Swing 15 – Manejo Básico de Imágenes

Se comenzará cargando el proyecto llamado **SW15Base** el cual contiene dos ***JFrames***, ***ImagenFrm*** y ***GraficaFrm***, uno para el despliegue de imágenes y otro para el despliegue de gráficas usando la biblioteca ***JFreeChart***. Además de un ***JFrame*** llamado ***MenuFrm*** para la pantalla inicial.

Despliegue de una Imagen

Se utilizará el ***JFrame ImagenFrm*** para presentar una imagen a partir de un archivo incluido en el directorio ***imgs*** en el directorio del proyecto, **logo_uam.jpg**.

Se tiene un JPanel (pnlContenedor) para desplegar la imagen y un botón para ajustarla.

ImagenFrm.java

```
        ImageIcon icon;

public ImagenFrm() {
        initComponents();
        cargarImagen();
    }
```

Una de las formas más sencillas de colocar una imagen es asignarla a otro elemento y desplegarlo, en este caso se asignará a un JLabel.

ImagenFrm.java

```
private void cargarImagen(){
    icon = new ImageIcon("imgs/logo_uam.jpg");

    Image img = icon.getImage();
    int alto = img.getHeight(pnlContenedorImagen);
    int ancho = img.getWidth(pnlContenedorImagen);

    System.out.println(ancho + " " + alto);

    JLabel lbl   = new JLabel();
    lbl.setIcon(icon);
    pnlContenedorImagen.add(lbl);

    lbl.setBounds(5, 5, ancho, alto);

 }
```

Con esto la imagen se visualiza al abrir la pantalla correspondiente, sin embargo no se muestra de manera correcta, ya que no se ha ajustado su tamaño.

ImagenFrm.java

```
private void btnAjustarActionPerformed(java.awt.event.ActionEvent evt)
{

        Dimension d = pnlContenedorImagen.getSize();
        int alto = d.height;
        int ancho = d.width;

        Image img = icon.getImage();
        Image nuevaImagen = img.getScaledInstance(ancho, alto,
java.awt.Image.SCALE_SMOOTH);
        Icon nuevoIcono = new ImageIcon(nuevaImagen);

        JLabel lbl   = new JLabel();
        lbl.setIcon(nuevoIcono);
        pnlContenedorImagen.removeAll();
        lbl.setBounds(0, 0, ancho, alto);
        pnlContenedorImagen.add(lbl);

        repaint();

    }
```

Se debe recuperar la imagen y posteriormente darle el tamaño que tiene el contenedor (o uno determinado por el usuario), posteriormente se vuelve a asignar al componente.

Desplegando una Gráfica

Para desplegar una gráfica, se hará uso de la biblioteca ***JFreeChart*** que permite manejar de manera sencilla las gráficas en Java.

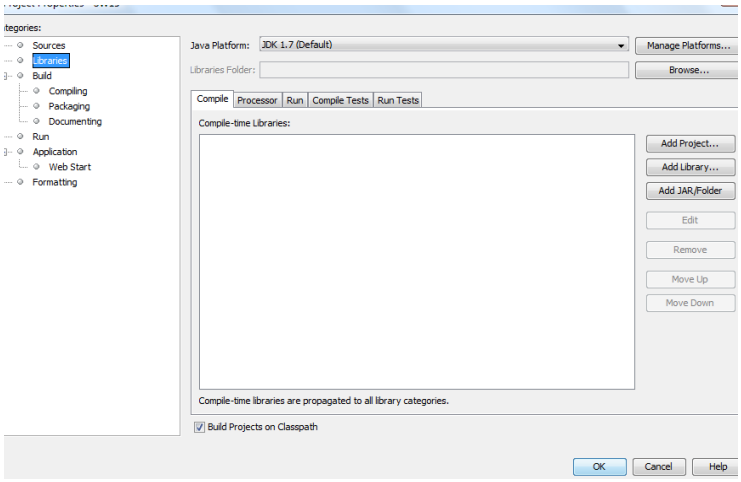
La biblioteca completa se puede descargar de la siguiente dirección:

http://www.jfree.org/jfreechart/download.html

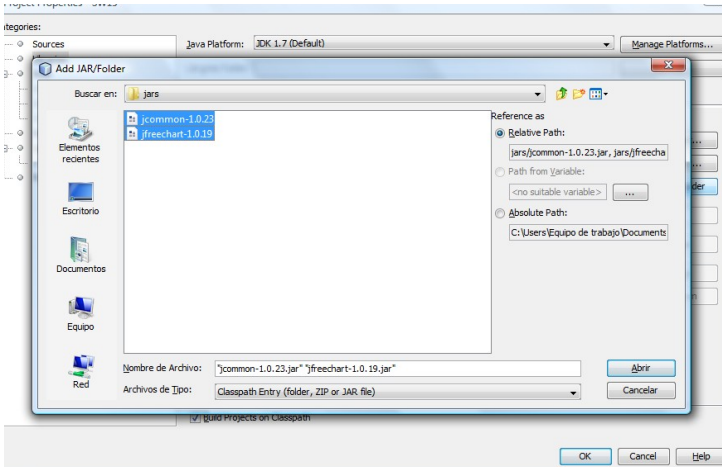
Después de descomprimir el archivo, se usarán para esta práctica solamente los archivos ***jcommon-1.0.23.jar*** y ***jfreechart-1.0.19.jar*** que se colocarán en un directorio llamado ***jars*** dentro del directorio del proyecto.

Estas dos bibliotecas se deben importar al proyecto. Se debe dar ***clic derecho*** sobre el nombre del proyecto, posteriormente seleccionar la opción de ***Properties***.

