).setMaxWidth(width[i] + 20);

}

// set phone column background

TableColumn phoneColumn = dataTable.getColumnModel().getColumn(2);

DefaultTableCellRenderer renderer = new DefaultTableCellRenderer();

renderer.setBackground(new Color(245, 245, 245));

phoneColumn.setCellRenderer(renderer);

}

public static void main(String args[]) {

try{

for(UIManager.LookAndFeelInfo laf : UIManager.getInstalledLookAndFeels()){

//Metal , Nimbus, CDE/Motif, Windows, Windows Classic

if(laf.getName().equals("Nimbus"))

UIManager.setLookAndFeel(laf.getClassName());

//System.out.println(laf.getName());

}

}catch(Exception e){

}

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

@Override

public void run() {

new Form().setVisible(true);

}

});

}

private class PersonModel extends AbstractTableModel{

private Vector<Person> items;

private String[] columns = {

"Name","City","Phone"

};

public static final int NAME = 0;

public static final int CITY = 1;

public static final int PHONE = 2;

public PersonModel(Vector items) {

this.items = items;

}

@Override public int getRowCount() {

return (items == null ? 0 : items.size());

}

@Override public int getColumnCount() {

return columns.length;

}

@Override public String getColumnName(int columnIndex) {

return columns[columnIndex];

}

@Override public boolean isCellEditable(int rowIndex, int columnIndex) {

return (columnIndex != PHONE);

}

@Override public void setValueAt(Object aValue, int rowIndex, int columnIndex) {

if(aValue.toString().equals(""))

return;

Person row = items.elementAt(rowIndex);

switch(columnIndex){

case NAME :

row.setName((String)aValue);

break;

case CITY :

row.setCity((String)aValue);

break;

case PHONE :

row.setPhone((String)aValue);

break;

}

// update database

try{

boolean edit = update(new Object[] {

getValueAt(rowIndex, NAME),

getValueAt(rowIndex, CITY),

getValueAt(rowIndex, PHONE)

});

if(!edit)

throw new Exception("an error occure at updating!");

}catch(Exception e){

JOptionPane.showMessageDialog(Form.this, e.getMessage());

}

}

@Override public Object getValueAt(int rowIndex, int columnIndex) {

Person info = items.elementAt(rowIndex);

switch(columnIndex){

case NAME : return info.getName();

case CITY : return info.getCity();

case PHONE : return info.getPhone();

}

return "NULL";

}

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton Backup;

private javax.swing.JTextField City;

private javax.swing.JButton DeleteAll;

private javax.swing.JButton Insert;

private javax.swing.JTextField Name;

private javax.swing.JTextField Phone;

private javax.swing.JButton Print;

private javax.swing.JButton Reset;

private javax.swing.JCheckBox cb\_horizLine;

private javax.swing.JCheckBox cb\_selCol;

private javax.swing.JCheckBox cb\_selRow;

private javax.swing.JCheckBox cb\_shwDlg;

private javax.swing.JCheckBox cb\_verticLine;

private javax.swing.JTable dataTable;

private javax.swing.JComboBox fieldName;

private javax.swing.JPanel infoPanel;

private javax.swing.JLabel label1;

private javax.swing.JLabel label2;

private javax.swing.JLabel label3;

private javax.swing.JLabel label4;

private javax.swing.JPanel mainPanel;

private javax.swing.JScrollPane scrollPane;

private javax.swing.JPanel searchPanel;

private javax.swing.JPanel settingPanel;

private javax.swing.JSlider slider;

private javax.swing.JPanel tablePanel;

private javax.swing.JTextField value;

// End of variables declaration//GEN-END:variables

}

Output:

