

JFreechart

JFreeChart es un marco de software open source para el lenguaje de programación Java, el cual permite la creación de gráficos complejos de forma simple.

JFreeChart es compatible con una serie de gráficas diferentes, incluyendo cuadros combinados. Después de tipos de gráficos son compatibles:

Tipos de graficas

- Gráficos XY (línea, spline y dispersión). Es posible usar un eje del tiempo.
- Gráfico circular.
- Diagrama de Gantt.
- Gráficos de barras (horizontal y vertical, apiladas e independientes). También tiene incorporado un dibujador de histogramas.
- Single valued (termómetro, brújula, indicador de velocidad) que luego se pueden colocar sobre el mapa.
- Varias gráficas específicas (tabla de viento, gráfica polar, burbujas de diferentes tamaños, etc.)

Además, los gráficos, es posible colocar varios marcadores en el área de gráfica.

Series de graficas

Las características que tendrá nuestra gráfica serán las siguientes:

- Cada serie (de tiempos por vuelta de un piloto) se representará de forma lineal.
- El eje Y de la gráfica mostrará los tiempos en segundos en un rango de 120 a 135. El número de segundos del eje se visualizará de dos en dos (120, 122, 124...).
- El eje X de la gráfica mostrará el número de vueltas, de la vuelta 1 a la vuelta 5.
- Cada punto de la gráfica debe de mostrarse con un círculo (evidentemente los círculos serán unidos por la línea que representará la serie).
- Los colores de las series deben ser distintos, así como las líneas de la gráfica que hacen de guía (líneas discontinuas que salen en el fondo de la gráfica).
- En el pie de la gráfica debe figurar una leyenda con los nombres de los pilotos.

Colecciones de series

JFreeChart dibuja automáticamente las escalas de los ejes y leyendas. Con el ratón informático se puede hacer zoom en la interfaz de la gráfica automáticamente y cambiar algunos ajustes a través del menú local. Las

tablas existentes pueden actualizarse fácilmente a través de los oyentes (listeners) que la biblioteca tiene en sus colecciones de datos.

Frames con la gráfica incluida

```
1. JFrame frame = new JFrame("CrearGraficos - LineaDeCodigo");
2. frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
3. frame.setLayout(new GridLayout(2, 1));
4. frame.add(crearGraficoTorta(array));
5. frame.add(crearGraficoBarras(array));
6. frame.pack();
7. frame.setLocationRelativeTo(null);
8. frame.setVisible(true);
```

En este código hemos añadido los gráficos mediante el método `.add()` del `JFrame`. Adicionalmente hemos **redimensionado el tamaño del frame, al mínimo posible con el método `.pack()` y centrado el frame con `.setLocationRelativeTo(null)`.**

```
public class <b>ventanaGrafica</b> extends java.awt.Frame {
```

```
    BufferedImage grafica = null;
```

```
    /** Creates new form ventanaGrafica */
```

```
    public <b>ventanaGrafica</b>() {
```

```
        initComponents();
```

```
    }
```

JSlider

Constructor	Descripción
-------------	-------------

JSlider ()	crea un control deslizante con el valor inicial de 50 y un rango de 0 a 100.
JSlider (orientación int)	crea un control deslizante con la orientación especificada establecida por JSlider.HORIZONTAL o JSlider.VERTICAL con el rango de 0 a 100 y el valor inicial 50.
JSlider (int min, int max)	crea un control deslizante horizontal utilizando los valores mínimos y máximos dados.
JSlider (int min, int max, int value)	crea un control deslizante horizontal utilizando los valores mínimos, máximos y mínimos dados.
JSlider (orientación int, int min, int max, valor int)	crea un control deslizante utilizando la orientación dada, min, max y value.

Métodos de uso común de la clase JSlider

Método	Descripción
public void setMinorTickSpacing (int n)	se usa para establecer el espaciado menor de ticks en el control deslizante.
public void setMajorTickSpacing (int n)	se usa para establecer el espaciado de tick principal en el control deslizante.
setPaintTicks vacío público (boolean b)	se utiliza para determinar si se pintan las marcas de graduación.
setPaintLabels vacío público (boolean b)	se usa para determinar si las etiquetas están pintadas.
setPaintTracks vacío público (boolean b)	se usa para determinar si la pista está pintada.

Ejemplo de Java JSlider

1. **import** javax.swing. *;
2. **público de clase** SliderExample1 **extiende** JFrame {
3. **public** SliderExample1 () {
4. Control deslizante JSlider = **nuevo** JSlider (JSlider.HORIZONTAL, 0 , 50 , 25);
5. Panel JPanel = **nuevo** JPanel ();
6. panel.add (control deslizante);
7. agregar (panel);
8. }
- 9.
10. **public static void** main (Cadena s []) {
11. SliderExample1 frame = **new** SliderExample1 ();
12. frame.pack ();
13. frame.setVisible (**verdadero**);
14. }
15. }