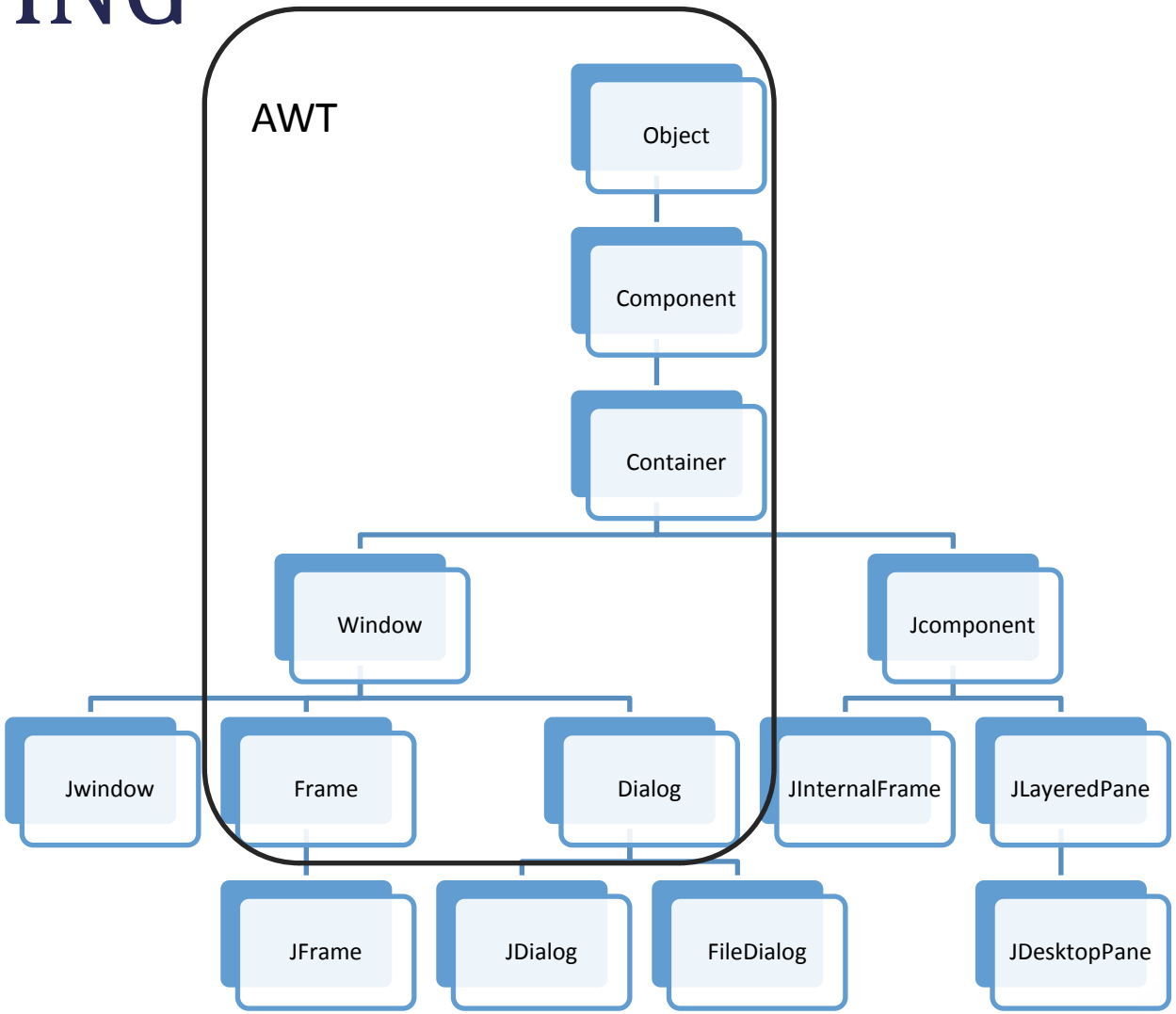
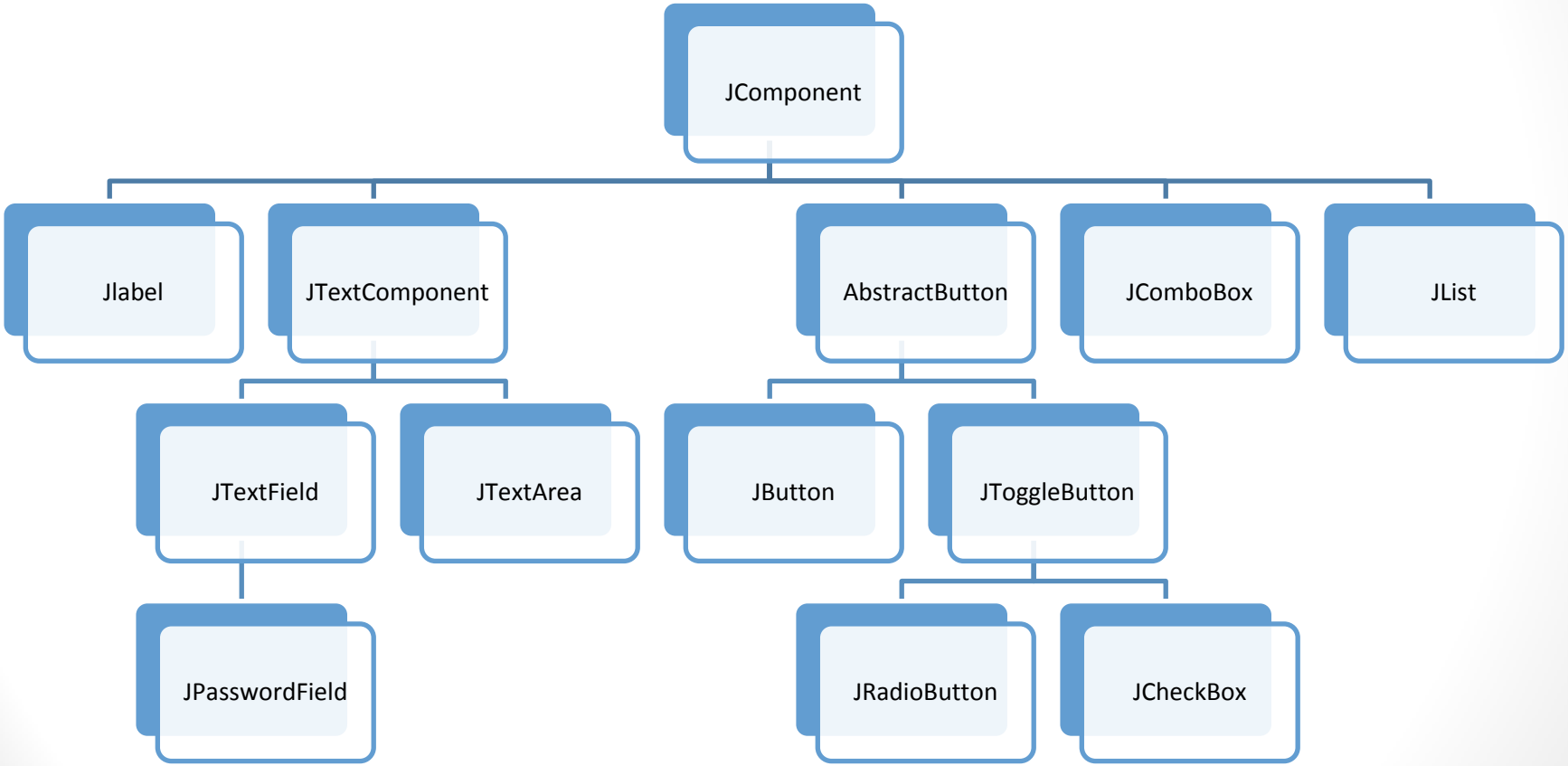
COURS – TD 3: COMPOSANTS SIMPLES

