

Estructura condicionales y componentes swing en Java

Carrera: Computación e Informática

Semestre: 2016 - I

Nombre de Unidad Didáctica: Lógica de Programación



Estructura selectiva simple y doble

- Muchas veces tenemos que decidir y realizar una u otra tarea dependiendo de una condición, en la programación existe una estructura que permite evaluar una condición (expresión lógica que devuelve verdadero o falso) y determina que instrucción o instrucciones se debe ejecutar si la condición es verdadera o si la condición es falsa.

Sintaxis en java:

If <expr. Log> (para una instrucción)

- <instrucción 1>;

If <expr. Log> (para varias instrucciones)

- <instruccion 2>;
- <instruccion n>;

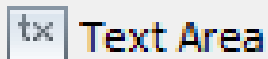
```
if(sueldo>350){  
    descuento=0.30*sueldo  
}
```

```
if( (a>b)  && (b>c) ){  
    menor=c ;  
}
```

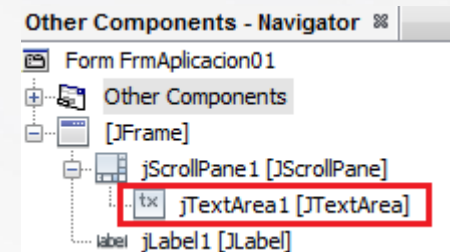
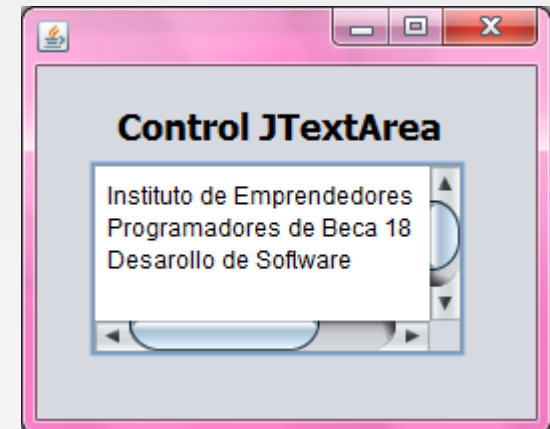
```
if((a%2==0) || (a%5==0) ){  
    System.out.println("estos son los numeros divisibles por 2 y 5 ");  
}
```

Control JTextArea

JTextArea es un control para pedir varias líneas de texto, que a diferencia de **TextField**, solo nos permitía ingresar una línea de texto.



```
public class FrmAplicacion01 extends javax.swing.JFrame {  
  
    public FrmAplicacion01() {  
        initComponents();  
        jTextArea1.setText("Instituto de Emprendedores \n"+  
            "Programadores de Beca 18 \n"+  
            "Desarrollo de Software");  
    }  
}
```



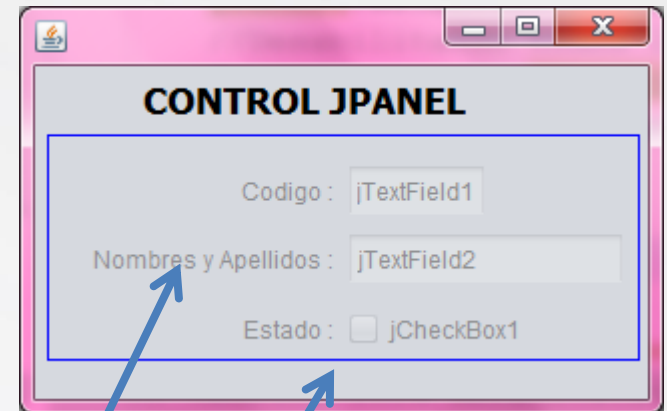
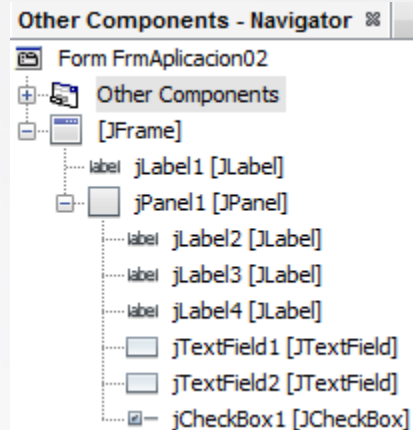
Control JPanel

JPanel son objetos contenedores, la finalidad de estos objetos es la agrupación de otros objetos tales como botones, campos de texto, etiquetas, selectores, etc, una gran ventaja de usar **Jpanel** en java es que podemos manejar la agrupación de una mejor forma.

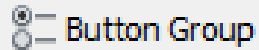
Swing Containers

Panel

Controles dentro del JPanel



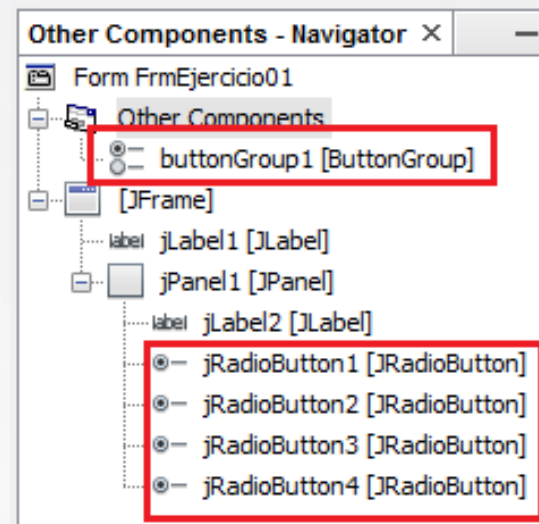
```
public class FrmAplicacion02 extends javax.swing.JFrame {  
    public FrmAplicacion02 () {  
        initComponents();  
        //Estableciendo una linea en el control JPanel  
        jPanel1.setBorder(BorderFactory.createLineBorder(Color.BLUE));  
        //Desabilitando todos los controles del JPanel  
        for(Component c: jPanel1.getComponents()) {  
            c.setEnabled(false);  
        }  
    }  
}
```



