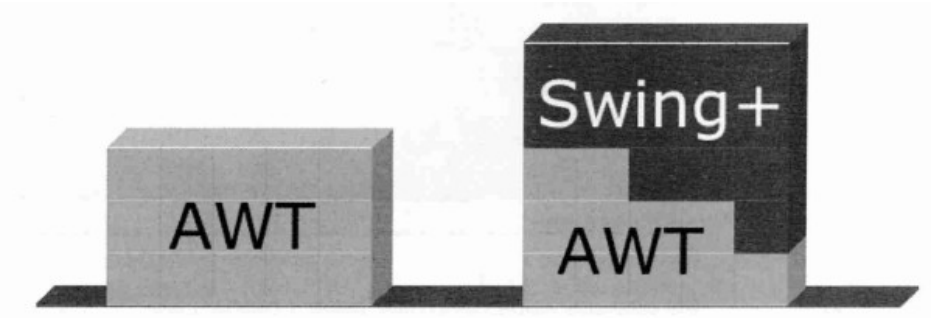
GUI en Java

Swing

Swing

- Swing es una colección de clases, divididas en 14 package que permiten crear interfaces gráficas.
- Swing es la segunda generación de GUI, AWT fue su predecesora.
- Hoy coexisten AWT y Swing con clases equivalente y solo diferenciadas por un prefijo J, indicando que esta clase pertenece a Swing.



Swing

- A diferencia de AWT, en Swing todos sus componentes son “lightweight”.
- Se recomienda encarecidamente no mesclar componentes de AWT y Swing, solo use swing.

Swing y threads

- Un thread es un proceso lightweight.
- La mayoria de los componentes de Swing no son thread-safe.
- La solucion para esto es asegurarnos que todo el codigo que crea y modifica componentes de Swing en nuestro programa se ejecute en el mismo 'event-dispatching' thread.
- Debe lanzar una programa en con Swing usando el siguiente código.

Thread:Lanzar una aplicación Swing

```
public static void main(String[] args) {
 SwingUtilities.invokeLater(new Runnable()
{
    public void run()
{
          new Ventana().setVisible(true);
           }
       });
   }
```

Clase Ventana

```
public class Ventana extends JFrame{
//Constructor
public Ventana() {
    initComponents();
}
private void initComponents() {
   jLabel1 = new Jlabel();
    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    jLabel1.setText("Holi");
    getContentPane().add(jLabel1);
    pack();
}
}
```

Layout Manager: Administrador de disposición

- La mayoria de interfaces Swing utiliza un ***LayoutManager*** para controlar la disposición de los componentes de Swing.
- Algunos de los LayoutManager que podemos encontrar estan:
  - FlowLayout
  - GridLayout
  - GroupLayout
- El GrupoLayout es el administrador de distribución por defecto ocupado por netbeans.

Listener:responder a las acciones de los usuarios

- Basado en un modelo de manejo de eventos.
- Nuevos componentes, como un boton deberian tener un atendedor (Listener) especificado.
- Los objetos Listener son programados para responder a los objetos eventos (Event) que vienen desde el componente.
- Para usar los objetos Listener se debe implementar las interfaces apropiadas.

Interfaces

- Una interfaz es un conjunto de metodos
- Por ejemplo la interfaz ActionListener tiene solo un método