Rappel: hiérarchie de classes SWING

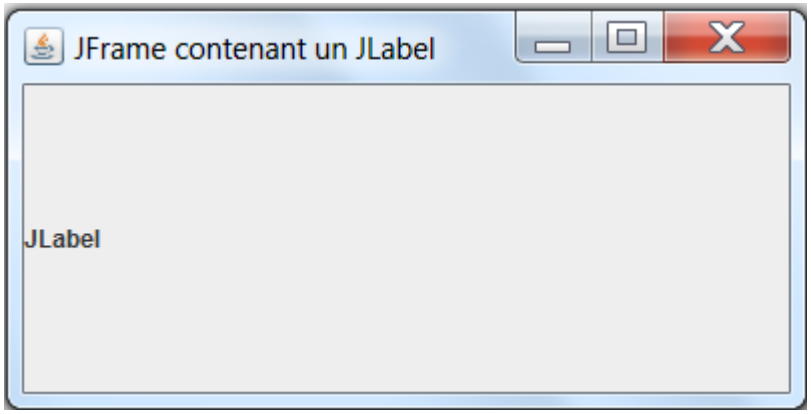


JComponent


JComponent
Jlabel
JTextComponent
AbstractButton
JComboBox
JList
JTextField
JTextArea
JButton
JToggleButton
JPasswordField
JRadioButton
JCheckBox

