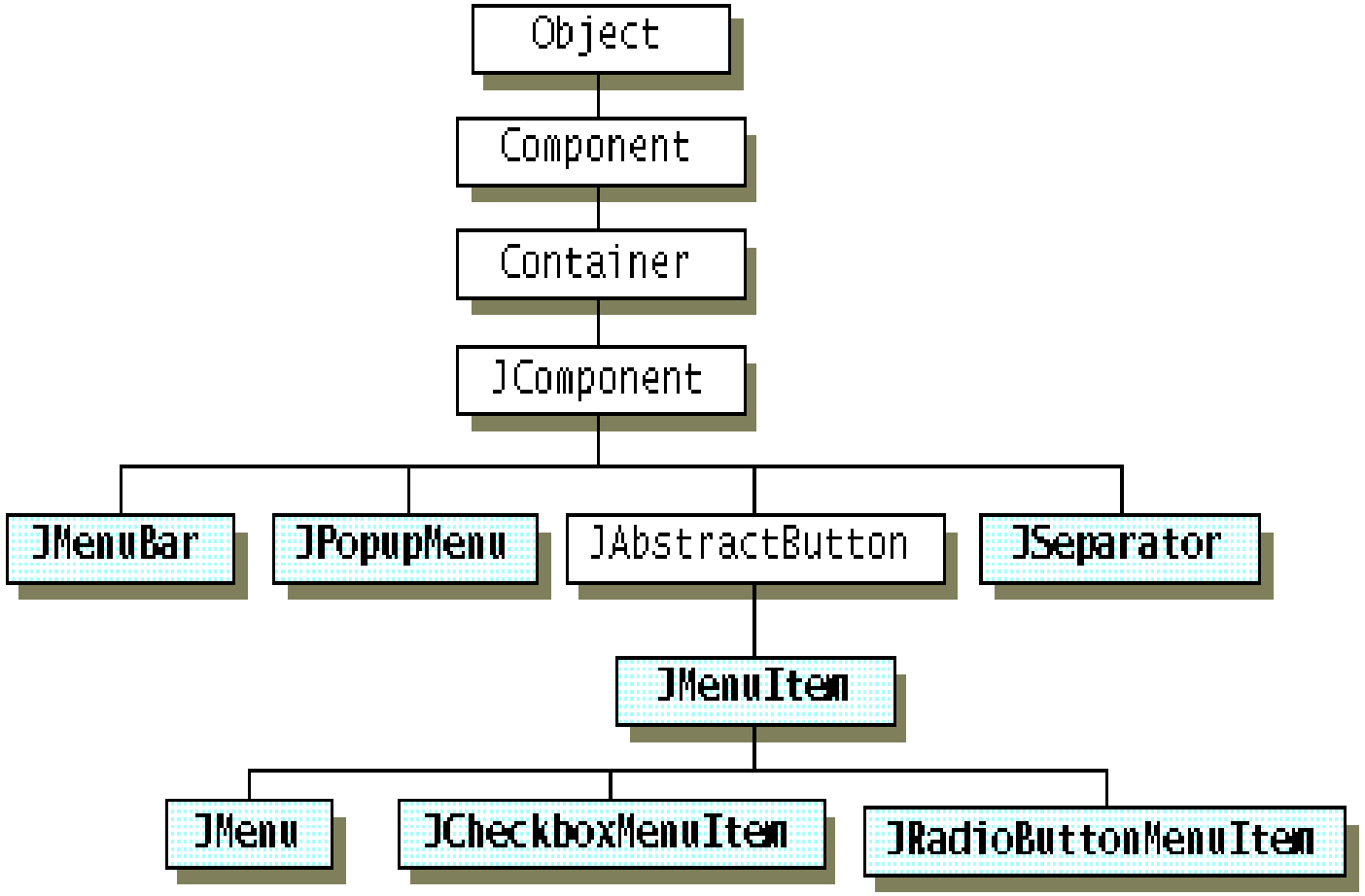
Les menus : Hiérarchie des classes



Elément de menu : Construction

La classe JMenuItem fournit une implémentation d'un élément de menu en swing.

```
JMenuItem()
//Creates a JMenuItem with no set text or icon.
JMenuItem(Icon icon)
//Creates a JMenuItem with the specified icon.
JMenuItem(String text)
//Creates a JMenuItem with the specified text.
JMenuItem(String text, Icon icon)
//Creates a JMenuItem with the specified text and icon.
JMenuItem(String text, int mnemonic)
//Creates a JMenuItem with the specified text and keyboard mnemonic.
MenuElement[] getSubElements()
//This method returns an array containing the sub-menu components for this menu component.
void  setEnabled(boolean b)
//Enables or disables the menu item.
```

