
TD 7 IJA

Les interfaces graphiques

JAVA

AWT & SWING

- Première bibliothèque graphique JAVA: AWT
 - Package java.awt
 - Utilisation de code natif
 - Composants limités
- Nouvelle bibliothèque: SWING
 - Package javax.swing
 - Plus riche et plus personnalisable
 - Ne remplace pas AWT mais fournit des composants plus performants

Hiérarchie des composants

Les composants graphiques sont placés dans des conteneurs

(Containers):

- Composants élémentaires

- JLabel
- JButton
- JCheckBox
- JRadioButton
- JTextField
- JTextArea

- Composants complexes

- ButtonGroup
- JComboBox
- JList
- JScrollBar
- JMenuBar
- JPopupMenu

- Les containers

- JWindow
 - JFrame
 - JDialog
 - JFileDialog
- JPanel
 - Applet
- JTabbedPane
- JScrollPane

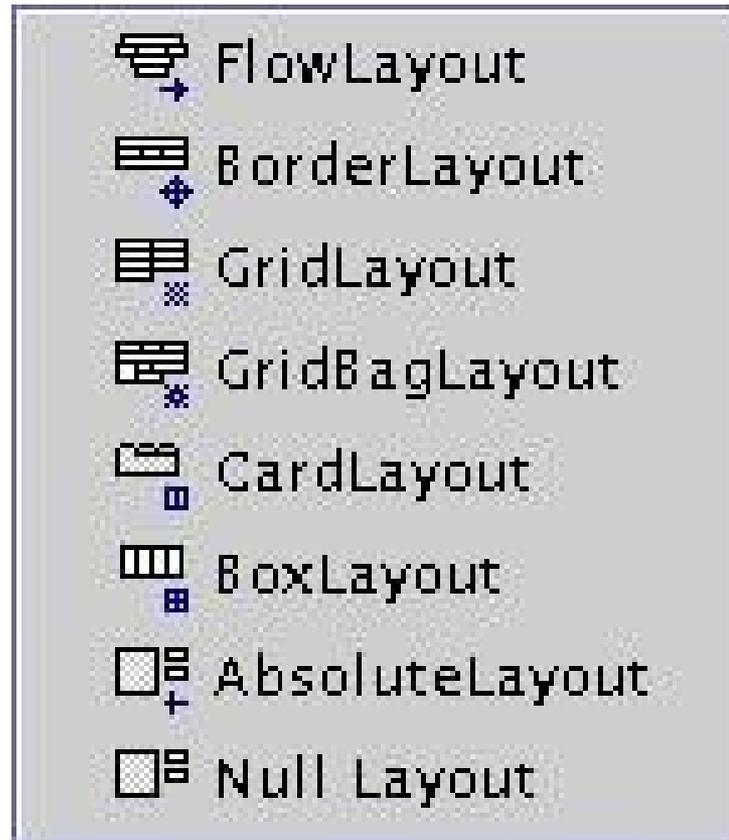
<http://java.sun.com/docs/books/tutorial/uiswing/components/components.html>

Exercice

- Réaliser une classe Appli qui affiche une fenêtre (JFrame) contenant un JButton

Disposition des composants (1/2)

Chaque conteneur utilise un gestionnaire de placement (Layout) pour la disposition des composants qu'il contient.



<http://java.sun.com/docs/books/tutorial/uiswing/layout/visual.html>

Disposition des composants (2/2)

