

# Cours n°8

## Interfaces graphiques I

## Sommaire

1. Composants d'applications graphiques
  1. Affichage et gestion de structures de données
  2. Composants de choix
2. Objets graphiques
  1. Modèle de construction
  2. Objets graphiques atomiques

## INTRODUCTION

### Bibliographie

#### La programmation

V. Berthié et J.-P. Briaud , «Swing la synthèse : développement des interfaces graphiques en Java », Dunod, 2003.

#### Les API

M. Loy, R. Eckstein, D. Wood, J. Elliot et B. Cole, «Java Swing, 2nd Edition », O'Reilly, 2002.

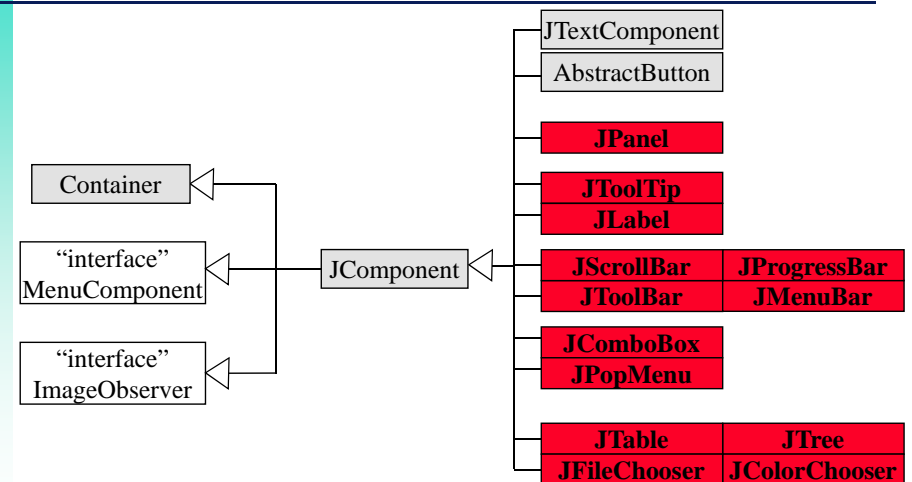
#### Sites

[java.sun.com/docs/books/tutorial/uiswing/](http://java.sun.com/docs/books/tutorial/uiswing/)

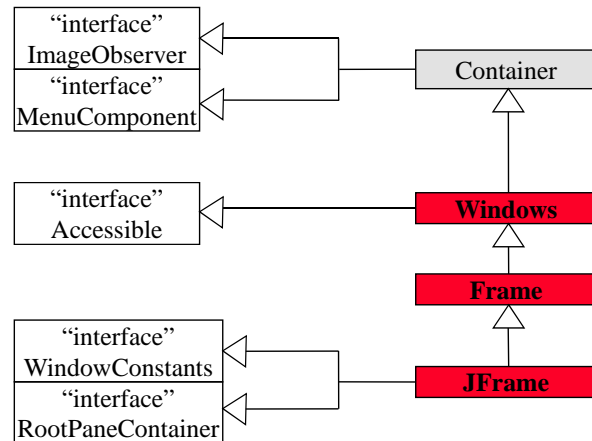
(site officiel)

## INTRODUCTION

### Classe JComponent



## Classe JFrame



## Généralités

### Applications graphiques directement intégrables

#### Affichage et gestion de structures de données

**JList** Affichage de listes

**JTable** Editeur de table de données

**JTree** Editeur d'arborescences

Méthodes graphiques de sélection, de copie et de collage de d'éléments

#### Composants de choix

**JFileChooser** Recherche graphique d'un fichier ou d'un dossier

**JColorChooser** Recherche graphique et/ou édition d'une couleur

## testJList.java

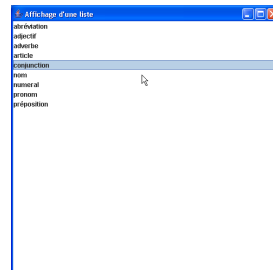
### JList – Programme

```

// valeurs des éléments de la liste
Object[] tagSyntaxe = { "abréviation", "adjectif", "adverbe",
    "article", "conjunction", "nom", "numeral", "pronom",
    "préposition" };

JList list = new JList(tagSyntaxe);

// affichage de la liste
JFrame fen = new JFrame("Affichage d'une liste"); fen.setSize(500,
    500);
Container cpfen = fen.getContentPane();
cpfen.add(list);
fen.setVisible(true);
  