Menu

La classe JMenu fournit une implémentation d'un menu : Fenêtre pop-up avec collection de JMenuItem.

```
JMenu()
//Constructs a new JMenu with no text.
JMenu(String s)
//Constructs a new JMenu with the supplied string as its text.
JMenu(String s, boolean b)
//Constructs a new JMenu with the supplied string as its text and specified as a tear-off menu or not.
JMenuItem    add(JMenuItem menuItem)
//Appends a menu item to the end of this menu.
JMenuItem    add(String s)
//Creates a new menu item with the specified text and appends it to the end of this menu
void  addSeparator()
//Appends a new separator to the end of the menu.
```

Menu/Exemple

```
1    public JMenuBar createMenuBar() {
2        JMenuBar menuBar;
3        JMenu menu, submenu;
4        JMenuItem menuItem;
5        JRadioButtonMenuItem rbMenuItem;
6        JCheckBoxMenuItem cbMenuItem;
7
8        //Create the menu bar.
9        menuBar = new JMenuBar();
10       //Build the first menu.
11       menu = new JMenu("Un menu");
12       menuBar.add(menu);
13       //a group of JMenuItems
14       menuItem = new JMenuItem("Element de menu textuel");
15       menu.add(menuItem);
16       ImageIcon icon = createImageIcon("images/middle.gif");
17       menuItem = new JMenuItem("Texte et icone", icon);
18       menu.add(menuItem);
```

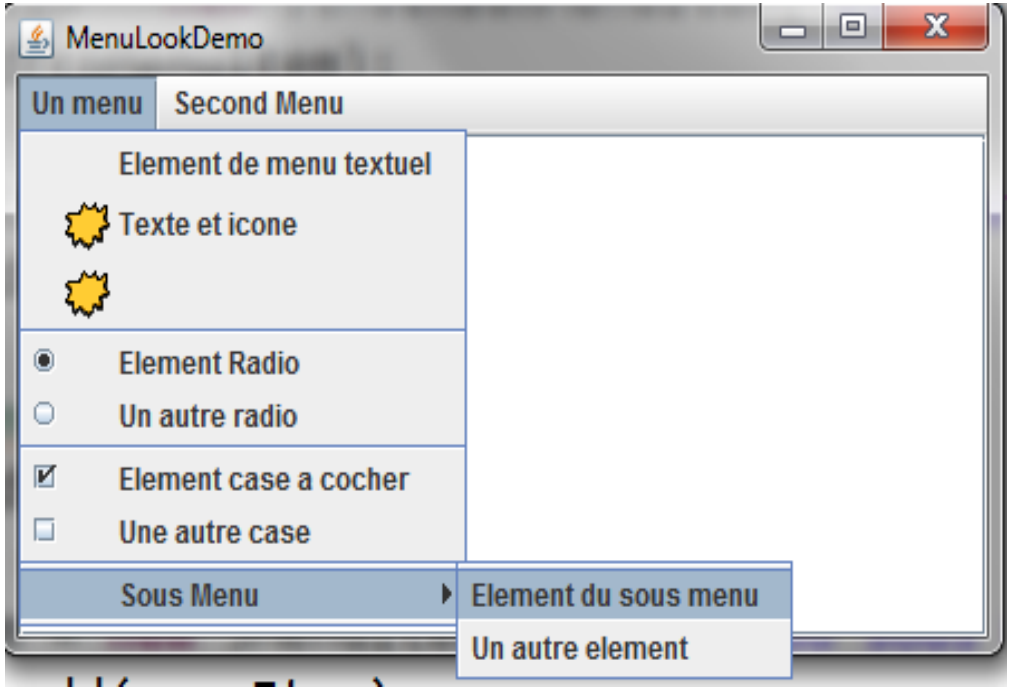
Menu/Exemple (suite)

```
1        ImageIcon icon = createImageIcon("images/middle.gif");
2        menuItem = new JMenuItem("Texte␣et␣icone", icon);
3        menu.add(menuItem);
4        menuItem = new JMenuItem(icon);
5        menu.add(menuItem);
6        //a group of radio button menu items
7        menu.addSeparator();
8        ButtonGroup group = new ButtonGroup();
9        rbMenuItem = new JRadioButtonMenuItem("Element␣Radio");
10       rbMenuItem.setSelected(true);
11       group.add(rbMenuItem);
12       menu.add(rbMenuItem);
13       menu.addSeparator();//a group of check box menu items
14       cbMenuItem = new JCheckBoxMenuItem("Element␣case␣a␣cocher");
15       menu.add(cbMenuItem);
16       menu.addSeparator();//a submenu
17       submenu = new JMenu("Sous␣Menu");
18       menuItem = new JMenuItem("Element␣du␣sous␣menu");
19       submenu.add(menuItem);
20       menuItem = new JMenuItem("Un␣autre␣element");
21       submenu.add(menuItem);
22       menu.add(submenu);
23   }
```