JLabel

- Javax.swing.JLabel
- Texte statique et/ou image
- Description d'information




```
JLabel myLabel = new JLabel("JLabel");
```

Caractéristiques communes à tous les composants

- Apparence (Bordure, Couleur, Font)
- Taille, position
- Disposition (layout)
- Etat (nom, visible/invisible, actif/inactif,...)
- Gestion d'évènements
- Dessin (surcharge de la méthode paint, rafraîchissement)
- Hiérarchie de contenants

Couleur

- Arrière plan

Color getBackground()

void setBackground(Color)

- Couleur d'écriture, de dessin

Color getForeground()

void setForeground(Color)

- Transparence

void setOpaque(boolean)

boolean isOpaque()

Color

- java.awt.Color

- public Color (int r, int g, int b)

Trois composantes rouge, vert, bleu (entiers de 0-255)

- Constantes de classes prédéfinies

black, white, red, blue, yellow, orange, grey, lightgrey,...

Font - Police

- Par défaut, police du composant contenant
- void setFont(Font)
- Font getFont()

- Style, taille du texte, effets (gras, italique, souligné)

- Font [] allFonts = GraphicsEnvironment.getLocalGraphicsEnvironment().getAllFonts();

Aide - Tooltip

- public void setToolTipText(String text)

Bordure - Cadre

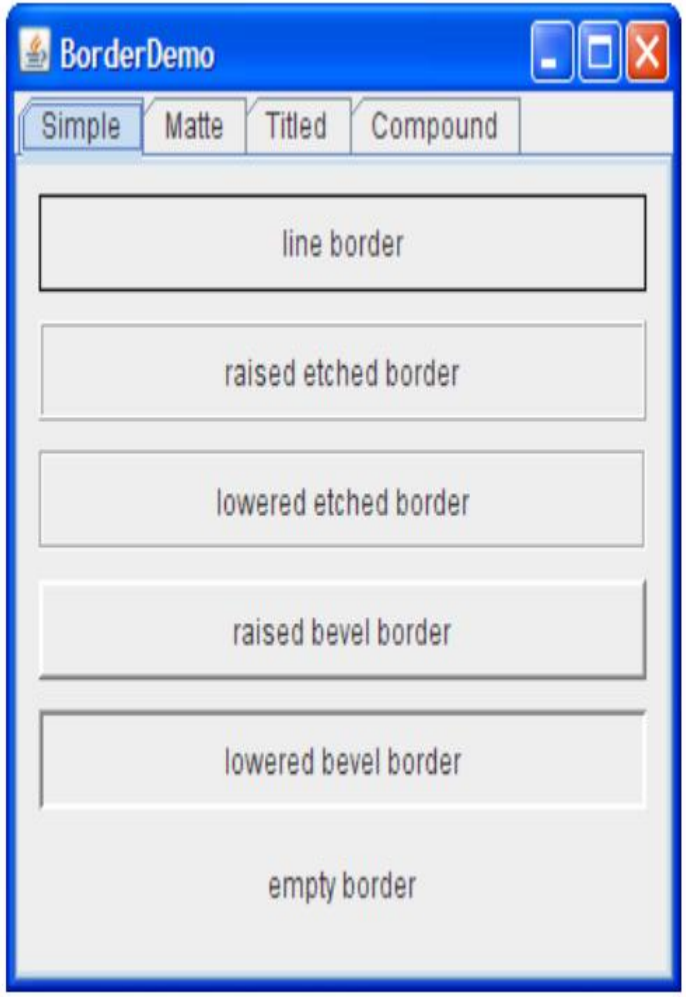
- Initialiser/configurer une bordure

void setBorder(Border)

Border getBorder()

