

يستخدم هذا الإنترفيس للتنبيه لأي شيء يحدث للنافذة مثل عند تصغيرها, تكبيرها, فتحها, خروجها, ناشطة, غير ناشطة.

الإنترفيس `WindowListener` يحتوي على خمس دوال يجب أن تفعل لها **Override** عند إنشاء كائن منه:

- الدالة `windowOpened(WindowEvent e)` : يتم إستدعاءها بعد أن يتم فتح النافذة.

- الدالة `windowClosing(WindowEvent e)` : يتم إستدعاءها أثناء إغلاق النافذة

- الدالة `windowClosed(WindowEvent)` : يتم إستدعاءها بعد أن يتم إغلاق النافذة.

- الدالة `windowIconified(WindowEvent)` : يتم إستدعاءها بعد أن يتم إخفاء النافذة بالنقر على زر التصغير الموجود أعلاها.

- الدالة `windowDeiconified(WindowEvent e)` : يتم إستدعاءها بعد أن يتم إظهارها من جديد.

- الدالة `windowActivated(WindowEvent)` : يتم إستدعاءها في حال كانت النافذة ناشطة, أي في حال لم يكن هناك أي نافذة أخرى أو تطبيق مفتوح فوقها.

- الدالة `windowDeactivated(WindowEvent e)` : يتم إستدعاءها في حال كانت النافذة غير ناشطة, أي في حال كان هناك نافذة أخرى أو تطبيق مفتوح فوقها.

هنا قلنا أنه كلما تفاعل المستخدم مع النافذة سيتم إظهار الحدث الذي فعله كرسالة //

```
frame.addListener(new WindowListener() {

    @Override
    public void windowOpened(WindowEvent e) {
        JOptionPane.showMessageDialog(frame, "Event: Window Opened");
    }

    @Override
    public void windowClosing(WindowEvent e) {
        JOptionPane.showMessageDialog(frame, "Event: Window Closing");
    }

    @Override
    public void windowClosed(WindowEvent e) {

    }

    @Override
    public void windowIconified(WindowEvent e) {
        JOptionPane.showMessageDialog(null, "Event: Window Iconified");
    }

    @Override
    public void windowDeiconified(WindowEvent e) {
        JOptionPane.showMessageDialog(null, "Event: Window Deiconified");
    }

    @Override
    public void windowActivated(WindowEvent e) {

    }

    @Override
    public void windowDeactivated(WindowEvent e) {

    }

});
```

addWindowListener: Diese Methode wird aufgerufen, um einem JFrame (oder einer anderen Fenster-Komponente) einen WindowListener hinzuzufügen. Dieser Listener reagiert auf verschiedene Ereignisse, die mit dem Fenster verbunden sind, wie das Öffnen, Schließen, Minimieren oder Wiederherstellen des Fensters.

new WindowListener(): WindowListener ist ein Interface, das mehrere Methoden enthält, die jeweils auf unterschiedliche Fensterereignisse reagieren. Diese Methoden müssen überschrieben werden, um festzulegen, wie das Programm auf die verschiedenen Ereignisse reagieren soll. In deinem Code werden alle relevanten Methoden des WindowListener-Interfaces überschrieben.

WindowEvent e: WindowEvent ist eine Klasse, die Informationen über das Fenstereignis enthält. Es gibt Auskunft darüber, welches Fenster das Ereignis ausgelöst hat und welcher spezifische Typ von Fensterevent aufgetreten ist (z.B. ob das Fenster geöffnet, geschlossen, minimiert usw. wurde).

Die überschriebenen Methoden:

- **windowOpened(WindowEvent e):** Diese Methode wird aufgerufen, wenn das Fenster geöffnet wird. In deinem Code wird ein Dialogfeld mit der Nachricht "Event: Window Opened" angezeigt.
- **windowClosing(WindowEvent e):** Diese Methode wird aufgerufen, wenn das Fenster geschlossen wird, aber bevor das Schließen abgeschlossen ist. In deinem Code wird ein Dialogfeld mit der Nachricht "Event: Window Closing" angezeigt.
- **windowClosed(WindowEvent e):** Diese Methode wird aufgerufen, nachdem das Fenster geschlossen wurde. (In deinem Code passiert hier aktuell nichts.)
- **windowIconified(WindowEvent e):** Diese Methode wird aufgerufen, wenn das Fenster minimiert (in die Taskleiste verschoben) wird. Ein Dialogfeld mit der Nachricht "Event: Window Iconified" wird angezeigt.
- **windowDeiconified(WindowEvent e):** Diese Methode wird aufgerufen, wenn das Fenster aus dem minimierten Zustand wiederhergestellt wird. Ein Dialogfeld mit der Nachricht "Event: Window Deiconified" wird angezeigt.
- **windowActivated(WindowEvent e):** Diese Methode wird aufgerufen, wenn das Fenster den Fokus erhält, also aktiv wird. (In deinem Code passiert hier aktuell nichts.)
- **windowDeactivated(WindowEvent e):** Diese Methode wird aufgerufen, wenn das Fenster den Fokus verliert, also inaktiv wird. (In deinem Code passiert hier aktuell nichts.)