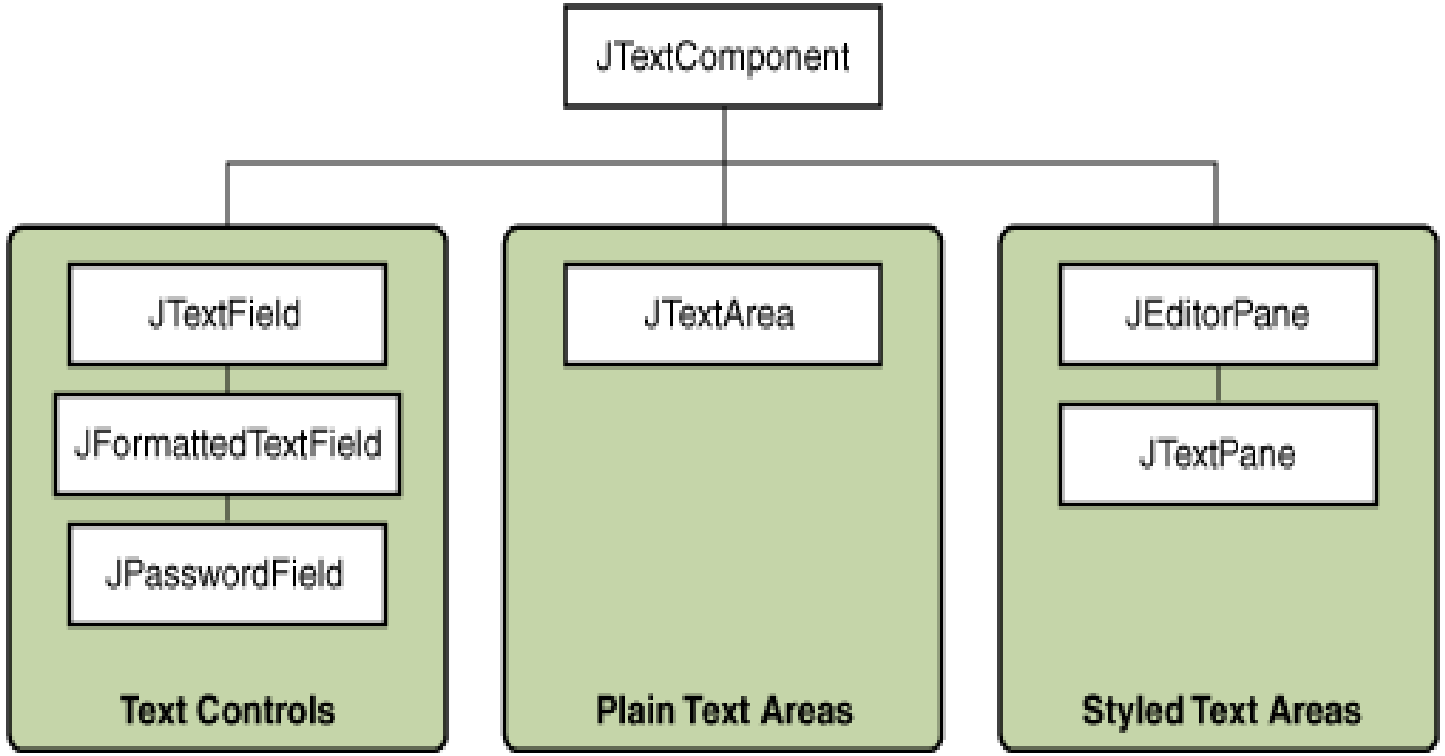
Menu/Exemple(Suite)

```
1        //Build second menu in the menu bar.
2        menu = new JMenu("Second Menu");
3        menuBar.add(menu);
4        return menuBar;
```

Menu/Exemple : Résultat



Hiérarchie des champs textuels : Résultat



Les zones de textes

- Instances de JTextField à créer avec l'un des constructeurs :

```
JTextField()
//Constructs a new TextField.
JTextField(Document doc, String text, int columns)
//Constructs a new JTextField that uses the given text storage model and the given number of columns.
JTextField(int columns)
//Constructs a new empty TextField with the specified number of columns.
JTextField(String text)
//Constructs a new TextField initialized with the specified text.
JTextField(String text, int columns)
//Constructs a new TextField initialized with the specified text and columns.
```

