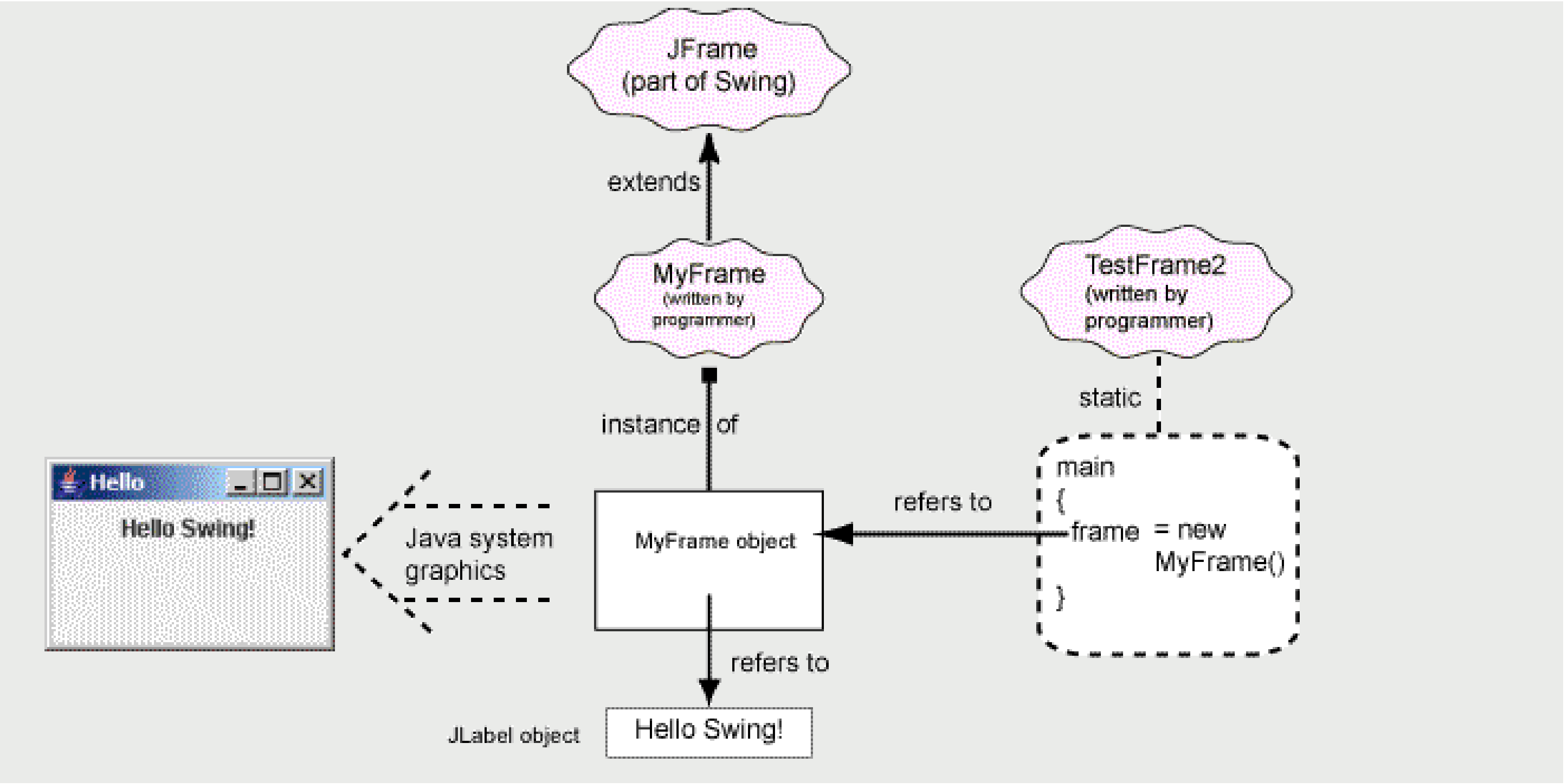
SIMPLON.CO

Week#2

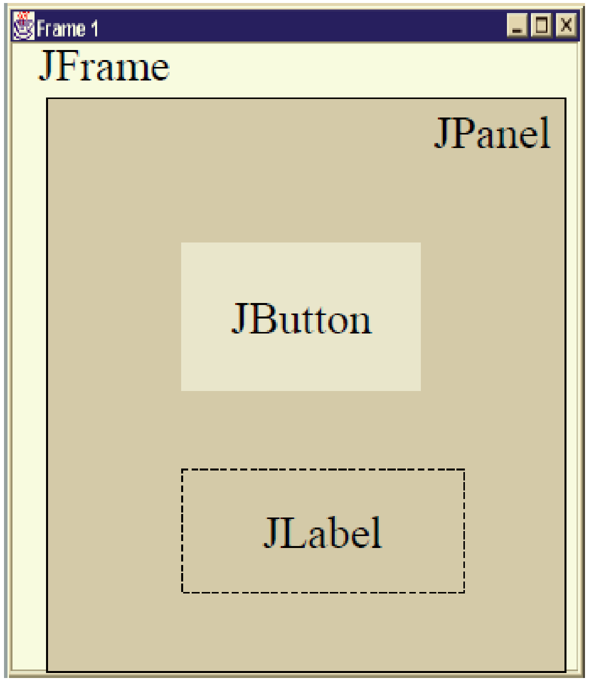
Parts of the GUI
Swing #1

Athanasia Katsouraki

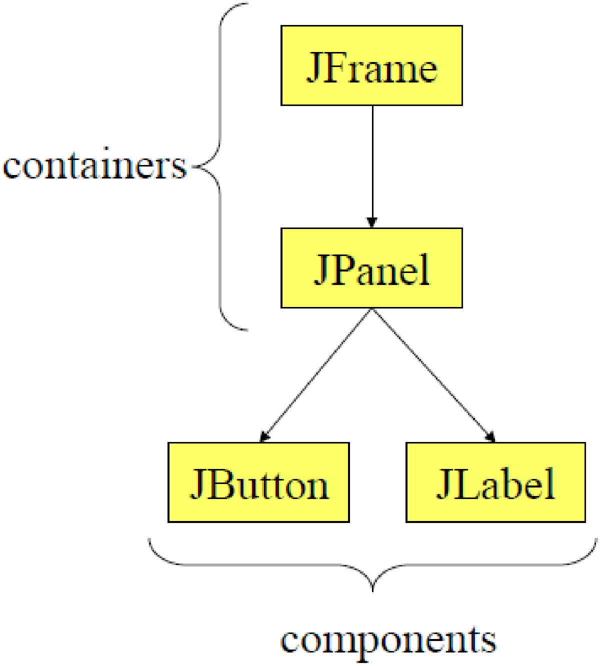
18/10/2017

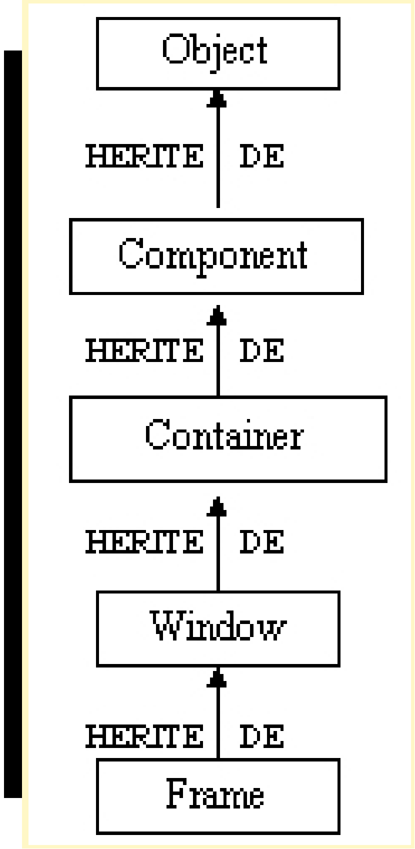

JFrame
(part of Swing)
extends
MyFrame
(written by
programmer)
TestFrame2
(written by
programmer)
static
instance of
Hello
Hello Swing!
Java system
graphics
MyFrame object
refers to
main
{
frame = new
MyFrame()
}
refers to
JLabel object
Hello Swing!

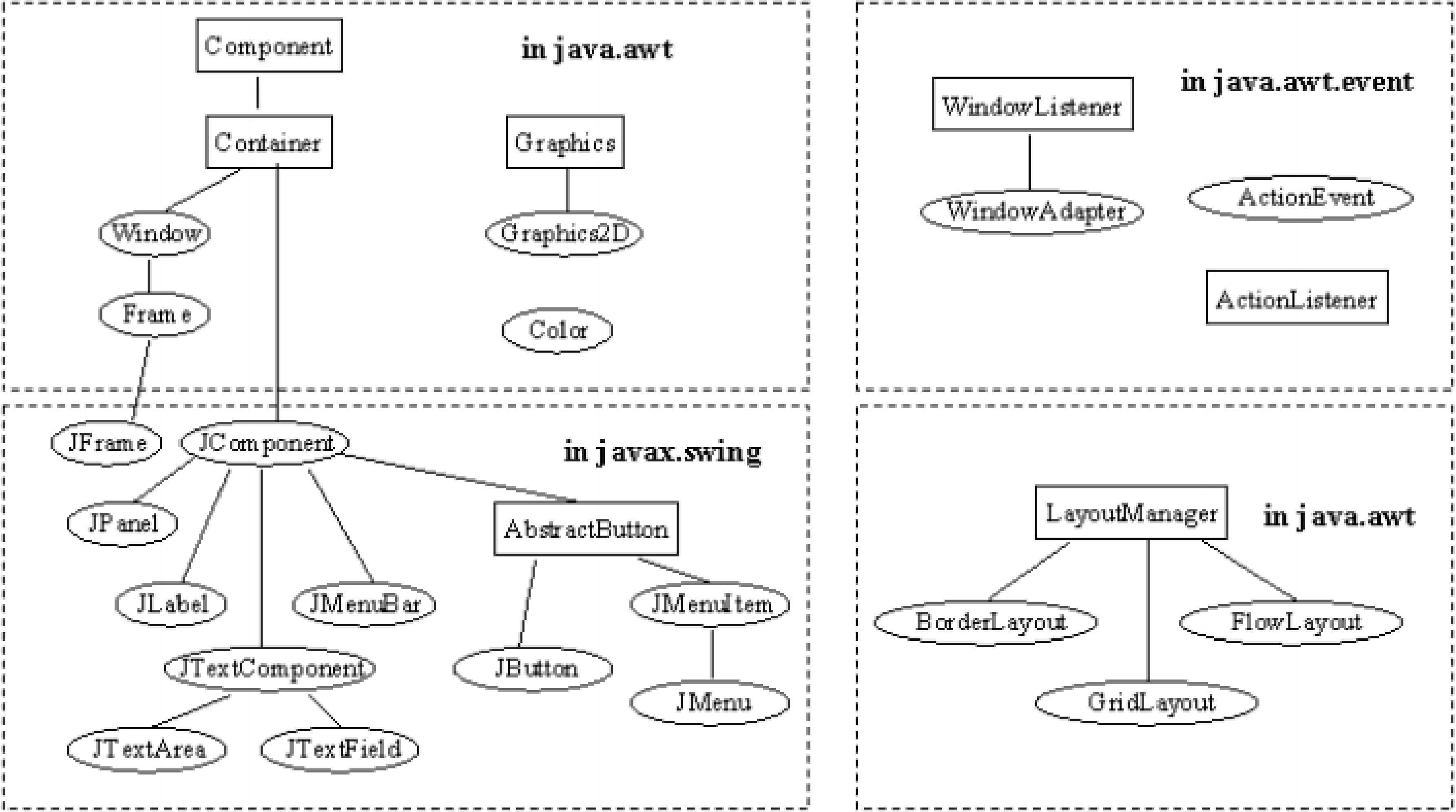
GUI


Frame 1
JFrame
JPanel
JButton
JLabel

Classes


containers
JFrame
JPanel
JButton
JLabel
components


Object
HERITE DE
Component
HERITE DE
Container
HERITE DE
Window
HERITE DE
Frame


in java.awt
Component
Container
Graphics
Window
Graphics2D
Frame
Color
in java.awt.event
WindowListener
WindowAdapter
ActionEvent
ActionListener
in javax.swing
JFrame
JComponent
JPanel
AbstractButton
JLabel
JMenuBar
JMenuItem
JTextComponent
JButton
JMenu
JTextArea
JTextField
in java.awt
LayoutManager
BorderLayout
FlowLayout
GridLayout

