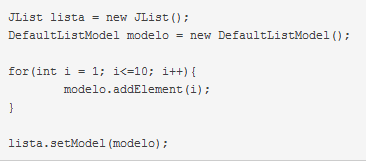


importar la librería javax.swing.\*

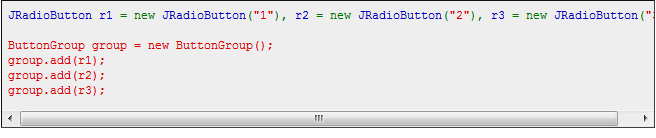




int index = list.getSelectedIndex();   
listModel.remove(index);

ListModel modelo=list.getModel();

DefaultListModel modelo = (DefaultListModel) jlist.getModel();   
int index = jlist.getSelectedIndex()   
modelo.remove( index )

DefaultListModel modelo = new DefaultListModel();   
JList lista = new JList (modelo);   
...   
  
modelo.addElement (unElemento);   
modelo.removeElementAt (3);   


import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.text.DateFormat;

import java.util.Date;

import java.util.Locale;

import javax.swing.DefaultListModel;

import javax.swing.JButton;

import javax.swing.JFrame;

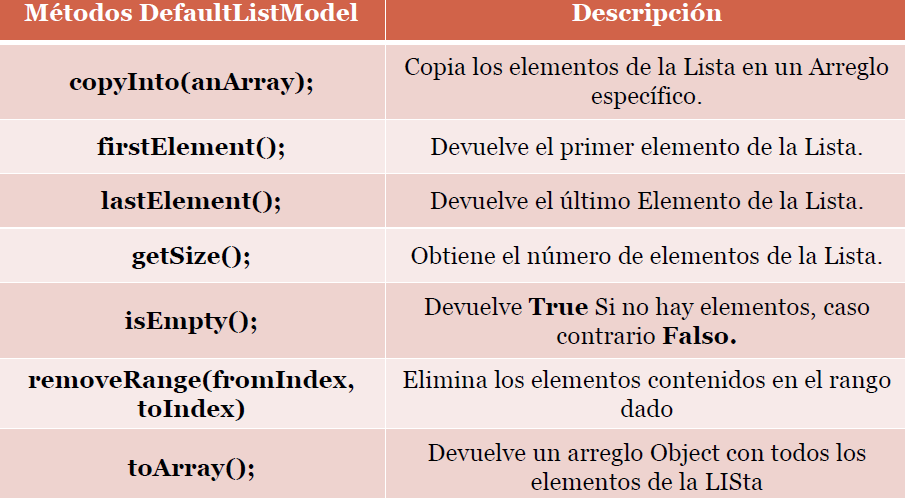
import javax.swing.JList;

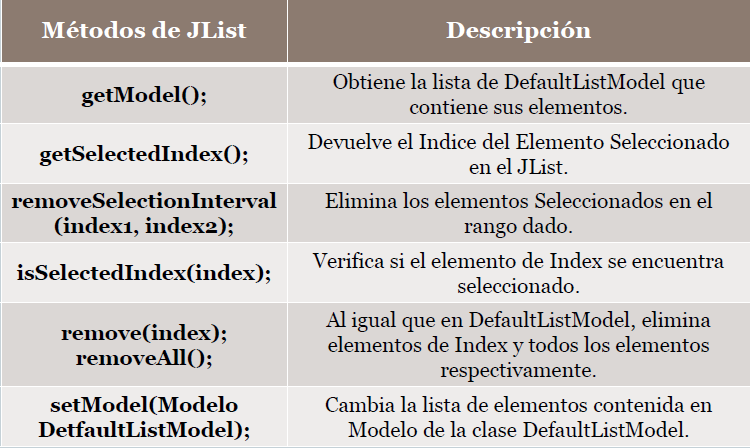
import javax.swing.JPanel;

import javax.swing.SwingUtilities;

**DefaultListModel**

**DefaultListModel** modelo = **new DefaultListModel**();





modelo.clear(); // para borrar toda la lista

modelo.addElement("un item"); // para añadir un item a la lista

modelo.removeElement(0); // Borra el item de la posición 0

DefaultListModel modelo = new DefaultListModel();

// Se puede hacer esto:

JList lista = new JList(modelo);

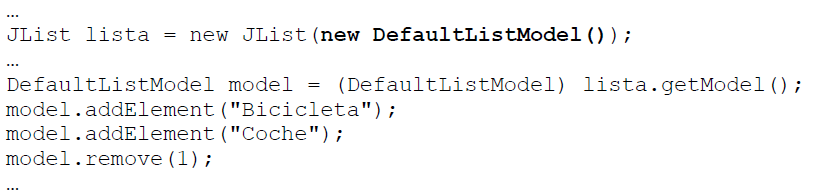
// o bien esto:

