1. **ancestorAdded**

The ancestorAdded event is triggered when a component is added to an ancestor (like a JPanel or JFrame) that is part of the component hierarchy. This event occurs when the component becomes part of a displayable hierarchy that has a Window as its root.

### ancestorMoved

The ancestorMoved event is triggered when an ancestor of a component is moved. This typically happens when a container (like a JFrame or JPanel) containing the component is moved to a new location on the screen.

### ancestorRemoved

The ancestorRemoved event is triggered when a component is removed from its ancestor hierarchy. This occurs when the component is no longer part of a displayable hierarchy with a Window as its root.

### ancestorResized

The ancestorResized event is triggered when an ancestor of a component is resized. This happens, for example, when the window or panel containing the component changes size.

### caretPositionChanged

The caretPositionChanged event is triggered when the position of the caret (the text cursor) changes in a text component like a JTextField or JTextArea. This event is useful for tracking where the user is typing or editing in the text component.

### componentAdded

The componentAdded event is triggered when a component is added to a container. This event is part of the ContainerListener interface rather than ComponentListener, and it allows you to respond whenever a component is added to a container like a JPanel or JFrame.

### componentHidden

The componentHidden event is triggered when a component becomes invisible. This event is part of the ComponentListener interface and occurs when the setVisible(false) method is called on a component.

### componentMoved

The componentMoved event is triggered when a component's position changes. This can occur when the component is manually repositioned within its parent container or when the container itself is moved.

### componentRemoved

The componentRemoved event is triggered when a component is removed from a container. Like componentAdded, this event is part of the ContainerListener interface and allows you to detect when components are dynamically removed from their parent container.

### componentResized

The componentResized event is triggered when a component's size changes. This event is part of the ComponentListener interface and occurs when the dimensions of a component are altered, either programmatically or by user interaction.

### componentShown

The componentShown event is triggered when a component becomes visible to the user. This event is part of the ComponentListener interface and occurs when the setVisible(true) method is called on a component, or when the component is made visible as part of its container.

### focusGained

The focusGained event is triggered when a component gains focus, meaning it becomes the active component that can receive keyboard input. This event is part of the FocusListener interface.

### focusLost

The focusLost event is triggered when a component loses focus, meaning it is no longer the active component that can receive keyboard input. This event is part of the FocusListener interface.

### hierarchyChanged

The hierarchyChanged event is triggered when there is a change in the hierarchy of a component, such as when a component is added, removed, or moved within its container hierarchy. This event is part of the HierarchyListener interface.

### inputMethodTextChanged

The inputMethodTextChanged event is triggered when text is input or modified using an input method, such as when typing in languages that require complex character composition (e.g., Chinese, Japanese). This event is part of the InputMethodListener interface.

### keyPressed

The keyPressed event is triggered when a key is pressed down on the keyboard. This event is part of the KeyListener interface and is commonly used to respond to specific key actions.

### keyReleased

The keyReleased event is triggered when a key is released after being pressed. This event is part of the KeyListener interface and can be used to detect when a key press is finished.

### keyTyped The keyTyped event is triggered when a key is typed. This event occurs when a key press generates a character (e.g., letters or numbers) and is part of the KeyListener interface. It’s often used to capture text input.

### mouseClicked

The mouseClicked event is triggered when a mouse button is clicked (pressed and released) on a component. This event is part of the MouseListener interface and is often used to detect mouse interactions, such as selecting items.

### mouseDragged

The mouseDragged event is triggered when a mouse button is pressed on a component and then dragged (moved while holding the button). This event is part of the MouseMotionListener interface and is used for operations like dragging objects.

### mouseEntered

The mouseEntered event is triggered when the mouse pointer enters a component's area. This event is part of the MouseListener interface and is used to detect when the mouse cursor moves over a component.

### mouseExited

The mouseExited event is triggered when the mouse pointer leaves a component's area. This event is part of the MouseListener interface and is used to detect when the mouse cursor moves away from a component.

### mouseMoved

The mouseMoved event is triggered when the mouse pointer moves within a component's area without pressing any buttons. This event is part of the MouseMotionListener interface and is used to track mouse movements.

