

# Programmation Graphique en Swing PO2 (2015-2016)

## Chapitre 4 Événements

# Événements

- Événements
- Événements et Ecouteurs
  - Exemple : écouteur avec classe anonyme
  - Exemple : écouteur avec classe interne
  - Exemple : MouseEvent et MouseListener

# Événements

- Au sens large, un événement désigne un changement d'état d'un objet.
- Au sens propre à une GUI, un événement désigne une action provoqué par l'utilisateur, par exemple, en cliquant un bouton, le souri, en appuyant une touche du clavier, etc.

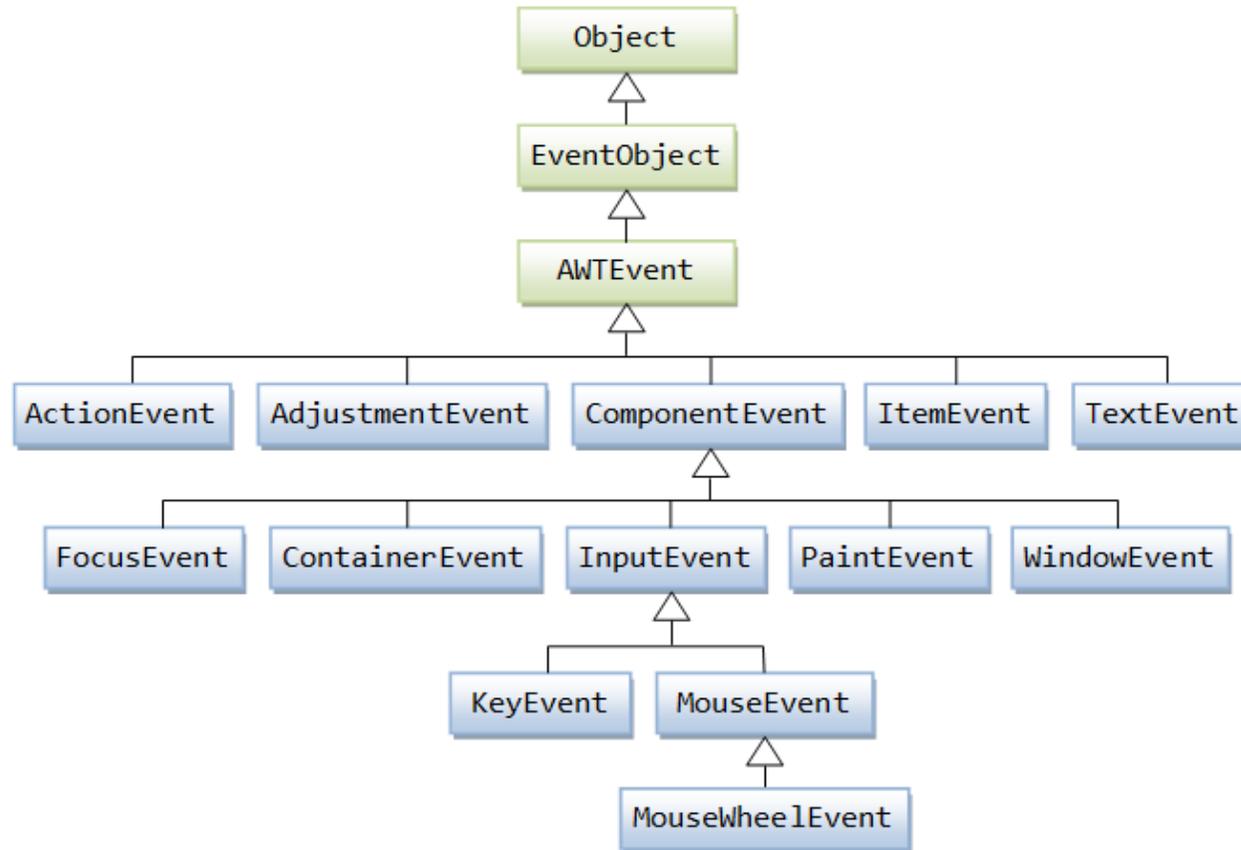
# Événements/Ecouteurs

- Composant : il produit un événement en conséquence d'une action de l'utilisateur.
- Ecouteur : il traite cet événement en exécutant une méthode en rapport avec la nature précise de cet événement.

# Événements/Ecouteurs

| Composant  | Événement                 | Ecouteur Interface              |
|--|---------------------------|---------------------------------|
| JButton  | ActionEvent               | ActionListener                  |
| JFrame   | WindowEvent               | WindowListener                  |
| Mouse  | MouseEvent                | MouseListener                   |
| JTextField                                       | TextEvent                 | TextListener                    |
| Key  | KeyEvent                  | KeyListener                     |
| Choice, JCheckbox,<br>JRadioButton,<br>JComboBox | ItemEvent,<br>ActionEvent | ItemListener,<br>ActionListener |

# Événements/Ecouteurs



# Exemple : JetDeEventAnonyme

```
// classes anonyme
jetButton1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("ActionEvent : "+e.toString());
        deGauche.jet();
        int nombre = deGauche.getVal();
        textDeGauche.setText(""+nombre);
    }
});

jetButton2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("ActionEvent : "+e.toString());
        deDroit.jet();
        int nombre = deDroit.getVal();
        textDeDroit.setText(""+nombre);
    }
});
```

- JetListener sous forme de classe interne anonyme
- Avantage :
  - accède facilement aux variables d'instance de la classe JetDePanel
- Inconvénient :
  - écouteur ne peut pas être partagé
  - code moins lisible

# Exemple : JetDeEventIntern

```
// créer une instance de JetListener
    JetListener jetListener = new JetListener();
    jetButton1.addActionListener(jetListener);
    jetButton2.addActionListener(jetListener);

...
// classe interne
private class JetListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String nomCom = e.getActionCommand();
        if (nomCom.equals("Joueur1")) {
            deGauche.jet();
            int nombre = deGauche.getVal();
            textDeGauche.setText(""+nombre);
        }
        else {
            deDroit.jet();
            int nombre = deDroit.getVal();
            textDeDroit.setText(""+nombre);
        }
    }
}
```

- JetListener sous forme de classe interne
- Méthodes permettant d'obtenir des informations sur ce qui est survenu
  - getSource() donne le composant source de l'événement
  - getActionCommand() donne le nom d'action, par défaut le texte du bouton, du menu, ...
- Avantage :
  - permet plusieurs instances

# MouseEvent et MouseListener

## Exemple : EventMouse

- L'écouteur MouseListener
  - mouseEntered() : entrée de souris
  - mouseExited() : sortie de souris
  - mousePressed() : bouton pressé
  - mouseReleased() : bouton relâché
  - mouseClicked() : bouton pressé puis relâché dans sa zone graphique
  - mouseMoved() : mouvement de souris
  - mouseDragged() : mouvement de souris avec bouton enfoncé
- L'écouteur MouseAdapter