



Camaleón
Human Computer Interaction
Research Group

Introducción a las GUI

PROGRAMACIÓN INTERACTIVA

ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN



Universidad del Valle

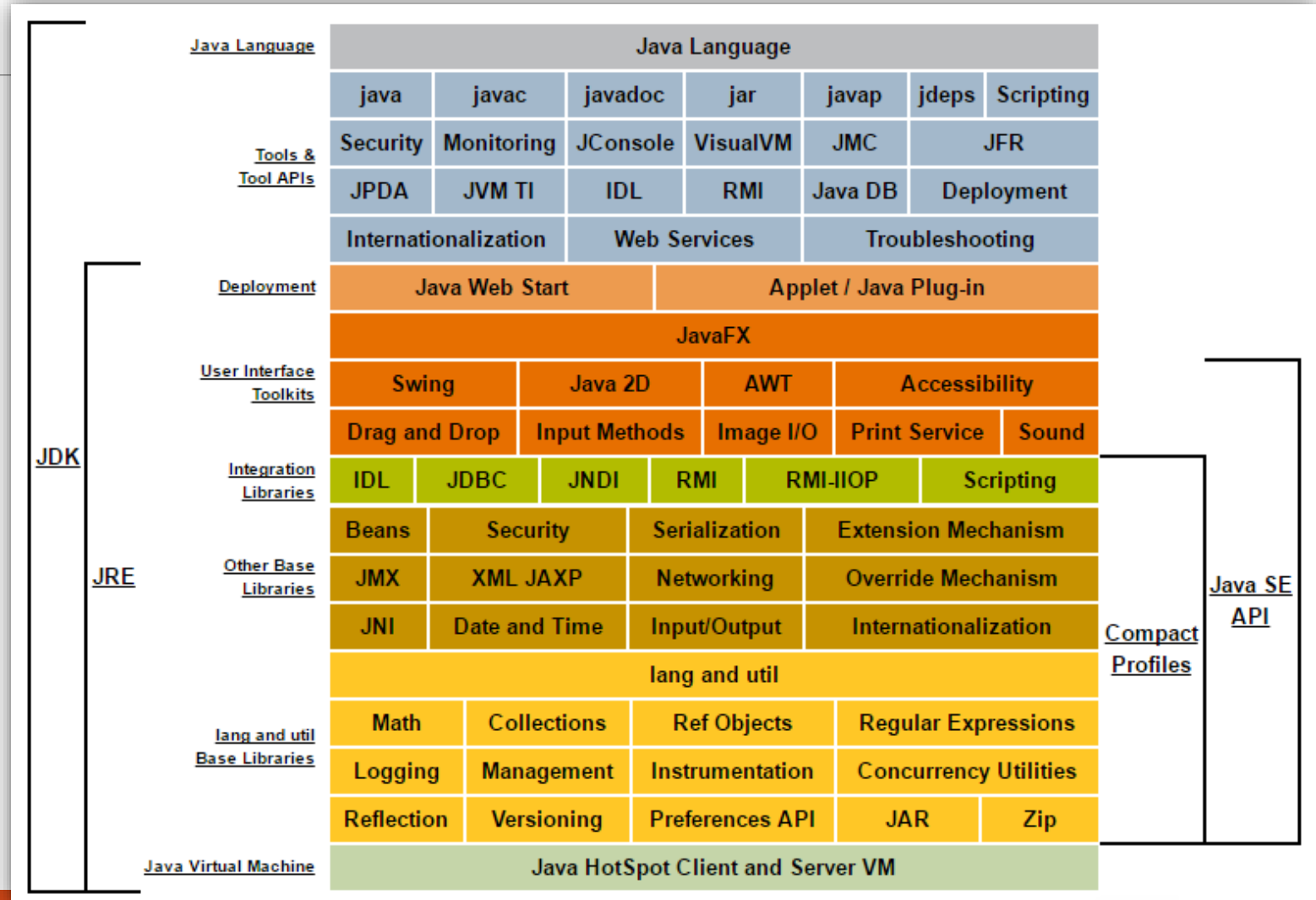


Qué debes saber al finalizar la clase!!

- ✓ *Entender el manejo de ventanas*
- ✓ *Entender el manejo de eventos*
- ✓ *Entender el Concepto de “Interfaces” en Java*

¿Qué es Swing?

