Swing 07 – Otros elementos de Selección

En esta práctica se revisarán conceptos básicos sobre el manejo de otros elementos de selección , como son las Listas, *Spinners* y *Sliders*. Se creará un proyecto llamado SW07 sin crear una clase Principal. Crear también un paquete llamado ***uam.pvoe.sw.otros.formas*** y uno llamado ***uam.pvoe.sw.otros.modelo***

En el paquete para las formas, se creará un nuevo ***Frame*** llamado ***FrmOtrosElementos***

Configurar la interfaz con las siguientes propiedades:

Ancho Máximo: **800 x 600**
Ancho Mínimo: **800 x 600**
No permitir **redimensionar**
Título: **Otros Elementos de Selección**

Colocar una etiqueta (***lblTitulo***) con la leyenda "Otros Elementos de Selección"

Colocar un elemento del tipo Spinner cuyo nombre será (***spnrSencillo***), agregarle una etiqueta (***lblSpnrSencillo***) con la leyenda "***Spinner Sencillo***"

Se colocará un botón (***btnLeer***) con la leyenda "Leer Datos" para ir leyendo los valores introducidos en los diferentes elementos.

Por el momento se leerá solo el dato del spinner sencillo y se imprimirá.

FrmOtrosElementos.java

```
private void btnLeerActionPerformed(java.awt.event.ActionEvent evt) {
    Integer datoSencillo = (Integer)spnrSencillo.getValue();
    System.out.println("Dato Leído y Multiplicado por 10: " + datoSencillo * 10);
}
```

El dato se recupera como un objeto con el método ***getValue***, por lo que hay que pasarlo a entero (***Integer*** o ***int***) para poder tratarlo como un número.

Rangos e Incremento

Es posible agregar un valor mínimo, un valor máximo, un valor inicial y un incremento a un ***spinner***.
Colocar un elemento del tipo Spinner cuyo nombre será (***spnrRangos***), agregarle una etiqueta (***lblSpnrRangos***) con la leyenda "***Spinner Rangos***"

FrmOtrosElementos.java

```
public FrmOtrosElementos() {
    initComponents();
    asignaRangos();
}
```

FrmOtrosElementos.java

```
public void asignaRangos(){
        SpinnerModel spnrModelo = new SpinnerNumberModel(10,5,20,3);
        spnrRangos.setModel(spnrModelo);
    }
```

En este caso se crea un objeto del tipo ***SpinnerModel*** que indica los siguientes valores:

- Valor inicial: 10
- Valor mínimo: 5
- Valor máximo: 20
- Incremento 3

Después de creado el modelo, se le asigna al *spinner*.

Spinner con Flotantes

Es posible crear un *spinner* que acepte incrementos con punto decimal (flotante), para eso se debe construir un modelo como en el caso de los rangos, especificando que el incremento es con punto decimal.

Colocar un elemento del tipo Spinner cuyo nombre será (***spnrFlotante***), agregarle una etiqueta (***lblSpnrFlotante***) con la leyenda "***Spinner Flotante***"

FrmOtrosElementos.java

```
public FrmOtrosElementos() {
        initComponents();
        asignaRangos();
        crearFlotante();
    }
```

FrmOtrosElementos.java

```
public void crearFlotante(){
        SpinnerModel spnrModelo = new SpinnerNumberModel(0,-2,2,0.05);
        spnrFlotante.setModel(spnrModelo);
    }
```

Spinner y Cadenas

Aunque no es común, también pueden crearse spinners que almacenen cadenas.

Colocar un elemento del tipo Spinner cuyo nombre será (***spnrCadenas***), agregarle una etiqueta (***lblSpnrCadenas***) con la leyenda "***Spinner Cadenas***"

FrmOtrosElementos.java

```
    public FrmOtrosElementos() {
        initComponents();
        asignaRangos();
        crearFlotante();
        llenarCadenas();
    }
```

FrmOtrosElementos.java

```
public void llenarCadenas(){
        String cadenas[] = {"HOLA","MUNDO","CRUEL"};
    SpinnerModel modeloCadenas = new SpinnerListModel(cadenas);
    spnrCadenas.setModel(modeloCadenas);
    }
```

Se crea un arreglo de cadenas con los valores a agregar. Se crea un modelo nuevo con el constructor ***SpinnerListModel*** y se asigna al *spinner*.

