



APRENDERAPROGRAMAR.COM

INGRESAR DATOS CON UN
CONTROL VISUAL EN JAVA.
CLASE JTEXTFIELD Y
MÉTODO GETTEXT.
EJEMPLOS (CU00928C)

Sección: Cursos

Categoría: Lenguaje de programación Java nivel avanzado I

Fecha revisión: 2029

Resumen: Entrega nº28 curso "Lenguaje de programación Java Nivel Avanzado I".

Autor: Walter Sagástegui

INSTRUCCIÓN PEDIR CON JAVA. EL JTEXTFIELD. EL MÉTODO GETTEXT.

La forma de pedir datos con Java a través de un control visual, resulta un tanto distinta a la inserción de datos a través de consola. Una petición de datos se puede realizar a través de un objeto prefabricado de la clase JTextField.



Mostraremos el siguiente código y la salida respectiva de entrada de datos, con un control visual en Java, y luego lo explicaremos y comentaremos:

```
//(1) PAQUETE
import java.awt.*;
import javax.swing.*;

//(2) FORMULARIO
public class Programa extends JFrame {

    //(3) CONTROLES DEL FORMULARIO
    JPanel jpanel = (JPanel) this.getContentPane();
    JTextField jtextfield = new JTextField();

    //(4) CONSTRUCTOR DEL FORMULARIO
    public Programa() {

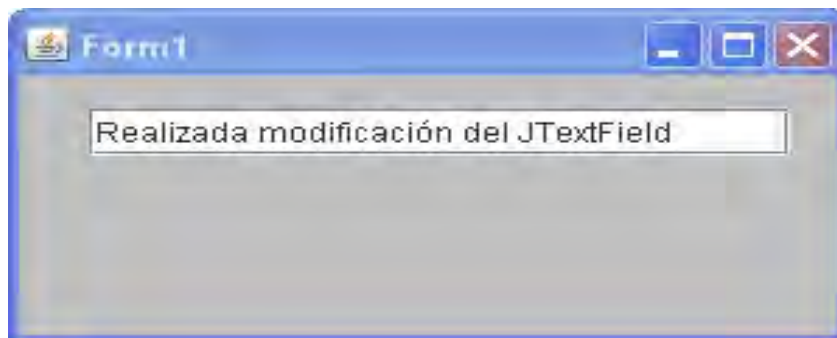
        //(5) PROPIEDADES DEL CONTENEDOR
        jpanel.setLayout(null);
        jpanel.setBackground(Color.lightGray);

        //(6) PROPIEDADES DE LOS CONTROLES
        jtextfield.setBounds(new Rectangle(25, 15, 250, 21));
        jtextfield.setText("Realizada modificación del JTextField");
        jtextfield.setEditable(false);
        jtextfield.setHorizontalAlignment(JTextField.LEFT);

        //(7) ADICION DE LOS CONTROLES AL CONTENEDOR
        jpanel.add(jtextfield, null);

        //(8) PROPIEDADES DEL FORMULARIO
        setSize(300,150);
        setTitle("Form1");
        setVisible(true);
    }

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



Ahora explicaremos en detalle el código. Para su mejor entendimiento lo hemos numerado y lo asumiremos como una secuencia de pasos a seguir para construir una estructura visual en Java. Para este ejemplo, hemos tomado la entrada de datos a través de un control visual, pero el tratamiento será similar para cualquier otro control visual como veremos en otros artículos.

(1) Paquete

En este punto, se usa la sentencia **import**, para declarar los paquetes que emplearemos en nuestro programa. Explicar, que un paquete es algo similar a una ruta, donde se encuentran las clases que utilizaremos en el programa. En nuestro caso, **JTextField**, **JPanel**, **JFrame**, son clases que pertenecen al paquete **javax.swing**, así como la clase **Color** pertenece al paquete **java.awt**. Los paquetes forman parte de java, lo único que hacemos nosotros es cargarlos para poder usarlos en nuestro programa.

(2) Formulario

Programa extends JFrame. La idea de esta sentencia, es hacer que nuestro programa tenga el comportamiento de un formulario (ventana Windows) y para ello debemos heredar (extends) de JFrame, sus particularidades. JFrame, es una clase que tiene todas las características propias de una ventana en Windows. A partir de este punto, nuestro programa deja de ser un programa de consola DOS y pasa a ser un programa visual tipo ventana Windows.

(3) Controles del formulario

Aquí, se crean los objetos de los controles visuales que se mostrarán en el formulario. El primer objeto que vemos es *jpanel*, mencionar, que es un nombre cualquiera y pertenece a la clase **JPanel**. EL objeto *jpanel*, es lo que se llama un *contenedor*, que como su propio nombre indica, va a contener a los demás controles visuales. Es decir, que los controles visuales no se ponen directamente en el formulario, sino en el contenedor, colocado éste encima del formulario. El siguiente objeto es *jtextfield*, perteneciente a la clase **JTextField**; este objeto *jtextfield*, contiene el control visual para pedir un dato al usuario y tiene la apariencia de una caja para ingresar texto (textbox).

(4) Constructor del formulario

Si se observa, es una estructura igual a un método, que se inicia con una apertura de llave "{" y termina con la clausura de la llave "}". Entre dichas llaves se procede a dar a los objetos, que representan a los controles visuales, los atributos. También añade los controles visuales al contenedor, además de establecer los atributos del formulario. Con más detalle, veremos todo esto en los siguientes puntos.

(5) Propiedades del contenedor del formulario

Lo que es un contenedor, ya fue explicado en el punto 3. Ahora aquí, explicaremos las siguientes instrucciones relacionadas al contenedor:

.- **jpanel.setLayout(null)**. Esta instrucción significa, que al pasarle **null** como parámetro al método **setLayout**, nuestro contenedor, representado a través del objeto **jpanel**, no administrará la forma de colocar los controles en el contenedor, sino que dejará que esa labor la realice el programador a través de coordenadas.

.- **jpanel.setBackground(Color.lightGray)**. Al pasarse **Color.lightGray**, como parámetro del método **setBackground**, le decimos al contenedor, representado en el objeto *jpanel*, que tome un color de fondo gris suave. Ahora, si quisiéramos usar otro color de fondo, como el color verde, usaríamos el parámetro **Color.green**, y de igual manera para otros colores.

(6) Propiedades de los controles.

En este punto, estableceremos a través de propiedades, la apariencia de nuestros controles visuales. En este ejemplo, el control visual será una caja de texto para ingresar datos, representados en el objeto **jtextfield**, para lo cual explicaremos las instrucciones siguientes:

.- **jtextfield.setBounds(new Rectangle(25, 15, 250, 21))**. Cada uno de los valores del parámetro (25,15,250,21) es igual a $(x,y,width,height)$, donde x,y , es la coordenada en la que se ubica el control, dentro del contenedor del formulario. *Width* es el ancho que tendrá el control, o sea, 250 píxeles y *height* sería la altura, en este caso de 21 píxeles.

.- **jtextfield.setText("Realizada modificación del JTextField")**. En esta instrucción, queda claro que la cadena "Realizada modificación del JTextField", que se le pasa como parámetro al método **setText**, aparece dentro de la caja de texto durante la ejecución del programa.

.- **jtextfield.setEditable(false)**. Si esta instrucción, está establecida en *true*, permite que se pueda escribir sobre el *JTextField*. Si está establecida en *false*, impide que el usuario pueda modificar el contenido del *JTextField*.

.- **jtextfield.setHorizontalAlignment(JTextField.LEFT)**. El parámetro **LEFT**, permite que el texto en la caja de texto, se alinee a la izquierda, el parámetro **CENTER** al centro y el **RIGHT** a la derecha.

(7) Adición de los controles al contenedor

.- **jpanel.add(jtextfield, null)**. El método **add** pertenece a la clase **Jpanel**. Este método, es usado para añadir un control al contenedor, representado con el objeto *jpanel*. El primer parámetro, el objeto *jtextfield*, se añade al contenedor y el segundo parámetro, **null**, indica que la posición, dónde colocar el controlador *jtextfield* dentro del contenedor, será determinado por el programador, a través de coordenadas que fueron explicadas en el punto 5 y 6.

(8) Propiedades del formulario

Un formulario tiene una apariencia visual por defecto, por ejemplo el tamaño, el color de fondo, entre otros. Estas propiedades, las podemos cambiar, a través de una serie de métodos, como los siguientes:

.- **setSize(300,150)**. Ubica la esquina superior izquierda del formulario en la pantalla, en la coordenada (300,150)=(x,y)=(columna,fila).

.- **setTitle("Form1")**. La cadena "Form1", como parámetro del método **setTitle**, significa que se establecerá como título del formulario la cadena "Form1".

.- **setVisible(true)**. Este parámetro **true**, del método **setVisible**, determina que el formulario sea visible en la pantalla, ya que si ponemos **false**, el formulario está en la pantalla de forma invisible.

(9) Métodos del formulario

En este punto se definen los métodos que se necesitan para realizar las tareas que diseñan el formulario. También es el lugar del método *main*, que hace que nuestra clase Programa se pueda ejecutar, ya que si dejamos de poner el método *main*, no podrá ejecutarse, porque dentro de este método, es dónde creamos un objeto del formulario, a través de la instrucción **new Programa ()**. Por tanto debe quedar claro que el formulario se crea siempre y cuando creamos un objeto de la clase **Programa**, lo cual declaramos como **new Programa()**;

EL MÉTODO GETTEXT

El contenido que introduce un usuario a través de un *JTextField*, tiene carácter de texto. Para trasladar dicho contenido a una variable, se usa la *función getText*. La sintaxis a emplear es:

```
String cadena = jtextfield.getText();
```


Si deseamos convertir el valor retornado por la instrucción **jtextfield.getText()**, en forma de texto, que se recibió en la variable *cadena*, a un valor en forma de número, la sintaxis es la siguiente:

```
float a = Float.parseFloat(cadena);
```

Esto supone, que la variable *a*, que ha sido definida como tipo *float*, adopta el contenido numérico del *JTextField*.

La función *getText*, no elimina los espacios en blanco, ni los tabuladores ni los avances de línea. Para suprimirlos, podemos aplicar el ejemplo siguiente:

```
String cadena = " 38205 ";
String cadenalimpia = cadena.trim(); //Devuelve "38205" como cadena
float a = Float.parseFloat(cadenalimpia); //Devuelve 38205 como número real
```

 Otra forma relevante de pedir datos con *Java*, consiste en el uso de ventanas de dialogo de petición de datos (*JOptionPane.showInputDialog*). Cuando se invoca la función *JOptionPane.showInputDialog*, se despliega una ventana independiente, con un mensaje definido de solicitud de datos al usuario, que dispone de una caja de texto asociado para responder y de botones para aceptar o cancelar. La sintaxis es:

```
String s = JOptionPane.showInputDialog("Introduzca un número","Introducción número");
```

donde "s" es una variable, que toma el valor que haya introducido el usuario en forma de cadena.

Te animamos a que, valiéndote de los ejemplos que hemos visto, crees tus propios programas de prueba y veas por tí mismo los distintos formatos de presentación y resultados que se pueden obtener.

Próxima entrega: CU00929C

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