Una vez en la ventana, seleccionar la opción de ***Libraries***



Posteriormente, dar clic en el botón de Add JAR/Folder, buscar los dos archivos, seleccionarlos y dar clic en Abrir (Open).



Finalmente dar clic en ***OK*** para que queden cargadas al proyecto.

Se creará un nuevo paquete llamado ***uam.pvoe.graficas.operaciones*** con la clase ***DatosGraficas*** para llenar el conjunto de datos que necesita cada una de las gráficas a desplegar.

DatosGraficas.java

```
public XYDataset datosGrafica() {
        DefaultXYDataset ds = new DefaultXYDataset();
        double[][] datos = { {0.1, 0.2, 0.3}, {1, 2, 3} };
        ds.addSeries("Mis Resultados", datos);
        return ds;
    }
```

Esto llenará el conjunto de datos para la gráfica, en este caso con valores en los ejes X y Y.

DatosGrafica.java

```
public DefaultCategoryDataset  datosBarra(){

       String elementoUno = "Elemento 1";
       String elementoDos = "Elemento 2";
       String elementoTres = "Elemento 3";
       String categoriaA = "Categoría 1";
       String categoriaB = "Categoría 2";
       String categoriaC = "Categoría 3";
       String CategoriaD = "Categoría 4";

       DefaultCategoryDataset ds = new DefaultCategoryDataset( );

      ds.addValue( 1.0 , elementoUno , categoriaA );
      ds.addValue( 3.0 , elementoUno , categoriaB );
      ds.addValue( 5.0 , elementoUno , categoriaC );
      ds.addValue( 5.0 , elementoUno , CategoriaD );

      ds.addValue( 5.0 , elementoDos , categoriaA );
      ds.addValue( 6.0 , elementoDos , categoriaB );
      ds.addValue( 10.0 , elementoDos , categoriaC );
      ds.addValue( 4.0 , elementoDos , CategoriaD );

      ds.addValue( 4.0 , elementoTres , categoriaA );
      ds.addValue( 2.0 , elementoTres , categoriaB );
      ds.addValue( 3.0 , elementoTres , categoriaC );
      ds.addValue( 6.0 , elementoTres , CategoriaD );

      return ds;
    }
```

Aquí se llena la información para una gráfica de barras colocando lo que sería un valor en el eje Y y las categorías de cada uno de los elementos del eje X.

DatosGrafica.java

```
public PieDataset datosPastel( )
   {
      DefaultPieDataset ds = new DefaultPieDataset( );
      ds.setValue( "Opción 1" , new Double( 20 ) );
      ds.setValue( "Opción 2" , new Double( 40 ) );
      ds.setValue( "Opción 3" , new Double( 50 ) );
      ds.setValue( "Opción 4" , new Double( 10 ) );

      return ds;
   }
```

Finalmente se llena la información para la gráfica de pastel, con la opción y su valor correspondiente.

Posteriormente se asignará funcionalidad a los botones para que cada uno de ellos mande a desplegar una gráfica diferente.

GraficaFrm.java

```
private void btnGraficaActionPerformed(java.awt.event.ActionEvent evt)
{
        crearGrafica();
    }

private void btnBarraActionPerformed(java.awt.event.ActionEvent evt) {
        crearBarra();
    }

private void btnPastelActionPerformed(java.awt.event.ActionEvent evt) {
        crearPastel();
    }
```

GraficaFrm.java

```
private void crearGrafica(){
        pnlContenedor.removeAll();
        DatosGraficas datos = new DatosGraficas();
        XYDataset ds = datos.datosGrafica();
                JFreeChart chart =
ChartFactory.createXYLineChart("Gráfica de Datos",
                        "Datos X", "Datos Y", ds);
```

```
                ChartPanel cp = new ChartPanel(chart);
                cp.setVisible(true);

                Dimension dim = pnlContenedor.getSize();

                cp.setBounds(2, 2, dim.width, dim.height);
                pnlContenedor.add(cp);
                repaint();
    }
```

En el constructor se indica:

- Título de la gráfica (String)
- Título del eje X (String)
- Título del eje Y (String)
- Conjunto de datos (dataset)

Se le asigna un tamaño de acuerdo al contenedor y se visualiza.

GraficaFrm.java

```
private void crearBarra(){
        pnlContenedor.removeAll();
        DatosGraficas datos = new DatosGraficas();
        DefaultCategoryDataset dataset = datos.datosBarra();

        JFreeChart barChart = ChartFactory.createBarChart(
         "Mi Primera Gráfica de Barras",
         "Elementos", "Valor",dataset);

        ChartPanel cp = new ChartPanel(barChart);
        cp.setVisible(true);
       Dimension dim = pnlContenedor.getSize();

                cp.setBounds(2, 2, dim.width, dim.height);
                pnlContenedor.add(cp);
                repaint();

    }
```

En el método se indica:

- Título de la gráfica (String)
- Título del eje X (String)
- Título del eje Y (String)
- Conjunto de datos (dataset)

GraficaFrm.java

```
private void crearPastel(){
        pnlContenedor.removeAll();
        DatosGraficas datos = new DatosGraficas();
        PieDataset dataset = datos.datosPastel();

        JFreeChart pieChart = ChartFactory.createPieChart(
         "Mi Primer Gráfica de Pastel",dataset);

        ChartPanel cp = new ChartPanel(pieChart);
        cp.setVisible(true);
        Dimension dim = pnlContenedor.getSize();
        cp.setBounds(2, 2, dim.width, dim.height);
        pnlContenedor.add(cp);
        repaint();

    }
```

En el constructor se indica:

- Título de la gráfica (String)
- Conjunto de datos (dataset)