Abstract Window Toolkit

java.awt.*, java.awt.event.*

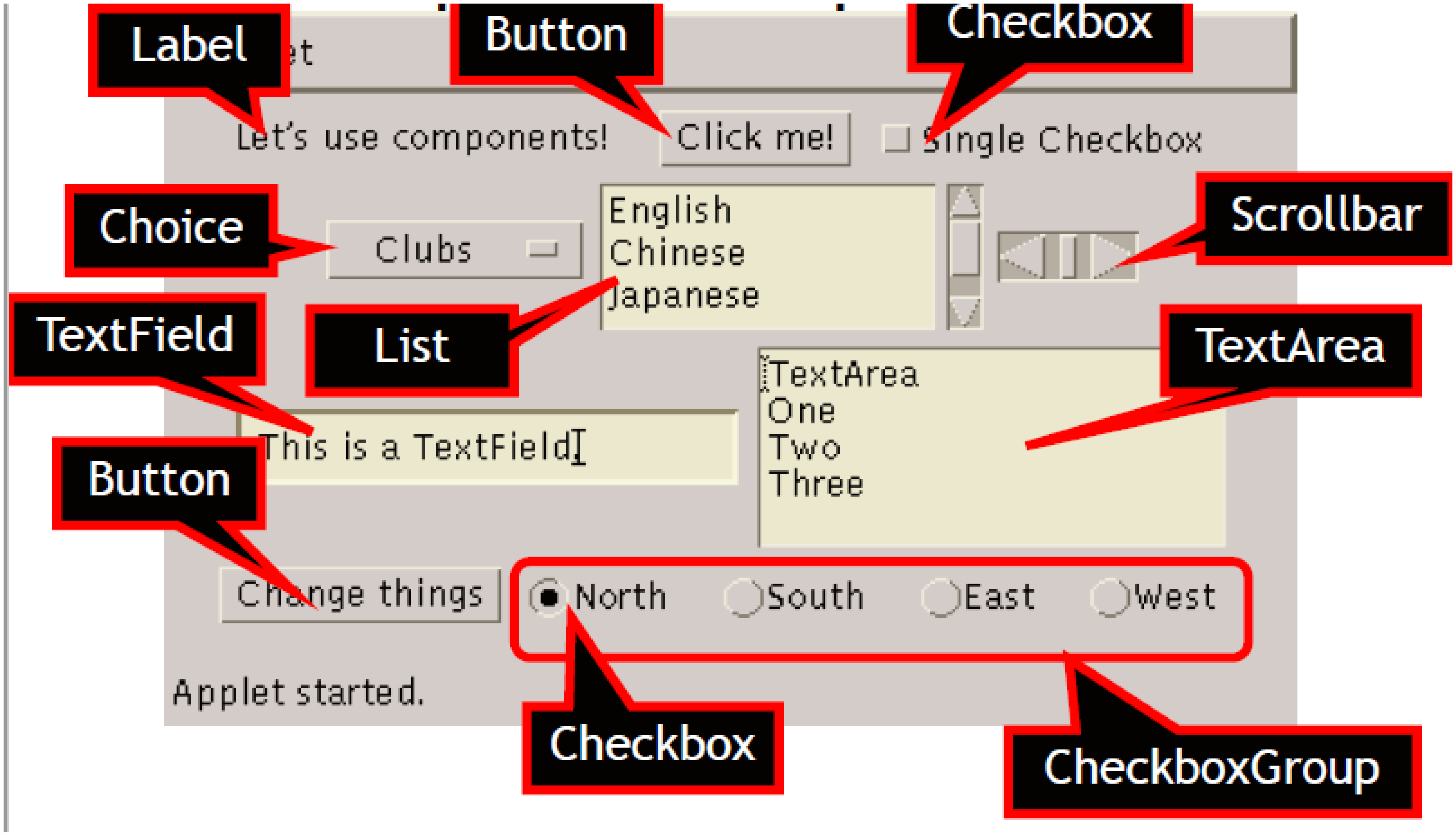
Window (Containers): Frame, Window, Panel, Applet

Elements (Components): Button, Checkbox, Label, Scrollbar, TextField, TextArea

Managers (LayoutManagers): FlowLayout, BorderLayout, GridLayout

(Listener interfaces):
ActionListener, TextListener κλπ.

Exemple



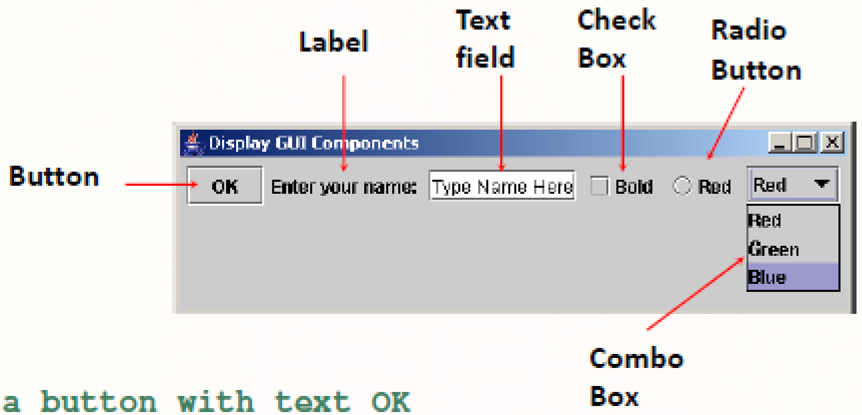
Let's use components!

This is a TextField

TextArea
One
Two
Three

Applet started.

Exemple



Label

Text field

Check Box

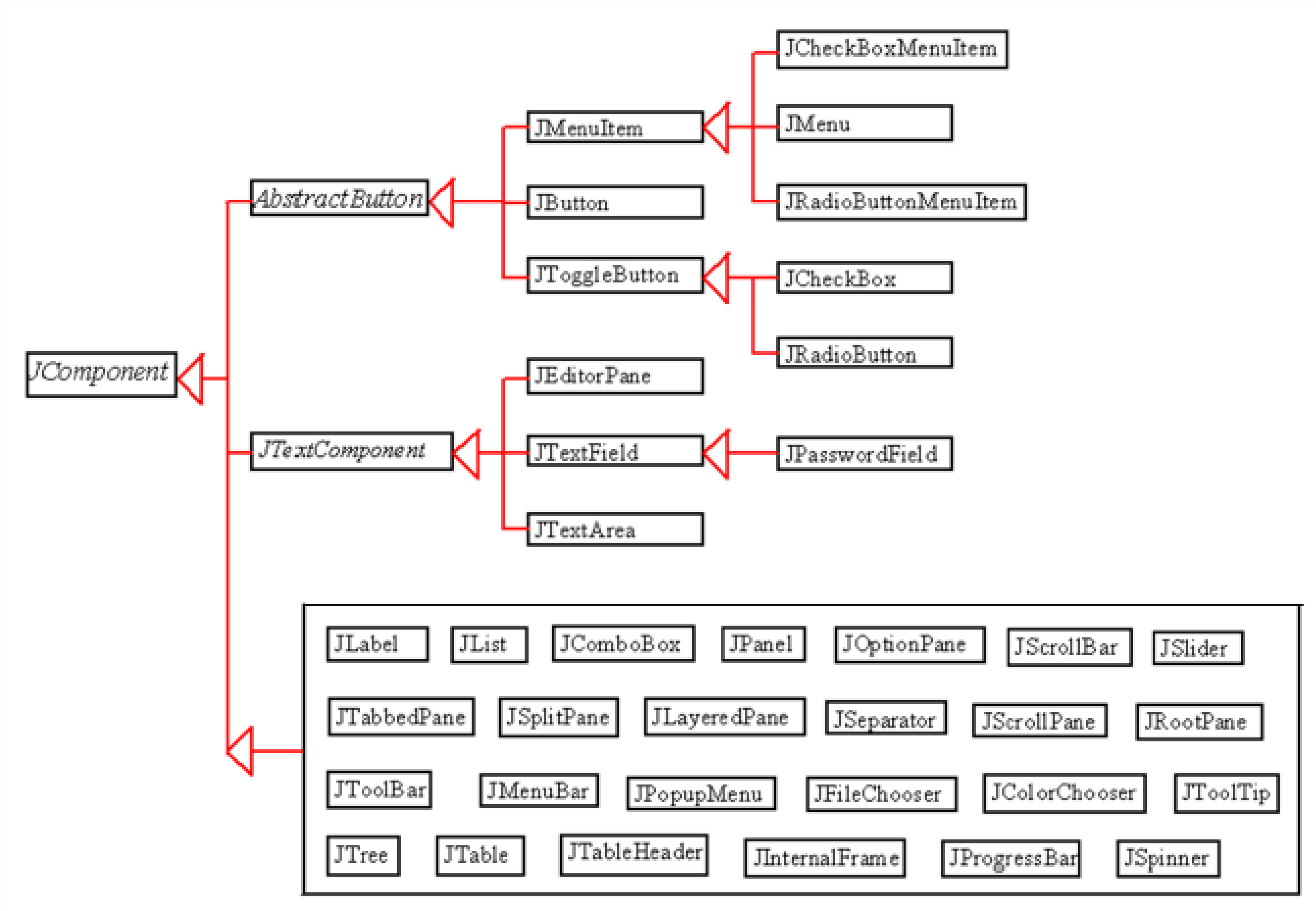
Radio Button

Button

Combo Box

```
// Create a button with text OK
JButton jbtOK = new JButton("OK");
// Create a label with text "Enter your name: "
JLabel jlblName = new JLabel("Enter your name: ");
// Create a text field with text "Type Name Here"
JTextField jtfName = new JTextField("Type Name Here");
// Create a check box with text bold
JCheckBox jchkBold = new JCheckBox("Bold");
// Create a radio button with text red
JRadioButton jrbRed = new JRadioButton("Red");
// Create a combo box with choices red, green, and blue
JComboBox jcbColor = new JComboBox(new String[] { "Red","Green","Blue"});
```

Elements GUI (Swing)



On commence par…

1. Nous créons

   JFrame

   JPanel

   Composants (JButton, JLabel)

   Le Listener pour Jbutton

2. On ajoute (add)

   Listener pour Jbutton

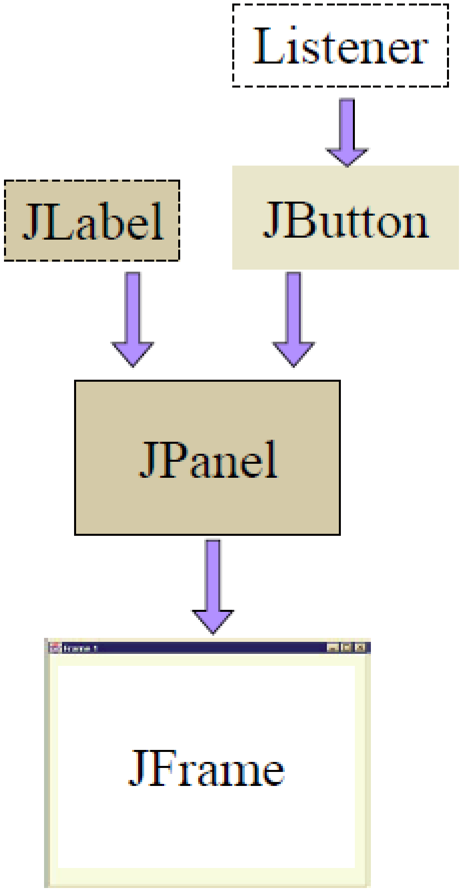
   Ajouter (ajouter la méthode)

   Les composants de Jpanel

   JPanel à JFrame

3. Nous nous présentons

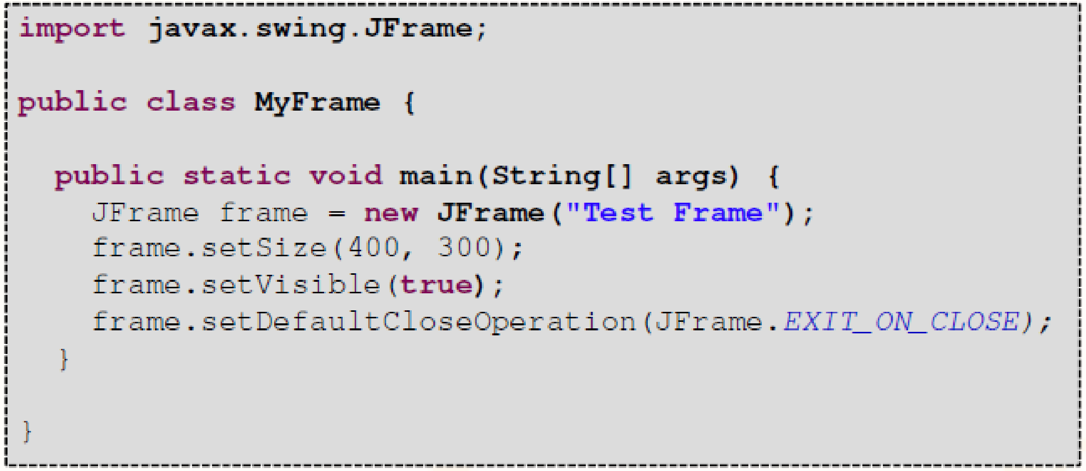
JFrame (méthode show)