***Spinner* y Objetos**

Es posible llenar un spinner con objetos de una determinada clase, para esto en el paquete ***uam.pvoe.sw.otros.modelo*** se crearán dos clases: ***DatoSpinner*** y ***Operaciones***.

DatoSpinner.java

```
package uam.pvoe.sw.otros.modelo;

public class DatoSpinner {

    private String llave;
    private String valor;

    public DatoSpinner(String k, String v){
        llave = k;
        valor = v;
    }

    public String getLlave() {
        return llave;
    }

    public void setLlave(String llave) {
        this.llave = llave;
    }
```

```
    public String getValor() {
        return valor;
    }

    public void setValor(String valor) {
        this.valor = valor;
    }

    public String toString(){
        String mensaje = "";
        return valor;
    }

}
```

Operaciones.java

```
package uam.pvoe.sw.otros.modelo;

import java.util.LinkedList;

public class Operaciones {

    public LinkedList<DatoSpinner> llenarListaSpinner(){
        LinkedList<DatoSpinner> listaObjetos = new LinkedList();
        DatoSpinner dato1 = new DatoSpinner("KEY1","Valor 1 en el Spinner");
        DatoSpinner dato2 = new DatoSpinner("KEY2","Valor 2 en el Spinner");
        DatoSpinner dato3 = new DatoSpinner("KEY3","Valor 3 en el Spinner");

        listaObjetos.add(dato1);
        listaObjetos.add(dato2);
        listaObjetos.add(dato3);

        return listaObjetos;

    }
}
```

Colocar un elemento del tipo ***Spinner*** cuyo nombre será (***spnrObjetos***), agregarle una etiqueta (***lblSpnrObjetos***) con la leyenda "***Spinner Objetos***"

FrmOtrosElementos.java

```
    public FrmOtrosElementos() {
        initComponents();
        asignaRangos();
```

```
        crearFlotante();
        llenarCadenas();
        llenarObjetosSpinner();
    }
```

FrmOtrosElementos.java

```
    public void llenarObjetosSpinner(){
        Operaciones operaciones = new Operaciones();
        LinkedList<DatoSpinner> listaObjetos = operaciones.llenarListaSpinner();

        SpinnerModel modeloListas = new SpinnerListModel(listaObjetos);
        spnrObjetos.setModel(modeloListas);

    }
```

Se crea nuevamente un modelo con el método ***SpinnerListModel***, pero en esta ocasión recibe un objeto del tipo ***LinkedList***. Posteriormente se agrega al *spinner* correspondiente.

Notar que el texto que aparece en el *spinner* es el que se regresa en el método ***toString()*** de la clase de la cuál se crean los objetos que llenan la lista asignada en el modelo.

Leyendo el resto de los Spinners

Finalmente se leerán los datos introducidos en los spinners que faltaban, esto será al presionar el botón de captura.

FrmOtrosElementos.java

```
private void btnLeerActionPerformed(java.awt.event.ActionEvent evt) {
        Integer datoSencillo = (int)spnrSencillo.getValue();
        Double datoFloat = (double)spnrFlotante.getValue();
        String cadena = (String)spnrCadenas.getValue();
        DatoSpinner dato = (DatoSpinner)spnrObjetos.getValue();
        System.out.println("Dato Leído y Multiplicado por 10: " + datoSencillo * 10);
        System.out.println("El Flotante tiene: " + datoFloat);
        System.out.println("El de Cadenas tiene: " + cadena);
        System.out.println("El de Objetos: " + dato.getLlave() + " " + dato.getValor());

    }
```

Manejo de Sliders

Se colocará un elemento del tipo Slider, junto con una etiqueta (“Slider Sencillo”) y un campo de texto, los nombres de las variables serán

- ***sldrSencillo***
- ***lblSliderSencillo***
- ***txtSliderSencillo***

Para comprobar la funcionalidad del *Slider*, se le asignará un evento del tipo ***StateChanged***, al mover el punto de desplazamiento, se desplegará en la caja de texto el valor correspondiente.