```



### JList – Méthodes de sélection

#### Sélection par programme

**void setSelectedIndex(int i)** Sélection du i<sup>ème</sup> élément

**void setSelectedIndices(int[] ti)** Sélection d'un ensemble d'éléments

#### Modèle de sélection par la vue

**void setSelectionMode(int selectionMode)**

**SINGLE\_SELECTION** sélection d'un seul élément

**SINGLE\_INTERVAL\_SELECTION** sélection d'un ensemble contiguë d'éléments

**MULTIPLE\_INTERVAL\_SELECTION** pas de contraintes (valeur par défaut)

#### Lecture d'un sélection par la vue

**int getSelectedIndex()** indice du 1er élément sélectionné

**int[] getSelectedIndices()** tableau des indices des éléments sélectionnés

**Object getSelectedValue()** 1er élément sélectionné

**Object[] getSelectedValues()** éléments sélectionnés

## JTable

```
// noms des champs
String[] nomChamps = {"Nom", "Marque", "Cylindrée"};

// valeurs des champs
Object[][] garage = {{ "Twingo", "Renault", "1239"}, {"Marbella",
    "Seat", "903"}, {"Corsa", "Opel", "1686"} };

// affichage de la table
JTable tb = new JTable(garage, nomChamps);
JScrollPane tableView = new JScrollPane (tb);
JFrame fen = new JFrame("Affichage d'une table"); fen.setSize(500,
    100);
Container cpfen = fen.getContentPane();
cpfen.add(tableView);
fen.setVisible(true);
```

Nom	Marque	Cylindrée
Twingo	Renault	1239
Marbella	Seat	903
Corsa	Opel	1686

## JTree (1/2)

```
String[] ts = {"NP", "NP0", "Det", "the", "N", "boy", "VP",
    "V", "delivers"};

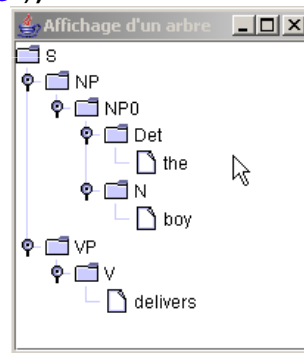
//création des nœuds
DefaultMutableTreeNode root, tn[];
root = new DefaultMutableTreeNode("S");
tn = new DefaultMutableTreeNode[ts.length];
for (int i = 0; i < ts.length; i++)
    tn[i] = new DefaultMutableTreeNode(ts[i]);

// création des liens entre noeuds
root.add(tn[0]);
tn[0].add(tn[1]); tn[1].add(tn[2]); tn[2].add(tn[3]);
tn[1].add(tn[4]); tn[4].add(tn[5]);
root.add(tn[6]);
tn[6].add(tn[7]); tn[7].add(tn[8]);
```

## JTree (2/2)

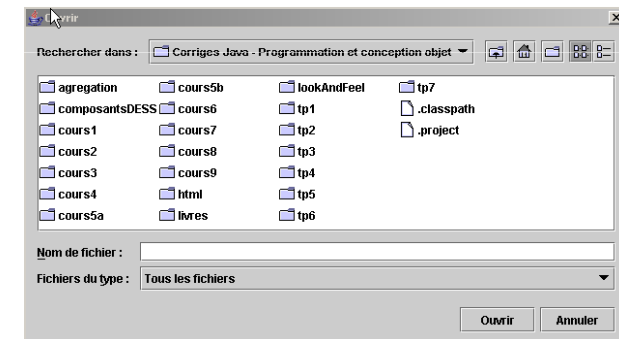
```
//affichage de l'arbre
JTree jt = new JTree(root);
JScrollPane treeView = new JScrollPane (jt);

JFrame fen = new JFrame("Affichage d'un arbre");
fen.setSize(500, 100);
Container cpfen = fen.getContentPane();
cpfen.add(treeView);
fen.setVisible(true);
```



## JFileChooser

```
JFileChooser fc = new JFileChooser(".");
JFrame fen = new JFrame();
fc.showOpenDialog(fen);
File selFile = fc.getSelectedFile();
```