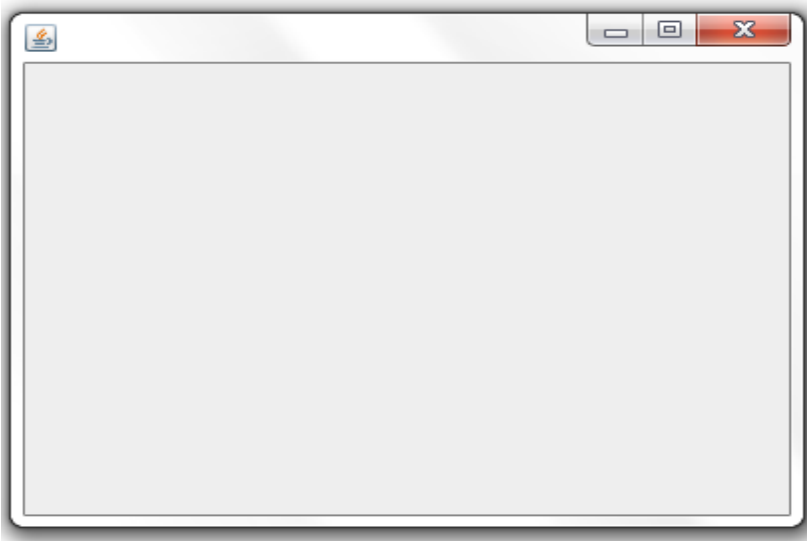
Swing nace en 1997 como parte del JFC



Introducción a Swing



✓ *Swing - JFrame*



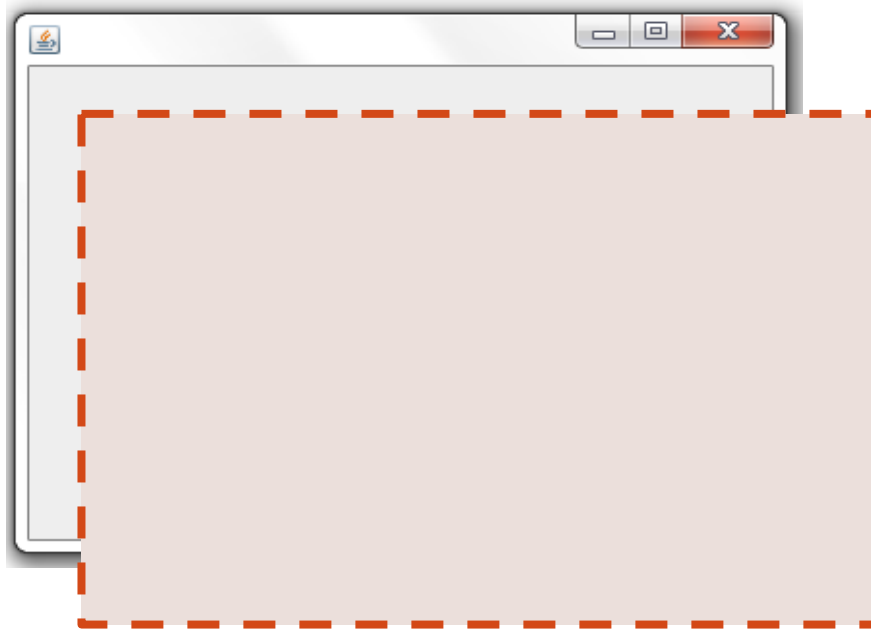
Configuración base de una ventana:

- ✓ *Darle un título*
- ✓ *Establecer el tamaño*
- ✓ *Establecer la Acción de Cierre*
- ✓ *Establecer ubicación*
- ✓ *Hacerla redimensionable*
- ✓ *Hacerla visible o invisible*

