

Clase adicional 5

Temas

- Interfaces
 - Qué es una interfaz
 - ActionListener
- Introducción a Java Swing
 - Qué es Swing
 - SwingApplication
 - Guía paso a paso
- Problemas de diseño

Interfaz

Qué es una interfaz

- Una interfaz es un conjunto de declaraciones de métodos que pueden ser implementadas por clases
- Una interfaz describe qué clases deberían hacerlo, sin especificar la forma
- La forma se definirá en las clases que implementen la interfaz
- Cada clase define la implementación de forma distinta
- Una clase puede implementar una o más interfaces
- Una interfaz puede contener tanto métodos como constantes

Para utilizar una interfaz, la clase debe

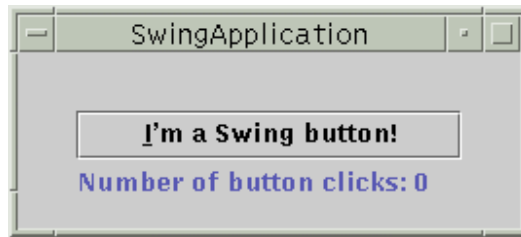
- Implementar dicha interfaz
- Definir TODOS los métodos incluidos en dicha interfaz

ActionListener

ActionListener es una interfaz de Java implementada por muchos componentes GUI, como los botones. Sólo tiene un método: actionPerformed. La definición es la siguiente:

```
public Interface ActionListener {  
  
    public void actionPerformed(ActionEvent e)  
  
}
```

El ejemplo que se muestra a continuación es una aplicación GUI de Java llamada SwingApplication que tiene un contador de "clics de botones": cada vez que el usuario hace clic en el botón, el contador se incrementa en 1. Explicaremos cómo crear la GUI en la siguiente sección. Aquí, nos limitaremos a mostrar cómo se debe implementar ActionListener.



```

class SwingApplication implements ActionListener
{
    .....
    .....

    int numClicks = 0; //"contador de clics"
    JLabel label = new JLabel("Número de clics: " + numClicks); JButton
    button = new JButton("¡Soy un botón Swing!");
    button.addActionListener(this); //Añade un actionListener al botón

    .....

    public void actionPerformed(ActionEvent e)
    {
        numClicks++;
        label.setText("Número de clics: " + numClicks);
    }
}

```

Aquí, la clase `SwingApplication` implementa la interfaz `ActionListener`. Dentro de la definición de la clase, ha redefinido el método `actionPerformed`: cada vez que el usuario hace clic en el botón, el contador del mensaje se incrementa en 1.

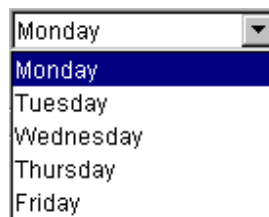
Introducción a Java Swing

Qué es Swing

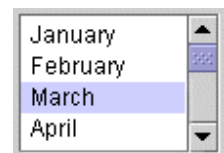
El paquete `Swing` forma parte de las *Java™ Foundation Classes (JFC)* de la plataforma de Java. Las JFC acompañan a un grupo de funciones que ayudan al usuario a construir las GUI. Aquí se muestran algunos de los componentes Swing utilizados con más frecuencia:



Botones



Cuadro combinado



Lista



Cuadro de diálogo



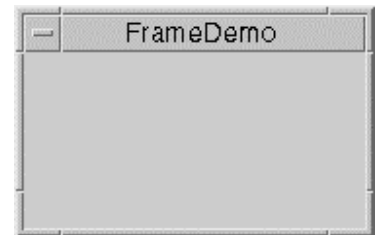
Panel de desplazamiento



Menú

First Name	Last Name	Sport	# of Years	Vegetarian
Mary	Campione	Snowboarding	5	<input type="checkbox"/>
Alison	Huml	Rowing	3	<input checked="" type="checkbox"/>
Kathy	Walrath	Chasing toddl...	2	<input type="checkbox"/>
Mark	Andrews	Speed reading	20	<input checked="" type="checkbox"/>

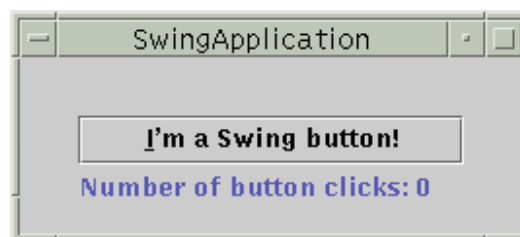
Tabla



Marco

SwingApplication

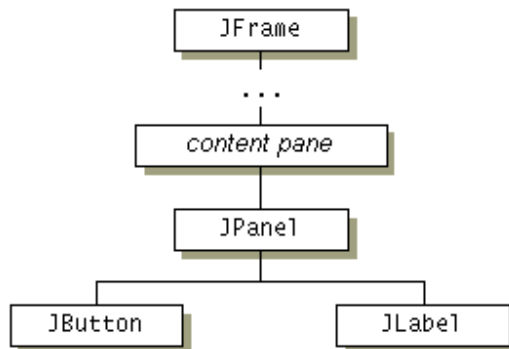
SwingApplication es un ejemplo extraído de la clase adicional de Java. Tal como mencionamos anteriormente, se trata de un contador de "clics de botones": cada vez que el usuario hace clic en el botón, la etiqueta se actualiza y muestra el contador con un incremento de 1.



