

Informatique 1

# 14. Interfaces graphiques avec *Swing*



# Objectifs de la thématique

## **Pouvoir réaliser une GUI**

- ▶ Composants graphiques et démarche
- ▶ Le modèle événementiel

# Graphical User Interface (GUI)

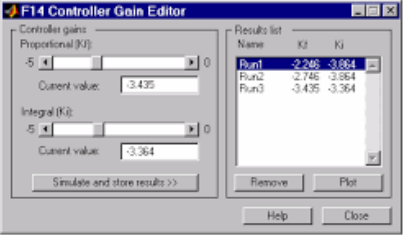
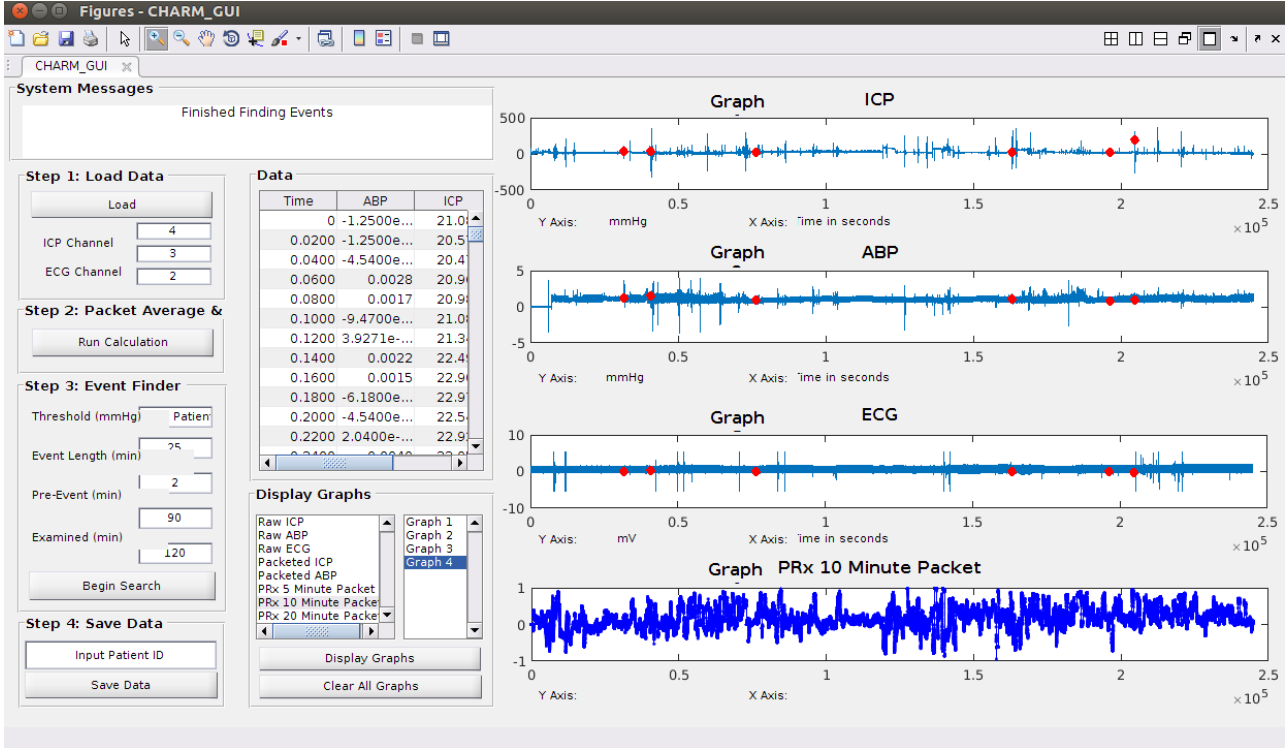
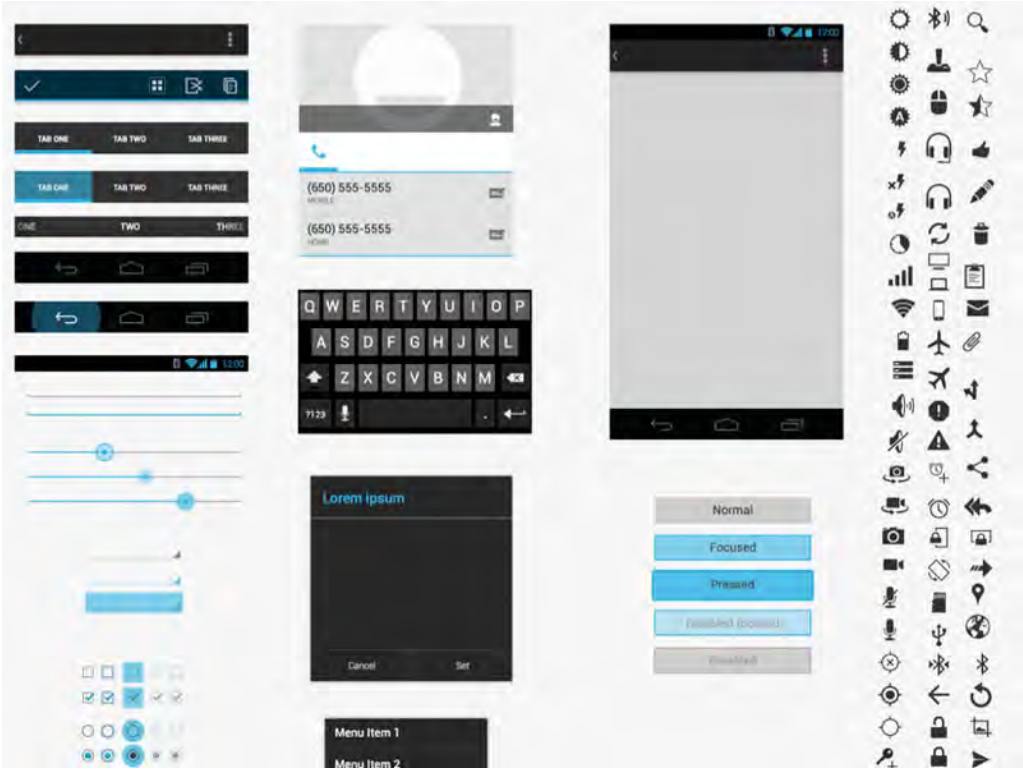
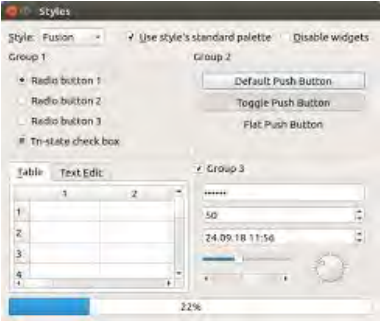
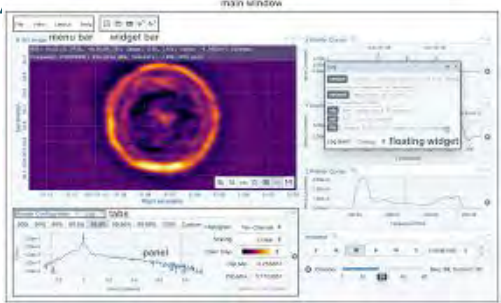
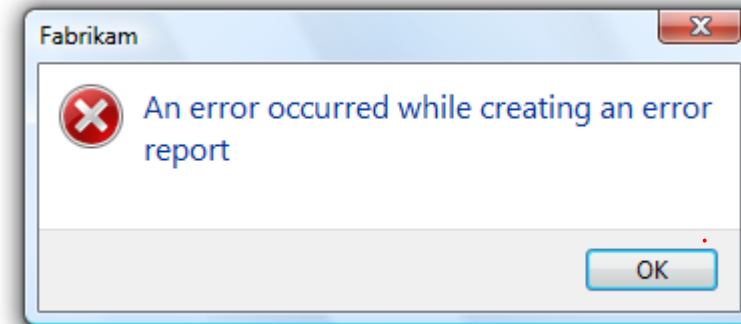
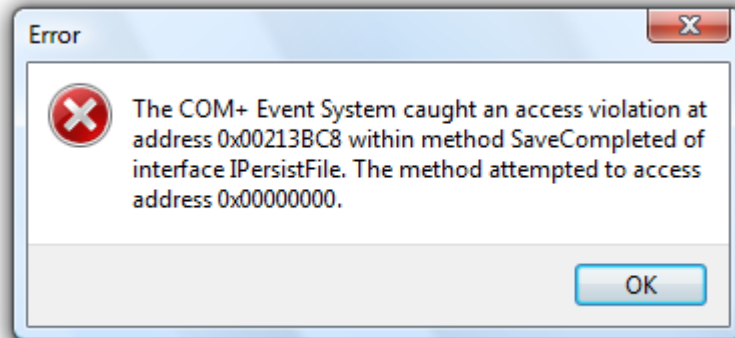


