

M03

Programació

CFGS Desenvolupament d'Aplicacions Web



INSTITUT

THOS I CODINA

Events Esdeveniments

Definició

Els esdeveniments són accions o successos que es generen en aplicacions gràfiques definides pels components i ocasionat pels usuaris, com pressionar un botó , ingressar un text, canviar de color, etc.

Definició

- Els esdeveniments li corresponen a les interaccions de l'usuari amb els components.
- Els components estan associats a diferents tipus d'esdeveniments.
- Un esdeveniment serà un objecte que representa un missatge asíncron que té un altre objecte com a destinatari.

Definició

Components



Events

Listeners



mètodes

Canvi de Paradigma

Quan es programa en mode consola, el `main()` determina l'ordre en que s'executen les operacions del programa.

En mode GUI, les operacions dependran de les accions de l'usuari: cada acció porta associada un o varis events que haurem de gestionar, definint el corresponent gestor d'events.

Canvi de Paradigma

Cada llenguatge gestiona el gestor d'events de manera diferent:

- **VB:** Cada component defineix els seus events i la resposta que ha de donar. Simple però poc flexible.
- **C:** Els events es situen en cua i s'han d'anar gestionant a mida que es treuen de la cua. Tediós de programar.
- **Java:** Es defineixen unes classes auxiliars EventListeners que reben i gestionen events específics. Metodologia molt flexible i cada component es pot tractar per separat.

Canvi de Paradigma

- Els gestors d'events (***EventListeners***) registren les fonts d'events (***EventSources***).
- Les fonts d'events envien un tipus d'objecte (***EventObject***) a tots els gestors registrats quan hi ha un event.
- Cada gestor realitza les accions adequades amb la informació de l'*EventObject* rebuda.


```
package cat.ieshoscodina;
import java.awt.*;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.*;
```

```
public class Window extends JFrame {
    public Window() {
        JPanel panel = new JPanel();
        JLabel etiqueta = new JLabel("DAW2 M03");
        etiqueta.setFont(new Font("Tahoma", 1, 18));
        etiqueta.setForeground(new java.awt.Color(51, 51, 255));
        panel.add(etiqueta);
```

EventSource

```
@Override
public void mouseClicked(MouseEvent e) {
    etiqueta.setText("Toni!");
    etiqueta.setFont(new Font("Tahoma", 1, 30));
    etiqueta.setForeground(new Color(255, 102, 102));
}
```

EventObject

MouseEvent

```
public void mouseEntered(MouseEvent e) {}
public void mouseExited(MouseEvent e) {}
public void mousePressed(MouseEvent e) {}
public void mouseReleased(MouseEvent e) {}
};
```

EventListener

```
getContentPane().add(panel);
setSize(300, 120);
setVisible(true);
}
```

```
public static void main(String args[]) {
    Window ventana = new Window();
    ventana.setTitle("Event de Ratolí ");
}
}
```



tipus

- **ActionEvent** : generat per activació de components
- **AdjustmentEvent** : generat per ajust de components ajustables com a barres de desplaçament
- **ContainerEvent** : generat quan els components s'agreguen o es lleven d'un contenidor
- **FocusEvent** : generat quan un component entra o surt del focus
- **ItemEvent** : generat quan un article se selecciona d'una llista, opció, o caixa de revisió mèdica

tipus

- **KeyEvent** : generat per activitat del teclat
- **MouseEvent** : generat per activitat del ratolí
- **PaintEvent** : generat quan un component es pinta
- **TextEvent** : generat quan un component del text es modifica
- **WindowEvent** : generat per activitat de la finestra (com tancar, obrir, minimitzar)

Implementacions

La classe *java.util.EventObject* és la classe base de tots els esdeveniments en Java.

La subclasse *java.awt.AWTEvent* és la classe base de tots els esdeveniments que s'utilitzen en la construcció de GUIs.

Cada tipus d'esdeveniment *XxxEvent* té associada una interfície *XxxListener*, que és la que ens permet definir *handlers* d'esdeveniments.

Per a simplificar la implementació d'alguns handlers d'esdeveniments, el paquet `java.awt.event` inclou classes base *XxxAdapter* que implementen les interfícies *XxxListener*.

The image features a light gray background with three horizontal, overlapping yellow brushstrokes. A white rectangular frame is centered over the middle brushstroke. Inside the frame, the text "SAX i DOM" is written in a bold, white, sans-serif font.

SAX i DOM

SAX i DOM

DOM i SAX, són dues eines que serveixen per a analitzar el llenguatge XML i definir l'estructura d'un document.

Podem fer una distinció entre les eines que són *validants* i les que són *No validantes*. Les primeres verifiquen que el document, a més d'estar ben format d'acord amb les regles de XML, respongui a una estructura definida en una Definició del Tipus de Document (DTD).

Els parsers DOM i SAX són independents del llenguatge de programació i existeixen versions particulars per a Java, VisualBasic, C, etc.



INSTITUT

THOS I CODINA

SAX

El Simple API for XML (SAIX) és una interfície simple per a aplicacions XML.

S'encarrega de recórrer l'estructura del document generant esdeveniments que corresponen als elements que es va trobant.

No utilitza gairebé memòria.



DOM

DOM (Document Object Model): representa l'arxiu en una estructura tipus arbre que farà servir per a llegir el document.

Càrrega tot el contingut en la memòria i pot acabar la memòria fins i tot amb documents de grandària mitjana.

Però es pot usar xpath i recórrer l'arbre, etc.

SAX VS DOM

DOM

Carrega el document sencer.

Representa el resultat com un arbre.

Permet buscar a l'arbre.

Deixa modificar l'arbre.

Bo per lectura de fitxers de dades i configuració.