JList lista = new JList();

lista.setModel(modelo);

//O bien, directamente:

JList lista = new JList(new DefaultListModel());



Para que se entienda mejor:  
String item = jlist.getSelectedItem();  
Contact c = (Contact)hash.get(item);  
Ahora llenas los JTextField con los valores del contacto  
textField1.setText(c.getNombre());  
textField2.setText(c.getMail());  
....

package zetcode;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.text.DateFormat;

import java.util.Date;

import java.util.Locale;

import javax.swing.DefaultListModel;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JList;

import javax.swing.JPanel;

import javax.swing.SwingUtilities;

public class Example extends JFrame implements ActionListener {

private JList list;

private DefaultListModel model;

public Example() {

initUI();

}

public final void initUI() {

JPanel panel = new JPanel();

panel.setLayout(null);

model = new DefaultListModel();

list = new JList(model);

list.setBounds(150, 30, 220, 150);

JButton okButton = new JButton("Ok");

okButton.setBounds(30, 35, 80, 25);

okButton.addActionListener(this);

panel.add(okButton);

panel.add(list);

add(panel);

setTitle("Event object");

setSize(420, 250);

setLocationRelativeTo(null);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e) {

Locale locale = Locale.getDefault();

Date date = new Date(e.getWhen());

String s = DateFormat.getTimeInstance(DateFormat.SHORT,

locale).format(date);

if (!model.isEmpty()) {

model.clear();

}

if (e.getID() == ActionEvent.ACTION\_PERFORMED) {

model.addElement(" Event Id: ACTION\_PERFORMED");

}

model.addElement(" Time: " + s);

String source = e.getSource().getClass().getName();

model.addElement(" Source: " + source);

int mod = e.getModifiers();

StringBuffer buffer = new StringBuffer(" Modifiers: ");

if ((mod & ActionEvent.ALT\_MASK) > 0) {

buffer.append("Alt ");

}

if ((mod & ActionEvent.SHIFT\_MASK) > 0) {

buffer.append("Shift ");

}

if ((mod & ActionEvent.META\_MASK) > 0) {

buffer.append("Meta ");

}

if ((mod & ActionEvent.CTRL\_MASK) > 0) {

buffer.append("Ctrl ");

}

model.addElement(buffer);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

Example ex = new Example();

ex.setVisible(true);

}

});

}

}

**public** ArrayList darPalabras( )

{

**int** contador = 0;

ArrayList palabras = **new** ArrayList( );

**while**( contador < listaPalabras.getModel( ).getSize( ) )

{

palabras.add( listaPalabras.getModel( ).getElementAt( contador ) );

contador++;

}

**return** palabras;

}

/\*\*

\* Retorna la lista de teléfonos del contacto

\* @return - arrayList

\*/

**public** ArrayList darTelefonos( )

{

**int** contador = 0;

ArrayList telfs = **new** ArrayList( );

**while**( contador < listaTelefonos.getModel( ).getSize( ) )

{

telfs.add( listaTelefonos.getModel( ).getElementAt( contador ) );

contador++;

}

**return** telfs;

}

/\*\*

\* limpia los campos del panel

\*/

**public** **void** limpiarCampos( )

{

listaTelefonos.setModel( **new** DefaultListModel( ) );

listaPalabras.setModel( **new** DefaultListModel( ) );

campoNombre.setText( "" );

campoApellido.setText( "" );

campoDireccion.setText( "" );

campoCorreo.setText( "" );

}

/\*\*

\* Agrega un teléfono a la lista

\*/

**private** **void** agregarTelefono( )

{

String telefono = JOptionPane.showInputDialog( **this**, "Escriba el nuevo teléfono", "Nuevo Teléfono", JOptionPane.PLAIN\_MESSAGE );

DefaultListModel dlf = **new** DefaultListModel( );

**int** contador = 0;

**while**( contador < listaTelefonos.getModel( ).getSize( ) )

{

dlf.addElement( listaTelefonos.getModel( ).getElementAt( contador ) );

contador++;

}

dlf.addElement( telefono );

listaTelefonos.setModel( dlf );

}

/\*\*

\* Agrega una nueva palabra a la lista

\*/

**private** **void** agregarPalabra( )

{

String palabra = JOptionPane.showInputDialog( **this**, "Escriba la nueva palabra clave", "Nueva Palabra", JOptionPane.PLAIN\_MESSAGE );