## Arborescence d'objets graphiques

### Trois types d'objets graphiques

Objets conteneur-racines

JFrame, JApplet, JDialog, JWindows

Objets conteneurs non racines

JPanel,

Objets terminaux ou atomiques

Jbutton, JRadioButton, Jcheckbox, Jlabel, JtextField, Jlist, Jcombo

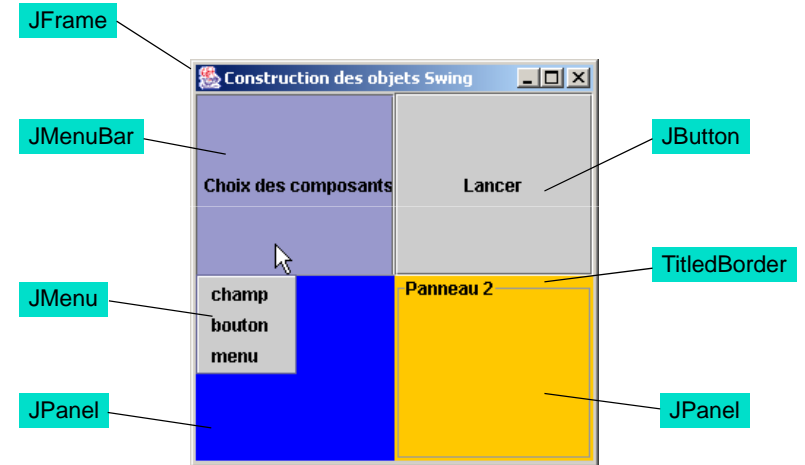
### Structure racine conseillée

un objet conteneur-racine contenant un JPanel incolore

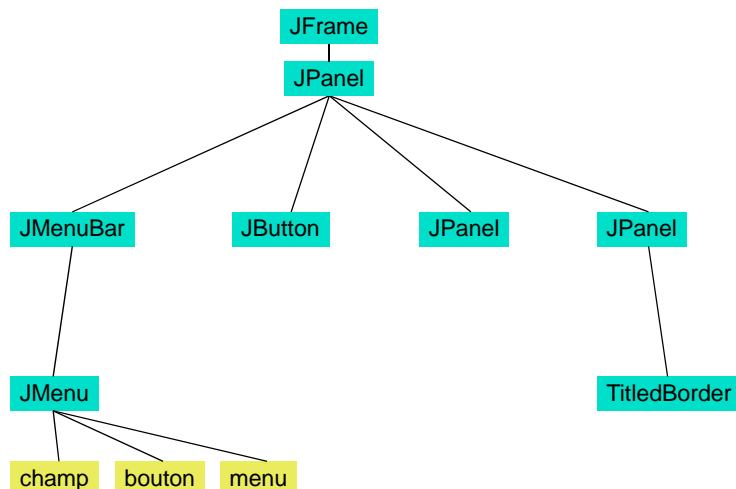
### Approche objet

Regroupement dans un JPanel des objets graphiques partageant la même sémantique

## Vue



## Représentation arborescente



## Code (1/2)

```
//Déclaration de la JFrame
JFrame fen = new JFrame("Construction des objets Swing");
fen.setSize(280, 280);

//Déclaration du Jpanel racine
JPanel p = new JPanel();
fen.getContentPane().add(p);
p.setLayout(new GridLayout(2,2));

//Déclaration du JMenuBar
JMenuBar mb = new JMenuBar();

// Déclaration du JMenu
JMenu m = new JMenu("Choix des composants");
m.add("champ"); m.add("bouton"); m.add("menu");
mb.add(m);
fen.getContentPane().add(mb);
```

## Code (2/2)

```
//Déclaration du JButton
JButton b1 = new JButton("Lancer");
p.add(b1);

//Déclaration du 1er JPanel
JPanel p1 = new JPanel(); p1.setBackground(Color.BLUE);
p1.setToolTipText("Etiquette sensitive");
p.add(p1);

//Déclaration du 2ème JPanel
JPanel p2 = new JPanel(); p2.setBackground(Color.ORANGE);
p2.setBorder(new TitledBorder("Panneau 2"));
p.add(p2);

//Affichage de la JFrame et de son contenu
fen.setVisible(true);
```

