

## Programmation événementielle

- ◆ Les programmes événementiels ne sollicitent pas l'utilisateur.
- ◆ Ils réagissent aux sollicitations de l'utilisateur qui peuvent :
  - être de natures différentes (souris, clavier ...)
  - survenir à un rythme très variable.

## En environnement graphique

- ◆ Différents événements peuvent survenir
  - Souris
    - Enfoncement ou relâchement d'un bouton
    - Clic ou multiple-clic
    - Déplacement ou glisser-déposer
  - Clavier
  - Fenêtre
    - Ouverture et fermeture
    - Redimensionnement
    - ...

---

## Réagir aux événements

- ◆ Un écouteur est une classe qui définit les méthodes correspondant à une ou plusieurs catégories d'événements.
- ◆ Un écouteur doit être recensé auprès de la source d'événements (qui le transmet comme argument d'une méthode **add...Listener**).
- ◆ Il peut alors recevoir les événements et les traiter.

---

## Interfaces d'écouteurs

- ◆ Pour chaque catégorie d'événements, il existe une interface qui spécifie les méthodes correspondant à ces événements
  - **MouseListener** pour les événements souris
  - **MouseMotionListener** pour les mouvements de la souris
  - **KeyListener** pour les événements clavier
  - ...

## Les classes d'événements

- ◆ Les méthodes de traitement d'événement reçoivent en argument des objets (instances de classes d'événements) qui renseignent sur l'événement à traiter.
  - `MouseEvent` → `int getX(), int getY()`
  - `KeyEvent` → `char getKeyChar()`
  - ...

## Exemple (trop) minimal

```
public class FenetreDeDessin extends JFrame
implements MouseListener, MouseMotionListener {
    private int xd, yd;
    public void FenetreDeDessin() {
        ...
        addMouseListener(this); addMouseMotionListener(this);
    }
    public void mouseEntered(MouseEvent e) { }
    public void mousePressed(MouseEvent e) {
        xd = e.getX(); yd = e.getY();
    }
    public void mouseReleased(MouseEvent e) { }
    public void mouseClicked(MouseEvent e) { }
    public void mouseExited(MouseEvent e) { }
    public void mouseMoved(MouseEvent e) { }
    public void mouseDragged(MouseEvent e) {
        getGraphics().drawLine(xd, yd, e.getX(), e.getY());
        xd = e.getX(); yd = e.getY();
    }
}
```

MouseListener

MouseMotionListener