Introducción a Swing



✓ *Swing – JFrame - Container*



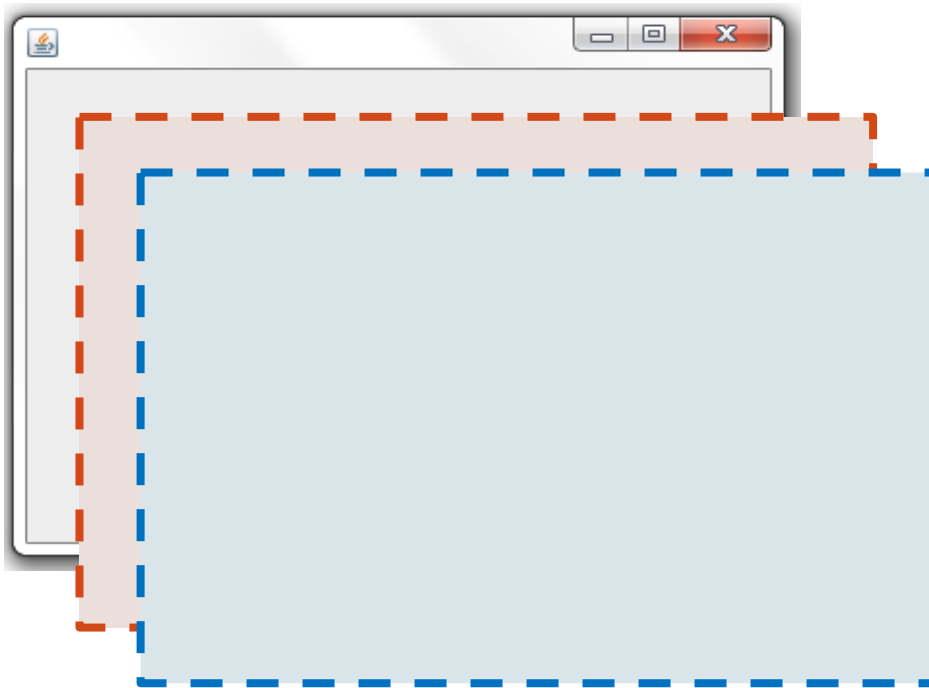
Todo JFrame tiene un **Container**

- ✓ Contenedor: Objeto cuya función es *agrupar componentes* (JButtons, JLabels, etc)

Introducción a Swing



✓ *Swing* – *Jframe* – *Container* - *Layout*



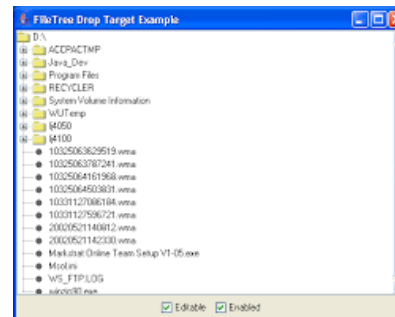
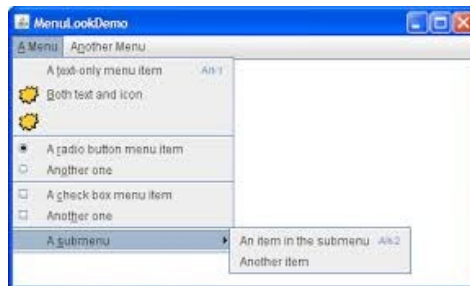
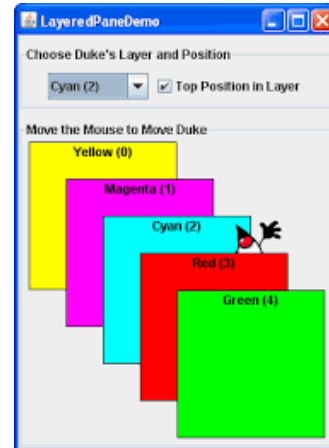
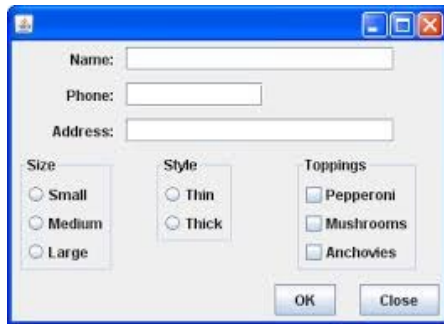
Todo Container *tiene asociado un Layout*
(Administrador de Esquema)

- ✓ Layout: Objeto cuya función es *gestionar la ubicación* de los componentes en un contenedor.

