

# El control barra de desplazamiento

El control barra de desplazamiento se emplea frecuentemente en los interfaces gráficos de usuario, cuando la figura o el texto que queremos ver es más grande que las dimensiones de la ventana. Dicho control es mucho más manejable que un control de edición, ya que basta arrastrar el dedo de la barra de desplazamiento, en vez de introducir un dato en un control de edición y pulsar el botón apropiado. En una barra de desplazamiento no es necesario verificar el dato que se ha introducido, ya que se trata de números enteros en un determinado intervalo.

## El control *Scrollbar*

Un control barra de desplazamiento es un objeto de la clase *Scrollbar*. Creamos un control *sbRadio* llamando al constructor por defecto

```
Scrollbar sbRadio = new Scrollbar();
```

Establecemos su orientación mediante la función miembro *setOrientation* pasándole un cero o un uno. La orientación horizontal es cero, o la constante *Scrollbar.HORIZONTAL*, y la orientación vertical es uno o la constante *Scrollbar.VERTICAL*

```
sbRadio.setOrientation(0);
```

El valor máximo y mínimo lo establecemos mediante las funciones miembro *setMaximum* y *setMinimum*, respectivamente.

```
sbRadio.setMaximum(110);  
sbRadio.setMinimum(10);
```

Cuando actuamos con el ratón sobre las flechas situadas en los extremos de la barra, el dedo se desplaza una determinada cantidad denominada unidad y que se establece mediante la función miembro *setUnitIncrement*.

```
sbRadio.setUnitIncrement(5);
```

Cuando se actúa con el ratón en las dos regiones de la barra situadas entre las flechas y el dedo, éste se desplaza una determinada cantidad denominada página o bloque, y se establece mediante llamada a la función miembro *setBlockIncrement*.

```
sbRadio.setBlockIncrement(25);
```

La posición del dedo en la barra de desplazamiento se establece en un valor entre el mínimo y el máximo

```
sbRadio.setValue(10);
```

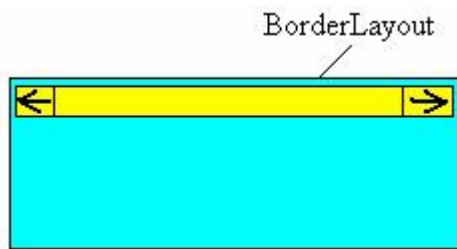
Para obtener la posición del dedo en la barra de desplazamiento se llama a la función miembro *getValue* que devuelve un entero

```
int pos=sbRadio.getValue();
```

## Propósito

En este applet hemos situado una barra de desplazamiento en la parte superior del applet. A medida que movemos el dedo de la barra de desplazamiento, el radio de la circunferencia que se dibuja aumenta.

## Diseño



Crear el applet y situar un control barra de desplazamiento (*Scrollbar*) en la parte superior

Establecer las propiedades del control en su hoja de propiedades

Establecer [\*BorderLayout\*](#) como gestor de diseño del applet de modo que la barra de desplazamiento quede al norte (NORTH).

Redefinir la función *paint* para dibujar una circunferencia (oval) centrada en el applet

Se recuerda al lector, que cada vez que se pretenda dibujar una circunferencia se debe llamar a la función *paint* mediante *repaint*.

## Respuesta a las acciones del usuario

La respuesta a la acción del usuario sobre una barra de desplazamiento la podemos resumir en los siguientes puntos:

- Se crea una clase que implemente el interface *AdjustmentListener*, y que ha de definir *adjustmentValueChanged*

```
class MueveBarra implements AdjustmentListener{
    private BarraApplet applet;
    public MueveBarra(BarraApplet applet){
        this.applet=applet;
    }
    public void adjustmentValueChanged(AdjustmentEvent ev){
        applet.funcionRespuesta (ev);
    }
}
```

- Mediante *addAdjustmentListener* se conecta el componente con un objeto de la clase que maneja los sucesos originados en dicho componente.

```
sbRadio.addAdjustmentListener(new MueveBarra(this));
```

- La información acerca del suceso viene encapsulada en un objeto de la clase *AdjustmentEvent*.

En la definición de la función respuesta *funcionRespuesta*, obtenemos el valor del dedo en la barra de desplazamiento mediante la función miembro *getValue*, y la guardamos en el miembro dato *radio*.

```
void funcionRespuesta(AdjustmentEvent ev) {
    radio=sbRadio.getValue();
    repaint();
}
```

Para dibujar una circunferencia cuyo centro está situado en el centro del applet, y cuyo radio sea el valor que indica el dedo en la barra de desplazamiento redefinimos la función miembro *paint*.

```
public void paint(Graphics g){
    int x1=getSize().width/2;
    int y1=getSize().height/2;
    g.setColor(Color.black);
    g.drawOval(x1-radio, y1-radio, 2*radio, 2*radio);
}
```

Las dimensiones del applet se obtienen mediante la función miembro *getSize*, que devuelve un objeto de la clase *Dimension*, que tiene dos miembros dato *width*, que proporciona la anchura del applet y *height* que proporciona la altura del applet.

El código completo de este ejemplo, es el siguiente

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class BarraApplet extends Applet {
    Scrollbar sbRadio = new Scrollbar();
    BorderLayout borderLayout1 = new BorderLayout();
    int radio=10;

    public void init() {
        sbRadio.setOrientation(0);
        sbRadio.setValue(10);
        sbRadio.setMaximum(110);
        sbRadio.setUnitIncrement(5);
        sbRadio.setBlockIncrement(25);
        sbRadio.setMinimum(10);
        sbRadio.addAdjustmentListener(new MueveBarra(this));
        this.setLayout(borderLayout1);
        this.add(sbRadio, BorderLayout.NORTH);
    }
}
```

```
void funcionRespuesta(AdjustmentEvent ev) {
    radio=sbRadio.getValue();
    repaint();
}

public void paint(Graphics g){
    int x1=getSize().width/2;
    int y1=getSize().height/2;
    g.setColor(Color.black);
    g.drawOval(x1-radio, y1-radio, 2*radio, 2*radio);
}
}
//*****
class MueveBarra implements AdjustmentListener{
    private BarraApplet applet;
    public MueveBarra(BarraApplet applet){
        this.applet=applet;
    }
    public void adjustmentValueChanged(AdjustmentEvent ev){
        applet.funcionRespuesta (ev);
    }
}
```