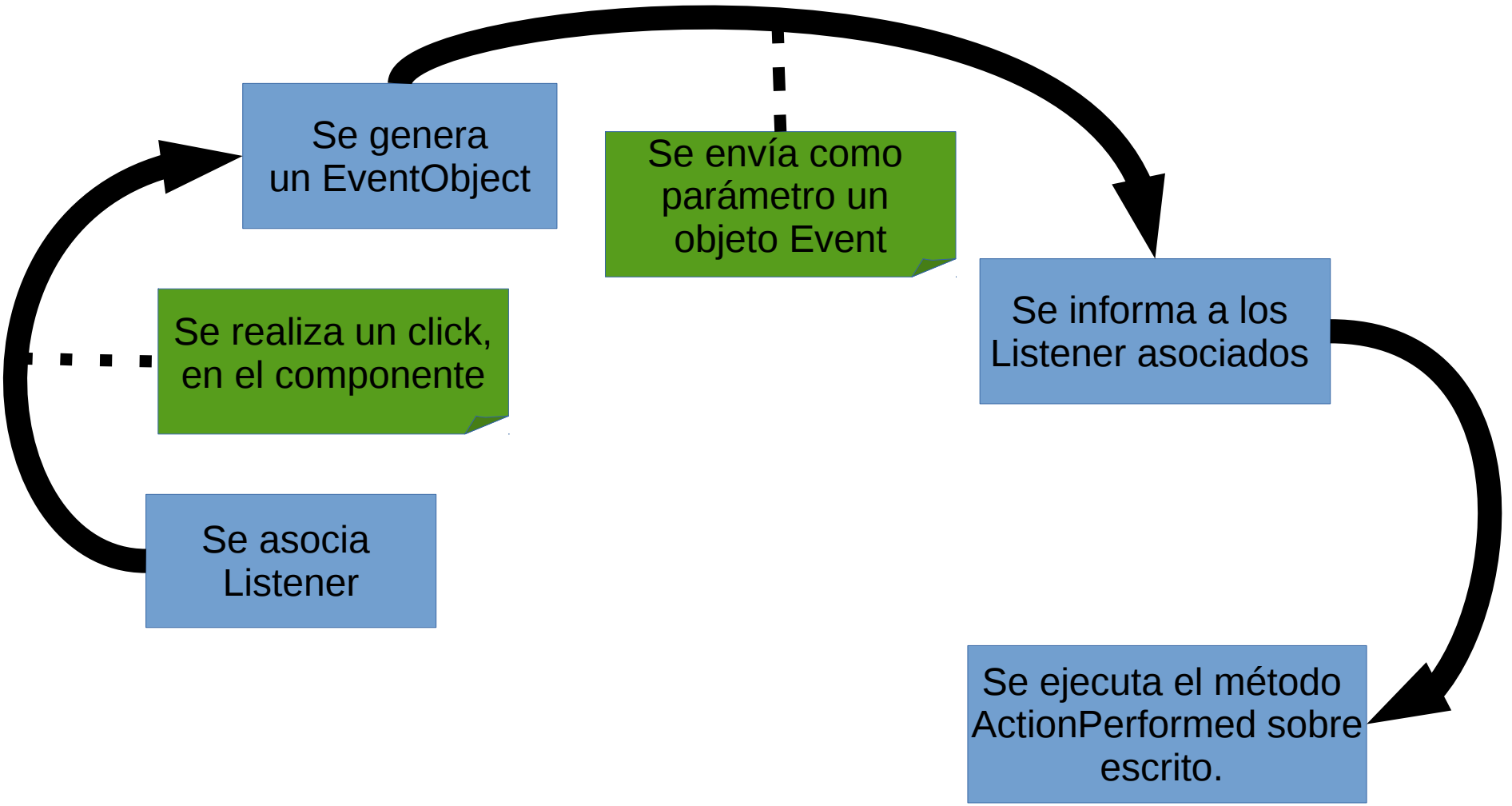
public void **actionPerformed**(**ActionEvent** e).

- Una clase puede declarar que implementa una interfaz, y con esto se obliga a sobre escribir sus metodos.

public class **Main** implements **ActionListener**

- Las clases pueden implementar multiples interfaces.

Listerner: como funciona?



Ejemplo de Listener

```
jButton1 = new JButton();
jButton1.setText("jButton1");
jButton1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
            JLabel1.setText(“Chaito”);
    }
});
```

Llamar a otro JFrame

- Para llamar a otra ventana y mostrarla escondiendo la anterior.
- Comúnmente se realiza mediante un evento de click.
- Ejemplo:

```
public void actionPerformed(ActionEvent e) {
        this.setVisible(false);
        new NuevaVentana().setVisible(true); // Muestra la nueva ventana
    }
```

Llamar a otro JFrame

- Una vez que hemos utilizado nuestro Jframe, y no lo volveremos a ocupar, debemos cerrar.
- Para esto debemos devolver los recursos ocupados en esta, esto se realiza con el método **dispose().**

Swing components: JPanel

- Es una subclases de JComponent
- Tal como todos los otro compoentes de Swing.
- Es una herremienta de división logica de la interfaz.
- Se pueden definir un borde, para remarcar mas su labor de divisor.

JPanel: ejemplo

```
JPanel myPanel = new JPanel();
myPanel.setBackground(new Color(255,3,25));
myPanel.setOpaque(true);
//Make it the content pane.
setContentPane(myPanel);
```

JDialog: enviar mensajes al usuario

- Un JDialog puede ser utilizado de forma modal o nonmodal.
  - Cuando abres un dialogo en modo modal se despliega una ventana informativa y el JFrame que lo llama se bloquea hasta que se entregue una respuesta desde el JDialog.
  - Cuando abres una dialogo en modo nonmodal este JDialog se ejecuta en un threads separado y el JFrame principal puede seguir su funcionamiento.
- Swing provee algunos modal JDialogs para su uso.
- EL programador también puede crear sus propios diálogos

JDialog: ejemplo

- Informativo

JoptionPane.**showMessageDialog**(frame,"Debe seleccionar una cuenta corriente");

- Advertencia

JOptionPane.**showMessageDialog**(frame,"Debe seleccionar una cuenta corriente", JoptionPane.WARNING_MESSAGE);

- Error

JoptionPane.**showMessageDialog**(frame, "Debe seleccionar una cuenta corriente", JOptionPane.ERROR_MESSAGE);