Introducción a Swing



✓ **Swing – JFrame – Container – Layout – Componentes Gráficos**



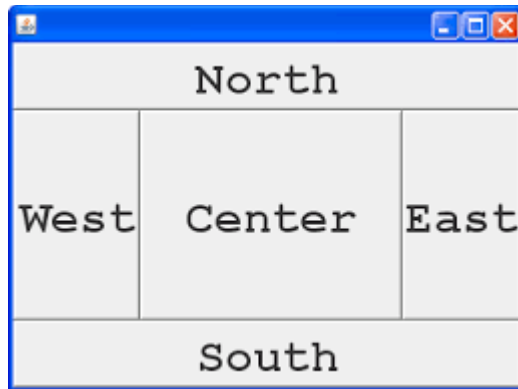
Todo Layout puede gestionar la posición de cualquier componente gráfico

- ✓ **Componente Gráfico:** Objetos que tiene una representación gráfica y responden a eventos (interacciones) del usuario.

Introducción a Swing



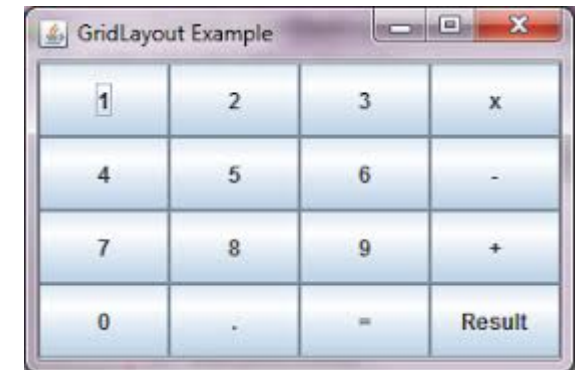
✓ **Swing** – **Jframe** – **Container** – **Layout** - **Básicos**



BorderLayout
Por defecto para JFrame



FlowLayout



GridLayout

Introducción a Swing



✓ Manejo de Eventos

En una Interfaz Gráfica de Usuario, *el orden en que se ejecutan las operaciones depende de las acciones del usuario* sobre la interfaz, es decir, los **eventos**.

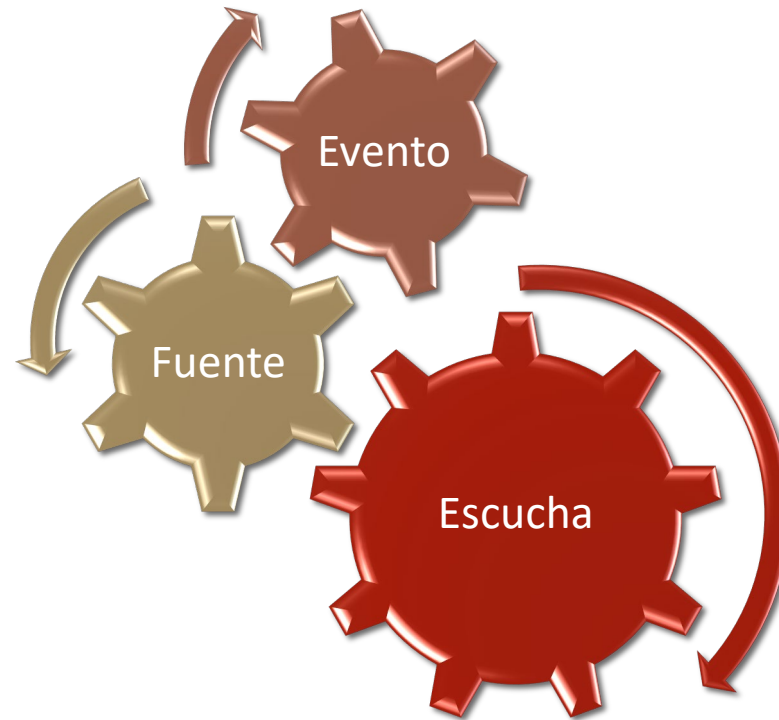
Un evento es un suceso que ocurre como consecuencia de la interacción del usuario con la interfaz gráfica, como por ejemplo **pulsar un botón**.

Los eventos se manejan por medio de la librería **java.awt.event** y por cada evento que se desee monitorear es necesario: **Adicionar el escucha (*listener*) del evento y su correspondiente *manejador***.