FrmOtrosElementos.java

```
private void sldrSencilloStateChanged(javax.swing.event.ChangeEvent evt) {
    int valor = sldrSencillo.getValue();
    txtSliderSencillo.setText(""+valor);
}
```

También se puede leer el valor al momento de presionar el botón de Lectura

FrmOtrosElementos.java

```
private void btnLeerActionPerformed(java.awt.event.ActionEvent evt) {
    Integer datoSencillo = (int)spnrSencillo.getValue();
    Double datoFloat = (double)spnrFlotante.getValue();
    String cadena = (String)spnrCadenas.getValue();
    DatoSpinner dato = (DatoSpinner)spnrObjetos.getValue();

    int valorSliderSencillo = sldrSencillo.getValue();

    System.out.println("Dato Leído y Multiplicado por 10: " + datoSencillo * 10);
    System.out.println("El Flotante tiene: " + datoFloat);
    System.out.println("El de Cadenas tiene: " + cadena);
    System.out.println("El de Objetos: " + dato.getLlave() + " " + dato.getValor());
    System.out.println("Del Slider Sencillo: " + valorSliderSencillo);

}
```

Slider con Marcas y Valores Máximos y Mínimos

Es posible crear un *Slider* y que las marcas sena visibles además de asignarle un valor máximo, un valor mínimo y un valor por defecto.

Se colocará un elemento del tipo Slider, junto con una etiqueta (“Slider Configurable”) y un campo de texto, los nombres de las variables serán

- ***sldrConfigurable***
- ***lblSliderConfigurable***
- ***txtSliderConfigurable***

Para comprobar la funcionalidad del *Slider*, se le asignará un evento del tipo ***StateChanged***, al mover el punto de desplazamiento, se desplegará en la caja de texto el valor correspondiente.

FrmOtrosElementos.java

```
private void sldrSliderConfigurableStateChanged(javax.swing.event.ChangeEvent evt) {
        int valor = sldrSliderConfigurable.getValue();
        txtSliderConfigurable.setText(""+valor);
    }
```

También se puede leer el valor al momento de presionar el botón de Lectura

FrmOtrosElementos.java

```
private void btnLeerActionPerformed(java.awt.event.ActionEvent evt) {
        Integer datoSencillo = (int)spnrSencillo.getValue();
        Double datoFloat = (double)spnrFlotante.getValue();
        String cadena = (String)spnrCadenas.getValue();
        DatoSpinner dato = (DatoSpinner)spnrObjetos.getValue();

        int valorSliderSencillo = sldrSencillo.getValue();
        int valorSliderConfigurable = sldrSliderConfigurable.getValue();


        System.out.println("Dato Leído y Multiplicado por 10: " + datoSencillo * 10);
        System.out.println("El Flotante tiene: " + datoFloat);
        System.out.println("El de Cadenas tiene: " + cadena);
        System.out.println("El de Objetos: " + dato.getLlave() + " " + dato.getValor());
        System.out.println("Del Slider Sencillo: " + valorSliderSencillo);
        System.out.println("Del Slider Configurable: " + valorSliderConfigurable);

    }
```

Configurando el *Slider*

Los métodos (*set*) para configurar el comportamiento de un slider son los siguientes:

FrmOtrosElementos.java

```
public void configurarSlider(){
   sldrSliderConfigurable.setMinimum(0);
   sldrSliderConfigurable.setMaximum(10);
   sldrSliderConfigurable.setValue(0);
   sldrSliderConfigurable.setMinorTickSpacing(2);
   sldrSliderConfigurable.setMajorTickSpacing(2);
   sldrSliderConfigurable.setPaintTicks(true);
   sldrSliderConfigurable.setPaintLabels(true);
   sldrSliderConfigurable.setSnapToTicks(true);
   }
```

- **setMinimum**() permite especificar el valor mínimo (incluyendo negativos)
- **setMaximum**() permite especificar el valor máximo
- **setValue**() especifica el valor por defecto
- **setMinorTickSpacing**() permite especificar el valor mínimo entre cada movimiento
- **setMajorTickSpacing**() permite especificar el valor máximo de separación (se observa en la escala)

Adicionalmente se puede especificar que se muestren los puntos, valores e incluso la recta

- **setPaintTicks**() permite mostrar o no los puntos
- **setPaintLabels**() permite mostrar o no los valores de la escala (setMajorTickSpacing)
- **setSnapToTicks**() hace que un cambio en la barra de desplazamiento tome el valor del señalador más cercano

Slider con Cadenas

No es posible crear un *slider* cadenas, lo que se debe realizar es manejar un mapa *hash* que asocie un valor a una determinada posición, además de otros métodos para asignar los valores al *slider*.