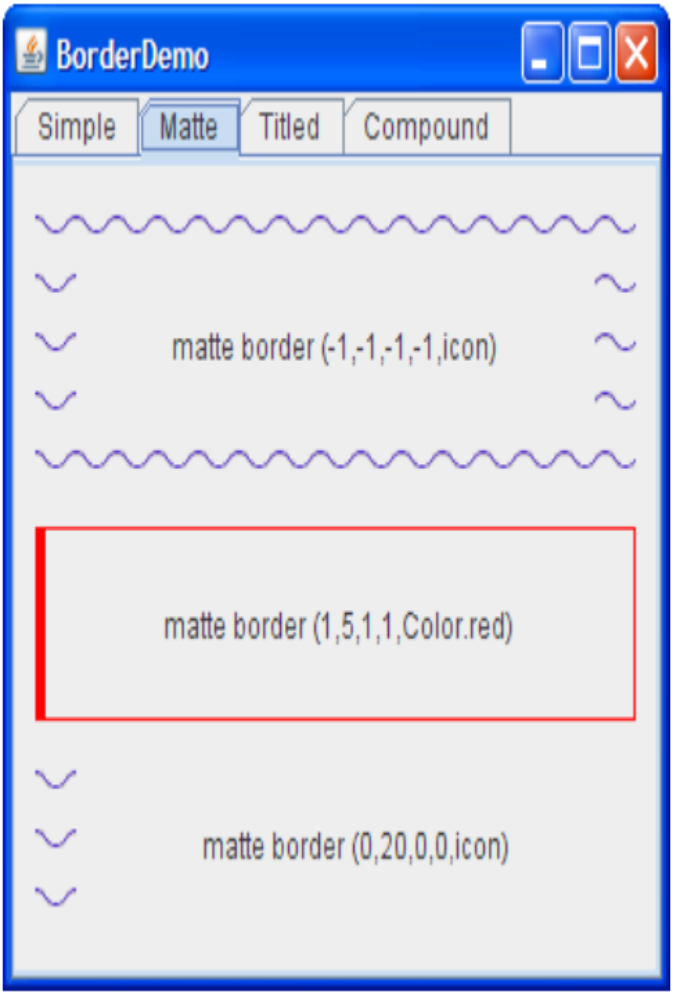
- BorderFactory
  - Simple
  - Matte (“Emmêlé”)
  - Titled
  - Compound

Simple border



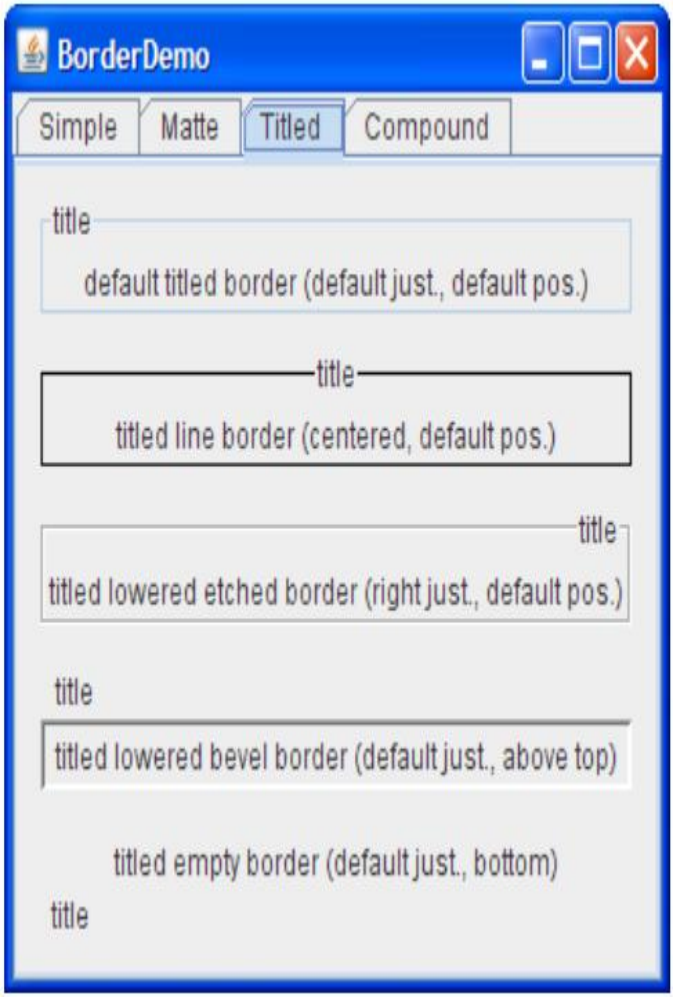
- static Border createLineBorder(**Color color**, int thickness, boolean rounded)
  - Couleur, épaisseur, coins arrondis
- static Border createEtchedBorder(int type, Color highlight, Color shadow)
  - Type (EtchedBorder.RAISED or EtchedBorder.LOWERED), couleurs (partie illuminée, partie ombragée)
- static Border createBevelBorder(**int type**, Color highlightOuter, Color highlightInner, Color shadowOuter, Color shadowInner)
  - Type (BevelBorder.LOWERED or BevelBorder.RAISED), couleurs