Introducción a Swing



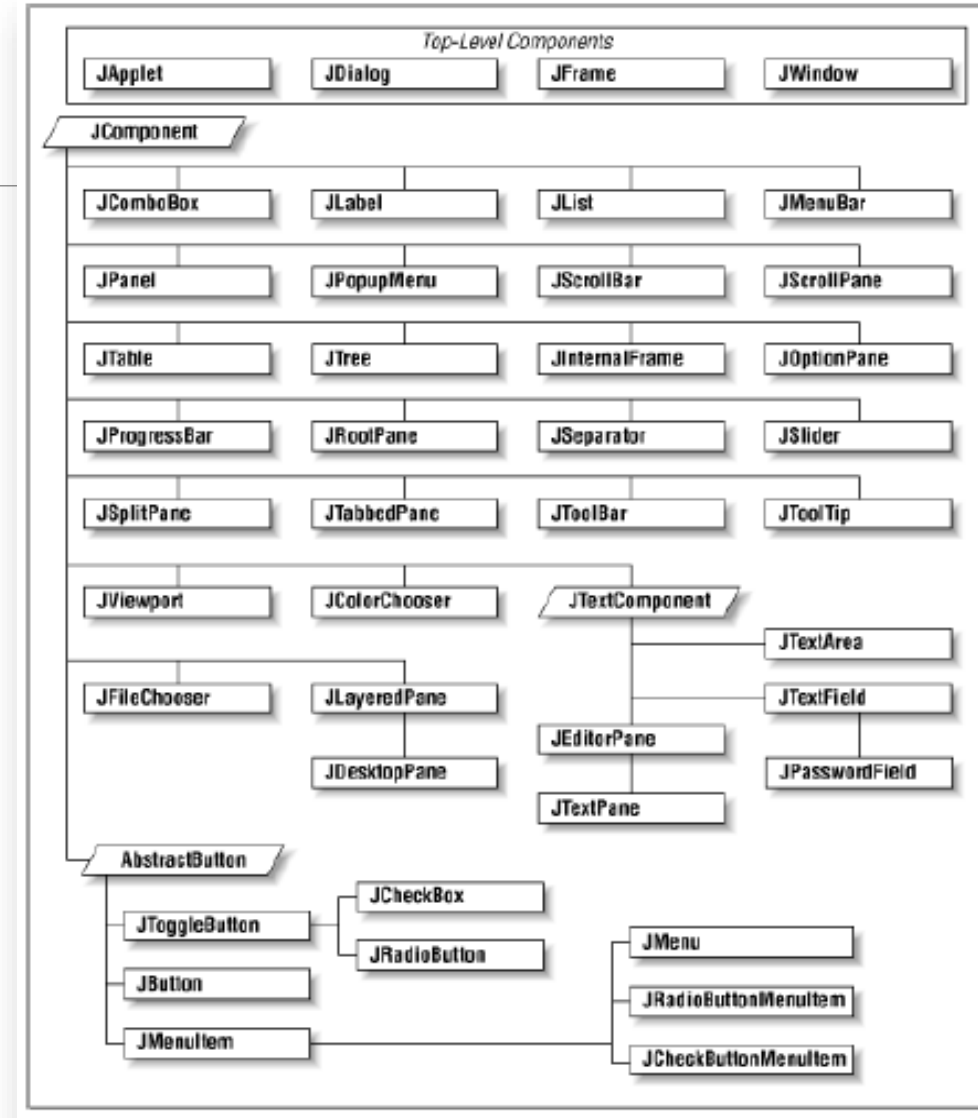
✓ Manejo de Eventos



Introducción a Swing

✓ Manejo de Eventos

Fuente del Evento: Es el componente de la GUI sobre el cuál se puede dar el evento y quien adiciona el Escucha