FIGURE 131. Hourglass



# Conception d'interfaces, principes (2)

- Principes (suite)
  - Aider l'utilisateur à résoudre les problèmes

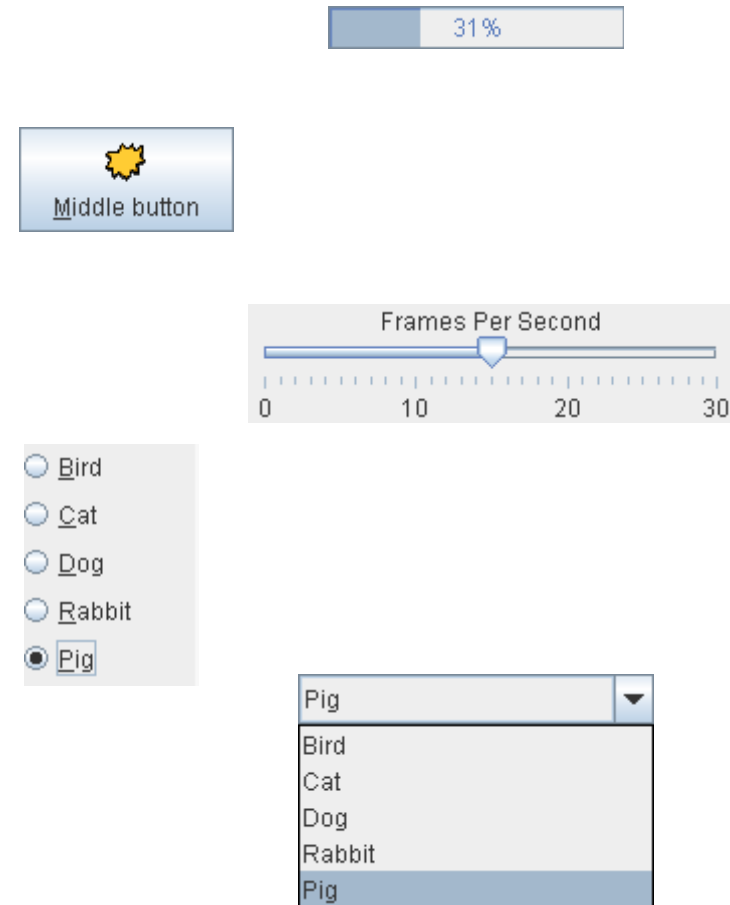


- Pour les intéressé-e-s : voir réf. [R1] et <https://userinyerface.com/>

# LES COMPOSANTS

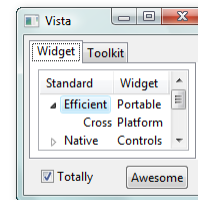
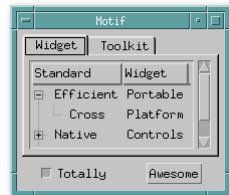
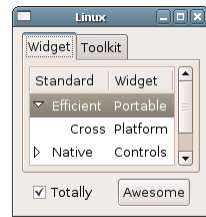
# Les composants

- GUI *principalement* combinaison éléments graphiques
- Permettent interaction
  - ▶ Saisie information
  - ▶ Exécution action
  - ▶ ...
- En Java, ces éléments sont nommés "**composants**"



