



# Évènement et composants

---

1



## Évènement et composants

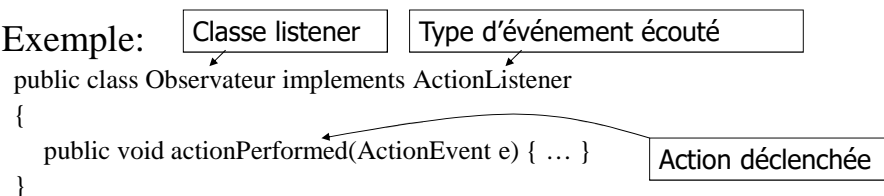
- Les évènements sont des objets issus de sous-classes de la classe abstraite `java.awt.AWTEvent`
- Les composants génèrent des évènements
- évènement: chaque interaction de l'utilisateur sur un composant génère un événement
  - Bouger la souris
  - Cliquer sur un bouton
  - Fermer une fenêtre
  - ...
- Un événement contient des informations: source, type d'événement, ...
  - Utile pour détecter d'où provient l'événement:  
`public Object getSource()`



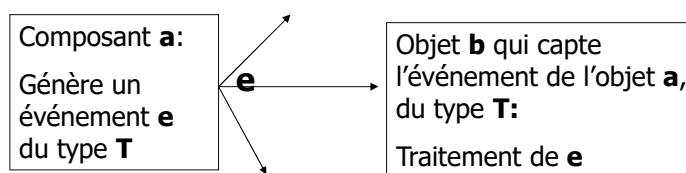
## Listener et Event handler: donner la capacité d'entendre un événement

- *Listener*: Un objet est intéressé à *écouter* l'événement produit (être signalé quand il y a un événement)
- Le listener doit implémenter l'interface *<event Listener interface>* associée à chaque type d'événement
- *eventHandler*: le programme qui lance un traitement suite à un événement

### Exemple:



## Propagation et traitement des événement



- Les événements sont générés et propagés
- Certains autres objets sont capables de capter des événements des types spécifiés, provenant de ces composants
  - **b** écoute les événements du type **T** venant de **a**
  - **b** est un *listener* de **a**
- On peut activer le traitement suite à la capture d'un événement
  - Le traitement lancé par l'objet **b**
- Programmation par événement
  - Le programme réagit aux événements



## Phase 1: coder la classe écouteur

```
class <class_name> implements <EventListener>
{
    <implementation of all the required methods>
    (eventHandler)
}
```

La classe est capable de capter les **évènement** du type  
<EventListener>

### Exemple:

```
public class Observateur implements ActionListener
{
    public void actionPerformed(ActionEvent e) { ... }
}
```



## Phase 2 : instancier l'écouteur



Phase 3: abonner l'écouteur au  
composant



END



XH

8