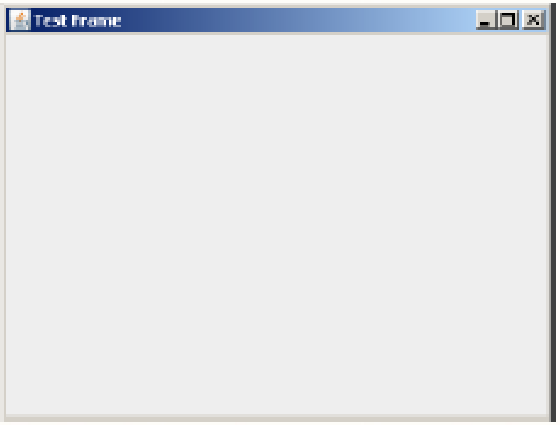
Frames

JFrame: est un genre fenêtre
C'est un conteneur: il peut contient d'autres éléments

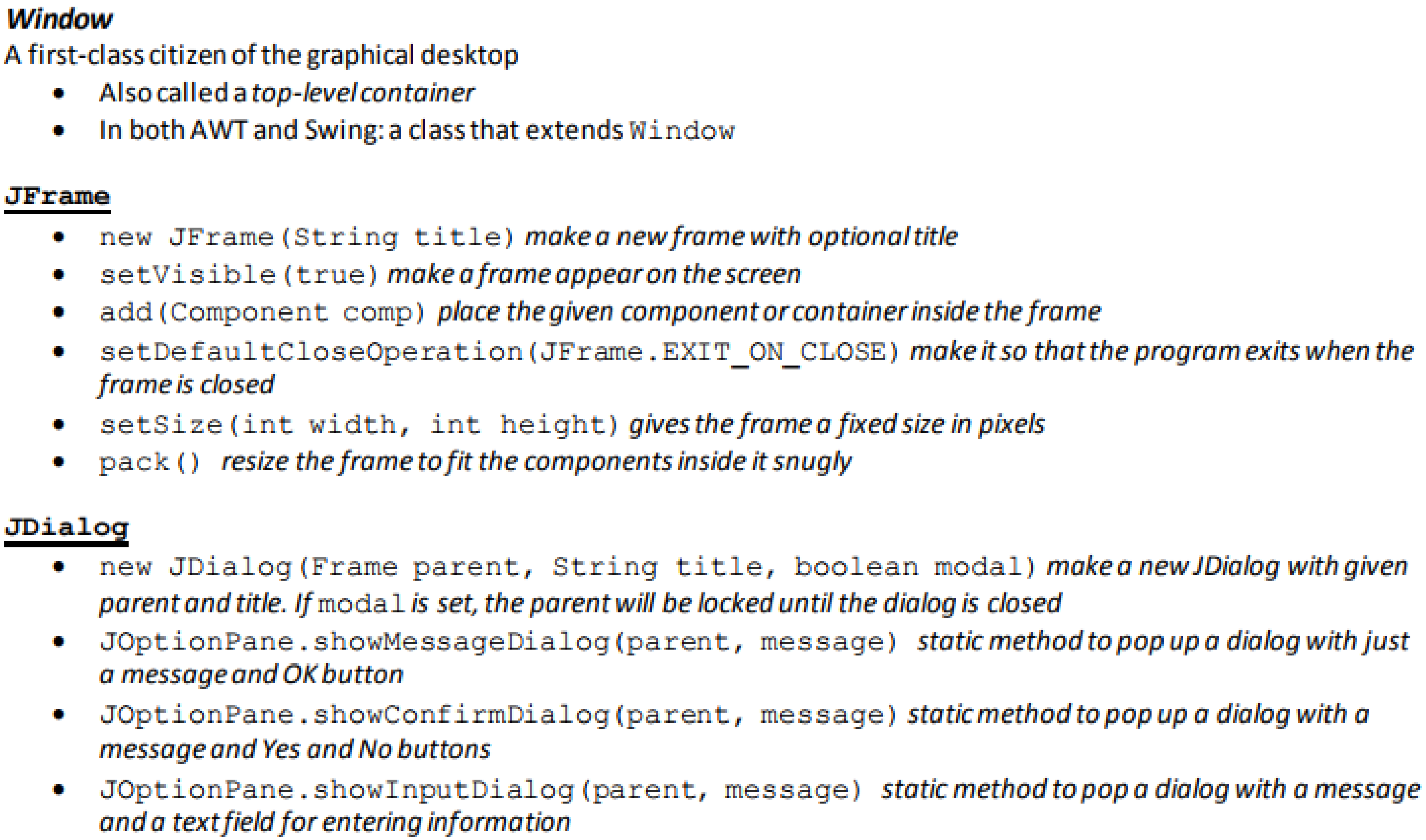
JFrame f=new JFrame(“title”);
f.getContentPane().add(myComponent);



```
import javax.swing.JFrame;

public class MyFrame {

  public static void main(String[] args) {
    JFrame frame = new JFrame("Test Frame");
    frame.setSize(400, 300);
    frame.setVisible(true);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }

}
```



Window



Window

A first-class citizen of the graphical desktop

- Also called a *top-level container*
- In both AWT and Swing: a class that extends `Window`

`JFrame`

- `new JFrame(String title)` *make a new frame with optional title*
- `setVisible(true)` *make a frame appear on the screen*
- `add(Component comp)` *place the given component or container inside the frame*
- `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)` *make it so that the program exits when the frame is closed*
- `setSize(int width, int height)` *gives the frame a fixed size in pixels*
- `pack()` *resize the frame to fit the components inside it snugly*

`JDialog`

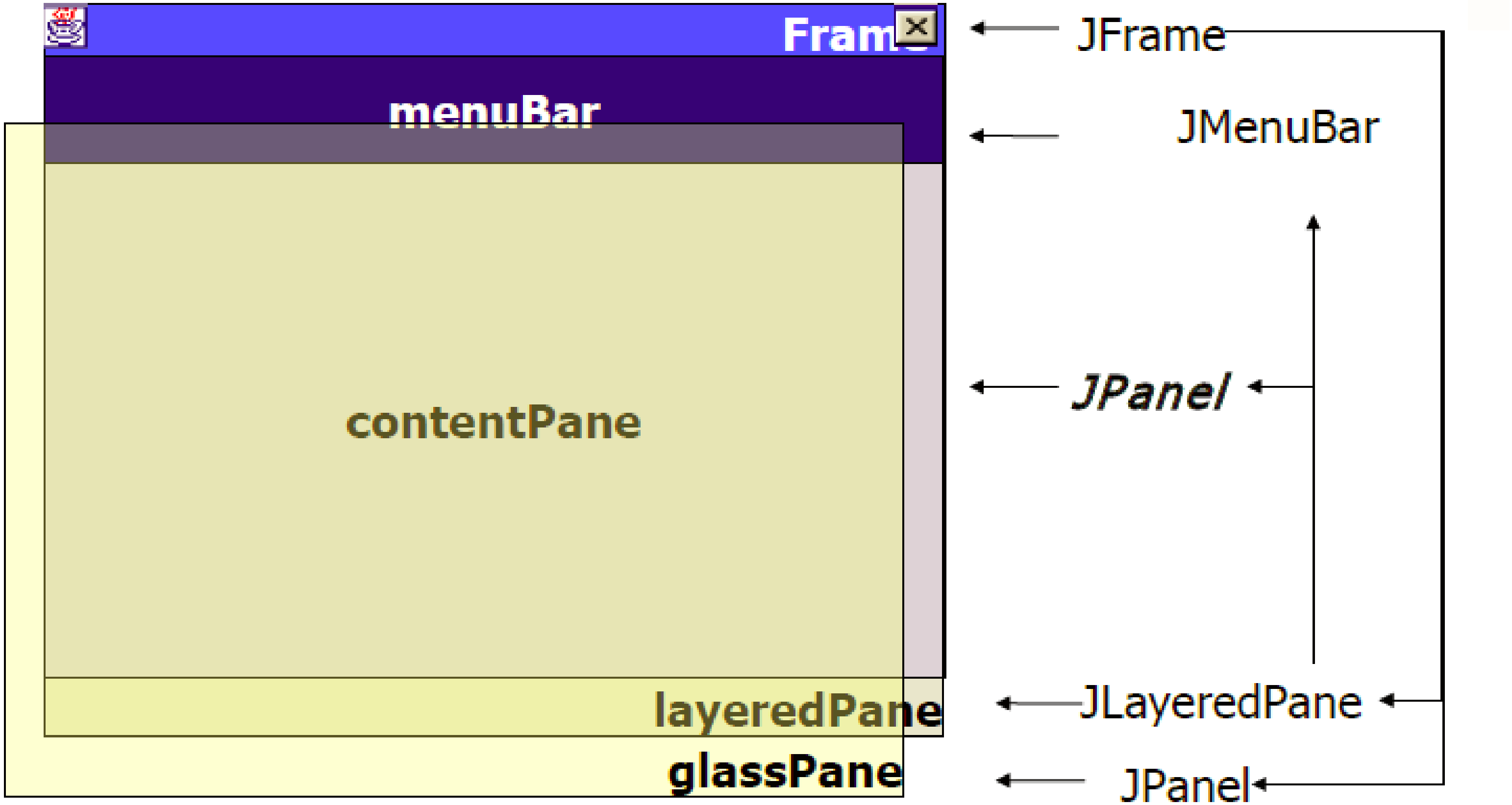
- `new JDialog(Frame parent, String title, boolean modal)` *make a new JDialog with given parent and title. If* `modal` *is set, the parent will be locked until the dialog is closed*
- `JOptionPane.showMessageDialog(parent, message)` *static method to pop up a dialog with just a message and OK button*
- `JOptionPane.showConfirmDialog(parent, message)` *static method to pop up a dialog with a message and Yes and No buttons*
- `JOptionPane.showInputDialog(parent, message)` *static method to pop a dialog with a message and a text field for entering information*

Fermer Frame..

En utilisant la methode :

setDefaultCloseOperation(int) avec les args:
WindowConstants.DISPOSE_ON_CLOSE – ferme le frame
WindowConstants.EXIT_ON_CLOSE – ferme l'application
WindowConstants.DO_NOTHING_ON_CLOSE – fait rien
WindowConstants.HIDE_ON_CLOSE – cacher le frame

```
f.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE)
```



Frame

menuBar

contentPane

layeredPane

glassPane

JFrame

JMenuBar

JPanel

JLayeredPane

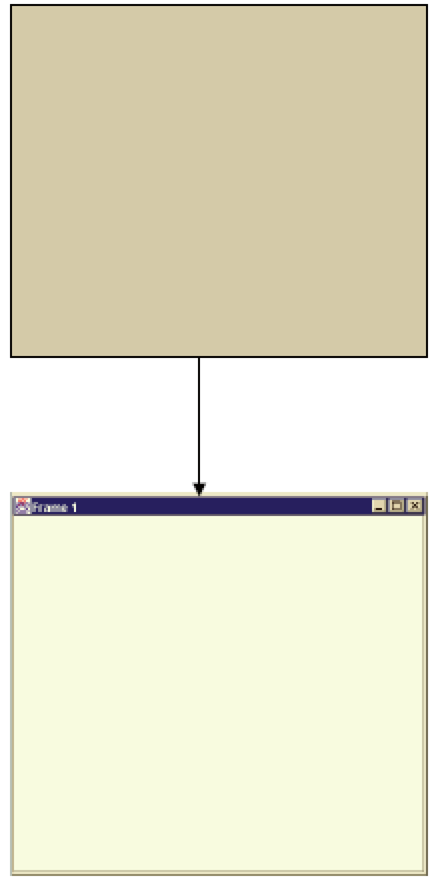
JPanel

Exemple

```
import javax.swing.*;
…
JFrame f = new JFrame(“title”);
JButton b = new JButton(“press me”);
f.getContentPane().add(b);
f.show();
```