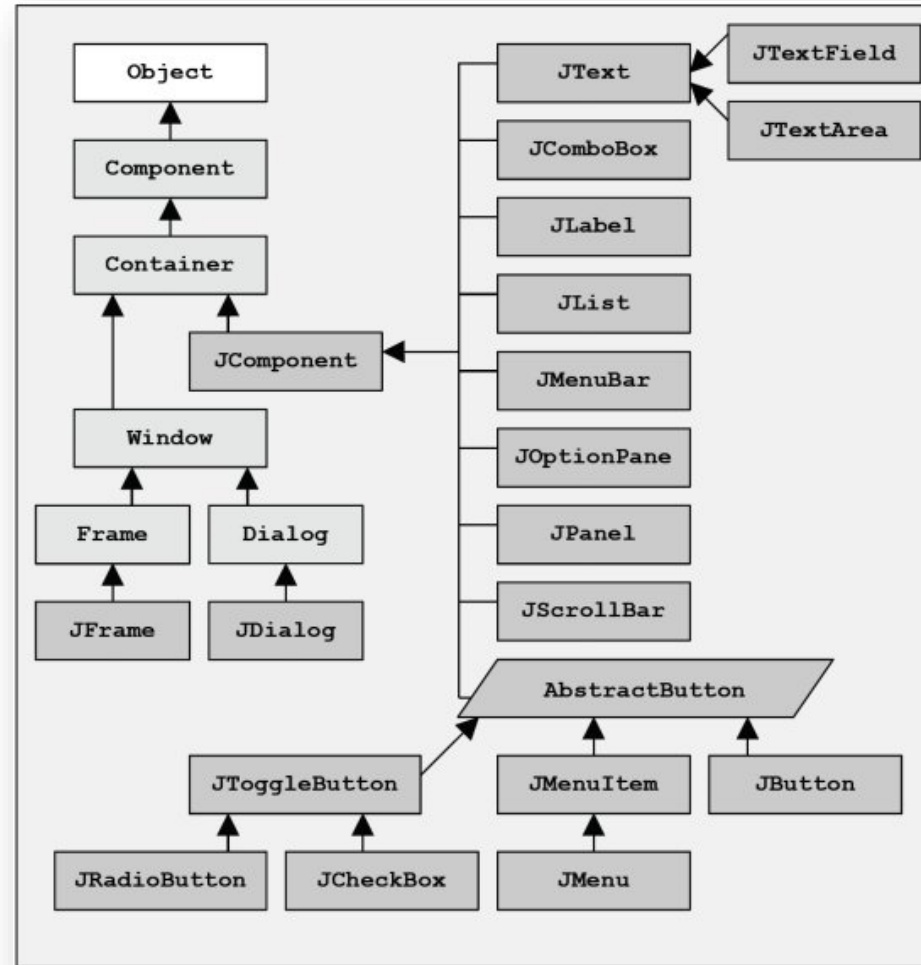
# Différentes APIS

- Fournies par le langage :
  - *AWT*
  - *Swing* (bâtie sur *AWT*)
  - *JavaFX*
- Il en existe d'autres :
  - *SWT*, utilisé notamment par Eclipse



-  Qt

# Hiérarchie de composants Swing



Guide visuel des composants :

<http://web.mit.edu/6.005/www/sp14/psets/ps4/java-6-tutorial/components.html>



# Autres éléments

- Fenêtre, menus
- Concepts non visibles
  - ▶ Agencement composants les uns par rapport aux autres
  - ▶ Gestionnaire de "layout"
    - Alignement
    - Dimensionnement

## Avec un titre...

- Utilisation de la classe `JFrame`

```
public class SimpleTitleWindow
{
    public static void main(String args[]){
        JFrame f = new JFrame("The title");
        f.setSize(200, 400);
        f.setLocation(100, 100);
        f.setVisible(true);
    }
}
```

# Meilleure implémentation

- On n'alourdit pas le main
- Bonne pratique :

**Avoir une classe qui étend JFrame**

```
public class GUIStarting
{
    public static void main(String args[]){
        AppFrame f = new AppFrame();
        f.setVisible(true);
    }
}

class AppFrame extends JFrame {
    AppFrame(){
        setTitle("A nice window");
        setSize(200, 400);
        setLocation(100, 100);
    }
}
```