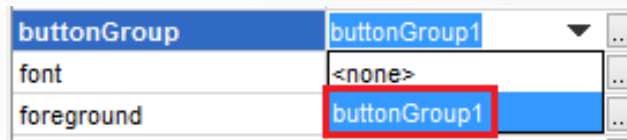
Control ButtonGroup y JRadioButton



Control **Swing**, el propósito de un grupo de botones es controlar que botón se ha seleccionado, y no tiene nada que ver con la interfaz grafica de usuario. Este tipo de control es mayor utilizado el control **JRadioButton** es utilizado para seleccionar solo un elemento u opción de un conjunto de elementos u opciones.



Enlazando el
JRadioButton al
control
buttonGroup1

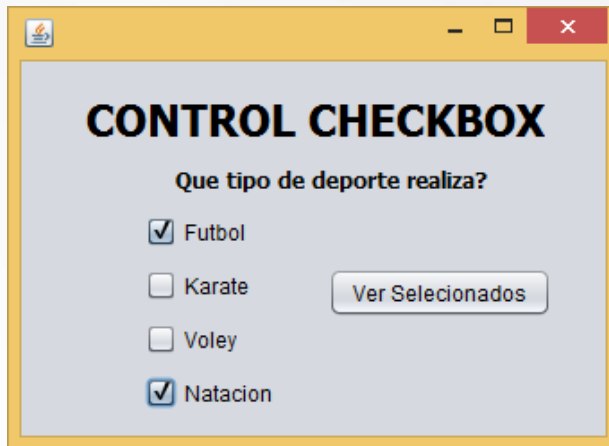


```
if(jRadioButton3.isSelected())  
{  
    JOptionPane.showMessageDialog(this,jRadioButton3.getText());  
}
```

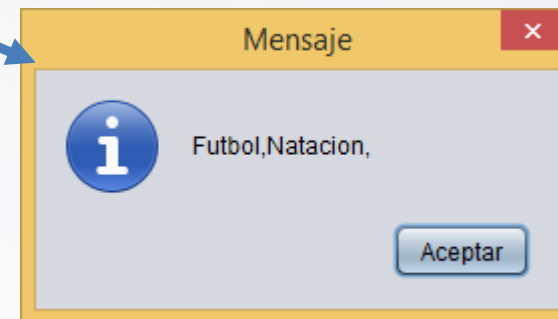


Control CheckBox

Control **Swing**, son botones de comprobación se utilizan generalmente como botones de estado, proporcionan información del tipo SI o NO. El argumento es de tipo booleano: verdadero (true) si la caja se ha seleccionado y falso (false) en otro caso.



```
String deporte="";  
for(int i=0;this.getContentPane().getComponents().length > i; i++)  
{ //Codigo para recuperar los JCheckBox seleccionados  
    if(this.getContentPane().getComponents()[i] instanceof JCheckBox){  
        JCheckBox jchk=(JCheckBox)this.getContentPane().getComponents()[i];  
        if( jchk.isSelected())  
        {  
            deporte+= jchk.getText()+",";  
        }  
    }  
}  
JOptionPane.showMessageDialog(this, deporte);
```

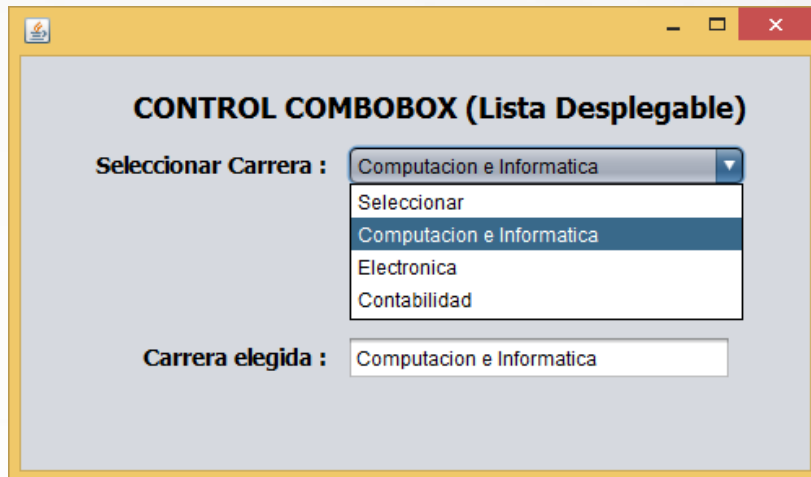


Control Combo Box

El control **JComboBox** permite seleccionar un **String** de una lista.

Para inicializar los **String** que contendrá el **JComboBox** debemos llamar al método **addItem** tantas veces como elementos queremos cargar.

Un evento muy útil con este control es cuando el operador selecciona un **Item** de la lista. Para capturar la selección de un **item** debemos implementar la interface **ItemListener** que contiene un método llamada **itemStateChanged**.



```
//Constructor
public FrmAplicacion03() {
    initComponents();
    jComboBox1.addItem("Seleccionar");
    jComboBox1.addItem("Computacion e Informatica");
    jComboBox1.addItem("Electronica");
    jComboBox1.addItem("Contabilidad");
}

@SuppressWarnings("unchecked")
Generated Code
//Evento del control JComboBox1
private void jComboBox1ActionPerformed(java.awt.event
    if (jComboBox1.getSelectedIndex() != 0) {
        txtcarrera.setText(jComboBox1.
            getSelectedItem().toString());
    }
}
```

MUCHAS GRACIAS

