



APRENDERAPROGRAMAR.COM

MATRICES, ARREGLOS O
ARRAYS DE ELEMENTOS
GRÁFICOS EN JAVA.
EJEMPLO CON JLABEL Y
JTEXTFIELD. (CU00930C)

Sección: Cursos

Categoría: Lenguaje de programación Java nivel avanzado I

Fecha revisión: 2029

Resumen: Entrega n°30 curso "Lenguaje de programación Java Nivel Avanzado I".

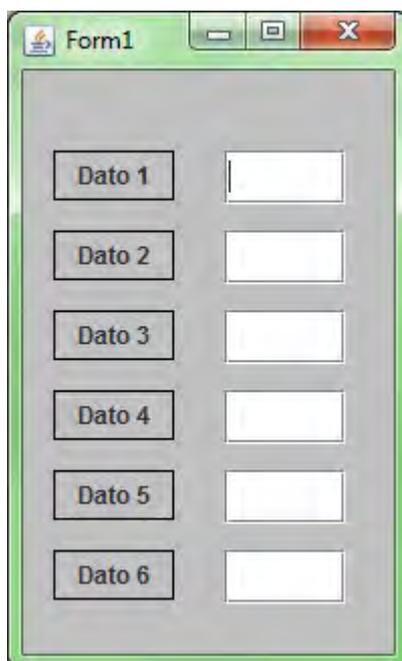
Autor: Walter Sagástegui

MATRICES O ARRAYS DE ELEMENTOS GRÁFICOS

Ya hemos visto la utilidad del uso de arrays de variables. Por otro lado, estamos viendo que el API Java nos provee de clases para mostrar elementos gráficos, con algunas propiedades que recuerdan a las variables. Por ejemplo, el contenido de un *JTextField*, "recuerda" al contenido de una variable.



Siendo esto así, se comprenderá que igual que es interesante trabajar con arrays de variables, puede resultar atractivo trabajar con arrays de elementos gráficos. A continuación veremos el código de un programa que hace uso de un array de controles, y que explicaremos detalladamente más abajo.



En este formulario, hemos definido 6 JLabels denominados `label[0]`, `label[1]`, `label[2]`, `label[3]`, `label[4]`, `label[5]` así como 6 JTextFields, denominados `text[0]`, `text[1]`, `text[2]`, `text[4]`, `text[5]`. También podríamos haberlos nombrado simplemente como `label0`, `label1`, `label2`, `label3`, `label4`, `label5`, etc. Pero al igual que ocurre con las variables, resultará mucho más cómodo de cara al tratamiento de datos el nombrarlas como una matriz de elementos gráficos, que nos permite un tratamiento automatizado.

Escribe el código, compílalo y comprueba sus resultados.

```
//(1) PAQUETE
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JPanel;
import javax.swing.JFrame;
import javax.swing.SwingConstants;
import javax.swing.BorderFactory;
import javax.swing.border.Border;
import java.awt.Color;
import java.awt.Rectangle;

//(2) FORMULARIO
public class Programa extends JFrame {
    //(3) CONTROLES DEL FORMULARIO
    JPanel jpanel = (JPanel) this.getContentPane();

    JLabel[] label = new JLabel[6];    //Declaración del array de etiquetas
    JTextField[] text = new JTextField[6];    //Declaración del array de cajas de texto

    Border border = BorderFactory.createLineBorder(Color.black, 1);

    //(4) CONSTRUCTOR DEL FORMULARIO
    public Programa() {
        //(5) PROPIEDADES DEL CONTENEDOR
        jpanel.setLayout(null);
        jpanel.setBackground(Color.lightGray);

        for(int i = 0; i < label.length; i++) {
            label[i] = new JLabel();    //Llenamos el array de etiquetas
            text[i] = new JTextField();    //Llemanos el array de cajas de texto
            //(6) PROPIEDADES DE LOS CONTROLES
            label[i].setBounds(new Rectangle(15, (i+1)*40, 60, 25));
            label[i].setText("Dato " + (i+1));
            label[i].setBorder(border);
            label[i].setHorizontalAlignment(SwingConstants.CENTER);

            text[i].setBounds(new Rectangle(100, (i+1)*40, 60, 25));
            //(7) ADICION DE LOS CONTROLES AL CONTENEDOR
            jpanel.add(label[i], null);
            jpanel.add(text[i], null);
        }

        //(8) PROPIEDADES DEL FORMULARIO
        setSize(200,330);
        setTitle("Form1");
        setVisible(true);
    }

    //(9) METODOS DEL FORMULARIO
    public static void main(String arg[]) {
        new Programa();
    }
}
```

Ahora nos centraremos en explicar el código del programa expuesto anteriormente.

```
//(1) PAQUETE
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JPanel;
import javax.swing.JFrame;
import javax.swing.SwingConstants;
import javax.swing.BorderFactory;
import javax.swing.border.Border;
import java.awt.Color;
import java.awt.Rectangle;
```

En este primer bloque de código, declaramos las rutas donde están ubicadas todas las clases que estamos usando en el programa .Y como recordatorio de otros artículos, **import java.swing.JLabel**, se interpreta de la siguiente manera “importar la clase JLabel ubicado en la ruta /java/swing/, siendo java y swing carpetas”.

```
JLabel[] label = new JLabel[6]; //Declaración del array de etiquetas
JTextField[] text = new JTextField[6]; //Declaración del array de cajas de texto
```