### mousePressed

The mousePressed event is triggered when a mouse button is pressed down on a component. This event is part of the MouseListener interface and is used to detect the start of a mouse click.

### mouseReleased

The mouseReleased event is triggered when a mouse button is released after being pressed. This event is part of the MouseListener interface and is used to detect the end of a mouse click.

### mouseWheelMoved

The mouseWheelMoved event is triggered when the mouse wheel is moved. This event is part of the MouseWheelListener interface and is used to respond to mouse wheel scrolling actions.

### propertyChange

The propertyChange event is triggered when a bound property of a bean changes. This event is part of the PropertyChangeListener interface and is commonly used to listen for changes in properties of JavaBeans.

### vetoableChange

The vetoableChange event is triggered when a bound property change is proposed but can be vetoed by the listener. This event is part of the VetoableChangeListener interface and allows listeners to prevent property changes.

### windowActivated

The windowActivated event is triggered when a window is activated, meaning it becomes the currently focused window. This event is part of the WindowListener interface.

### windowClosed

The windowClosed event is triggered when a window is closed. This event is part of the WindowListener interface and is used to handle actions that need to occur when a window is closed.

### windowClosing

The windowClosing event is triggered when a user initiates the closing of a window. This event is part of the WindowListener interface and is used to handle cleanup or prompts before a window is actually closed.

### windowDeactivated

The windowDeactivated event is triggered when a window loses focus and is no longer the active window. This event is part of the WindowListener interface and is used to handle actions when a window is deactivated.

### windowDeiconified

The windowDeiconified event is triggered when a window is restored to its normal size after being minimized (iconified). This event is part of the WindowListener interface and is used to handle actions when a window is restored from its minimized state.

### windowGainedFocus

The windowGainedFocus event is triggered when a window gains focus, meaning it becomes the currently active window. This event is part of the WindowFocusListener interface and is used to handle actions when a window gains focus.

### windowIconified

The windowIconified event is triggered when a window is minimized (iconified). This event is part of the WindowListener interface and is used to handle actions when a window is minimized.

### itemStateChanged

The itemStateChanged event is triggered when the state of an item in a component such as a checkbox or radio button changes. This event is part of the ItemListener interface and is used to respond to changes in selection state.

### popupMenuCanceled

The popupMenuCanceled event is triggered when a popup menu is canceled. This event is part of the PopupMenuListener interface and is used to handle scenarios where a popup menu is dismissed without selection.

### popupMenuWillBecomeInvisible

The popupMenuWillBecomeInvisible event is triggered when a popup menu is about to become invisible. This event is part of the PopupMenuListener interface and is used to handle actions just before the menu is hidden.

### popupMenuWillBecomeVisible

The popupMenuWillBecomeVisible event is triggered when a popup menu is about to become visible. This event is part of the PopupMenuListener interface and is used to handle actions just before the menu is shown.

### adjustmentValueChanged

The adjustmentValueChanged event is triggered when the value of an adjustable component (like a scrollbar) changes. This event is part of the AdjustmentListener interface and is used to respond to changes in scroll positions or other adjustable values.

### menuKeyPressed

The menuKeyPressed event is triggered when a key is pressed while the menu is active. This event is part of the MenuKeyListener interface and is used to handle key presses in menus.

### menuKeyReleased

The menuKeyReleased event is triggered when a key is released while the menu is active. This event is part of the MenuKeyListener interface and is used to handle key releases in menus.

### menuKeyTyped

The menuKeyTyped event is triggered when a key is typed while the menu is active. This event is part of the MenuKeyListener interface and is used to handle key typing in menus.

### windowStateChanged

The windowStateChanged event is triggered when the state of a window changes (e.g., maximized, minimized, or restored). This event is part of the WindowListener interface and is used to respond to changes in the window's state.

### windowOpened

The windowOpened event is triggered when a window is opened for the first time. This event is part of the WindowListener interface and is used to perform actions when a window is initially shown.

### windowLostFocus

The windowLostFocus event is triggered when a window loses focus, meaning it is no longer the active window. This event is part of the WindowFocusListener interface and is used to handle actions when a window loses focus.