package GUI;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.filechooser.FileNameExtensionFilter;

// @author LuisR

public class FileView extends javax.swing.JFrame {

/\*\*

\* Creates new form FileView

\*/

public FileView() {

initComponents();

}

/\*\*

\* 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() {

lblResultados = new javax.swing.JLabel();

btnSelect = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

taArchivo = new javax.swing.JTextArea();

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

btnSelect.setText("Seleccionar Archivo");

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

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

btnSelectActionPerformed(evt);

}

});

taArchivo.setEditable(false);

taArchivo.setColumns(20);

taArchivo.setRows(5);

jScrollPane1.setViewportView(taArchivo);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(14, 14, 14)

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 572, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addComponent(lblResultados, javax.swing.GroupLayout.PREFERRED\_SIZE, 411, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(btnSelect, javax.swing.GroupLayout.PREFERRED\_SIZE, 135, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(14, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(14, 14, 14)

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

.addComponent(btnSelect, javax.swing.GroupLayout.DEFAULT\_SIZE, 35, Short.MAX\_VALUE)

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

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 325, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(14, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void clean() {

String vacio = "";

lblResultados.setText(vacio);

taArchivo.setText(vacio);

}

private void cantLineasYPalabras(File archivo) {

int nPalabras = 0;

int nLineas = 0;

try {

BufferedReader lector = new BufferedReader(new FileReader(archivo));

StringBuilder textoArchivo = new StringBuilder();

String linea = lector.readLine();

while (linea != null) {

nLineas++;

textoArchivo.append(linea).append("\n");

linea = lector.readLine();

}

if (textoArchivo.length() != 0) {

int letraFinal = 0;

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

if (textoArchivo.charAt(i) >= 65 && textoArchivo.charAt(i) <= 90

|| textoArchivo.charAt(i) >= 97 && textoArchivo.charAt(i) <= 122

|| textoArchivo.charAt(i) >= 48 && textoArchivo.charAt(i) <= 57) {

letraFinal = i;

}

}

int temp = 0;

while (true) {

for (int i = temp; i <= letraFinal; i++) {

if (textoArchivo.charAt(i) >= 65 && textoArchivo.charAt(i) <= 90

|| textoArchivo.charAt(i) >= 97 && textoArchivo.charAt(i) <= 122

|| textoArchivo.charAt(i) >= 48 && textoArchivo.charAt(i) <= 57) {

temp = i;

break;

} else {

temp = i;

}

}

for (int i = temp; i <= letraFinal; i++) {

if (textoArchivo.charAt(i) == 32 || textoArchivo.charAt(i) == 10

|| i == letraFinal) {

nPalabras++;

temp = i;

break;

}

}

if (temp == letraFinal) {

break;

}

}

}

taArchivo.setText(textoArchivo.substring(0, textoArchivo.length() - 1));

lblResultados.setText("Número de Palabras: " + nPalabras + " | Número de líneas: " + nLineas);

} catch (FileNotFoundException ex) {

Logger.getLogger(FileView.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Seleccione sólo archivos de texto", "Error", JOptionPane.ERROR\_MESSAGE);

clean();

} catch (IOException ex) {

Logger.getLogger(FileView.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Seleccione sólo archivos de texto", "Error", JOptionPane.ERROR\_MESSAGE);

clean();

}

}

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

/\*

\* Muestra el cuadro de diálogo de archivos, para la selección del

\* archivo de texto que se desee abrir

\*/

JFileChooser selectorArchivos = new JFileChooser();

selectorArchivos.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);

// Filtra los tipos de archivos que se mostrarán en el selector de archivos

FileNameExtensionFilter filtro = new FileNameExtensionFilter("Archivos de Texto",

"java", "txt", "csv", "html",

"htm", "css", "js", "xml",

"json", "yaml", "yml", "md",

"bat", "cmd", "srt", "sql",

"ini", "properties", "rtf",

"log", "syslog", "htaccess",

"lnk");

selectorArchivos.setFileFilter(filtro);

// Indica cual fue la acción del usuario sobre el JFileChooser

int resultado = selectorArchivos.showOpenDialog(this);

// Obtiene el archivo seleccionado por el usuario

File archivo = selectorArchivos.getSelectedFile();

if (filtro.accept(archivo)) {

cantLineasYPalabras(archivo);

} else {

JOptionPane.showMessageDialog(null, "Seleccione sólo archivos de texto", "Error", JOptionPane.ERROR\_MESSAGE);

clean();

}

}

/\*\*

\* @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 | InstantiationException | IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {

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

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new FileView().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnSelect;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JLabel lblResultados;

private javax.swing.JTextArea taArchivo;

// End of variables declaration

}