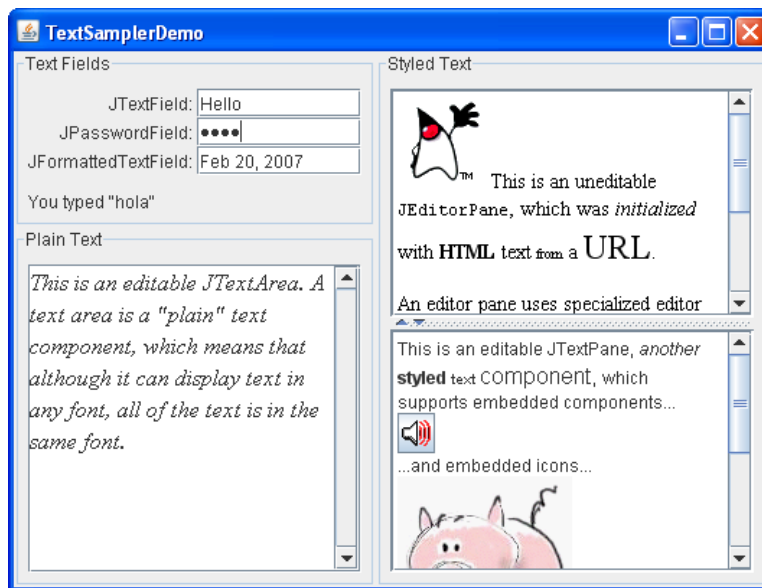


# IFT1025 été 2010

Miklós Csűrös

2 juillet 2010

## 11 Affichage de texte

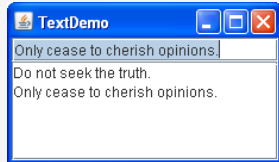


### 11.1 JLabel



Le but d'un JLabel est l'affichage d'un texte court et/ou une petite image, avec aucune interaction usager. Instanciation : **JLabel**(String txt). Si on veut une étiquette opaque, alors appeler **setOpaque(true)**.

## 11.2 JTextField et JTextArea



Un objet `JTextField` (en haut du panneau à côté) affiche une ligne de texte. Pour afficher plusieurs lignes, on utilise `JTextArea` (en bas du panneau à côté). Par défaut, l'utilisateur peut éditer le texte affiché. Pour le rendre non-éditable, il faut appeler `setEditable(false)`. On peut examiner le contenu du champ par la méthode `String getText()`. Dans le cas d'un `JTextField`, on peut y associer un `ActionListener` qui capture les `ActionEvents` lancés quand l'utilisateur change le texte. Dans le cas d'un `JTextArea`, la méthode `append(String txt)` ajoute `txt` à la fin du texte affiché.

## 11.3 JEditorPane

La classe `JEditorPane` permet l'affichage de texte dans plusieurs formats, y compris HTML et RTF (*Rich Text Format*).

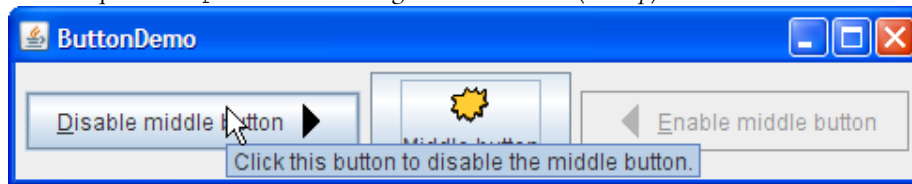
- \* constructeur `JEditorPane(String type, String contenu)`. `type` peut être "text/plain", "text/html" ou "text/rtf".
- \* constructeur `JEditorPane(String url)` throws `IOException` permet l'instanciation avec un page Web.

La même classe implémente aussi l'édition des documents affichés mais on n'en parle pas ici.

```
String t = "<h1>Titre</h1><p>Ceci est un document en HTML</p>";
JEditorPane ep = new JEditorPane("text/html", t);
ep.setEditable(false); // pas d'édition
// scrolling est nécessaire quand on affiche du texte long
JScrollPane sp = new JScrollPane(ep);
this.add(sp);
...
```

## 11.4 Infobulles

Chaque `JComponent` en Swing a une infobulle (*tool tip*) associée.



- \* `setToolTipText(String t)` pour l'affectation (null pour désactivation)
- \* `String getToolTipText()` retourne la valeur sélectionnée par `setToolTipText`. Dans des sous-classes, on peut redéfinir la méthode.
- \* `String getToolTipText(MouseEvent e)` retourne `getToolTipText()` par défaut. Dans des sous-classes, on peut redéfinir la méthode.