SwingApplication tiene cuatro componentes Swing:

- Un marco (JFrame). El marco es un contenedor de nivel superior. Ofrece un espacio para que otros componentes Swing se puedan dibujar a sí mismos. Los otros contenedores de nivel superior que se utilizan con más frecuencia son los cuadros de diálogo (JDialog) y los *applets* (JApplet).
- Un panel (JPanel). El panel es un contenedor intermedio. Su única finalidad es simplificar la ubicación del botón y la etiqueta. Otros contenedores intermedios Swing son JScrollPane (paneles de desplazamiento) y JTabbedPane (paneles tabulados)
- Un botón (JButton) y una etiqueta (JLabel). El botón y la etiqueta son componentes atómicos, componentes que no existen para alojar a otros componentes Swing sino para interactuar con el usuario. La API Swing ofrece muchos componentes atómicos entre los que se incluyen cuadros combinados (JComboBox), campos de texto (JTextField) y tablas (JTable).

A continuación se incluye un diagrama de la jerarquía de componentes para la ventana mostrada por `SwingApplication`



Aquí está el código que agrega el botón y la etiqueta al panel, y el panel al panel de contenido:

```
frame = new JFrame(...);
```

```
pane = new JPanel();
```

```
button = new JButton(...);
```

```
label = new JLabel(...);
```

```
pane.add(button);
```

```
pane.add(label);
```

```
frame.getContentPane().add(pane, BorderLayout.CENTER);
```

Guía paso a paso

Ahora mostramos una guía paso a paso para crear esta `SwingApplication`:

- Definición del contenedor de nivel superior
- Definición de botones y etiquetas
- Incorporación de componentes a los contenedores
- Gestión de eventos

A continuación analizaremos cada paso en profundidad.

- Definición del contenedor de nivel superior

```
//Crear el contenedor de nivel superior llamado "SwingApplication"  
JFrame frame = new JFrame("SwingApplication");
```

```
.....  
frame.pack();  
frame.setVisible(true);
```

- o Definición de botones y etiquetas

```
//Crear un botón  
JButton button = new JButton("¡Soy un botón Swing!");
```

```
//Crear una etiqueta  
JLabel label = new JLabel("Número de clics de botones: " + "0 ");  
//Definir el texto de la etiqueta  
int numClicks = 0;  
label.setText("Número de clics de botones: " + numClicks);
```

- o Incorporación de componentes a los contenedores

```
JPanel pane = new JPanel();  
pane.setLayout(new GridLayout(0, 1));  
pane.add(button);  
pane.add(label);  
frame.getContentPane().add(pane, BorderLayout.CENTER);
```

- o Gestión de eventos

Hablaremos de esta función en la siguiente clase adicional.

Problema de diseño

Interfaz

Han contratado sus servicios para desarrollar un programa de contabilidad que calcule el salario semanal de tres tipos de empleados: administrativos, sindicalistas y contratistas.

Cada empleado administrativo tiene un número de seguridad social (SSN), un nombre, título y salario. No se contempla el abono de horas extra para este tipo de empleados

Cada empleado sindicalista tiene un SSN, un nombre, una tarifa por hora y un número de horas trabajadas. Los estatutos del sindicato exigen el abono de horas extra a este tipo de empleados, que es 1,5 veces la tarifa por hora cuando éstas superen las 40

Cada empleado contratista tiene un SSN, un nombre, una agencia, un número de horas trabajadas y una tarifa por hora. No es preciso pagar a los contratistas una tarifa más elevada por las horas extra. Sin embargo, hay un límite para las horas que cada contratista puede trabajar a la semana. Actualmente, este límite está en $60 * \text{tarifa}$ por hora.

Defina un conjunto de clases que den forma a esta situación. Abajo se definen los requisitos:

Crear una clase abstracta llamada *Employee* (Empleado) que tenga un método llamado *print* (imprimir)

Crear una interfaz *Compensation* (Compensación) que tenga un método llamado *calculatePay* (calcularSalario)

Crear tres subclases: *Management* (Administrativo), *UnionWorker* (Sindicalista) y *Contractor* (Contratista) que hereden de *Employee*

Cada subclase implementará la interfaz *Compensation*

Cada subclase, a su vez, deberá imprimir toda su información. A modo de ejemplo, ésta es la información que se obtendría para un director:

SSN: 1111 Nombre: Wen Xiao

Salario: \$9500

Componente básico Swing

Descargue el código de `SwingApplication.java` y agregue una barra de menú al marco. Esta barra de menú tiene dos menús: *File* (Archivo) y *Edit* (Edición). En *File*, hay 3 elementos de menú: *Open* (Abrir), *Save* (Guardar) y *Exit* (Salir). Dentro de *Edit*, hay dos elementos de menú: *Copy* (Copiar) y *Paste* (Pegar). No es necesario que implemente ninguna función de gestión de eventos, sólo la GUI. (Puede consultar un curso breve sobre el modo de utilizar menús en <http://java.sun.com/docs/books/tutorial/uiswing/components/menu.html>)

Compruebe que el programa se puede compilar y ejecutar.