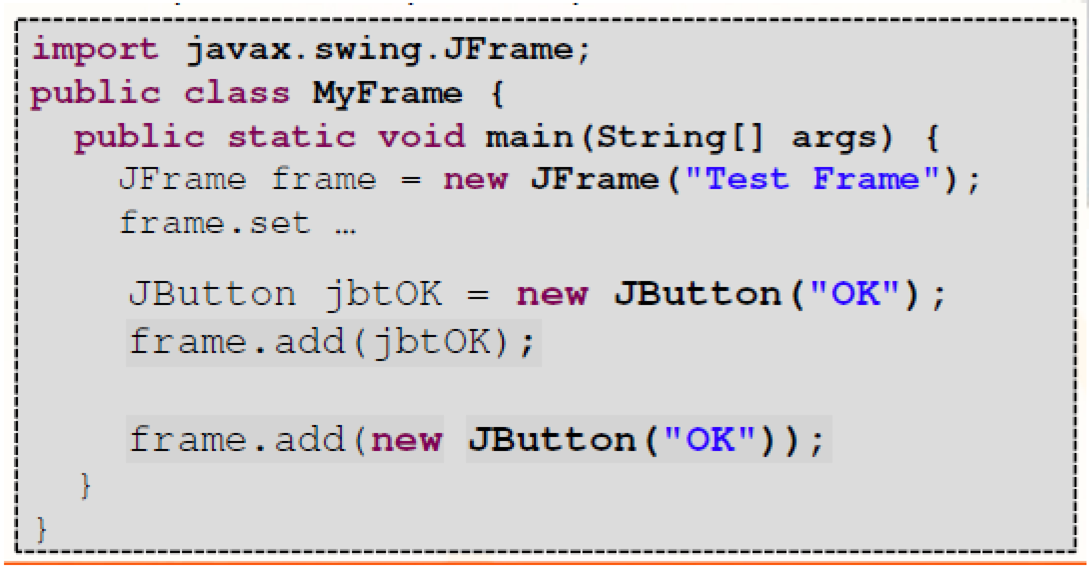
- Παρόμοια

```
JFrame f = new JFrame(“title”);
JPanel p = new JPanel( );
JButton b = new JButton(“press me”);
p.add(b);
f.setContentPane(p);
f.show();
```

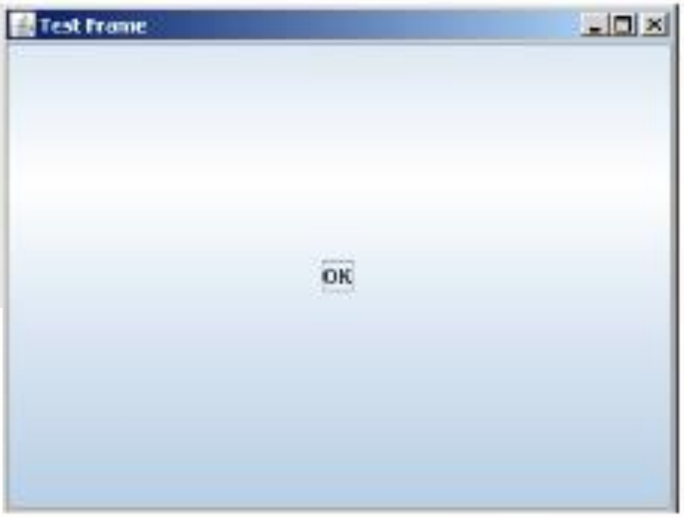


Ajouter des elements…

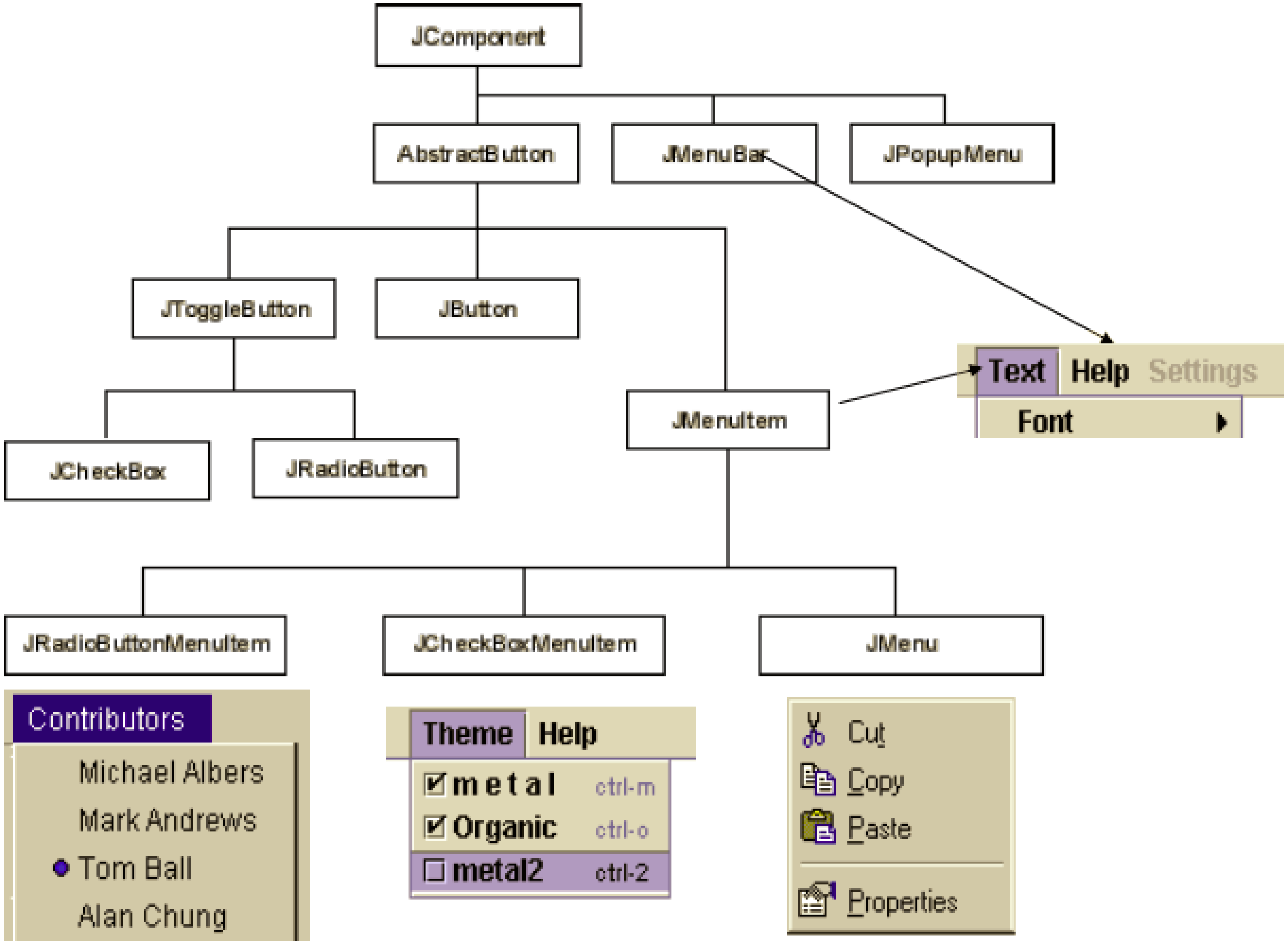
Ajouter des elements :
function frame.add ()
(frame.getContentPane () .add ())



```
import javax.swing.JFrame;
public class MyFrame {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Test Frame");
    frame.set …

    JButton jbtOK = new JButton("OK");
    frame.add(jbtOK);

    frame.add(new JButton("OK"));
  }
}
```



MENUS



JComponent

AbstractButton

JMenuBar

JPopupMenu

JToggleButton

JButton

JMenuItem

JCheckBox

JRadioButton

JRadioButtonMenuItem

JCheckBoxMenuItem

JMenu

Menu Bar

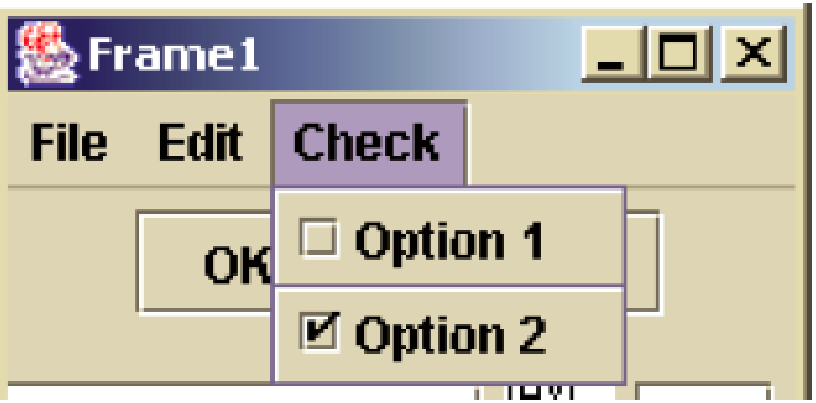
La barre de menu n'est pas obligatoire pour un JFrame
Nous pouvons l'ajouter -> un objet JMenuBar

```
JMenuBar menu = new JMenuBar ();
```

et l'assigner à Jframe

```
f.setJMenuBar (menu);
```

```
JMenu file = new JMenu(“File");
file.add(new JMenuItem(“Open"));
file.add(new JMenuItem(“Close"));
JMenu edit = new JMenu(“Edit");
edit.add(new JMenuItem(“Copy"));
edit.add(new JMenuItem(“Paste"));
JMenu check = new JMenu(“Check");
check.add(new JCheckBoxMenuItem(“Option 1"));
check.add(new JSeparator());
check.add(new JCheckBoxMenuItem(“
MenuBar mb = new MenuBar();
mb.add(file);
mb.add(edit);
mb.add(check);
myFrame.setJMenuBar(mb);
```



Quoi d’autre?

Image

```
        ImageIcon image = new ImageIcon("spiral.gif");
        f.setIconImage(image.getImage());
```

Taille init.

```
         f.setSize(100,100);
```

Position (par defaut) (0,0) = haut a gauche

```
f.setLocation(50, 100);
```

Taille et Position :

```
f.setBounds(120,120,300,300);
```

Analys. d’ecran:

```
Dimension dim = f.getToolkit().getScreenSize(); int
screenwidth=dim.width; int screenlegthn=dim.length;
```

JPanel

JPanel est le conteneur le plus basique dans un JFrame.

Un JPanel peut contenir des composants ou d'autres JPanel permettant ainsi une meilleure organisation dans la fenêtre.

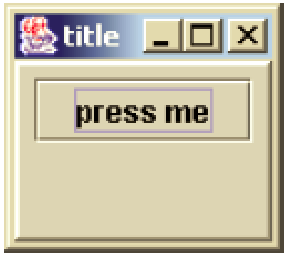
Chaque JPanel a un gestionnaire de placement de données (LayoutManager).

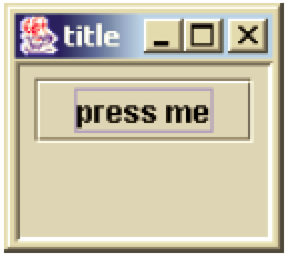
JPanel : on peut definir un type de bordure
(Border)

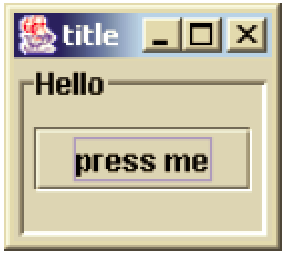
Borders

javax.swing.border
BevelBorder (élevé ou submergé)
CompoundBorder (double)
EmptyBorder (transparent)
TitledBorder, et ainsi de suite.

```
Panneau p = nouveau panneau ();
BevelBorder bb = new BevelBorder (BevelBorder.RAISED);
p.setBorder (bb);
```







Layout Managers

Layout Managers : spécifient comment l'application apparaîtra
sur n'importe quelle plateforme -gérer le placement des
éléments dans la fenêtre

Certains sont contenus dans java.awt
alors que certains d'autres chez javax.swing

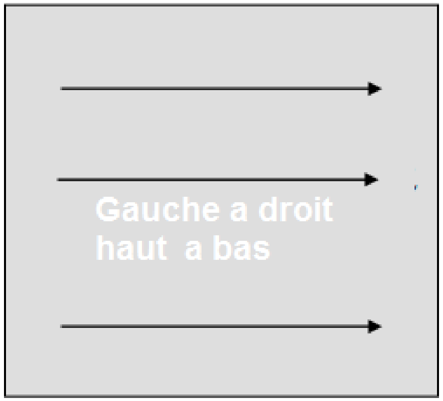
Chaque conteneur a un layout manager qui arrange les objets graphiques à l'intérieur.

- setLayout (LayoutManager)

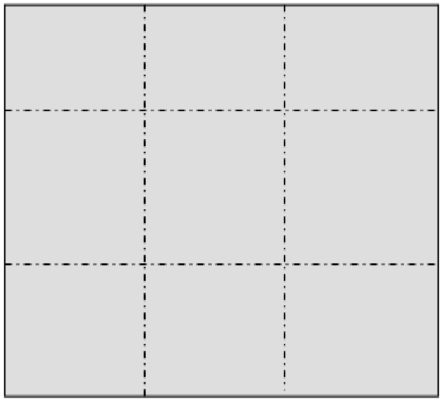
- Exemples LayoutManager:
- FlowLayout: un objet sous l'autre en série
- GridLayout: disposition dans une table
- BorderLayout: mise en page en Est, Sud, Ouest, Nord, Centre.