Matte border



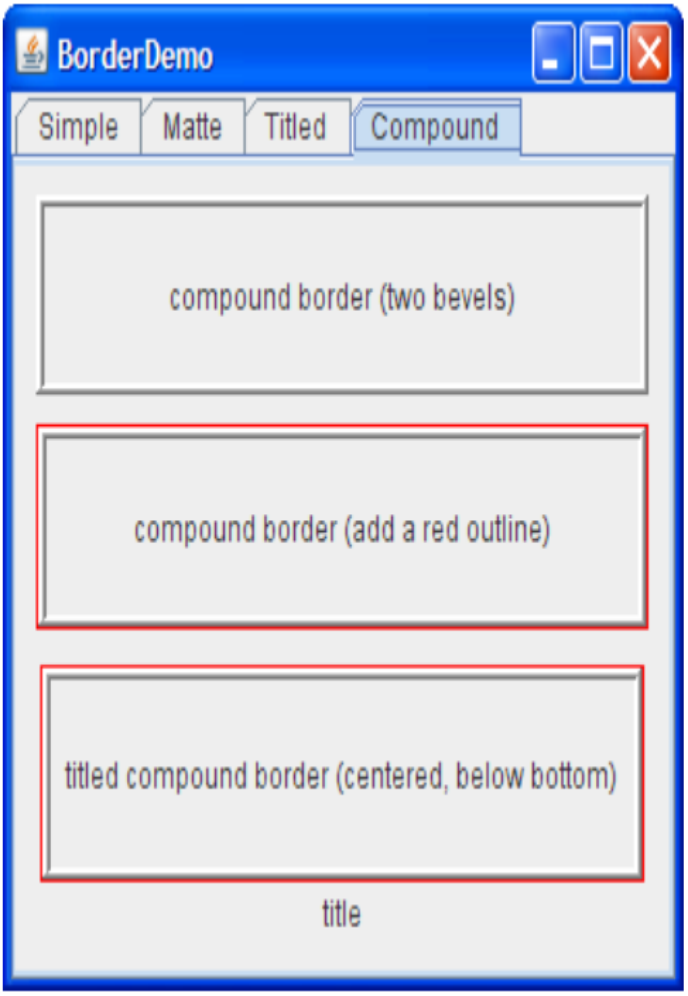
- Static Border createMatteBorder(**int top, int left, int bottom, int right**, Icon tileIcon)
  - Largeur de chaque bord (pixel)
  - Icône

Titled border



- Static Border createTitledBorder(**Border border**, **String title**, int titleJustification, int titlePosition, Font titleFont, Color titleColor)
  - Border ou titre
  - Justification du titre
    - TitledBorder.LEFT, TitledBorder.CENTER , TitledBorder.RIGHT , TitledBorder.LEADING , TitledBorder.TRAILING , TitledBorder.DEFAULT_JUSTIFICATION (leading)
  - Position
    - TitledBorder.ABOVE_TOP , TitledBorder.TOP (sitting on the top line), TitledBorder.BELOW_TOP , TitledBorder.ABOVE_BOTTOM, TitledBorder.BOTTOM (sitting on the bottom line), TitledBorder.BELOW_BOTTOM, TitledBorder.DEFAULT_POSITION (top)
  - Police
  - Couleur du titre

Compound border



- Static Border createCompoundBorder(Border outsideBorder, Border insideBorder)
  - Combinaison de deux types de bordures

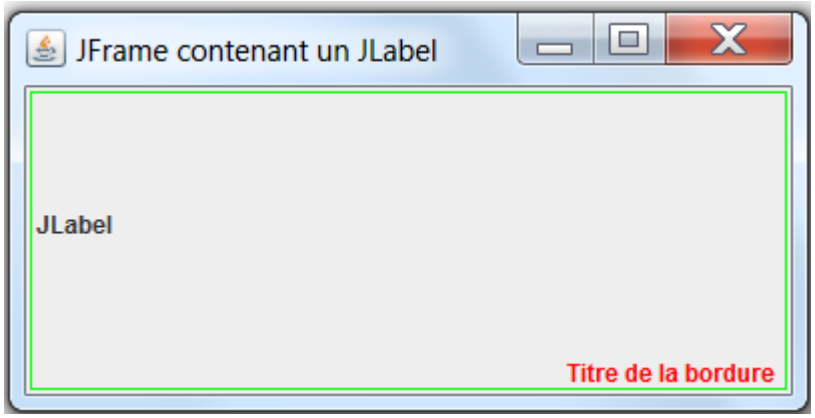
Exercice 1

- **A compléter pour afficher un label encadré d'une bordure rouge aux coins arrondis**

```
public class Exercice1 {
    public static void main(String[] args) {
        JFrame myFrame = new JFrame("JFrame contenant un JLabel");
        JLabel myLabel = ...
                    ...
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.setSize(400, 200);
        myFrame.setLocationRelativeTo(null);
        myFrame.getContentPane().add(myLabel);
        myFrame.setVisible(true);
    }
}
```