La sentencia **JLabel[] label = new JLabel[6]**, es una declaración de un array de controles de tipo JLabel y el array es de tamaño 6, es decir, estamos declarando 6 variables y cada una de estas variables almacena un control de tipo JLabel y el nombre de éstas adoptan la siguiente forma:

label[0], label[1], label[2], label[3], label[4], label[5]. De igual manera, se puede interpretar lo que hace la sentencia **JTextField[] text = new JTextField[6]**, en este caso, las variables presentan esta forma: text[0], text[1], text[2], text[3], text[4], text[5] y cada una almacena un control de tipo JTextField.

```
Border border = BorderFactory.createLineBorder(Color.black, 1);
```

Esta sentencia, es la declaración de una variable de nombre **border**, que almacena un borde, con la característica de que es de color negro y de un grosor de 1, pudiendo tanto el color como el grosor del borde cambiar a gusto del programador. Por ejemplo, si queremos un borde rojo en lugar de “black”, pondríamos “red” y si deseamos un grosor mayor, podemos poner “2” en lugar de “1”, hasta obtener la característica del borde que nos agrada para la etiqueta JLabel. Más adelante, en el código, se usa esta característica del borde para ponerlo como atributo de nuestras etiquetas.

```
for(int i = 0; i < label.length; i++) {  
    label[i] = new JLabel(); //Llenamos el array de etiquetas  
    text[i] = new JTextField(); //Llemanos el array de cajas de texto  
  
    ...  
}
```

La sentencia **for** en el programa es usada para manejar los arrays de controles que hemos declarado previamente. El uso del **for** en Java lo estudiaremos más adelante, por lo que no debemos preocuparnos ahora por entender la sintaxis. Simplemente decir que vamos a crear tantas etiquetas y tantas cajas de texto como hayamos definido; en nuestro caso, crearemos 6 etiquetas y 6 cajas de texto. Lo primero a destacar es el atributo **label.length**, que nos da de forma automática el tamaño de los arrays mencionados, que en este caso es de tamaño 6. También podíamos haber usado **text.length**, que nos da igualmente el valor 6, ya que **text** y **label** son arrays declarados, ambos de tamaño 6.

Por otro lado, la sentencia **label[i] = new JLabel()**; toma la siguiente forma por cada iteración de la sentencia **for**:

```
label[0] = new JLabel();  
label[1] = new JLabel();  
label[2] = new JLabel();  
label[3] = new JLabel();  
label[4] = new JLabel();  
label[5] = new JLabel();
```

Cada una de estas 6 variables **label[0]**, ... **label[5]**, toma como valor un objeto de tipo **JLabel** o lo que es lo mismo, un control de tipo **JLabel**. De la misma manera, podemos interpretar la sentencia **text[i] = new JTextField()**;

```
text[0] = new JTextField();  
text[1] = new JTextField();  
text[2] = new JTextField();  
text[3] = new JTextField();  
text[4] = new JTextField();  
text[5] = new JTextField();
```

```
for(int i = 0; i < label.length; i++) {  
    ...  
  
    //(6) PROPIEDADES DE LOS CONTROLES  
    label[i].setBounds(new Rectangle(15, (i+1)*40, 60, 25));  
    label[i].setText("Dato "+(i+1));  
    label[i].setBorder(border);  
    label[i].setHorizontalAlignment(SwingConstants.CENTER);  
  
    text[i].setBounds(new Rectangle(100, (i+1)*40, 60, 25));  
    ...  
}
```

Esta parte del código está dentro de la sentencia **for** que vimos antes; por lo tanto, en cada iteración, estas sentencias, toman la forma siguiente. Sólo analizaremos cuando $i = 0$. Para los siguientes valores 1, 2, 3, 4 y 5, se produce la misma acción.

- (1) `label[0].setBounds(new Rectangle(15, (0+1)*40, 60, 25));`
- (2) `label[0].setText("Dato "+(0+1));`
- (3) `label[0].setBorder(border);`
- (4) `label[0].setHorizontalAlignment(SwingConstants.CENTER);`
- (5) `text[0].setBounds(new Rectangle(100, (i+0)*40, 60, 27));`

Aclarar que las sentencias numeradas con (1), (2), (3), (4) y (5), ya fueron explicadas en anteriores entregas; vamos a explicar aspectos generales de lo que hacen en el programa. En (1), tenemos la sentencia encargada de ubicar la etiqueta `label[0]` dentro del formulario, en una coordenada determinada, así como con un ancho y una altura específica. Resaltaremos el artificio hecho para espaciar una etiqueta de otra, de tal forma, que estén en filas distintas y no se superpongan en el formulario. Por ejemplo, si $i = 0$, entonces $(i + 1) * 40 = 40$, es decir, que esta primera etiqueta `label[0]`, se ubica dentro del formulario en la fila 40 y si analizamos cuando $i = 1$, vemos que $(i + 1) * 40 = 80$, significa que la etiqueta `label[1]`, se ubica dentro del formulario en la fila 80.

En (2), tenemos la sentencia encargada de poner un texto a la etiqueta `label[0]` en el formulario, en este caso cuando $i = 0$, la etiqueta `label[0]` muestra el texto "Dato 1" y cuando $i = 1$, la etiqueta `label[1]` muestra el texto "Dato 2" y así sucesivamente con las demás valores para i .

En (3), tenemos la sentencia encargada de poner un borde a nuestra etiqueta `label[0]`, la definición de las características de este borde, como el color y grosor, ya fueron explicados más arriba.

En (4), tenemos la sentencia encargada de centrar el texto del punto (2).

En (5) tenemos la misma sentencia de (1), pero en este caso ubica la caja de texto dentro del formulario en una coordenada determinada, así como el ancho y altura de la etiqueta `text[i]`.

Próxima entrega: CU00931C

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:

http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=58&Itemid=180