Layout

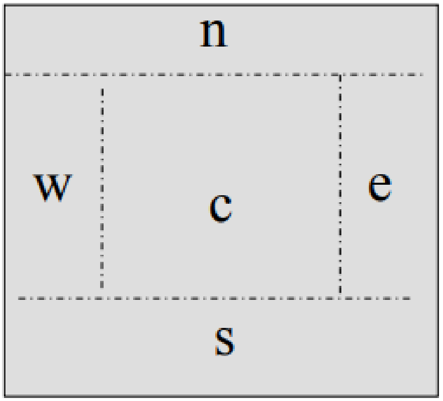
FlowLayout



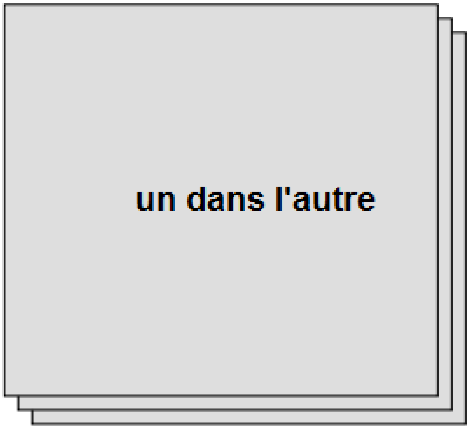
GridLayout



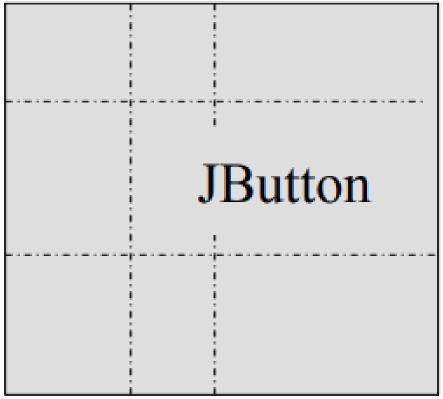
BorderLayout



CardLayout



GridBagLayout



Exemples

- FlowLayout(),

  myPanel.setLayout(new
     FlowLayout(FlowLayout.CENTER));

  myPanel.add(new JButton(“1”));

- GridLayout(int rows, int columns)

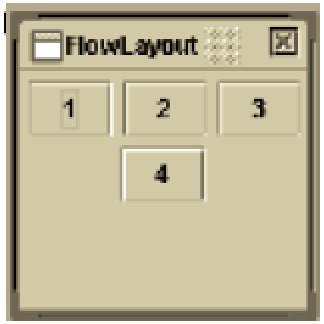
  myPanel.setLayout(new GridLayout(2,2));

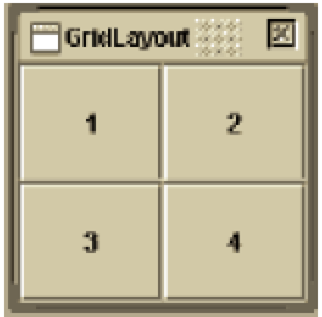
  myPanel.add(new JButton(“1”));

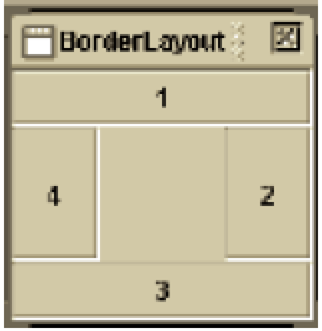
- BorderLayout()

  myPanel.setLayout(new BorderLayout());

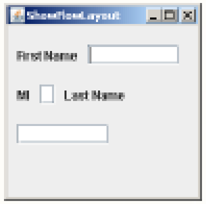
  add(new JButton(“1"), BorderLayout.NORTH);

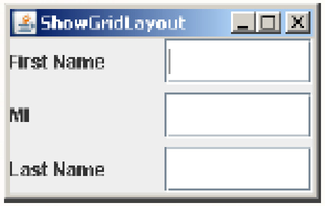


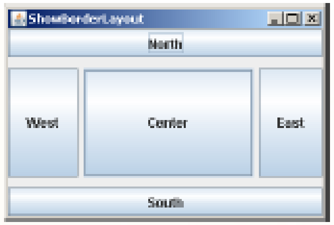


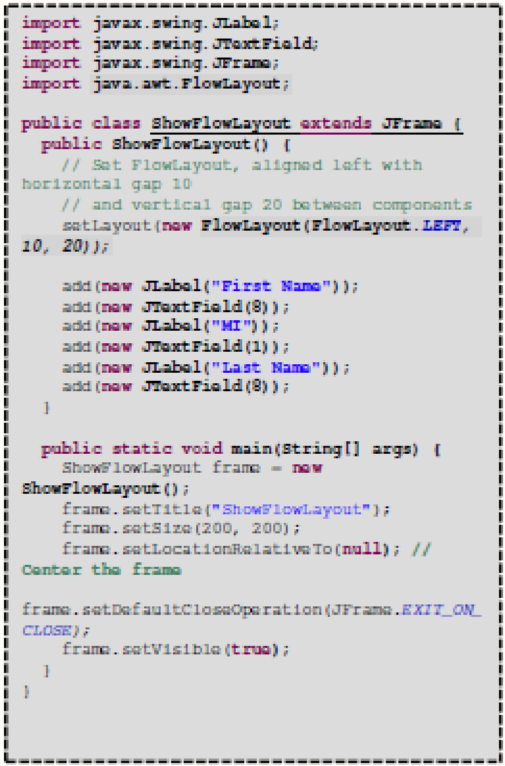


Exemples

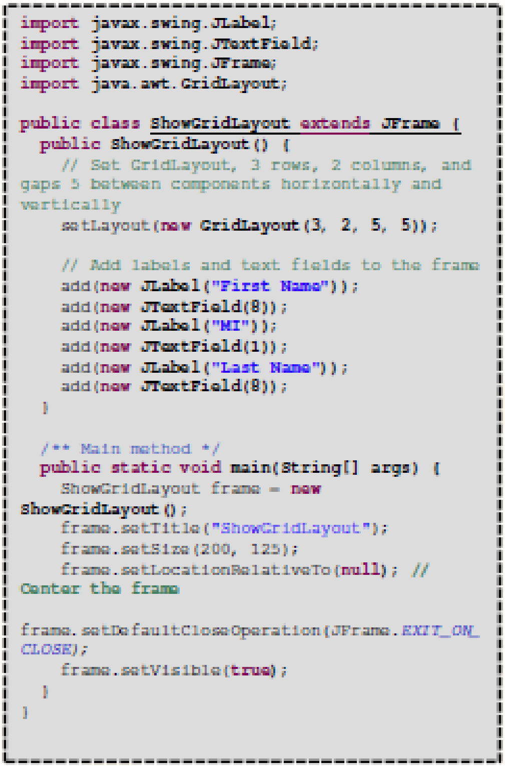




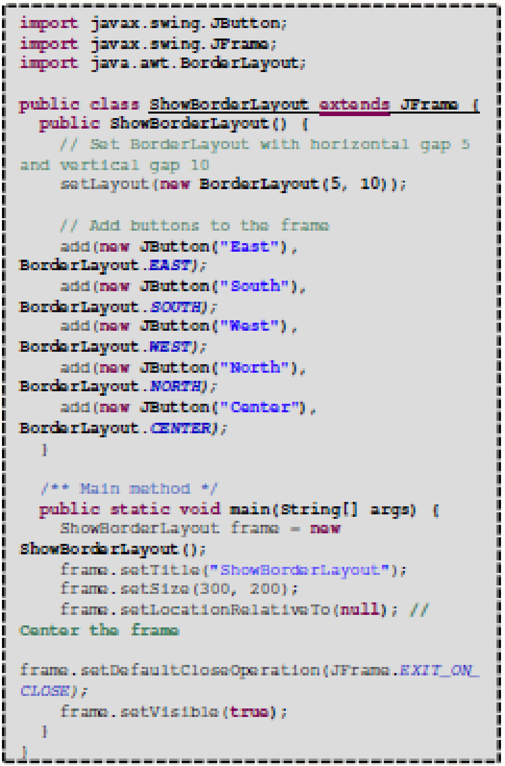




```
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JFrame;
import java.awt.FlowLayout;

public class ShowFlowLayout extends JFrame {
  public ShowFlowLayout() {
    // Set FlowLayout, aligned left with
horizontal gap 10
    // and vertical gap 20 between components
    setLayout(new FlowLayout(FlowLayout.LEFT,
10, 20));

    add(new JLabel("First Name"));
    add(new JTextField(8));
    add(new JLabel("MI"));
    add(new JTextField(1));
    add(new JLabel("Last Name"));
    add(new JTextField(8));
  }

  public static void main(String[] args) {
    ShowFlowLayout frame = new
ShowFlowLayout();
    frame.setTitle("ShowFlowLayout");
    frame.setSize(200, 200);
    frame.setLocationRelativeTo(null); //
Center the frame

frame.setDefaultCloseOperation(JFrame.EXIT_ON_
CLOSE);
    frame.setVisible(true);
  }
}
```



```
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JFrame;
import java.awt.GridLayout;

public class ShowGridLayout extends JFrame {
  public ShowGridLayout() {
    // Set GridLayout, 3 rows, 2 columns, and
gaps 5 between components horizontally and
vertically
    setLayout(new GridLayout(3, 2, 5, 5));

    // Add labels and text fields to the frame
    add(new JLabel("First Name"));
    add(new JTextField(8));
    add(new JLabel("MI"));
    add(new JTextField(1));
    add(new JLabel("Last Name"));
    add(new JTextField(8));
  }

  /** Main method */
  public static void main(String[] args) {
    ShowGridLayout frame = new
ShowGridLayout();
    frame.setTitle("ShowGridLayout");
    frame.setSize(200, 125);
    frame.setLocationRelativeTo(null); //
Center the frame

frame.setDefaultCloseOperation(JFrame.EXIT_ON_
CLOSE);
    frame.setVisible(true);
  }
}
```

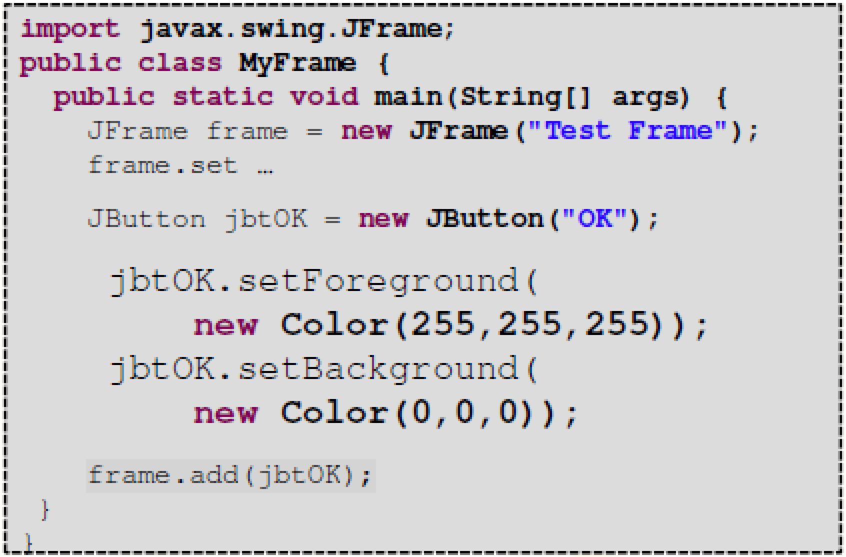


```
import javax.swing.JButton;
import javax.swing.JFrame;
import java.awt.BorderLayout;

public class ShowBorderLayout extends JFrame {
  public ShowBorderLayout() {
    // Set BorderLayout with horizontal gap 5
and vertical gap 10
    setLayout(new BorderLayout(5, 10));

    // Add buttons to the frame
    add(new JButton("East"),
BorderLayout.EAST);
    add(new JButton("South"),
BorderLayout.SOUTH);
    add(new JButton("West"),
BorderLayout.WEST);
    add(new JButton("North"),
BorderLayout.NORTH);
    add(new JButton("Center"),
BorderLayout.CENTER);
  }

  /** Main method */
  public static void main(String[] args) {
    ShowBorderLayout frame = new
ShowBorderLayout();
    frame.setTitle("ShowBorderLayout");
    frame.setSize(300, 200);
    frame.setLocationRelativeTo(null); //
Center the frame

frame.setDefaultCloseOperation(JFrame.EXIT_ON_
CLOSE);
    frame.setVisible(true);
  }
}
```

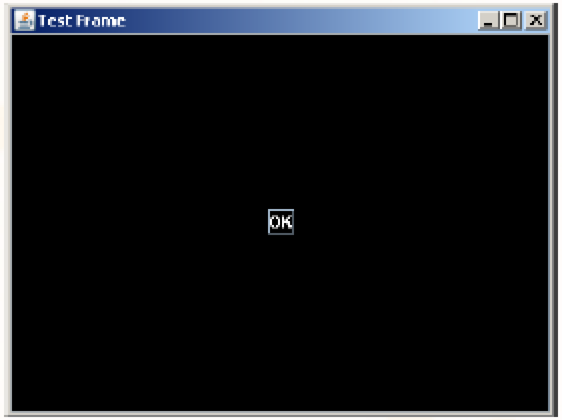
Couleurs

La détermination de la couleur s'effectue à l'aide de: java.awt.Color standard red-green-blue (RGB)

