

## Problema 18.

El paquete Swing también identificado como paquete javax.swing, está incluida en la JFC, esta provee facilidades para proporcionar ayuda a los programadores para construir GUIs, Swing abarca componentes como marcos, tablas, botones y más. Las componentes de Swing utilizan la infraestructura de AWT, incluyendo el modelo de eventos AWT, el cual rige cómo una componente reacciona a eventos tales como, eventos de teclado, mouse, etc... Es por esto, que la mayoría de los programas Swing necesitan importar dos paquetes AWT: java.awt.\* y java.awt.event.\*

Ejemplos de eventos y sus escuchadores	
Acción que gatilla un evento	Tipo de escuchador
El usuario hace un click, presiona Return en un área de texto o selecciona un menú	ActionListener
El usuario elige un frame (ventana principal)	WindowListener
El usuario hace un click sobre una componente	MouseListener
El usuario pasa el mouse sobre una componente	MouseMotionListener
Una componente se hace visible	ComponentListener
Una componente adquiere el foco del teclado	FocusListener
Cambia la selección en una lista o tabla	ListSelectionListener

## Problema 19.

**ABSTRACCIÓN** es un proceso de interpretación y diseño que implica reconocer y enfocarse en las características importantes de una situación u objeto, y filtrar o ignorar todas las particularidades no esenciales. Dejar a un lado los detalles de un objeto y definir las características específicas de éste, aquellas que lo distinguen de los demás tipos de objetos. Hay que centrarse en lo que es y lo que hace un objeto, antes de decidir cómo debería ser implementado. Se hace énfasis en el qué hace más que en el cómo lo hace

La **MODULARIDAD** es la propiedad que permite dividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. En el lenguaje de programación Java y en algunos entornos de desarrollo integrado (IDE), la modularidad radica en la separación de todos los posibles paquetes de código. La siguiente imagen muestra la división que se hace a partir de una aplicación base o un servicio general, que se divide en diferentes módulos con un propósito especial y finalmente una subdivisión en pequeños paquetes de código que integran las clases.

**ENCAPSULAMIENTO** es la propiedad que permite asegurar que la información de un objeto está oculta del mundo exterior. El encapsulamiento consiste en agrupar en una Clase las características (atributos) con un acceso privado y los comportamientos (métodos) con un acceso público. Acceder o modificar los miembros de una clase a través de sus métodos.