## Composant de contrôle et de saisie

**JButton** (rectangle portant un titre réagissant à un clic)

**JRadioButton** (rectangle portant un titre réagissant à un clic avec mémoire)

**JCheckbox** (boîte à cocher)

**JLabel** (affichage d'un texte dans un composant)

**TextField** (ligne de saisie texte)

**TextArea** (champ multi-ligne de saisie texte)

**PasswordField** (champ de saisie d'un mot de passe)

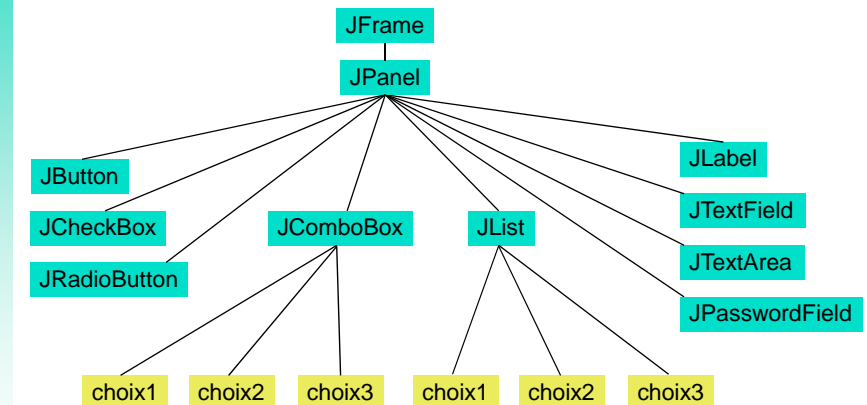
**List** (choix parmi une liste de valeurs)

**ComboBox** (choix parmi une liste de valeurs et affichage de la valeur choisie)

## Contrôle et saisie - Vue



## Contrôle et saisie – Représentation arborescente



## Contrôle et saisie - Code (1/2)

```
// Déclaration du conteneur racine
JFrame fen = new JFrame("Exemples de composants graphiques atomiques");
fen.setSize(500, 500);
JPanel p = new JPanel();
fen.getContentPane().add(p);
p.setLayout(new GridLayout(3,3));

// Déclaration de composants de choix
JButton bt; JRadioButton rbt; JCheckBox cb; JList ls;

bt = new JButton("rectangle à cliquer");
rbt = new JRadioButton("rectangle à cliquer avec mémoire");
cb = new JCheckBox("case à cocher");

String chx[] = {"choix1", "choix2", "choix3"};
ls = new JList(chx);
JComboBox cbo = new JComboBox(chx);
```

## Contrôle et saisie – Code (2/2)

```
// Déclaration de composants de texte
JLabel lb; JTextField tf; JTextArea ta; JPasswordField pf;

lb = new JLabel("affichage d'un texte dans un composant");
tf = new JTextField("ligne de saisie texte");
ta = new JTextArea("champ multi-ligne de saisie texte");
pf = new JPasswordField("champ de saisie d'un mot de passe");

// Ajouts des composants au JPanel racine
p.add(bt); p.add(rbt); p.add(cb); p.add(lb); p.add(tf); p.add(ta);
p.add(pf); p.add(ls); p.add(cbo);

fen.setVisible(true);
```