Swing

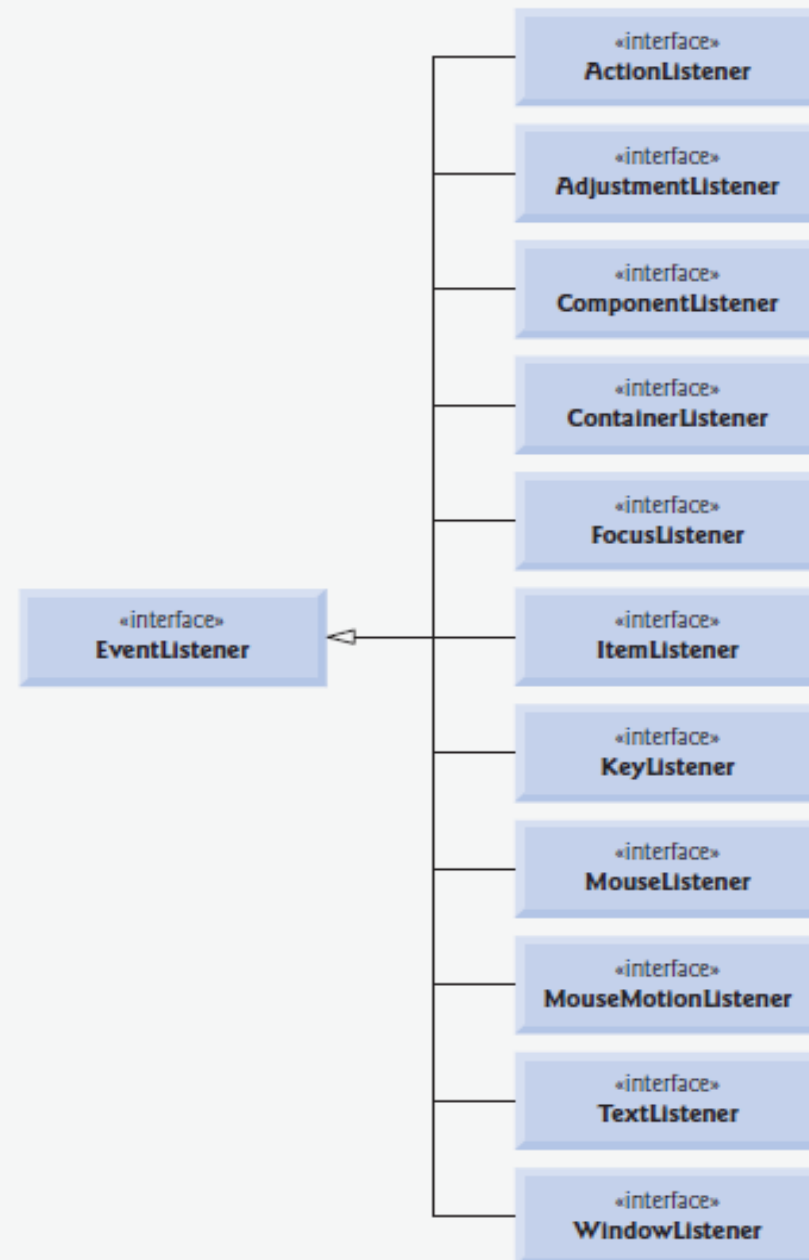
Introducción a Swing

✓ Manejo de Eventos

Escucha: **Interface** que provee el **método** encargado de manejar el evento, es decir, dónde el programador escribe las instrucciones a seguir en respuesta al evento.