Exercice 2

- Ecrire la portion de code permettant d'afficher le label contenu dans la JFrame suivante



JTextField

- Saisie d'un texte utilisateur
- public JTextField(String text)
  - Valeur initiale du texte

- public setText(String t)
- public String getText()

JPasswordField

- public JPasswordField()
- public JPasswordField(String text)

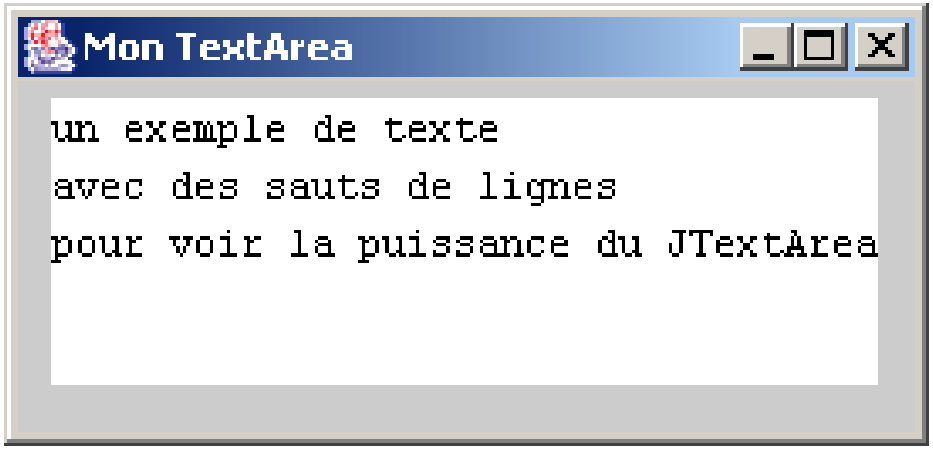
JTextArea

- Champ de saisie sur plusieurs lignes

JTextArea(int rows, int columns)

JTextArea(String text)

JTextArea(String text, int rows, int columns)



Boutons

- JButton
- JToggleButton
- JRadioButton
- JCheckBox

AbstractButton

- Activation par touches alt + mnemonic (autre touche clavier)

public void setMnemonic(int mnemonic)

java.awt.event.KeyEvent

VK_A à VK_Z

- Etat
  - Sélectionné ou non

  public void setSelected(boolean b)

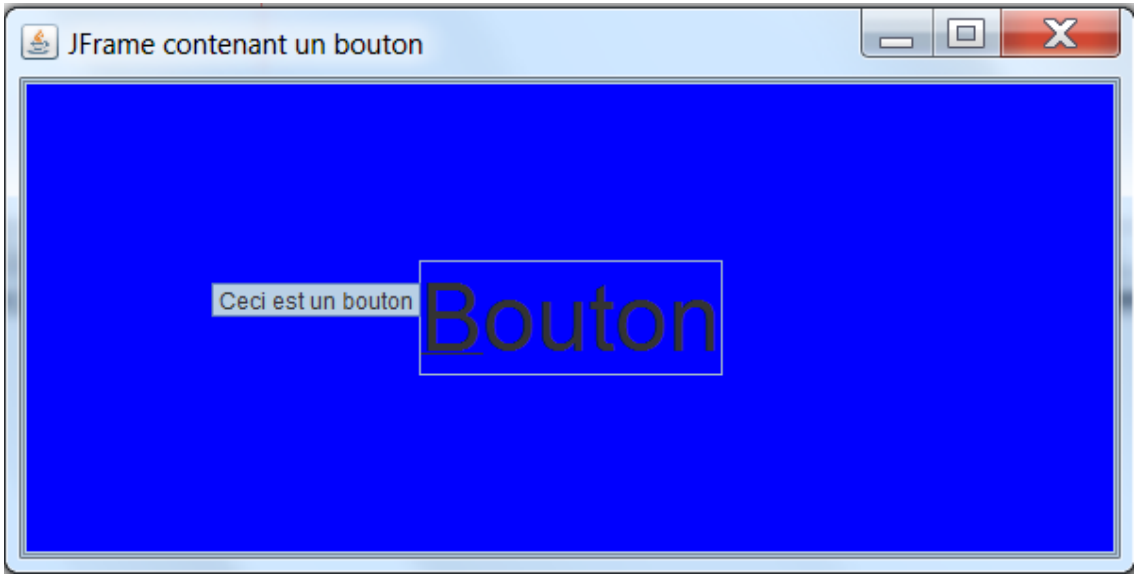
JButton

- Bouton simple avec ou sans image

- JButton(String text, Icon icon)