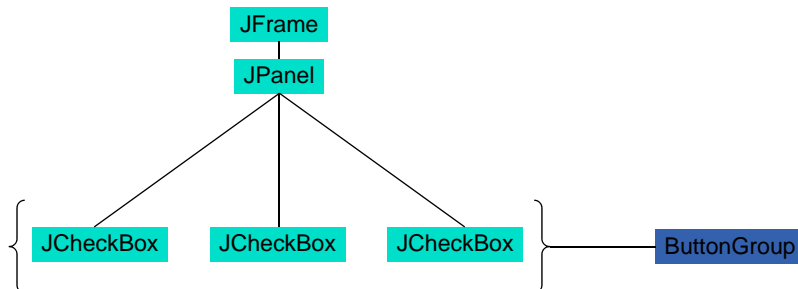
## Regroupement d'objets atomiques

## Activation d'un seul objet à la fois

Désactivation automatique des autres objets

**ButtonGroup** (groupe de JRadioButton)

**CheckboxGroup** (groupe de JCheckBox)



## Menus déroulants

**JMenuBar** (barre de menu)

**JMenu** (menu visible dans la barre de menu)

**JMenuItem** (options d'un menu)

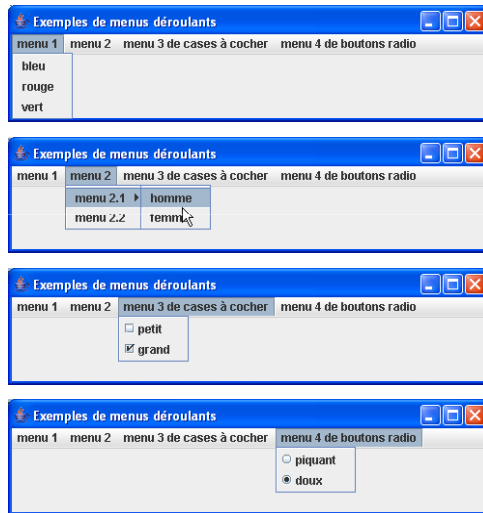
**JCheckBoxMenuItem** (options à cocher d'un menu)

**JRadioButtonMenuItem** (options boutons radio d'un menu)

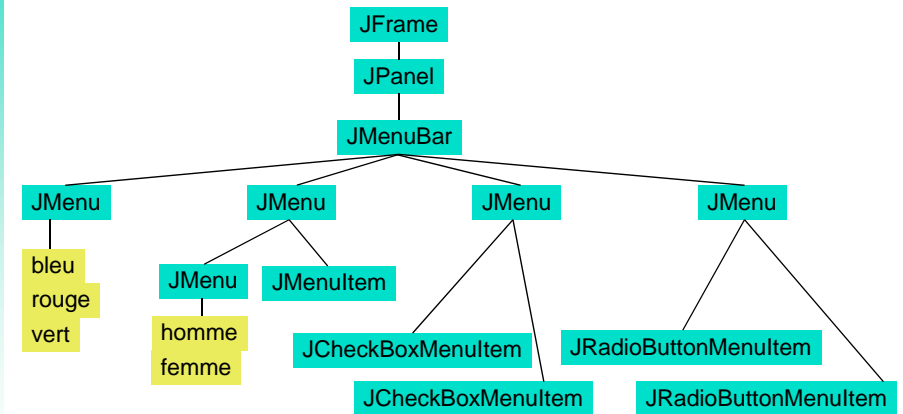
**JPopupMenu** (menu surgissant non relié à une barre de menu)

**JToolBar**

## Menus déroulants – Vue



## Menus déroulants – Représentation arborescente



## Menus déroulants – Code (1/2)

```
// Déclaration du conteneur racine
JFrame fen = new JFrame("Exemples de menus déroulants");
fen.setSize(500, 100);
JPanel p = new JPanel(); fen.getContentPane().add(p);
p.setLayout(new GridLayout(4,2));

// Déclaration de la barre de menu
JMenuBar mb = new JMenuBar ();

// Déclaration des différents menus
JMenu m1 = new JMenu("menu 1");
JMenu m2 = new JMenu("menu 2");
JMenu m3 = new JMenu("menu 3 de cases à cocher");
JMenu m4 = new JMenu("menu 4 de boutons radio");

// composants du menu 1
m1.add("bleu"); m1.add("rouge"); m1.add("vert");

// composants du menu 2
// composants du sous-menu 1 du menu 2
JMenu m21 = new JMenu("menu 2.1"); m21.add("homme"); m21.add("femme");
m2.add(m21);
JMenuItem mi = new JMenuItem("menu 2.2"); m2.add(mi);
```

## Menus déroulants – Code (2/2)

```
// composants du menu 3
JCheckBoxMenuItem cbmi1 = new JCheckBoxMenuItem("petit");
m3.add(cbmi1);
JCheckBoxMenuItem cbmi2 = new JCheckBoxMenuItem("grand");
m3.add(cbmi2);

// composants du menu 4
JRadioButtonMenuItem rbmi1, rbmi2;
rbmi1 = new JRadioButtonMenuItem("piquant"); m4.add(rbmi1); rbmi2 =
new JRadioButtonMenuItem("doux"); m4.add(rbmi2);

// ajout des menus à la barre de menu
mb.add(m1); mb.add(m2); mb.add(m3); mb.add(m4);

// ajout de la barre de menu au JPanel racine
p.add(mb);
fen.setVisible(true);

}
```