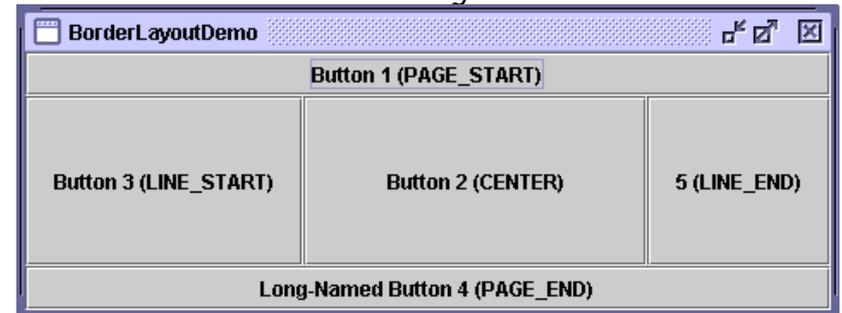
Exemples de dispositions

GridLayout



```
JFrame fenetre=new JFrame("GridLayoutDemo");
JPanel tmp = fenetre.getContentPane();
tmp.setLayout(new GridLayout(3,2));
tmp.add(new Button("Button 1"), 0,0);
tmp.add(new Button("Button 2"), 0,1);
tmp.add(new Button("Button 3"), 1,0);
tmp.add(new Button("Long-Named Button 4 "), 1,1);
tmp.add(new Button("5"), 2,0);
```

BorderLayout

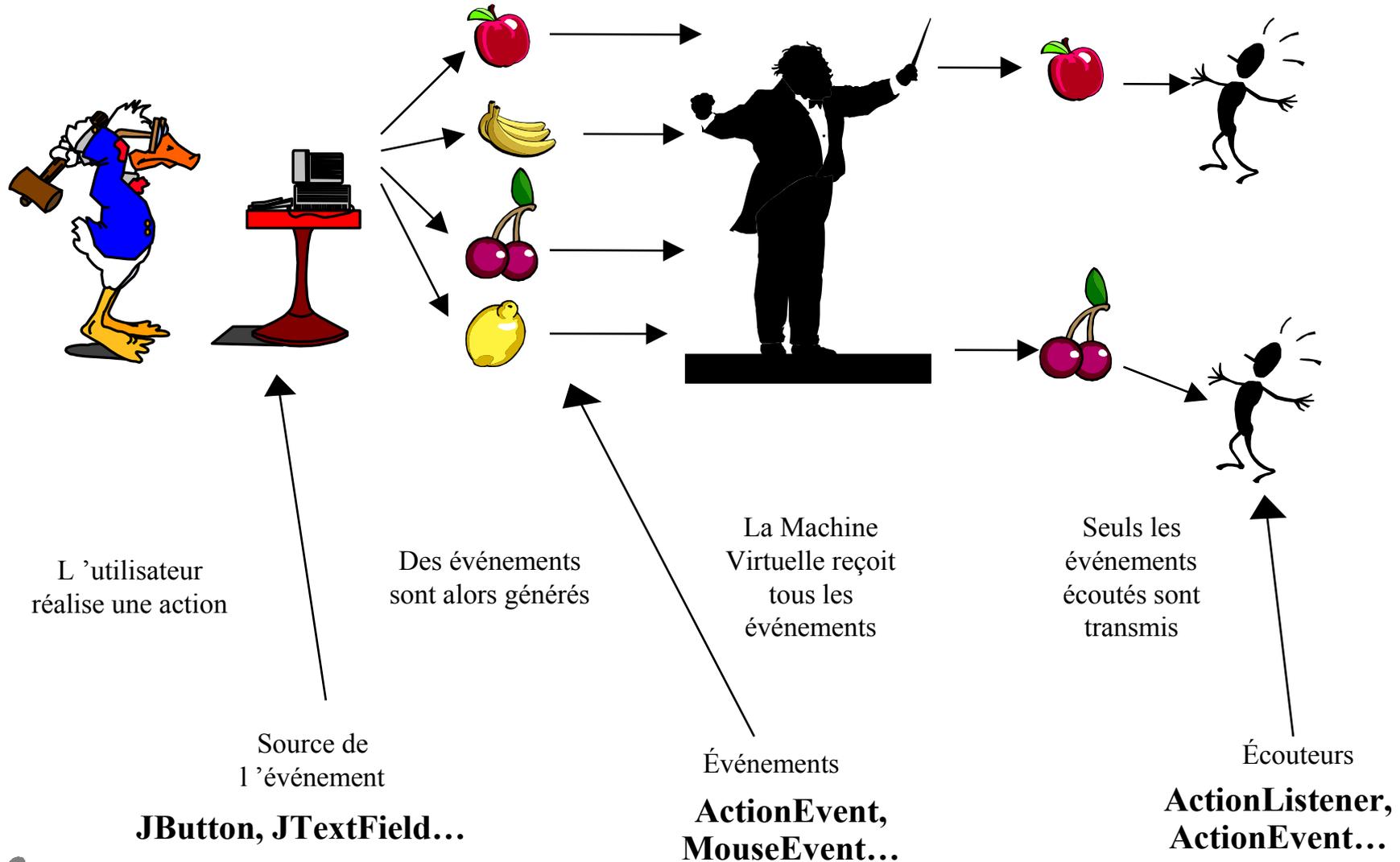


```
JFrame fenetre=new JFrame("BorderLayoutDemo");
JPanel tmp = fenetre.getContentPane();
tmp.setLayout(new BorderLayout());
tmp.add(new Button("Button 1 (PAGE_START)",
BorderLayout.NORTH));
tmp.add(new Button("Button 3 (LINE_START)",
BorderLayout.WEST));
tmp.add(new Button("Button 2 (CENTER)")
BorderLayout.CENTER);
tmp.add(new Button("5 (LINE_END)")
BorderLayout.EAST);
tmp.add(new Button("Long-Named Button 4
(PAGE_END)") BorderLayout.SOUTH);
```