Les champs textuels

```
int    getColumns()
//Returns the number of columns in this TextField.
protected int getColumnWidth()
//Returns the column width.
int    getHorizontalAlignment()
//Returns the horizontal alignment of the text.
void   setColumns(int columns)
//Sets the number of columns in this TextField, and then invalidate the layout.
void   setFont(Font f)
//Sets the current font.
void   setHorizontalAlignment(int alignment)
//Sets the horizontal alignment of the text.
```

Les champs textuels

Pour l'alignement les constantes suivantes peuvent être utilisées :

```
JTextField.LEFT
JTextField.CENTER
JTextField.RIGHT
JTextField.LEADING
JTextField.TRAILING
```

Les zones de textes avec format

- Instances de JFormattedTextField à créer avce l'un des constructeurs :

```
1   JFormattedTextField()
2  //Creates a JFormattedTextField with no AbstractFormatterFactory.
3  JFormattedTextField(Format format)
4  //Creates a JFormattedTextField.
5  JFormattedTextField(JFormattedTextField.AbstractFormatter formatter)
6  //Creates a JFormattedTextField with the specified ↙
       ↳ AbstractFormatter.
7  JFormattedTextField(JFormattedTextField.AbstractFormatterFactory ↙
       ↳ factory)
8  //Creates a JFormattedTextField with the specified ↙
       ↳ AbstractFormatterFactory.
9  JFormattedTextField(JFormattedTextField.AbstractFormatterFactory ↙
       ↳ factory, Object currentValue)
10 //Creates a JFormattedTextField with the specified ↙
       ↳ AbstractFormatterFactory and initial value.
11 JFormattedTextField(Object value)
12 //Creates a JFormattedTextField with the specified value.
```

Les champs textuels formattés

```
1  JFormattedTextField.AbstractFormatter getFormatter()
2  //Returns the AbstractFormatter that is used to format and parse the ↙
       ↳ current value.
3  JFormattedTextField.AbstractFormatterFactory getFormatterFactory()
4  //Returns the current AbstractFormatterFactory.
5  Object getValue()
6  //Returns the last valid value.
7  protected void invalidEdit()
8  //Invoked when the user inputs an invalid value.
9  boolean isEditValid()
10 //Returns true if the current value being edited is valid.
11 protected void setFormatter(JFormattedTextField.AbstractFormatter format)
12 //Sets the current AbstractFormatter.
13 void  setFormatterFactory(JFormattedTextField.AbstractFormatterFactory ↙
       ↳ tf)
14 //Sets the AbstractFormatterFactory.
15 void  setValue(Object value)
16 //Sets the value that will be formatted by an AbstractFormatter ↙
       ↳ obtained from the current AbstractFormatterFactory.
```

Exemples

```
1   private double amount = 100000;
2   private double rate = 7.5; //7.5%
3   private int numPeriods = 30;
4   private JLabel amountLabel;
5   private JLabel rateLabel;
6   private JLabel numPeriodsLabel;
7   private JLabel paymentLabel;
8   private static String amountString = "Montant␣du␣pret:␣";
9   private static String rateString = "Taux␣Annuel␣␣(%):␣";
10  private static String numPeriodsString = "Anees:␣";
11  private static String paymentString = "Paiement␣mensuel:␣";
12  private JFormattedTextField amountField;
13  private JFormattedTextField rateField;
14  private JFormattedTextField numPeriodsField;
15  private JFormattedTextField paymentField;
16
17  private NumberFormat amountFormat;
18  private NumberFormat percentFormat;
19  private NumberFormat paymentFormat;
```

Exemples

```
1
2  public FormattedTextFieldDemo() {
3          setUpFormats();
4          double payment = computePayment(amount,rate,numPeriods);
5          amountLabel = new JLabel(amountString);
6          rateLabel = new JLabel(rateString);
7          numPeriodsLabel = new JLabel(numPeriodsString);
8          paymentLabel = new JLabel(paymentString);
9          amountField = new JFormattedTextField(amountFormat);
10         amountField.setValue(new Double(amount));
```

