Les interface homme-machine et le langage Java

DUT première année
Henri Garreta, Faculté des Sciences (Luminy)
Cyril Pain-Barre & Sébastien Nedjar, IUT d'Aix-Marseille (Aix)

Cours 3: Interfaces graphiques

Bibliothèque *JFC* (*Java Foundation Classes*)

- *AWT* (*Abstract Windowing Toolkit*)
  - première version, rôle important dans le succès de Java
  - composants *lourds* (appariés avec des composants natifs)
- *Swing*
  - composants *légers*, « 100% pur Java »
  - plus nombreux, complexes et indépendants de la plate-forme
- en fait, on emploie :
  - les composants de *Swing*
  - certains éléments importants (*événements*, *gestionnaires de disposition*, etc.) de *AWT*

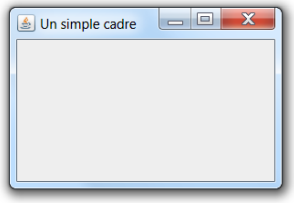
- attention aux noms des composants
  - *AWT* : `Frame`, `Button`, `Panel`
  - *Swing* : `JFrame`, `JButton`, `JPanel`
- *SWT* : développée (indépendamment de *Swing*) par *IBM*

La plus petite application avec *IUG*

```
import javax.swing.JFrame;

public class Simple {

    public static void main(String[] args) {
        JFrame cadre = new JFrame("Un simple cadre");
        cadre.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        cadre.setSize(300, 200);
        cadre.setVisible(true);
    }
}
```



- Attention, la case de fermeture ne termine pas l'application

Sous-classer

- une autre organisation de la même chose
- cela consiste à créer sa propre classe « Cadre »
- dont le constructeur met en place l'interface graphique

```
public class Simple extends JFrame {

    public Simple() {
        super("Un cadre");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        JFrame cadre = new Simple();
        cadre.setVisible(true);
    }
}
```

Le haut de la hiérarchie

`Object`
- `Component`
  - visible à l'écran — *paint(...)*
  - source d'événements — *addXXXListener(...)*
  - `Container`
    - contient d'autres composants — *add(...)*
    - a un gestionnaire de disposition — *setLayout(...)*
    - `Window`
      - composant de niveau supérieur — *setVisible(...)*
      - est rattaché à l'interface — *getOwner(...)*
      - `Frame`
        - a un bord, bandeau de titre, barre de menus
        - est souvent unique et permanent
      - `Dialog`
        - multiple et éphémère
        - peut être *modal* ou *non modal*

Programmation événementielle

Programmation « procédurale »
- le déroulement est contrôlé par une séquence écrite d'instructions
- la boucle principale « appartient » au programmeur

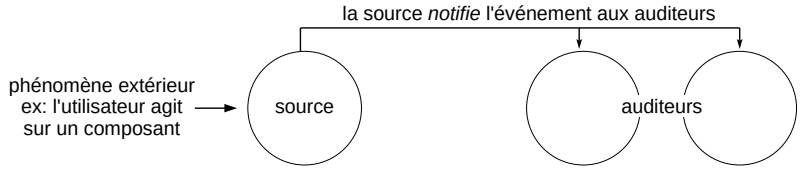
programme principal
initialisations
répéter
lire une commande
traiter une commande
jusqu'à une commande "finir"

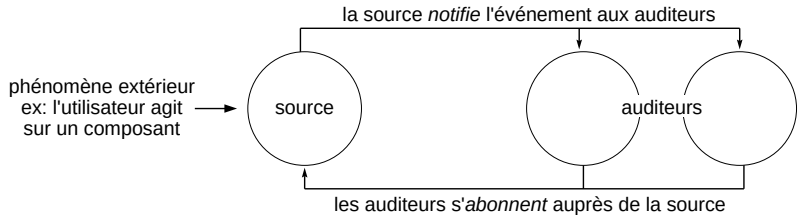
Programmation événementielle
- les événements (dont les actions de l'utilisateur) contrôlent le déroulement
- pas de boucle principale (elle est enfouie dans la bibliothèque)

fonctions de réaction aux événements

programme principal
initialisation : mise en place de l'interface graphique
(guetter des événements)

Modèle événementiel de JFC





- exemple : les événements souris sont notifiés par

```
void mousePressed(MouseEvent e)
void mouseReleased(MouseEvent e)
void mouseClicked(MouseEvent e)
void mouseEntered(MouseEvent e)
void mouseExited(MouseEvent e)
```


Principales catégories d'événements

- `MouseEvent` : actions *discrètes* sur la souris
  `mousePressed`, `mouseReleased`, `mouseClicked`
  `mouseEntered`, `mouseExited`
- `MouseMotionEvent` : actions *continues* sur la souris
  `mouseMoved`, `mouseDragged`
- `FocusEvent` : obtention et perte du clavier
  `focusGained`, `focusLost`
- `KeyEvent` : actions sur le clavier
  `keyPressed`, `keyReleased` — *événements de bas niveau*
  `keyTyped` — *événement élaboré*
- `ActionEvent` : pression d'un bouton, choix dans un menu, etc.
  `actionPerformed`
- `WindowEvent` : événements survenant sur une fenêtre
  `windowActivated`, `windowIconified`, `windowOpened`
  `windowDeactivated`, `windowDeiconified`, `windowClosed`
  `windowClosing`

La suite...

...se passe dans Eclipse :

- détection d'événements souris
- tracé d'une ligne polygonale
- gestionnaires de disposition