

Práctica 9: Programación Orientada a Objetos (Manejo de Archivos)

Moisés García Villanueva

Noviembre de 2012

1. Archivos de texto

En esta actividad, vamos a crear un objeto ventana, al que se le agregará un menú de opciones. El menú se compone de un sólo objeto (el elemento Archivo), el cual contiene a su vez dos objetos (items): Abrir y Salir. En la ventana se agregará un área de texto, la cual servirá para mostrar el contenido del archivo de texto que se seleccione con el item Abrir del menú archivo. El programa que realiza las tareas descritas es el siguiente:

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class MenuFrame extends JFrame implements ActionListener {
/* ----- FIELDS ----- */
// Menues
/** File menu. */
private JMenu fileMenu;
// Other components
/** Text Area. */
private JTextArea textArea;
private JFileChooser abrirArchivo;
/* ----- */
/* */
/* CONSTRUCTORS */
/* */
/* ----- */
public MenuFrame() {
super("Manipular Imágenes");
// Content pane
Container container = getContentPane();
//container.setBackground(Color.pink);
// Create menus
createFileMenu();
// Create menu bar
JMenuBar bar = new JMenuBar();
setJMenuBar(bar);
bar.add(fileMenu);
// Add text area
textArea = new JTextArea(40, 10);
textArea.setEditable(false);
container.add(new JScrollPane(textArea),BorderLayout.CENTER);
}
/* CREATE FILE MENU */
/** Creates file menu */
private void createFileMenu() {
// Create file menu
fileMenu = new JMenu("Archivo");
fileMenu.setMnemonic('A');
// Create file menu items
JMenuItem aboutItem = new JMenuItem("Abrir");
aboutItem.setMnemonic('A');
aboutItem.setEnabled(true);
aboutItem.addActionListener(this);
JMenuItem exitItem = new JMenuItem("Salir");
exitItem.setMnemonic('x');
exitItem.setEnabled(true);
exitItem.addActionListener(this);
// Add to menu
```

```

fileMenu.add(aboutItem);
fileMenu.addSeparator();
fileMenu.add(exitItem);
}

/* ----- METHODS ----- */
/* ACTION PERFORMED */ /** Item handlers. @param event the triggered event. */
public void actionPerformed(ActionEvent event) {
if (event.getActionCommand().equals("Abrir")) abrir();
else if (event.getActionCommand().equals("Salir")) exitSystem();
else JOptionPane.showMessageDialog(this,"Error in event handler",
"Error: ",JOptionPane.ERROR_MESSAGE);
}
/* ABOUT */
/** Outputs JOptionPane pane if about menu item selected. */
private void abrir() {
// Utilizar la ventana de dialogo para abrir un archivo
if( abrirArchivo == null ) abrirArchivo = new JFileChooser();
//Con esto solamente podemos abrir archivos
abrirArchivo.setSelectionMode( JFileChooser.FILES_ONLY );
int seleccion = abrirArchivo.showOpenDialog( this );
if( seleccion == JFileChooser.APPROVE_OPTION )
{
File f = abrirArchivo.getSelectedFile();
try{
String nombre = f.getName();
String path = f.getAbsolutePath();
String contenido = getArchivo( path );
//Colocamos en el titulo de la aplicacion el
//nombre del archivo
this.setTitle( nombre );
//En el editor de texto colocamos su contenido
textArea.setText( contenido );

}catch( Exception exp){}
}
}
//-----//

//-----Se obtiene el contenido del Archivo-----//
public String getArchivo( String ruta ){
FileReader fr = null;
BufferedReader br = null;
//Cadena de texto donde se guardara el contenido del archivo
String contenido = "";
try{
//ruta puede ser de tipo String o tipo File
fr = new FileReader( ruta );
br = new BufferedReader( fr );

String linea;
//Obtenemos el contenido del archivo linea por linea
while( ( linea = br.readLine() ) != null ){
contenido += linea + "\n";
}

}catch( Exception e ){ }
//finally se utiliza para que si todo ocurre correctamente o si ocurre
//algún error se cierre el archivo que anteriormente abrimos
finally{
try{
br.close();
}catch( Exception ex ){ }
}
return contenido;
}
}
//-----//

/* EXIT */
/** Exits system */
private void exitSystem() {
System.exit(0);
}
}

```