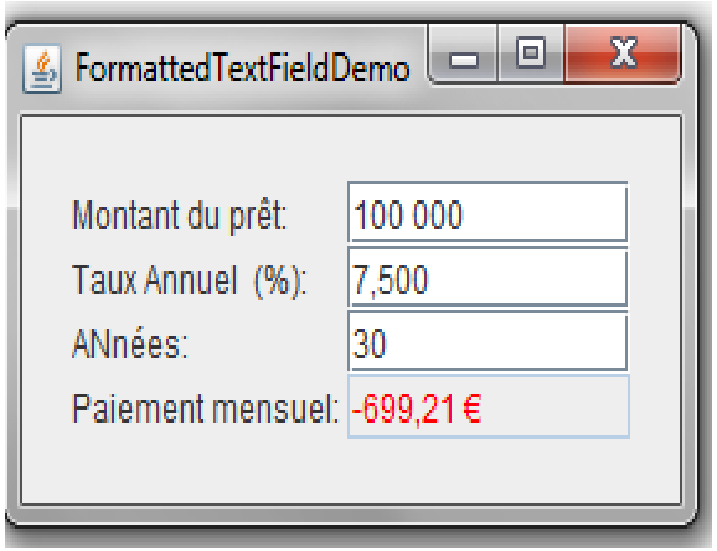
Exemples

```
1   amountField.setColumns(10);
2         amountField.addPropertyChangeListener("value", this);
3         rateField = new JFormattedTextField(percentFormat);
4         rateField.setValue(new Double(rate));
5         rateField.setColumns(10);
6         rateField.addPropertyChangeListener("value", this);
7         numPeriodsField = new JFormattedTextField();
8         numPeriodsField.setValue(new Integer(numPeriods));
9         numPeriodsField.setColumns(10);
10        numPeriodsField.addPropertyChangeListener("value", this);
11        paymentField = new JFormattedTextField(paymentFormat);
12        paymentField.setValue(new Double(payment));
13        paymentField.setColumns(10);
14        paymentField.setEditable(false);
15        paymentField.setForeground(Color.red);
16        amountLabel.setLabelFor(amountField);
17        rateLabel.setLabelFor(rateField);
18        numPeriodsLabel.setLabelFor(numPeriodsField);
19        paymentLabel.setLabelFor(paymentField);
```

Exemples

```
1          JPanel labelPane = new JPanel(new GridLayout(0,1));
2          labelPane.add(amountLabel);
3          labelPane.add(rateLabel);
4          labelPane.add(numPeriodsLabel);
5          labelPane.add(paymentLabel);
6          JPanel fieldPane = new JPanel(new GridLayout(0,1));
7          fieldPane.add(amountField);
8          fieldPane.add(rateField);
9          fieldPane.add(numPeriodsField);
10         fieldPane.add(paymentField);
11         setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
12         add(labelPane, BorderLayout.CENTER);
13         add(fieldPane, BorderLayout.LINE_END);
14     }
15     private void setUpFormats() {
16         amountFormat = NumberFormat.getNumberInstance();
17         percentFormat = NumberFormat.getNumberInstance();
18         percentFormat.setMinimumFractionDigits(3);
19         paymentFormat = NumberFormat.getCurrencyInstance();
20     }
21 }
```

Les zones de texte avec formattage : Résultat



Icônes

- La classe ImageIcon implémente l'interface Icon.

```
1  //Constructeurs
2  ImageIcon(String nomfichier)
3  ImageIcon(Image image)
4  ImageIcon(URL url)
5
6  //Methodes
7  int getIconHeight()
8  int getIconWidth()
9  Image getImage()
```

Les bordures

- Peut être associé à tout instance de JComponent
- Figurent dans javax.swing.border