- Création de couleurs: Couleur c = nouvelle couleur (r, g, b);
- Exemple de méthodes graphiques utilisant des couleurs:
- Pour l'arrière-plan: setBackground (Couleur c)
- Pour le premier plan: setForeground (Couleur c)



```
import javax.swing.JFrame;
public class MyFrame {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Test Frame");
    frame.set …

    JButton jbtOK = new JButton("OK");

     jbtOK.setForeground(
          new Color(255,255,255));
     jbtOK.setBackground(
          new Color(0,0,0));

    frame.add(jbtOK);
  }
}
```

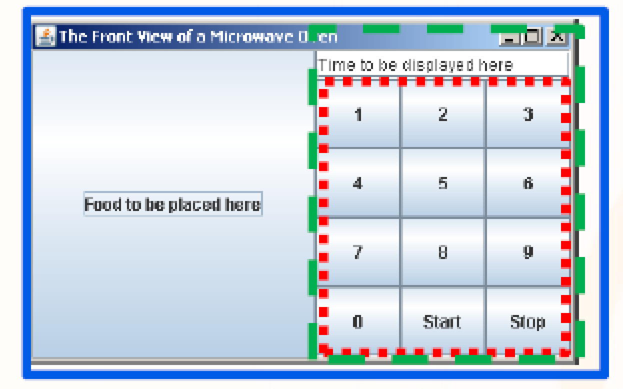


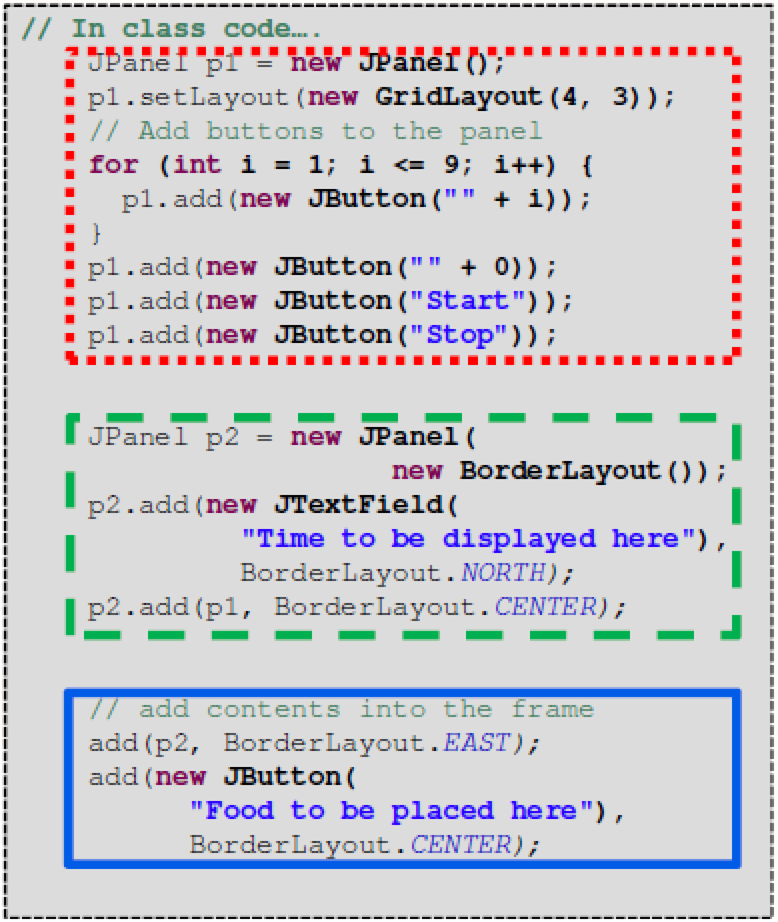
Panels

Panels: sont utilisés comme sous-conteneurs pour organisation

Exemple:

- JPanel p = nouveau JPanel ();
- p.add (nouveau JButton ("OK"));

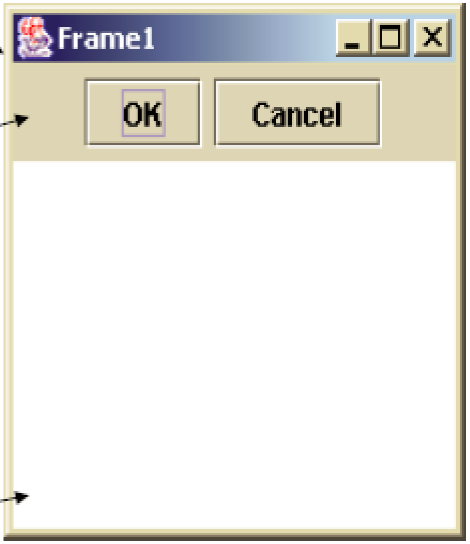

The Front View of a Microwave O..en
Time to be displayed here
1
2
3
4
5
6
7
8
9
0
Start
Stop
Food to be placed here



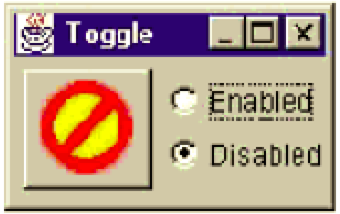
```
// In class code….
    JPanel p1 = new JPanel();
    p1.setLayout(new GridLayout(4, 3));
    // Add buttons to the panel
    for (int i = 1; i <= 9; i++) {
      p1.add(new JButton("" + i));
    }
    p1.add(new JButton("" + 0));
    p1.add(new JButton("Start"));
    p1.add(new JButton("Stop"));

    JPanel p2 = new JPanel(
                      new BorderLayout());
    p2.add(new JTextField(
                "Time to be displayed here"),
                BorderLayout.NORTH);
    p2.add(p1, BorderLayout.CENTER);

    // add contents into the frame
    add(p2, BorderLayout.EAST);
    add(new JButton(
          "Food to be placed here"),
          BorderLayout.CENTER);
```

Exemple – Frame,Panel,Button, Layout

```
JFrame f = new JFrame(“Frame1”);



JPanel p1 = new JPanel( );
JButton b1 = new JButton(“OK”);
JButton b2 = new JButton(“Cancel”);
p1.add(b1);
p1.add(b2);
JTextArea t=new JTextArea("");
f.getContentPane().setLayout(new BorderLayout());
f.getContentPane().add(p1,BorderLayout.NORTH);
f.getContentPane().add(t,BorderLayout.CENTER);
```



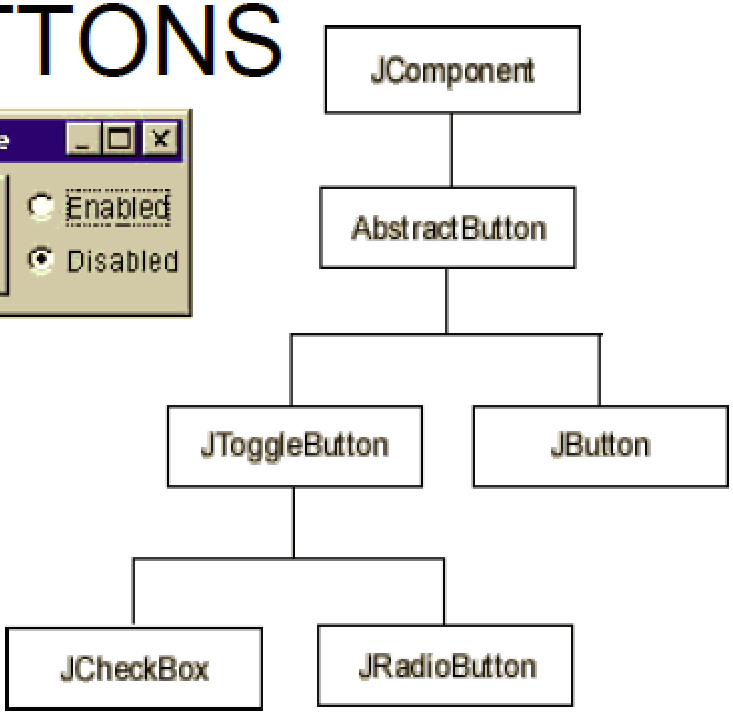
BUTTONS



Toggle

○ Enabled

● Disabled



JComponent

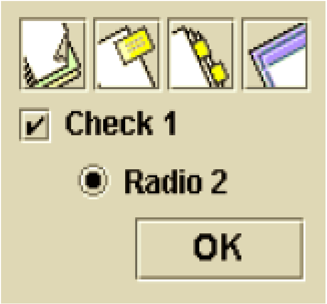
AbstractButton

JToggleButton

JButton

JCheckBox

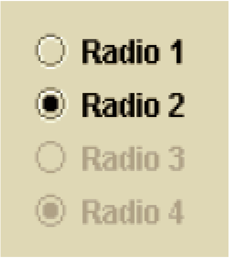
JRadioButton



☑ Check 1

● Radio 2

OK

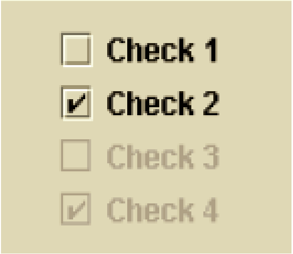


○ Radio 1

● Radio 2

○ Radio 3

● Radio 4



□ Check 1

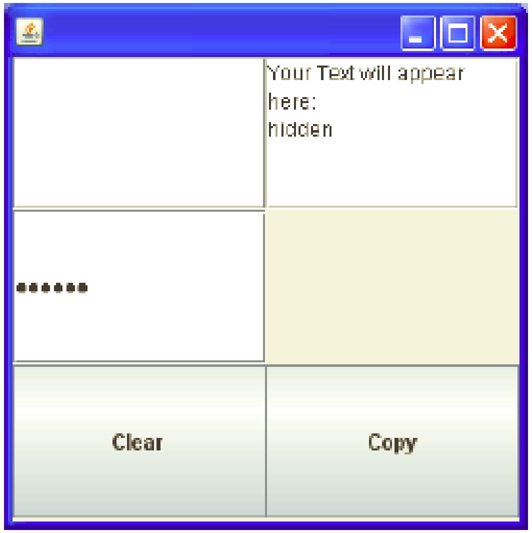
☑ Check 2

□ Check 3

☑ Check 4

Buttons

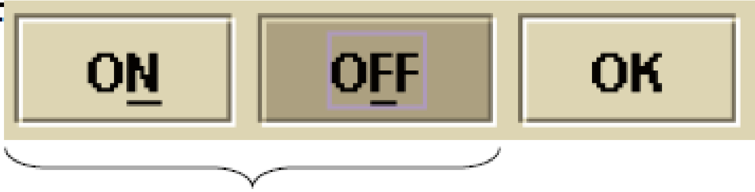
```
  ….
  ta.append(password);
    clearButton=new JButton("Clear");
    copyButton = new JButton("Copy");
    this.getContentPane().add(new JPanel());
    this.getContentPane().add(clearButton);
    this.getContentPane().add(copyButton);



    this.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
```



Your Text will appear
here:
hidden

••••••

```
JButton b= new JButton(“OK”);
JToggleButton tb1= new JToggleButton(“ON”);
JToggleButton tb2= new JToggleButton(“OFF”);
ButtonGroup buttonGroup = new ButtonGroup();
tb1.setMnemonic(‘n’); //  Activation            ALT+n
tb1.setToolTipText(“This is the ON button”);
tb2.setMnemonic(‘f’);  //  Activation           ALT+f
tb1.setToolTipText(“This is the OF
buttonGroup.add(tb1);
buttonGroup.add(tb2);
myPanel.add(tb1);
myPanel.add(tb2);
myPanel.add(b);
```



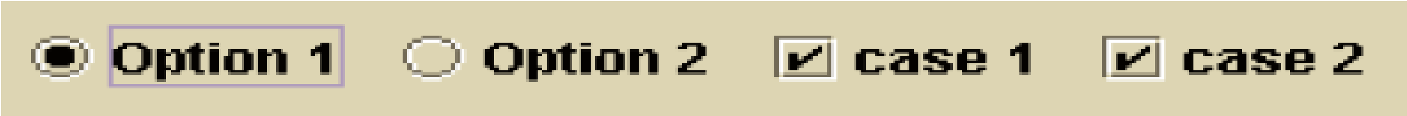
JCheckBox JRadioButton

JCheckBox **: On peut choisir tous en meme temps**

```
myPanel.add(new JCheckBox("case 1"));
myPanel.add(new JCheckBox("case 2"));
```