# Démarche de construction

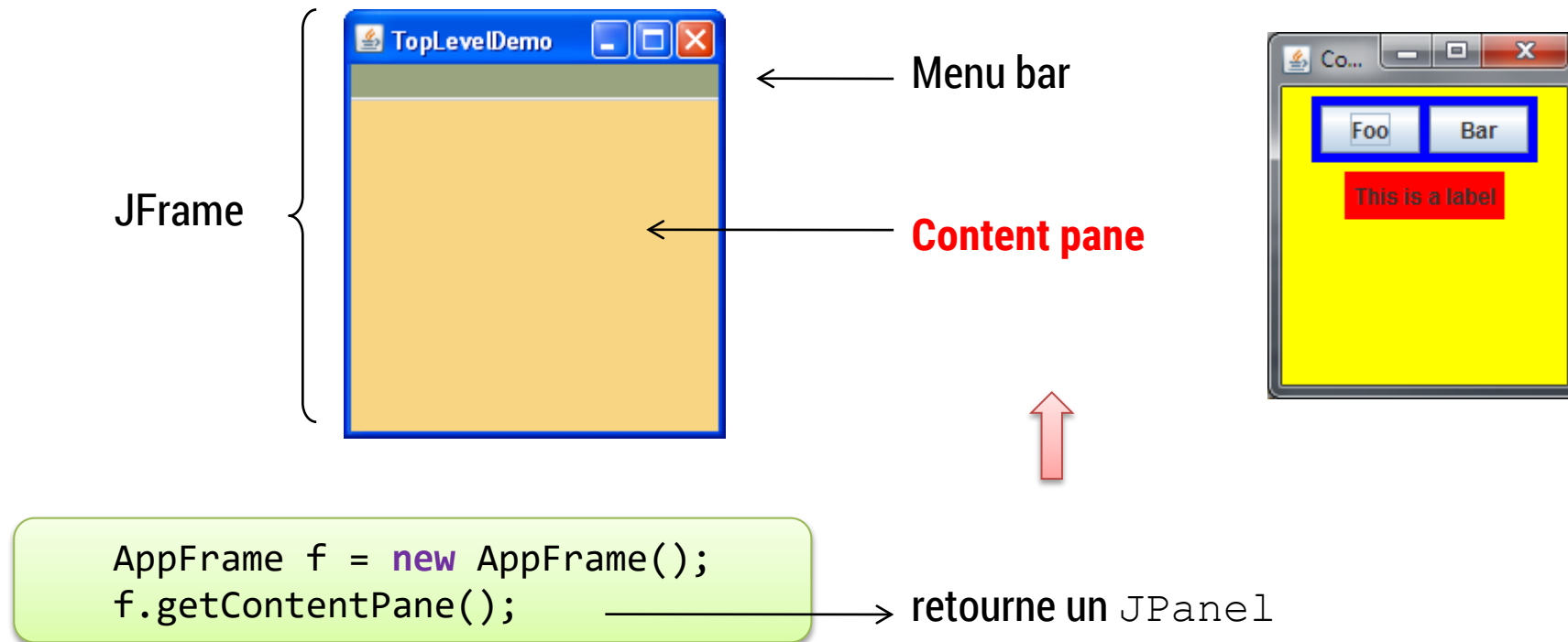
1. Construire les éléments graphiques
  - ▶ Fenêtre, composants
  - ▶ Panels (servent à contenir d'autres composants)
2. Construire et associer les *layout manager*
  - ▶ Un *layout manager* pour la fenêtre
  - ▶ Un *layout manager* pour chaque panneau
3. Attacher les éléments graphiques
  - ▶ Attacher les composants à une fenêtre ou panneau
  - ▶ Attacher chaque panneau à la fenêtre ou panneau

(tiré du cours C. Vanoirbeek, "Interface Utilisateur", S. 23, Version 2009)

**ComponentDemo.java**

# Remarque sur l'exemple

- Pour mettre du contenu, il faut des `JPanel`



# LE MODÈLE ÉVÉNEMENTIEL

# Gestion des évènements

- Les actions utilisateur → évènements Java
- Evènements ?
  - ▶ Classe Java de base *EventObject*



# Classes d'évènements

- Différents objets émis en fonction de l'interaction

Interaction de l'utilisateur	Evènement émis par Java
Passage du focus	FocusEvent
Clic sur un panel	MouseEvent
Frappe d'une touche clavier	KeyEvent
Iconification fenêtre	WindowEvent
Clic sur un bouton	ActionEvent
Déplacement d'un slider	ChangeEvent
...	...