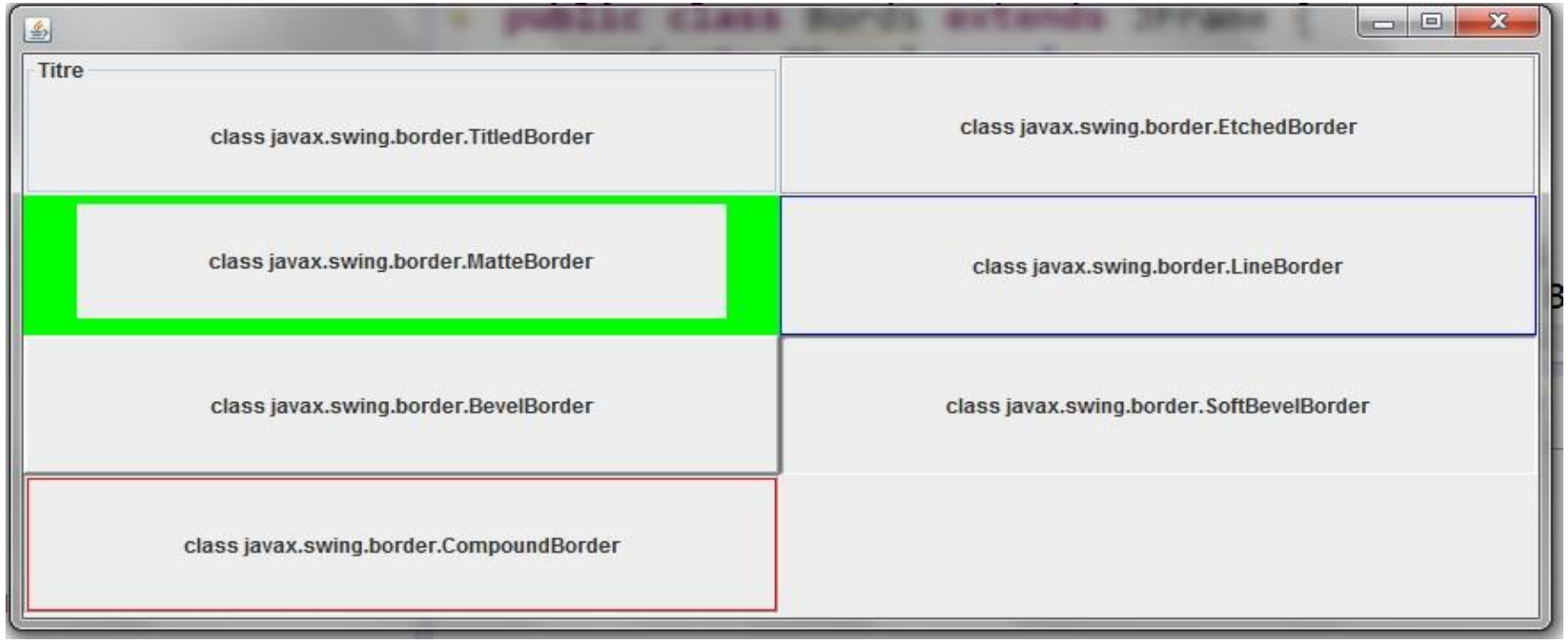
Les bordures (Exemple)

```
1  import java.awt.*;
2  import javax.swing.* ;
3  import javax.swing.border.*;
4  public class Bords extends JFrame {
5        private JPanel panel;
6        static JPanel panelBord(Border b) {
7           JPanel p = new JPanel();
8           p.setLayout(new BorderLayout());
9           String nom = b.getClass().toString();
10          p.add(new JLabel(nom,JLabel.CENTER), BorderLayout.CENTER);
11          p.setBorder(b);
12          return p;
13       }
14       public Bords(){
15          panel = (JPanel) getContentPane();
16          setLayout(new GridLayout(4,2));
17          add(panelBord(new TitledBorder("Titre")));
18          add(panelBord(new EtchedBorder()));
19          add(panelBord(new MatteBorder(5,30,10,30,Color.green)));
20          add(panelBord(new LineBorder(Color.blue)));
21          add(panelBord(new BevelBorder(BevelBorder.RAISED)));
22          add(panelBord(new
```

Les bordures (Exemple)

```
1   SoftBevelBorder(BevelBorder.LOWERED)));
2           add(panelBord(new CompoundBorder(new EtchedBorder(),
3           new LineBorder(Color.red))));
4           this.setVisible(true);
5       }
6       public static void main(String[] args) {
7           Bords b= new Bords();
8       }
9   }
```

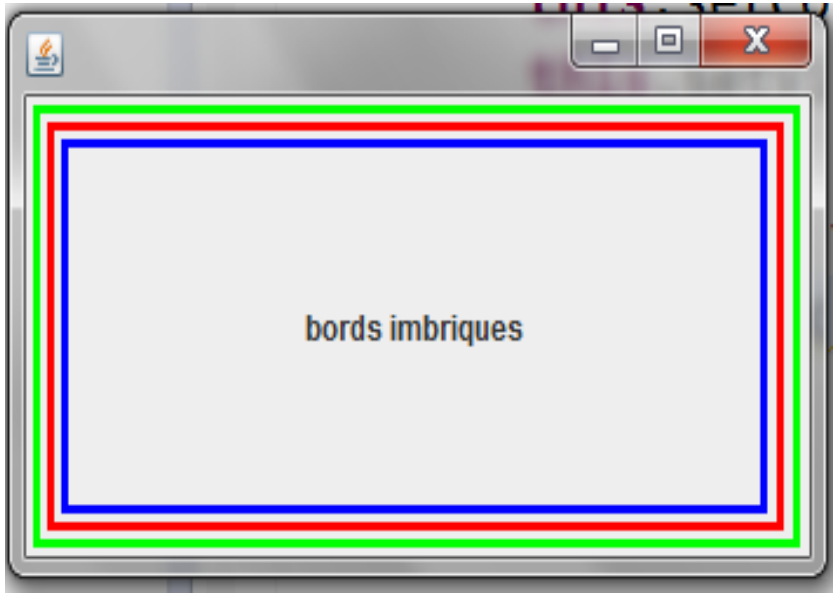
Les bordures (Exemple)