JRadioButton **: On peut choisit un / fois**

```
ButtonGroup options = new ButtonGroup( );
JRadioButton rb1= new JRadioButton("Option 1");
JRadioButton rb2= new JRadioButton("Option 2");
options.add(rb1); options.add(rb2);
myPanel.add(rb1); myPanel.add(rb2);
```



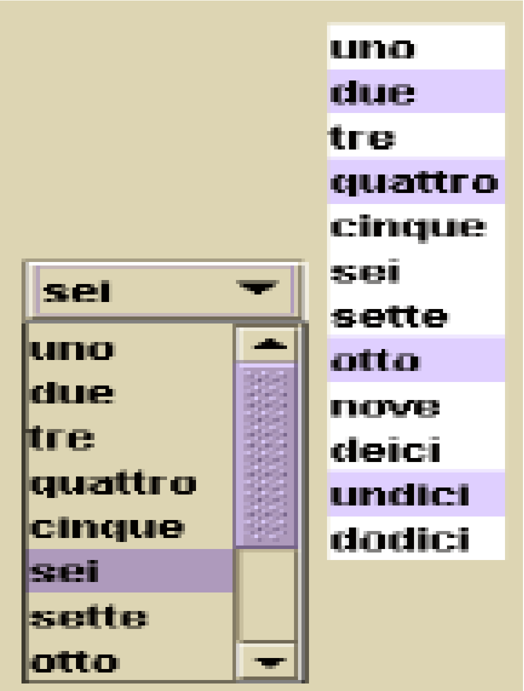
● Option 1 ○ Option 2 ☑ case 1 ☑ case 2

JComboBox

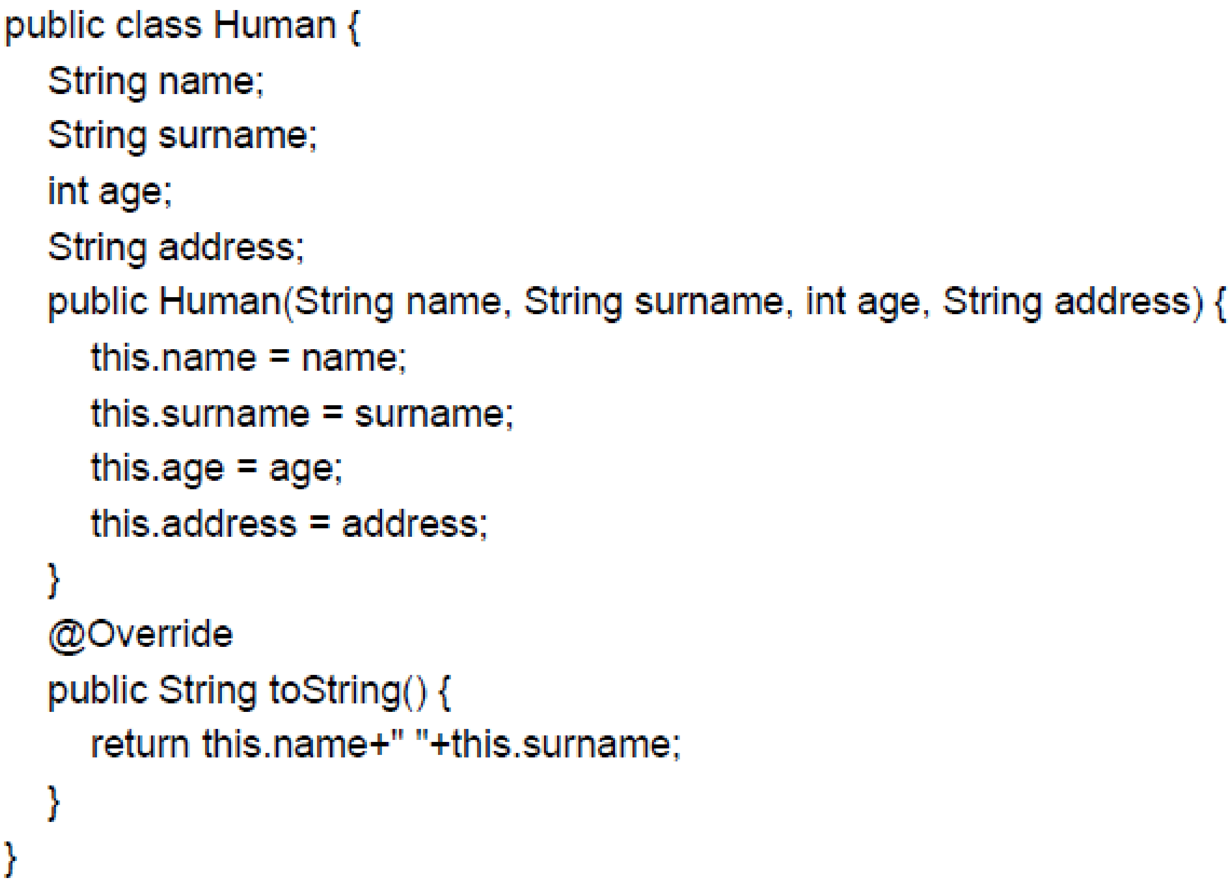
```
String [] items = { "uno", "due", "tre", "quattro", "cinque", "sei", "sette",
    "otto", "nove", "deici", "undici" };
JComboBox comboBox = new JComboBox(items);
comboBox.addItem("dodici");
comboBox.getSelectedItem() //           Object
comboBox.getSelectedObjects() //           Object[]
```

JList

```
JList list = new JList(comboBox.getModel( ));
list.getSelectedValues() //           Object[]
```



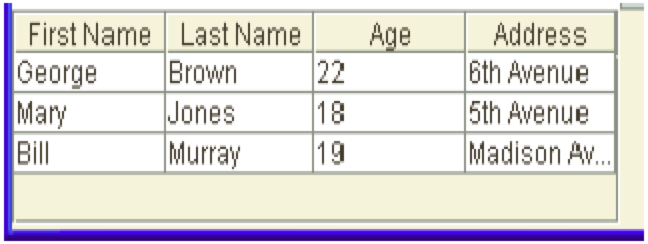
Exemple



```
public class Human {
    String name;
    String surname;
    int age;
    String address;
    public Human(String name, String surname, int age, String address) {
        this.name = name;
        this.surname = surname;
        this.age = age;
        this.address = address;
    }
    @Override
    public String toString() {
        return this.name+" "+this.surname;
    }
}
```

JTable

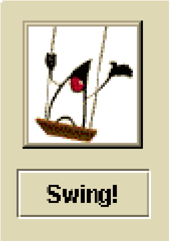
```
String[] columnNames = {"First Name", "Last Name", "Age", "Address"};
Object[][] data = {{"George", "Brown", new Integer(22), "6th Avenue"},
          {"Mary", "Jones", new Integer(18), "5th Avenue"},
          {"Bill", "Murray", new Integer(19), "Madison Avenue"}};
dtm = new DefaultTableModel(data,columnNames);
studentsTable=new JTable(dtm);
JScrollPane jsp2 = new JScrollPane();
jsp2.getViewport().add(studentsTable);
this.getContentPane().add(jsp2);
```



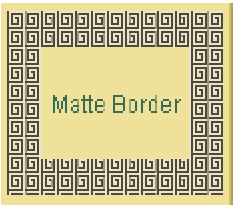
First Name	Last Name	Age	Address
George	Brown	22	6th Avenue
Mary	Jones	18	5th Avenue
Bill	Murray	19	Madison Av...

```
JTable(Object[][] rowData, Object[] columnNames);
    JTable(Vector rowData, Vector columnNames);
               JTable(DefaultTableModel dtm);
```

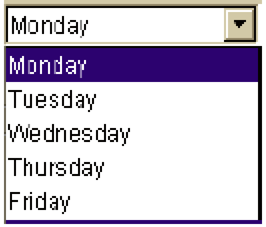
COMPONENTS



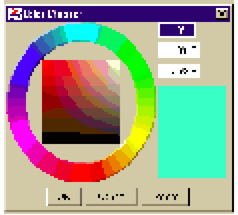
JApplet



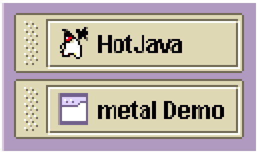
Border Interface



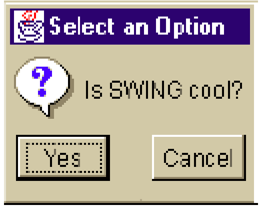
JComboBox



JColorChooser

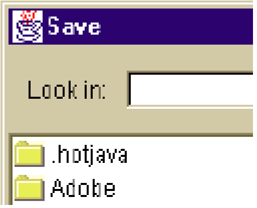


ImageIcon

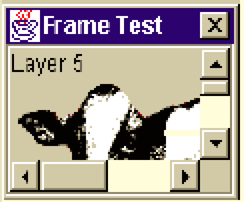


Is SWING cool?

JDialog

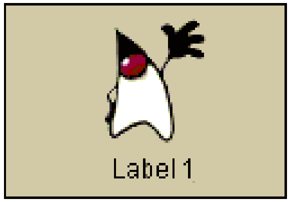


JFileChooser

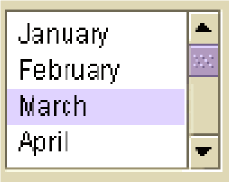


JInternalFrame

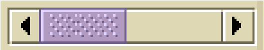
COMPONENTS



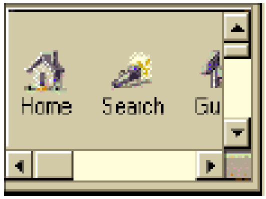
JLabel



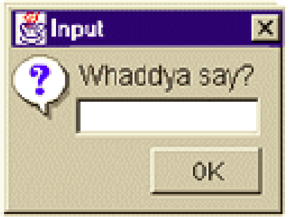
JList



JScrollBar

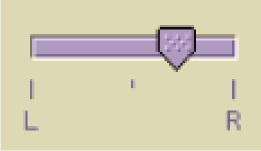


JScrollPane

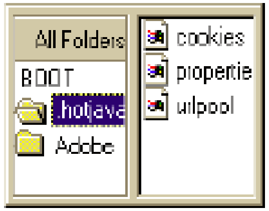


Whaddya say?

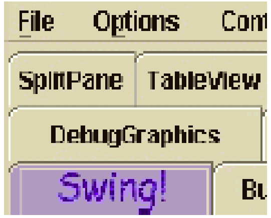
JOptionPane



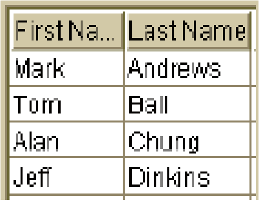
JSlider



JSplitPane

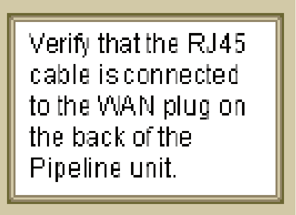


JTabbedPane



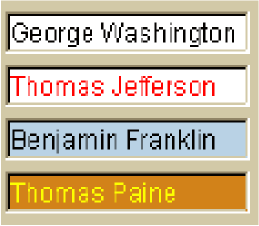
First Na...	Last Name
Mark	Andrews
Tom	Ball
Alan	Chung
Jeff	Dinkins

JTable



Verify that the RJ45 cable is connected to the WAN plug on the back of the Pipeline unit.

JTextArea



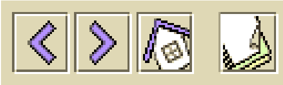
George Washington

Thomas Jefferson

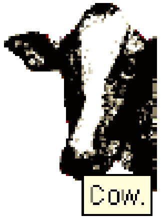
Benjamin Franklin

Thomas Paine

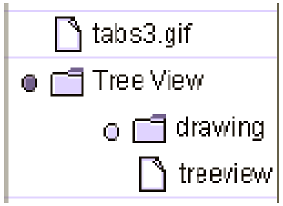
JTextField



JToolBar



JToolTip



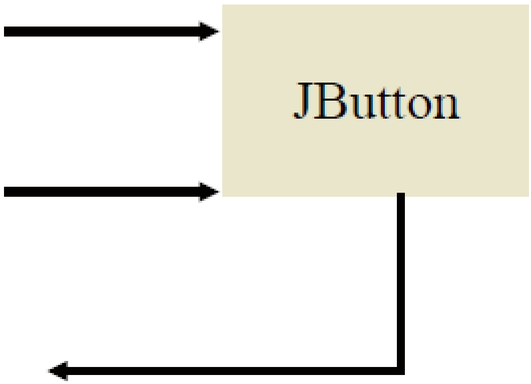
- tabs3.gif
- Tree View
  - drawing
    - treeview

JTree

JComponent

Chaque JComponent est un objet d'une classe.