Exercice

- Réaliser un pavé numérique de 10 touches en utilisant les gestionnaires de position suivants :
 - BorderLayout
 - FlowLayout
 - GridLayout
- Quelle différence remarquez-vous entre Border et FlowLayout

Propagation des événements



Les acteurs

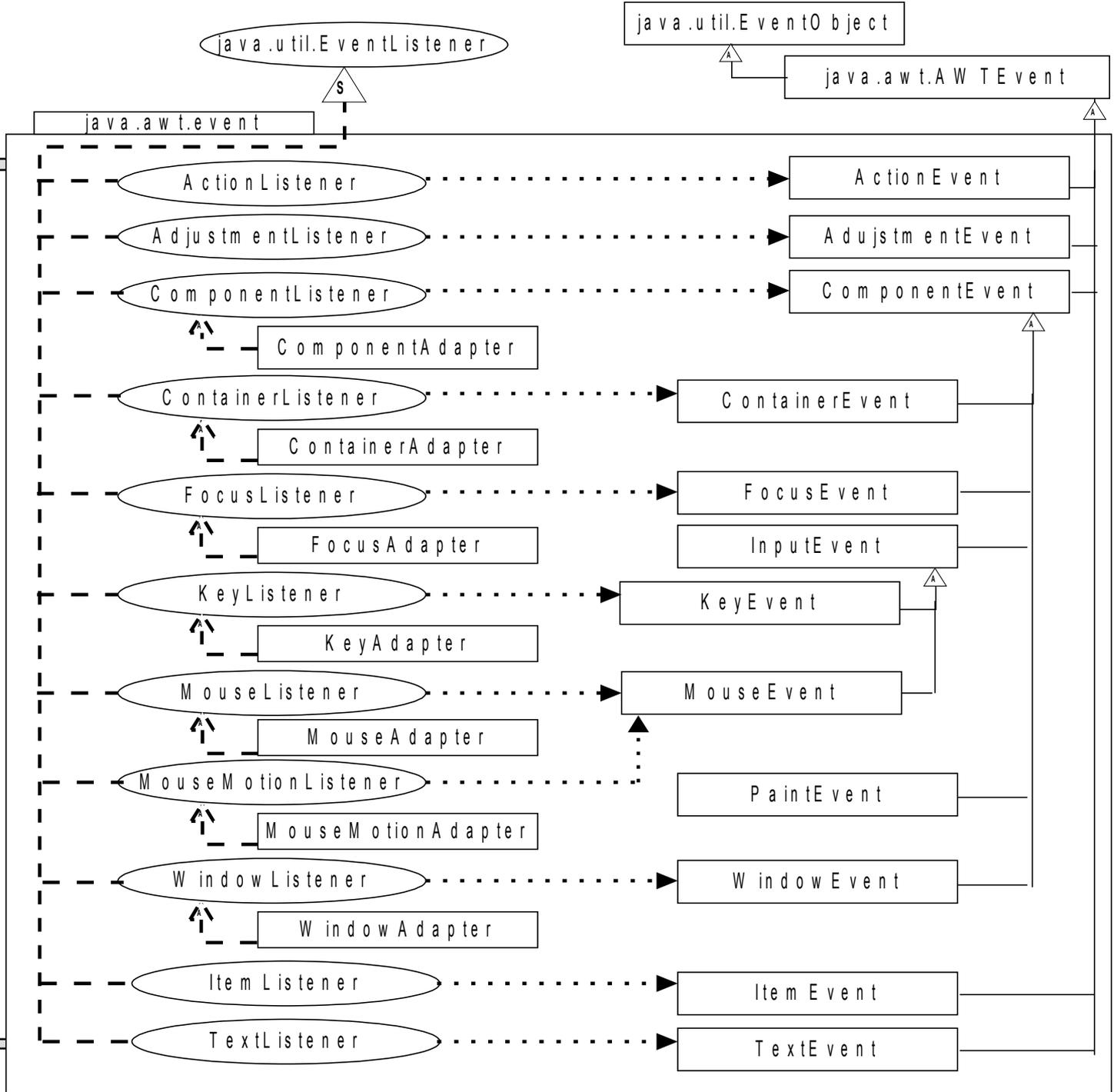
- Le Composant :
 - Indique les événements qu'il peut générer.
 - Button : MouseEvent, ActionEvent, ComponentEvent...
- L'événement :
 - Indique l'action que l'utilisateur a générée.
 - Ex : MouseEvent
- Le listener :
 - Il indique le traitement à faire sur une catégorie d'événements
 - MouseListener, ActionListener...

Exercice

- Ajouter un JTextField à la fenêtre précédente et un ActionPerformed afin qu'à chaque click sur un bouton la valeur soit affichée

Les composants et leurs événements

- Tous les composants génèrent des événements
 - Car il dérivent de la classe Component qui génère des événements
- Tous les composants ne génèrent pas tous les événements
 - Un bouton ne génère pas d'événements de type text
- Il existe pour les composants élémentaires un événement de sémantique générale appelé `ActionEvent`, qui représente l'interaction standard avec l'utilisateur
 - Click sur bouton ==> `ActionEvent`
 - `DoubleClick` sur une liste ==> `ActionEvent`
 - Click sur un élément de liste ==> `ActionEvent`
 - `<Return>` à la fin d'une saisie dans un `TextField` ==> `ActionEvent`



Un exemple

- Réaliser un listener change la couleur du bouton qui possède le focus
`java.awt.event.FocusListener`

```
public void focusGained(FocusEvent e){  
    }  
public void focusLost(FocusEvent e){  
    }
```

- Modifiez votre classe Appli afin que tous les boutons soient abonnés à une instance de votre FocusListener

```
addFocusListener(<unFocusListener>);
```

Exercice à la maison

I) L'exemple inévitable (HelloWorld)

- 1) développer une fenêtre HelloWorld qui affiche « Hello !! » dans un label
- 2) Ajouter un bouton à la fenêtre. Le label affichera « Hello (n) » où n est le nombre de clicks sur le bouton

II) Interface graphique pour la gestion des étudiants

Développer une interface graphique pour ajouter, supprimer et afficher un étudiant à la classe GesEtudHash du TD précédent.