Un bord composé

```
1   JPanel p = new JPanel();
2   p.setLayout(new BorderLayout());
3   p.add(new JLabel("bords␣imbriques",JLabel.CENTER), BorderLayout.CENTER);
4   Border emptyBorder = new EmptyBorder(3,3,3,4);
5   Border b = new CompoundBorder(emptyBorder, new LineBorder(Color.blue,3) );
6   b = new CompoundBorder(new LineBorder(Color.red,3),b);
7   b = new CompoundBorder(emptyBorder, b);
8   b = new CompoundBorder(new LineBorder(Color.green,3), b);
9   b = new CompoundBorder(emptyBorder, b);
10  p.setBorder(b);
```

Les bordures (Exemple)



Curseur

- Un curseur (slider) permet d'entrer une valeur entre deux bornes selon une échelle linéaire.
- Construction

```
1        JSlider cur = new JSlider(min, max, valeurInitiale);
```

- Si les valeurs extrêmes ne sont pas données, elles prennent les valeurs par défaut 0, 100, et 50 pour la valeur initiale.
- Un curseur vertical est construit avec

```
1        JSlider vcur = new JSlider(JSlider.VERTICAL,min, max, ↙
             ↳ valeurInitiale);
```

Curseur

- Marques d'espacement :

```
1    setMajorTickSpacing(int);
2    setMinorTickSpacing(int);
```

- Affichage de ces marques :

```
1    setPaintTicks(boolean);
```

- Arrêt sur graduation :

```
1    setSnapToTicks(boolean);
```

- Libellés des graduations :

```
1    setPaintLabels(boolean);
```

- Inversion de direction :

```
1    setInverted(boolean);
```

Les arbres

- Instance de la classe JTree qui fournit une représentation hiérarchique de données
- Dépend des classes suivantes :
  - ✓ TreeModel : contient les données représenté dans l'arbre
  - ✓ TreeNode : implémentation des noeuds et de la structure d'arbre
  - ✓ TreeSelectionModel : Contient le ou les noeuds sélectionnés
  - ✓ TreePath : Contient le chemin de la racine de l'arbre.
  - ✓ TreeCellRenderer : Appelé pour dessiner un noeud
  - ✓ TreeCellEditor : Editeur pour un noeud (éditable)
  - ✓ TreeUI : look-and-feel

Les arbres

- Une instanec de TreeModel sert de base pour créer Un arbre
- Divers modèles de sélection sont disponibles :
  - ✓ sélection d'un seul élément
  - ✓ sélection de plusieurs éléments contigus
  - ✓ sélection de plusieurs éléments non contigus
- Pour personnaliser l'affichage des noeuds de l'arbre on passe par un objet TreeCellRenderer
- Pour personnaliser l'édition des noeuds de l'arbre on passe par un objet TreeCellEditor

Les arbres : Méthodes de TreeModel

```
1  //Retourne l'enfant d'ordre index
2  public Object getChild(Object parent, int index);
3
4  // Retourne le noeud parent
5  public Object getRoot();
6
7  //Pour verifier q'un objet est une feuille
8  public boolean isLeaf(Object node);
```

Les arbres/Construction

- La classe JTree fournit une vue du modèle.

```
1    JTree()
2    JTree(TreeNode racine)
3    JTree(TreeNode racine, boolean allowsChildren)
4    JTree(TreeModel modele)
5    JTree(TreeModel modele, boolean allowsChildren)
```

- La classe DefaultMutableTreeNode fournit le modèle d'un noeud.

```
1  DefaultMutableTreeNode()
2  DefaultMutableTreeNode(Object userObject)
3  DefaultMutableTreeNode(Object userObject, boolean allowsChildren)
```

- Le contenu d'un userObject est affiché via appel de la méthode toString().

Les arbres : Exemple

```
1
2  class Arbre extends JPanel {
3  JTree tree;
4  public Arbre() {
5  DefaultMutableTreeNode top, noeud, fils, n;
6  top = new DefaultMutableTreeNode("Top");
7  tree = new JTree(top);
8  noeud = new DefaultMutableTreeNode("Repertoire␣1");
9  top.add(noeud);
10 n = new DefaultMutableTreeNode("1a"); noeud.add(n);
11 n = new DefaultMutableTreeNode("1b"); noeud.add(n);
12
13 noeud = new DefaultMutableTreeNode("Repertoire␣2");
14 top.add(noeud);
15 n = new DefaultMutableTreeNode("2a"); noeud.add(n);
16
17 fils = new DefaultMutableTreeNode("2d"); noeud.add(fils);
18 n = new DefaultMutableTreeNode("3a"); fils.add(n);
19 }
```