Exercice 3

- Ecrire la portion de code permettant d'afficher le bouton suivant
  - Raccourci clavier avec la lettre B
  - Au passage du curseur, un texte d'aide s'affiche



Associer une icône

public void setIcon(Icon defaultIcon)

ImageIcon: Implémentation de l'interface Icon

ImageIcon(URL location, String description)

java.net.URL getClass().getResource(String path)

package monpackage contient la classe de code source

package monpackage.images contient le fichier pict.png

Exercice 4

- Ecrire la portion de code permettant d'afficher le bouton suivant

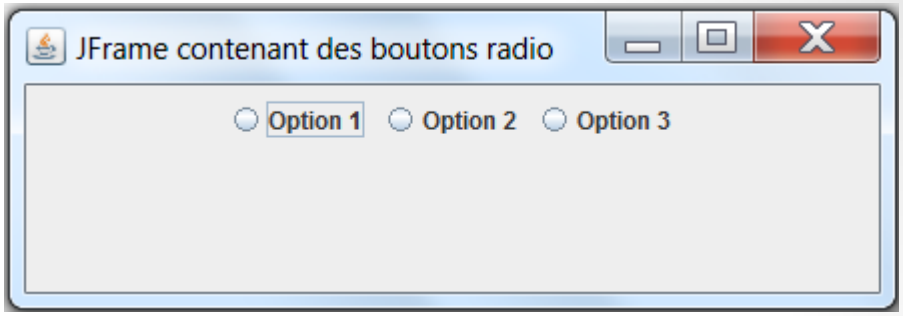
JToggleButton

- Boutons à état
  - Aspect visuel du bouton reflète l'état d'une fonctionnalité
  - Exemple: barres d'outils d'éditeurs de texte (gras, italique, surligné)
  - Deux aspects
    - Sélectionné
    - Non sélectionné

- JToggleButton(String text, Icon icon, boolean selected)

- Classe mère de JRadioButton et JCheckBox

JRadioButton



- Boutons radio

JRadioButton public JRadioButton(String text, Icon icon, boolean selected)

- Prévus pour être utilisés au sein d'un groupe

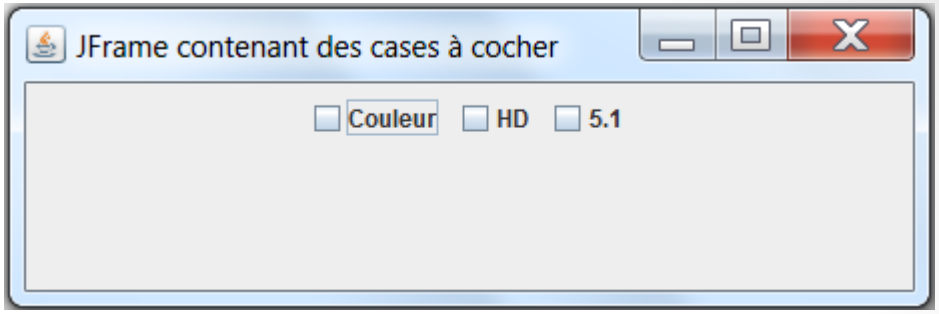
ButtonGroup public ButtonGroup

public void add(AbstractButton b)

Composant logique, doivent aussi être ajoutés à un container graphique (JPanel).

- 1 seul sélectionné à la fois

JCheckBox



- Cases à cocher
- Plusieurs cases peuvent être cochées simultanément

- Doivent être ajoutés à un container graphique (JPanel)

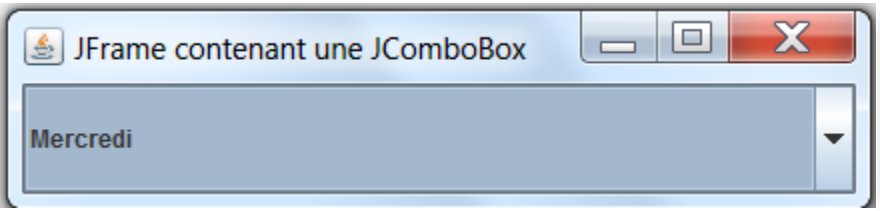
JCheckBox public JCheckBox(String text, Icon icon, boolean selected)