Par conséquent:
* Il a un etat (propriétés)
(par exemple actif, visible, sélectionné,
position, texte, image, etc.)
* Il a des méthodes
(par exemple, définir le texte / l'image, le fond)
* événements
(par exemple mouseClicked, mouseEntered, keyTyped, componentMoved)



JButton

Comment utiliser Jcomponent?

Creation :
Ex. JButton b = new JButton(“press me”);
Customization :
Properties : b.text = “press me”;

Methodes:
 b.setText(“press me”);

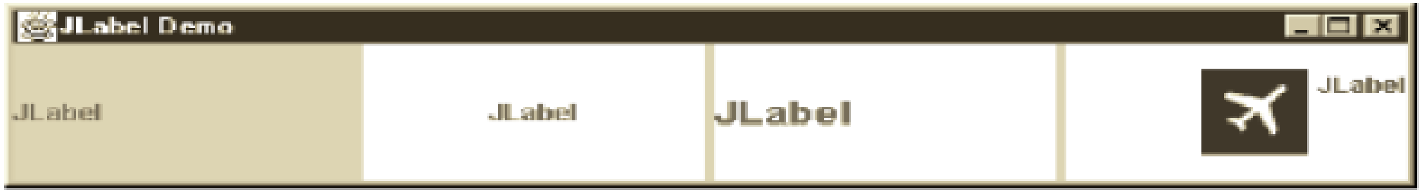
Ajouter components (si c’est container)

On l’ajoute dans un container :
 panel.add(b);

Events: Listeners

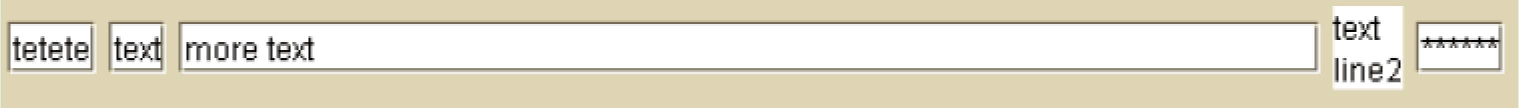
JLabel

```
JLabel label1 = new JLabel(“JLabel");
JLabel label2 = new JLabel("JLabel",SwingConstants.CENTER);
label2.setOpaque(true); label2.setBackground(Color.white);

JLabel label3 = new JLabel("JLabel",SwingConstants.CENTER);
label3.setFont(new Font("Helvetica", Font.BOLD, 18));
label3.setOpaque(true); label3.setBackground(Color.white);

ImageIcon icon = new ImageIcon("image.gif");
JLabel label4 = new JLabel("JLabel",image, SwingConstants.RIGHT);
label4.setVerticalTextPosition(SwingConstants.TOP);
label4.setOpaque(true); label4.setBackground(Color.white);

myPanel.add(label); myPanel.add(label2);
myPanel.add(label3); myPanel.add(label4);
```



JTextField

```
JTextField tf1 = new JTextField();
JTextField tf2 = new JTextField(“text”);
JTextField tf3 = new JTextField(“more text”,40);
tf2.getText();
tf1.setText(“empty”);
JTextArea ta= new TextArea();
ta.append(“text”); ta.append(“\nline2”);
JPasswordField pf=new JPasswordField(“hidden”);
```



tetete

text

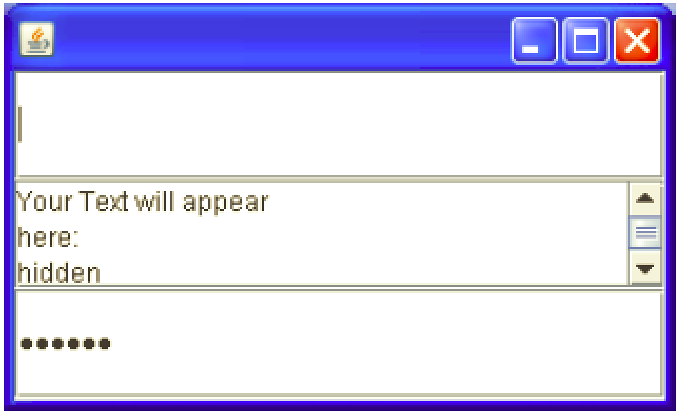
more text

text
line2

******

Exemple

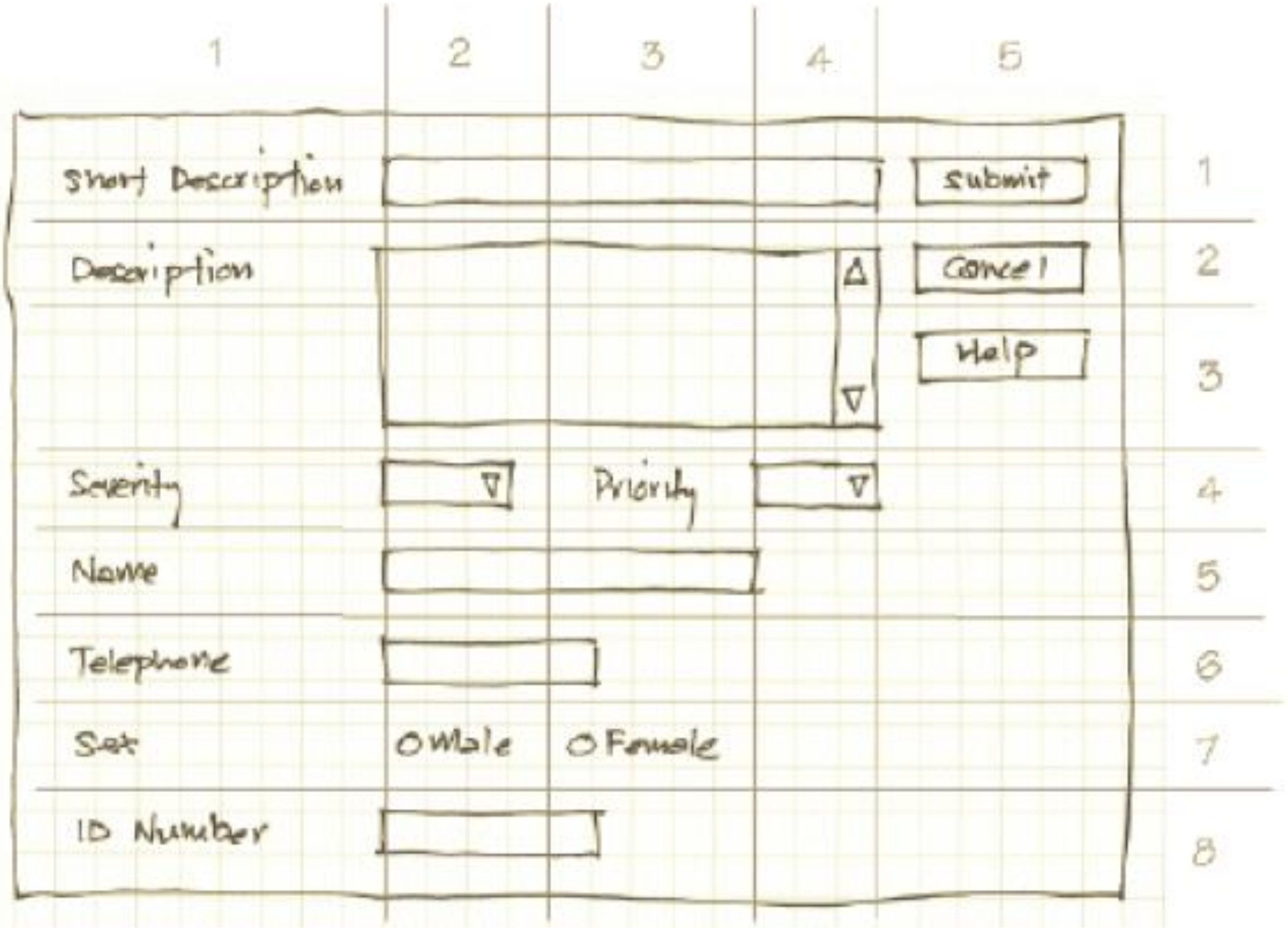
```
public class TestFrame extends JFrame {
    JTextField tf;     JTextArea ta;
    JPasswordField pf;    JScrollPane jsp;
    public TestFrame() {
        tf = new JTextField();   ta = new JTextArea();
        ta.setText("Your Text will appear");
        ta.append("\nhere:\n");
        pf = new JPasswordField("hidden");
        jsp = new JScrollPane();
        jsp.getViewport().add(ta);
        this.setLayout(new GridLayout(3, 3));
        this.getContentPane().add(tf);
        this.getContentPane().add(jsp);
        this.getContentPane().add(pf);
        String password=pf.getText();
        ta.append(password);
     this.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }
}
```

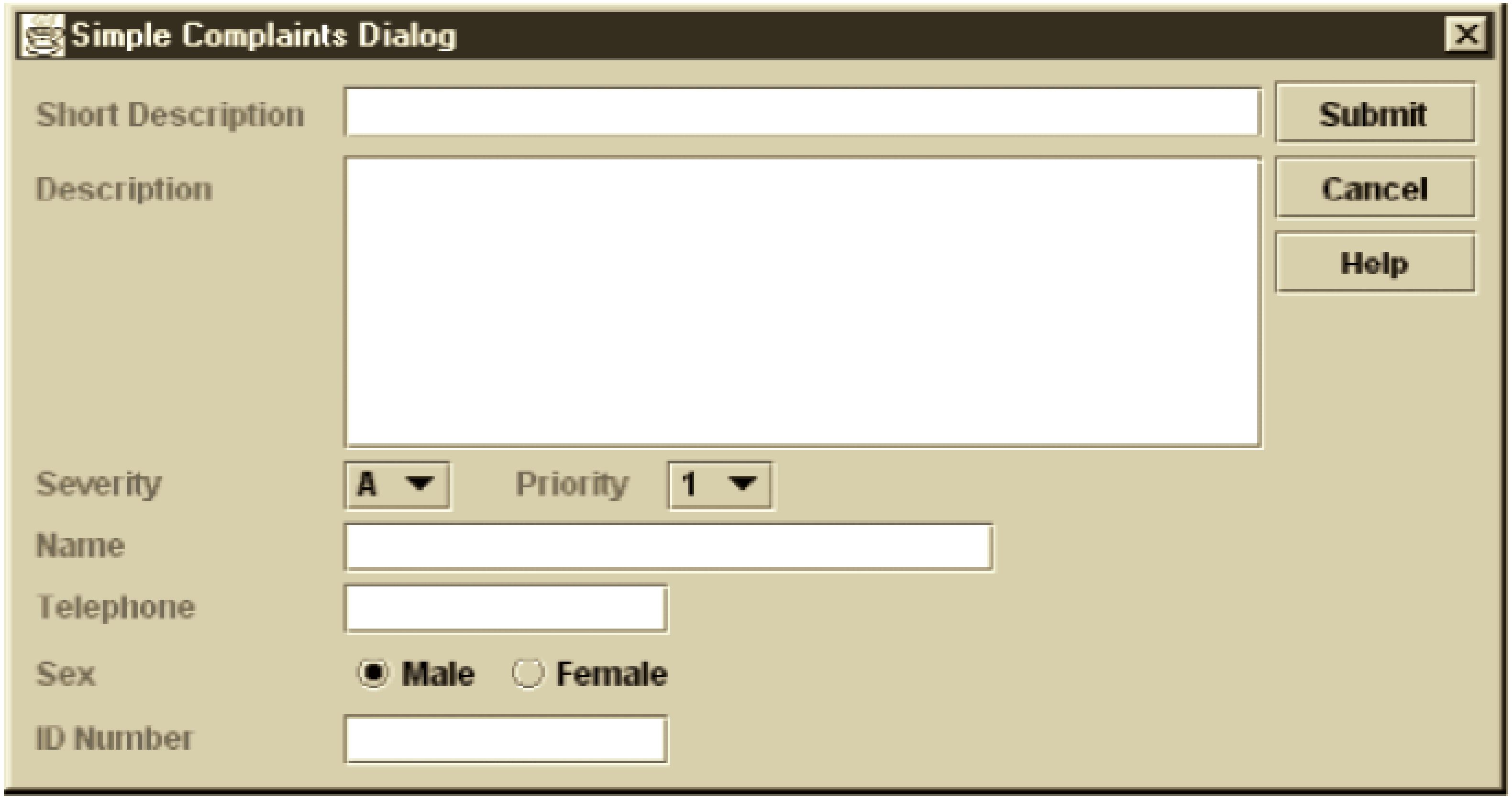


Your Text will appear
here:
hidden

••••••

A vous …


1
2
3
4
5
Short Description
Submit
Description
Cancel
Help
Severity
Priority
Name
Telephone
Sex
○ Male
○ Female
ID Number
1
2
3
4
5
6
7
8



Simple Complaints Dialog

Severity A

Priority 1

Sex ● Male ○ Female