En java Una **Clase puede implementar (implements)** uno o mas interfaces

Un componente puede tener varios tipos de escuchas

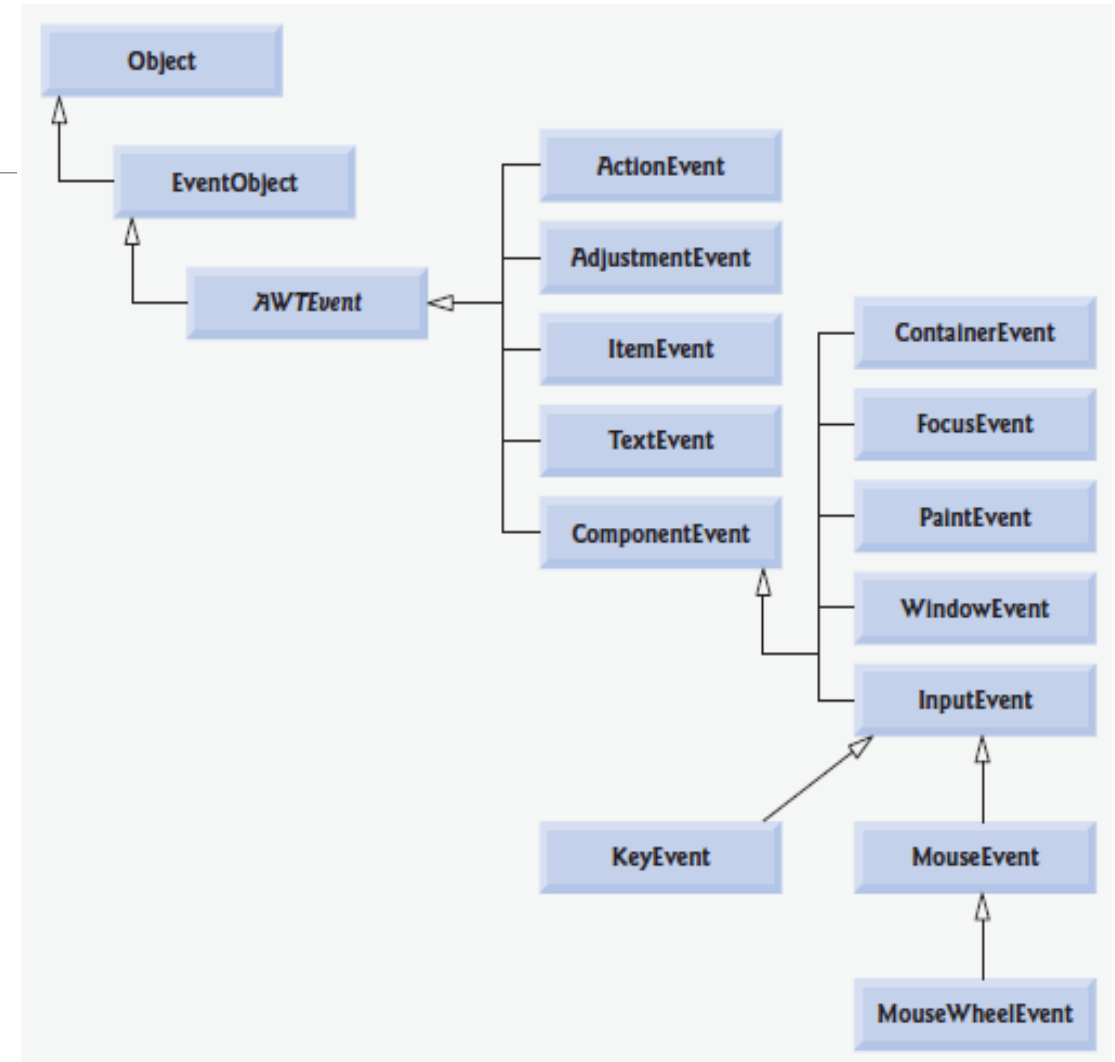


Introducción a Swing

✓ Manejo de Eventos

Evento: Es el objeto que representa un tipo de evento y permite referenciar la fuente del evento en el escucha.

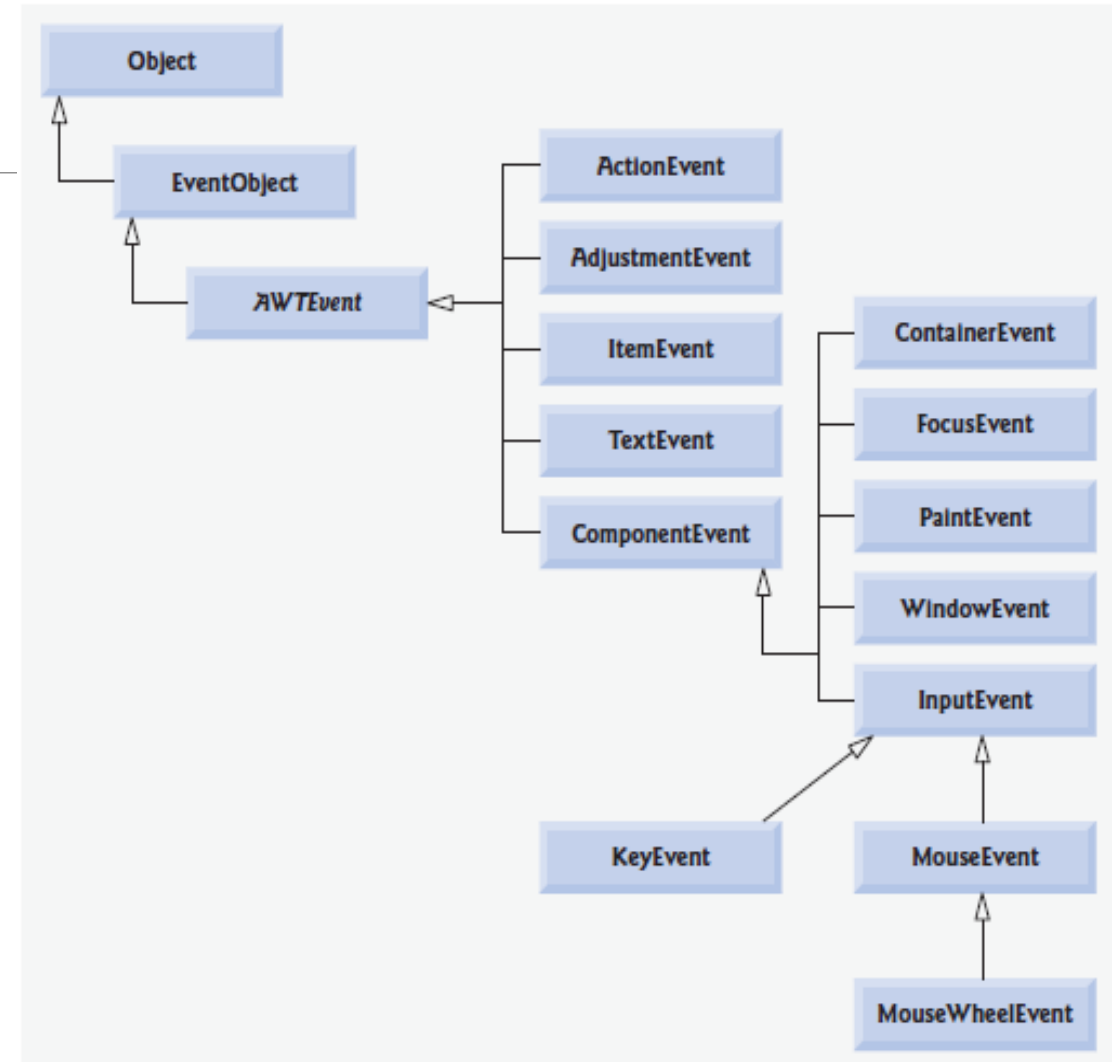
Todos los componentes implementan un único escucha de eventos.