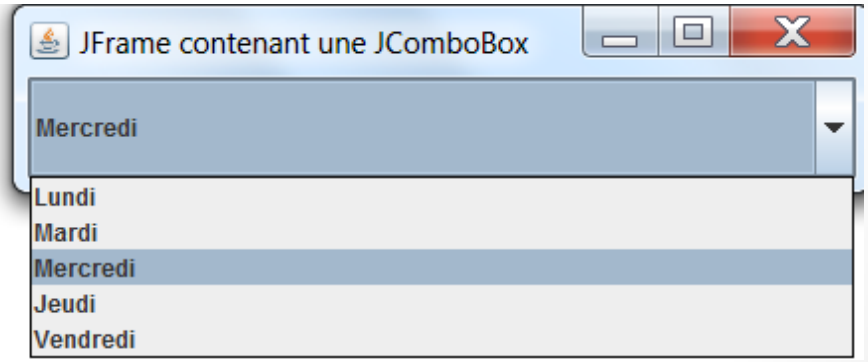
JComboBox

- Liste déroulante pour choisir un item (fermée, ouverte)

public JComboBox(E[] items)

public JComboBox(Vector<E> items)





JComboBox - suite

- Pré-selection

  setSelectedIndex

  public void setSelectedIndex(int anIndex)

- Nombre d'items à afficher quand déroulée

  setMaximumRowCount

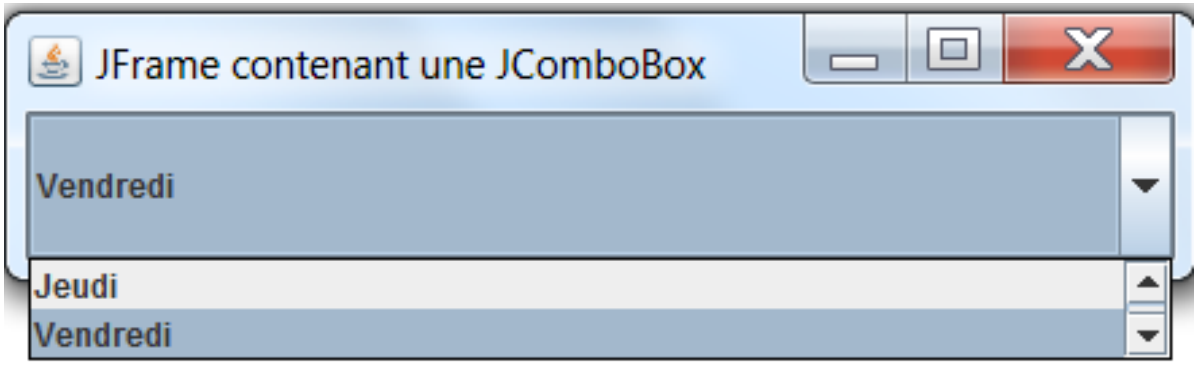
  public void setMaximumRowCount(int count)

- Récupération de l'élément sélectionné

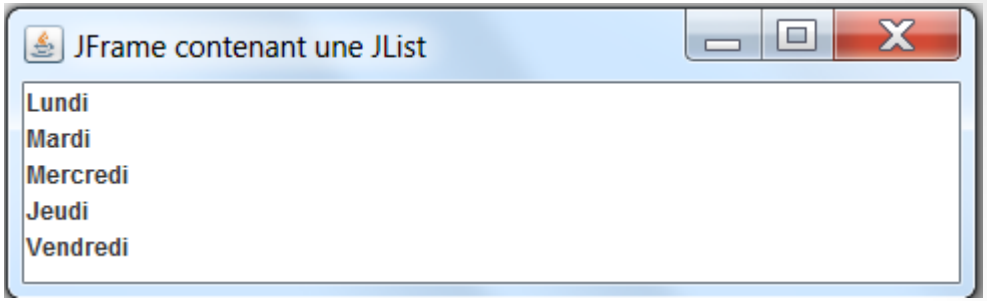
  public Object getSelectedItem()

Exercice 5

- Ecrire la portion de code permettant d'afficher la boîte déroulante suivante



JList



- Liste de valeurs

  public JList(E[] listData)

  public JList(Vector<? extends E> listData)

- Pré-selection

  setSelectedIndex

  public void setSelectedIndex(int anIndex)

- Doit être ajoutée à un container particulier pour être défilante (JScrollPane)

Exercice 6

- Ecrire la portion de code permettant d'afficher la liste suivante