SAX

Carrega fins que li dius de parar.

Dispara manegadors d'events per cada:

- Etiqueta d'inici (start tag)
- Etiqueta de cos (body tag)
- Etiqueta de final (end tag)

API de baix nivell.

Bo per documents molt grans, especialment si només necessites porcions petites del document.

API de SAX

SAX llegeix el document seqüencialment de principi a fi, sense carregar-lo en memòria, de manera que quan troba un element s'encarrega de llançar el seu esdeveniment associat.

Quan es llença l'esdeveniment, aquest pot ser capturat per a realitzar una funció determinada.

Aquesta API està definida en el paquet:

javax.xml.parsers

API de SAX

Perquè aquests esdeveniments es puguin capturar i realitzar les operacions que es desitgin s'ha d'usar un manejador d'esdeveniments (***Event Handler***)

Un manejador és una classe amb una sèrie de mètodes i cada mètode s'executarà quan l'analitzador capturi el seu esdeveniment associat. Aquests esdeveniments es produeixen en llegir un document (al començament del document, obertura o tancament d'un element, en trobar una instrucció de procés o un comentari, etc.)

API de SAX

SAX defineix quatre interfícies bàsiques de manejadors d'esdeveniments:

- ❖ **ContentHandler:** s'utilitza per a tractar esdeveniments generals del document, com a obertura i tancament d'etiquetes o quan apareixen blocs de text.
- ❖ **DTDHandler:** invocat per a tractar esdeveniments relacionats amb les DTD's.
- ❖ **EntityResolver:** s'utilitza per a resoldre referències a entitats externes.
- ❖ **ErrorHandler:** maneja els errors i warnings.

API de SAX

Per a poder fer ús de l'analitzador primer s'ha d'obtenir una instància d'una factoria d'analitzadors (**SAXParserFactory**). Amb aquesta factoria es crea l'analitzador (SAXParser, javax.xml.parsers) que encapsula un objecte de la interfície **Parser** (org.xml.sax).

A aquest analitzador se l'associa el/els manejador/s, que posseeixen els mètodes que s'han d'executar en capturar un esdeveniment llançat per l'analitzador.

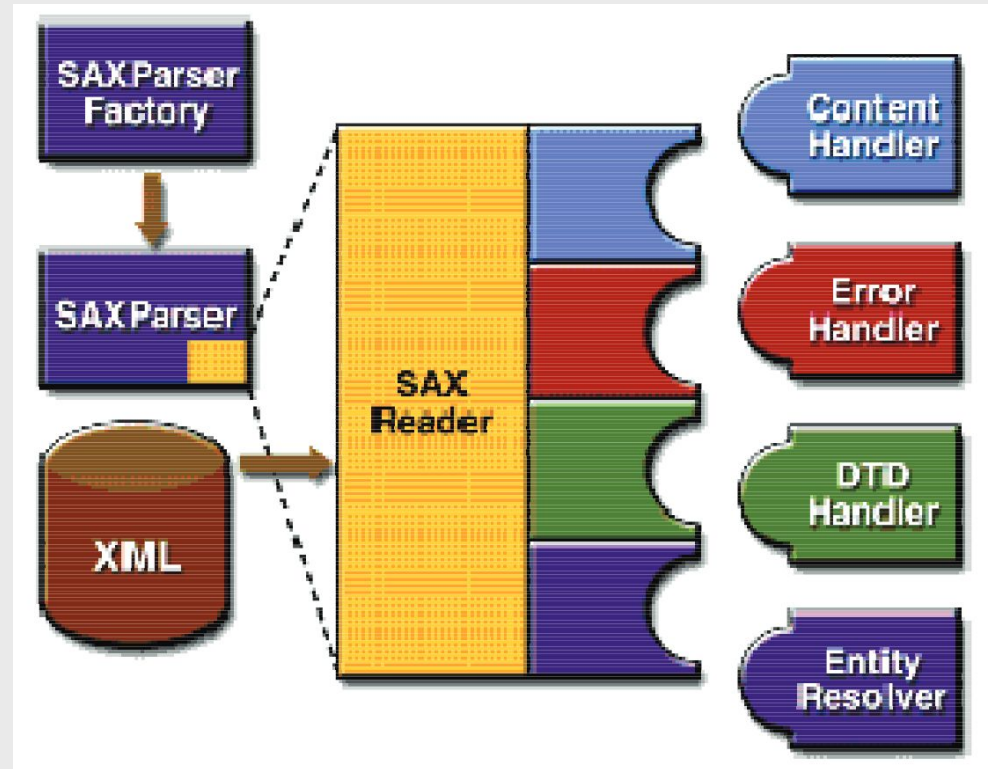
Finalment, se li passa a l'analitzador el document per a començar a llegir-lo i validar-lo.



INSTITUT

THOS I CODINA

API de SAX



API de SAX

Els mètodes del manejador no estan implementats, després s'hauran de codificar aquells que es vulguin utilitzar. Representen la lògica que s'ha d'executar en dur-se a terme cada esdeveniment, entre ells destaquen:

- ★ `startDocument()`
- ★ `endDocument()`
- ★ `startElement(nom, atributs)`
- ★ `endElement(nom)`
- ★ `characters(text)`



INSTITUT

THOSICODINA

API de SAX

startElement

<etiqueta>text amb contingut</etiqueta>



INSTITUT

THOS I CODINA

API de SAX

startElement

<etiqueta>text amb contingut</etiqueta>

characters



INSTITUT

THOSICODINA

API de SAX

startElement

endElement

<etiqueta>text amb contingut</etiqueta>

characters



INSTITUT

THOSICODINA

API de SAX

[Per més informació....](#)