Introducción a Swing

✓ Manejo de Eventos

Evento: Es el objeto que representa un tipo de evento y permite referenciar la fuente del evento con el escucha del evento



Introducción a Swing



✓ *Ventanas de Dialogo - JOptionPane*

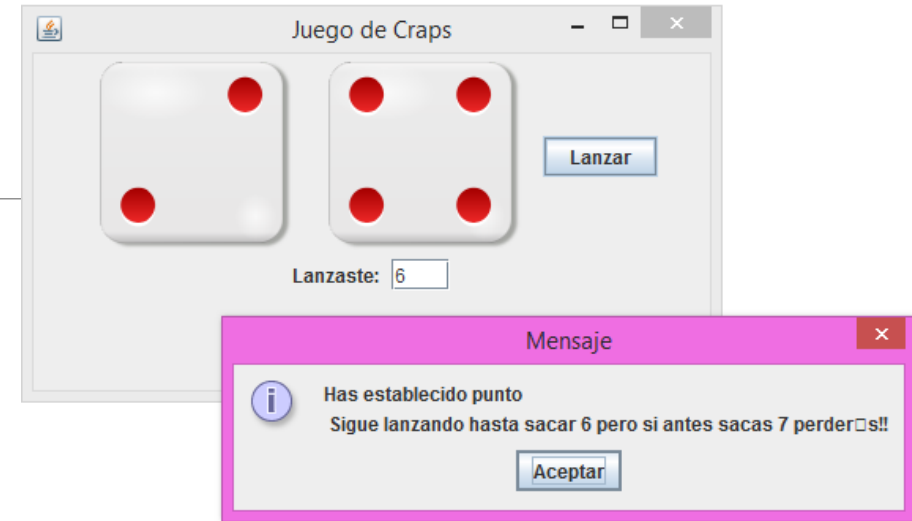


`JOptionPane.showMessageDialog(null,String)`

Introducción a Swing

✓ ¿Cómo se aplica esto?

Para el Juego de Craps, cree una GUI que responda a los siguientes bocetos:



Introducción a Swing



✓ ¿Cómo se aplica esto?

- **Fuente:** `JButton`
- **Escucha:** interfaz `ActionListener` obliga a incluir el método `actionPerformed()` que el desarrollador de programar.
- **Evento:** `ActionEvent`. Esta clase define el método que permite identificar la fuente origen del evento (`getSource()`) y las acciones a ejecutar ante la ocurrencia del evento.