Les arbres/Rendu

- Le rendu est réalisé par défaut par une instance de DefaultTreeCellRenderer

```
1  ...
2  JTree tree = new JTree(top);
3  DefaultTreeCellRenderer rd = (DefaultTreeCellRenderer) ↙
       ↳ tree.getCellRenderer();
```

- Pour modifier le rendu visuel on peut utiliser le snippet de code ci-dessous :

```
1   void setBackground(Color color)
2   void setBackgroundNonSelectionColor(Color newColor)
3   void setBackgroundSelectionColor(Color newColor)
4   void setBorderSelectionColor(Color newColor)
5   void setClosedIcon(Icon newIcon)
6   void setFont(Font font)
7   void setLeafIcon(Icon newIcon)
8   void setOpenIcon(Icon newIcon)
9   void setTextNonSelectionColor(Color newColor)
10  void setTextSelectionColor(Color newColor)
```

Les arbres/Sélection

- Un objet TreeSelectionListener informe sur les changements dans les sélections

```
1  class Selecteur implements TreeSelectionListener {
2      public void valueChanged( TreeSelectionEvent e ) {
3          message.setText( "Nouveau␣:␣" + e.getNewLeadSelectionPath() ↙
               ↳ );
4      }
5  }
6  tree.addTreeSelectionListener(new Selecteur());
```

Les arbres/Parcours

- Un arbre peut être parcouru de différentes manières

```
1 breadthFirstEnumeration
2 depthFirstEnumeration
3 postorderEnumeration
4 preorderEnumeration
```

- Exemple de parcours en largeur :

```
1 DefaultMutableTreeNode n, top;
2 Enumeration e;
3 top = (DefaultMutableTreeNode)tree.getModel().getRoot();
4 e = top.breadthFirstEnumeration();
5 while (e.hasMoreElements()) {
6   n = (DefaultMutableTreeNode) e.nextElement();
7   System.out.println(n.getUserObject()+" ");
8 }
```

JTree : Exemple

Lines 1–21 / 20

```
1 import javax.swing.* ;
2 import javax.swing.event.* ;
3 import javax.swing.tree.* ;
4
5 import java.awt.BorderLayout;
6 import java.io.File;
7
8 public class FileTreeDemo {
9  public static void main(String[] args) {
10  // Figure out where in the filesystem to start displaying
11  File root;
12  if (args.length > 0) root = new File(args[0]);
13  else root = new File(System.getProperty("user.home"));
14
15  // Create a TreeModel object to represent our tree of
16  files  FileTreeModel model = new FileTreeModel(root);
17
18 // Create a JTree and tell it to display our
19 model  JTree tree = new JTree();
20 tree.setModel(model);
```

JTree : Exemple

Lines 20–40 / 20

```
1 tree.setModel(model);
2 // The JTree can get big, so allow it to scroll
3 JScrollPane scrollpane = new JScrollPane(tree);
4 // Display it all in a window and make the window appear
5 JFrame frame = new JFrame("FileTreeDemo");
6 frame.getContentPane().add(scrollpane,
7 BorderLayout.CENTER);  frame.setSize(400,600);
8 frame.setVisible(true);
9
10
11  }
12 }
13
14
15 class FileTreeModel implements TreeModel {
16  protected File root;
17  public FileTreeModel(File root) { this.root = root; }
18
19  public Object getRoot() { return root; }
20
21  public boolean isLeaf(Object node) { return ((File)node).isFile(); }
```

JTree : Exemple

Lines 39–59 / 20

```
1
2  public boolean isLeaf(Object node) { return ((File)node).isFile(); }
3
4  public int getChildCount(Object parent) {
5   String[] children = ((File)parent).list();
6   if (children == null) return 0;
7   return children.length;
8  }
9  public Object getChild(Object parent, int index) {
10  String[] children = ((File)parent).list();
11  if ((children == null) || (index >= children.length)) return null;
12  return new File((File) parent, children[index]);
13 }
14
15 public int getIndexOfChild(Object parent, Object child) {
16  String[] children = ((File)parent).list();
17  if (children == null) return -1;
18  String childname = ((File)child).getName();
19  for(int i = 0; i < children.length; i++) {
20   if (childname.equals(children[i])) return i;
21  }
```

JTree : Exemple

Lines 58–78 / 20

```
1  if (childname.equals(children[i])) return i;
2  }
3  return -1;
4  }
5  public void valueForPathChanged(TreePath path, Object newvalue) {}
6  public void addTreeModelListener(TreeModelListener l) {}
7  public void removeTreeModelListener(TreeModelListener l) {}
8 }
```

JTree (Exemple)