# Réaction à des évènements

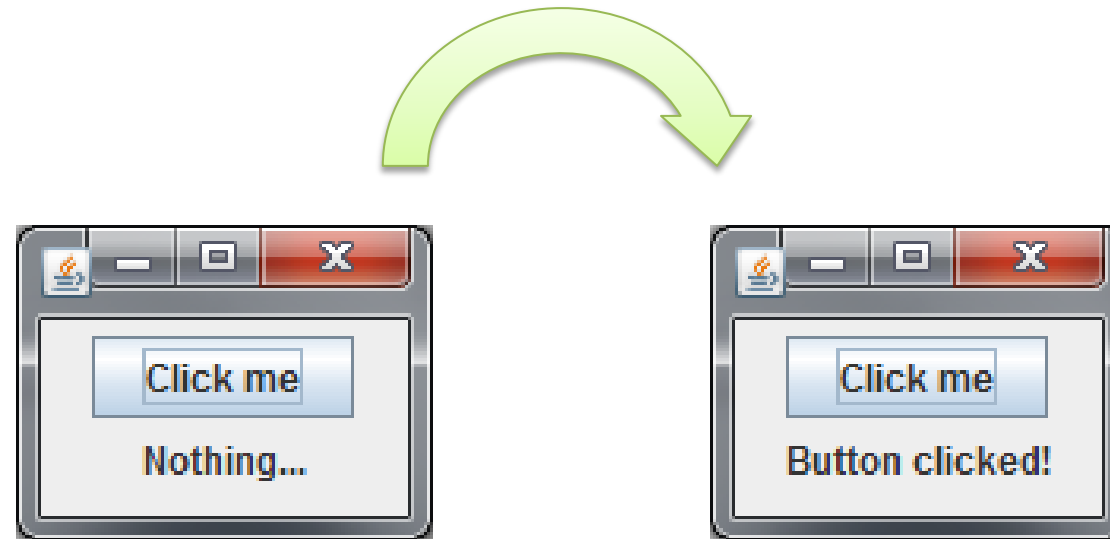


Nouveau modèle exécution code !

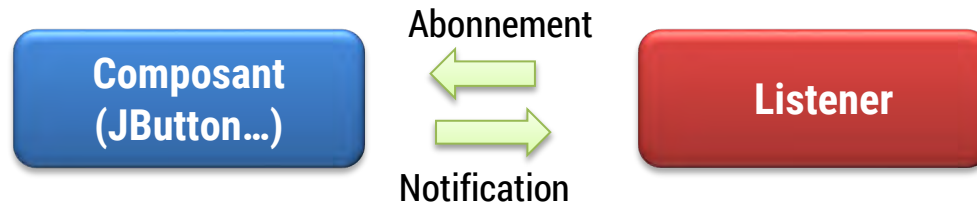
Modèle :



# Exemple simple



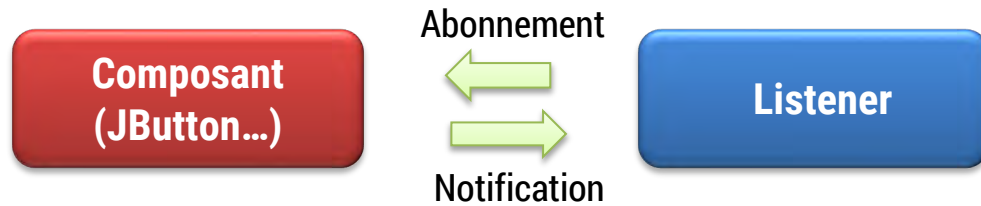
# Programmation événementielle



**Listener est  
une interface**

```
public class ClickListener implements ActionListener {  
    public static final String msg = "Button clicked!";  
  
    private JLabel target ;  
  
    public ClickListener(JLabel t){  
        target = t;  
    }  
  
    public void actionPerformed(ActionEvent e) {  
        target.setText(msg);  
    }  
}
```

# Partie composant



```
JButton jb1 = new JButton("Click me");  
  
// Register listener  
jb1.addActionListener(new ClickListener(jb1));
```

Voir code en annexe, *EventDemo.java*

# Conclusion



- *GUI : fait de vous des programmeur-se-s complet-ète-s*
- *Modèle objet (héritage, interfaces, ...) présent partout dans les GUI*

# Bibliographie



## [1] Livre de référence

- ▶ Matthew Robinson and Pavel Vorobiev, *Swing*, 2003, 2<sup>nd</sup> edition.

## [2] *The Swing Tutorial*

- ▶ <http://docs.oracle.com/javase/tutorial/uiswing/>

## [3] *Visual index of Swing components*

- ▶ <https://web.mit.edu/6.005/www/sp14/psets/ps4/java-6-tutorial/components.html>