DefaultListModel dlf = **new** DefaultListModel( );

**int** contador = 0;

**while**( contador < listaPalabras.getModel( ).getSize( ) )

{

dlf.addElement( listaPalabras.getModel( ).getElementAt( contador ) );

contador++;

}

dlf.addElement( palabra );

listaPalabras.setModel( dlf );

}

/\*\*

\* Elimina un teléfono de la lista

\*/

**private** **void** eliminarTelefono( )

{

String telefono = ( String )listaTelefonos.getSelectedValue( );

**if**( telefono == **null** )

{

JOptionPane.showMessageDialog( **this**, "Debe seleccionar un teléfono", "Lista de Contactos", JOptionPane.ERROR\_MESSAGE );

}

**else**

{

DefaultListModel dlf = **new** DefaultListModel( );

**int** contador = 0;

**while**( contador < listaTelefonos.getModel( ).getSize( ) )

{

String elemento = ( String )listaTelefonos.getModel( ).getElementAt( contador );

**if**( !elemento.equals( telefono ) )

dlf.addElement( listaTelefonos.getModel( ).getElementAt( contador ) );

contador++;

}

listaTelefonos.setModel( dlf );

}

}

/\*\*

\* Elimina una palabra de la lista

\*/

**private** **void** eliminarPalabra( )

{

String palabra = ( String )listaPalabras.getSelectedValue( );

**if**( palabra == **null** )

{

JOptionPane.showMessageDialog( **this**, "Debe seleccionar una palabra", "Lista de Contactos", JOptionPane.ERROR\_MESSAGE );

}

**else**

{

DefaultListModel dlf = **new** DefaultListModel( );

**int** contador = 0;

**while**( contador < listaPalabras.getModel( ).getSize( ) )

{

String elemento = ( String )listaPalabras.getModel( ).getElementAt( contador );

**if**( !elemento.equals( palabra ) )

dlf.addElement( listaPalabras.getModel( ).getElementAt( contador ) );

contador++;

}

listaPalabras.setModel( dlf );

}

}

/\*\*

\* Actualiza la lista con los elementos del arrayList

\* @param lista - la nueva lista de nombres para mostrar

\*/

**private** **void** actualizarListaTelefonos( ArrayList lista )

{

DefaultListModel dlf = **new** DefaultListModel( );

**int** contador = 0;

**while**( contador < lista.size( ) )

{

String tel = ( String )lista.get( contador );

dlf.addElement( tel );

contador++;

}

listaTelefonos.setModel( dlf );

}

/\*\*

\* Actualiza la lista con los elementos del arrayList

\* @param lista - la nueva lista de nombres para mostrar

\*/

**private** **void** actualizarListaPalabras( ArrayList lista )

{

DefaultListModel dlf = **new** DefaultListModel( );

**int** contador = 0;

**while**( contador < lista.size( ) )

{

dlf.addElement( lista.get( contador ) );

contador++;

}

listaPalabras.setModel( dlf );

}

/\*\*

\* Muestra la información del contacto dado

\* @param contacto contacto a mostrar

\*/

**public** **void** mostrarContacto( Contacto contacto )

{

campoNombre.setText( contacto.darNombre( ) );

campoApellido.setText( contacto.darApellido( ) );

campoDireccion.setText( contacto.darDireccion( ) );

campoCorreo.setText( contacto.darCorreo( ) );

actualizarListaPalabras( contacto.darPalabras( ) );

actualizarListaTelefonos( contacto.darTelefonos( ) );

}

/\*\*

\* Recibe los eventos realizados en los elementos de la interfaz y realiza acciones

\* @param evento El evento realizado por el usuario. evento != null

\*/

**public** **void** actionPerformed( ActionEvent evento )

{

String comando = evento.getActionCommand( );

**if**( comando.equals( AGREGAR ) )

{

ventanaPrincipal.agregarContacto( );

}

**else** **if**( comando.equals( MODIFICAR ) )

{

ventanaPrincipal.modificarContacto( );

}

**else** **if**( comando.equals( LIMPIAR ) )

{

limpiarCampos( );

}

**else** **if**( comando.equals( AGREGAR\_TELEFONO ) )

{

agregarTelefono( );

}

**else** **if**( comando.equals( AGREGAR\_PALABRA ) )

{

agregarPalabra( );

}

**else** **if**( comando.equals( ELIMINAR\_TELEFONO ) )

{

eliminarTelefono( );

}

**else** **if**( comando.equals( ELIMINAR\_PALABRA ) )

{

eliminarPalabra( );

}

}

}