Se creará un *slider* que permita seleccionar algún día de la semana.

Se colocará un elemento del tipo Slider, junto con una etiqueta (“Slider Semana”) y un campo de texto, los nombres de las variables serán

- ***sldrSemana***
- ***lblSliderSemana***
- ***txtSliderSemana***

Primero se creará un método para asignar un valor numérico a cada uno de los días de la semana:

Se configurará para que tenga valores de 1 a 7 (los días de la semana)

FrmOtrosElementos.java

```
public void configurarSemana(){
   sldrSemana.setMinimum(1);
   sldrSemana.setMaximum(7);
   sldrSemana.setValue(1);
   sldrSemana.setMinorTickSpacing(1);
   sldrSemana.setPaintTicks(true);
   sldrSemana.setPaintLabels(true);
   sldrSemana.setSnapToTicks(true);
   }
```

Operaciones.java

```
   public Hashtable<Integer,JLabel>diasSemana(){

       Hashtable<Integer,JLabel>diasSemana = new Hashtable<Integer,JLabel>();


   diasSemana.put(1, new JLabel("Lunes"));
   diasSemana.put(2, new JLabel("Martes"));
   diasSemana.put(3, new JLabel("Miércoles"));
   diasSemana.put(4, new JLabel("Jueves"));
   diasSemana.put(5, new JLabel("Viernes"));
   diasSemana.put(6, new JLabel("Sábado"));
   diasSemana.put(7, new JLabel("Domingo"));

   return diasSemana;

   }
```

Posteriormente se configurará el *slider* para que presente esos días.

FrmOtrosElementos.java

```
public void configurarSemana(){
   sldrSemana.setMinimum(1);
   sldrSemana.setMaximum(7);
   sldrSemana.setValue(1);
   sldrSemana.setMinorTickSpacing(1);
   sldrSemana.setPaintTicks(true);
   sldrSemana.setPaintLabels(true);
   sldrSemana.setSnapToTicks(true);
```

```
    Operaciones operaciones = new Operaciones();
    Hashtable<Integer,JLabel>diasSemana = operaciones.diasSemana();
    sldrSemana.setLabelTable(diasSemana);
  }
```

Se invoca el método para configurar el slider

FrmOtrosElementos.java

```
    public FrmOtrosElementos() {
        initComponents();
        asignaRangos();
        crearFlotante();
        llenarCadenas();
        llenarObjetosSpinner();
        configurarSlider();
        configurarSemana();
    }
```

Solo resta imprimir el valor (tanto en posición como en texto) del día seleccionado

FrmOtrosElementos.java

```
private void btnLeerActionPerformed(java.awt.event.ActionEvent evt) {
        Integer datoSencillo = (int)spnrSencillo.getValue();
        Double datoFloat = (double)spnrFlotante.getValue();
        String cadena = (String)spnrCadenas.getValue();
        DatoSpinner dato = (DatoSpinner)spnrObjetos.getValue();

        int valorSliderSencillo = sldrSencillo.getValue();
        int valorSliderConfigurable = sldrSliderConfigurable.getValue();

        Integer diaSeleccionado =(sldrSemana.getValue());
        Hashtable<Integer,JLabel>diasSemana = (Hashtable)sldrSemana.getLabelTable();
        String dia = diasSemana.get(diaSeleccionado).getText();

        System.out.println("Dato Leído y Multiplicado por 10: " + datoSencillo * 10);
        System.out.println("El Flotante tiene: " + datoFloat);
        System.out.println("El de Cadenas tiene: " + cadena);
        System.out.println("El de Objetos: " + dato.getLlave() + " " + dato.getValor());
        System.out.println("Del Slider Sencillo: " + valorSliderSencillo);
        System.out.println("Del Slider Configurable: " + valorSliderConfigurable);
        System.out.println("El día de la semana: " + diaSeleccionado + " " + dia);
    }
```