```

public class VerTextos {
/* ----- FIELDS ----- */
// No fields
/* ----- METHODS ----- */
/** Main method (to start the "ball rolling"). */
public static void main(String Args[]) {
// Creat menu frame
MenuFrame menuFrame = new MenuFrame();
menuFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// Set frame size
menuFrame.setSize(500,400);
// Display frame
menuFrame.setVisible(true);
}
}

```

Actividades

1. Agregar Un nuevo elemento al menú para obtener algunas estadísticas del archivo.
2. Agregar un nuevo método para contar las vocales del archivo actualmente abierto. Invocar este método por medio de un item del menú recién creado.
3. Indicar cuantas líneas de texto contiene el archivo de texto.

2. Archivos de Imágenes

Ahora veamos como abrir imágenes utilizando Java. El siguiente código nos permite seleccionar el archivo de imagen que deseamos mostrar en nuestra área de imágenes.

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.imageio.*;
import java.awt.image.*;

class MenuFrame extends JFrame implements ActionListener {
/* ----- FIELDS ----- */
// Menues
/** File menu. */
private JMenu fileMenu;
// Other components
/** Text Area. Selector de archivos. Label o etiqueta para insertar imágenes */
private JTextArea textArea;
private JFileChooser abrirArchivo;
private JLabel imagenArea;
private BufferedImage img;

/* ----- */
/* */
/* CONSTRUCTORS */
/* */
/* ----- */
public MenuFrame() {
super("Manipular Imágenes");
// Content pane
Container container = getContentPane();
//container.setBackground(Color.pink);
// Create menus
createFileMenu();
// Create menu bar
JMenuBar bar = new JMenuBar();
setJMenuBar(bar);
bar.add(fileMenu);
// Add text area
/* textArea = new JTextArea(40, 10);
textArea.setEditable(false);
container.add(new JScrollPane(textArea),BorderLayout.CENTER);

```

```

*/ // Agregar la etiqueta para insertar la imagen
imagenArea=new JLabel("Hola",SwingConstants.CENTER);
container.add(imagenArea,BorderLayout.CENTER);
}
/* CREATE FILE MENU */
/** Creates file menu */
private void createFileMenu() {
// Create file menu
fileMenu = new JMenu("Archivo");
fileMenu.setMnemonic('A');
// Create file menu items
JMenuItem aboutItem = new JMenuItem("Abrir");
aboutItem.setMnemonic('A');
aboutItem.setEnabled(true);
aboutItem.addActionListener(this);
JMenuItem exitItem = new JMenuItem("Salir");
exitItem.setMnemonic('x');
exitItem.setEnabled(true);
exitItem.addActionListener(this);
// Add to menu
fileMenu.add(aboutItem);
fileMenu.addSeparator();
fileMenu.add(exitItem);
}

/* ----- METHODS ----- */
/* ACTION PERFORMED */
/** Item handlers.
@param event the triggered event. */
public void actionPerformed(ActionEvent event) {
if (event.getActionCommand().equals("Abrir")) abrir();
else if (event.getActionCommand().equals("Salir")) exitSystem();
else JOptionPane.showMessageDialog(this,"Error in event handler",
"Error: ",JOptionPane.ERROR_MESSAGE);
}
/* ABOUT */
/** Outputs JOptionPane pane if about menu item selected. */
private void abrir() {
// Utilizar la ventana de dialogo para abrir un archivo
if( abrirArchivo == null ) abrirArchivo = new JFileChooser();
//Con esto solamente podemos abrir archivos
abrirArchivo.setSelectionMode( JFileChooser.FILES_ONLY );
int seleccion = abrirArchivo.showOpenDialog( this );
if( seleccion == JFileChooser.APPROVE_OPTION )
{
File f = abrirArchivo.getSelectedFile();
try{
String nombre = f.getName();
String path = f.getAbsolutePath();
img = ImageIO.read(new File(path));
System.out.println("Se abrirá el archivo: "+path);
imagenArea.setIcon(new ImageIcon(img));
//Colocamos en el titulo de la aplicacion el nombre del archivo
this.setTitle( nombre );
this.repaint();
}catch( Exception exp){}
}
}

/* EXIT */
/** Exits system */
private void exitSystem() {
System.exit(0);
}

public class abrirImagenes {
/* ----- FIELDS ----- */
// No fields
/* ----- METHODS ----- */
/** Main method (to start the "ball rolling"). */
public static void main(String Args[]) {
// Creat menu frame
MenuFrame menuFrame = new MenuFrame();

```

```
menuFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// Set frame size
menuFrame.setSize(500,400);
// Display frame
menuFrame.setVisible(true);
}
}
```

Actividades

1. Modificar el programa para que aparezca el nombre del archivo en la etiqueta o área de la imagen.
2. Agregar un nuevo elemento a la barra de menú, para incluir elementos de edición de la imagen.
3. Obtener los valores RGB de la imagen.
Para obtener los valores de un píxel de la imagen, el procedimiento puede ser:
 - a) Almacenar en una variable entera el valor del píxel en RGB mediante el método `getRGB(x,y)`
 - b) Crear una nueva variable del tipo `Color` y asignarle el resultado de instanciar el constructor con el valor obtenido en el paso anterior. `colores=new Color(valorRGB)`
 - c) Utilizar los métodos `getRed()`, `getGreen()` y `getBlue()` para obtener los valores de Rojo, Verde y Azul del píxel en cuestión (x,y).
4. Cuando el usuario de click sobre la imagen proporcionarle el valor de los colores Rojo, Verde y Azul del píxel.