

Programmation Java Avancé

Interfaces Graphiques



1

Interfaces Graphiques

- Introduction
- AWT
- Gestion des Evènements
- Swing

Interfaces Graphiques

I- Introduction

- Les programmes informatiques nécessitent une interaction avec leurs utilisateurs pour :
 - Entrée des données par l'utilisateur au moment de l'exécution
 - Affichage d'une partie des résultats obtenus par le traitement informatique
 - ...
- Cet échange d'informations peut s'effectuer avec console ou en mode graphique

Interfaces Graphiques

I- Introduction

- Une interface graphique (GUI) est formée d'une ou plusieurs fenêtres qui contiennent divers composants graphiques (widgets) tels que :
 - Boutons
 - Listes déroulantes
 - Menus
 - Champ texte
 - ...

Interfaces Graphiques

I- Introduction

- Pour créer des interfaces graphiques Java propose des API
 - AWT
 - Swing
 - SWT (Pour Créé par IBM pour Eclipse)
 - JFace

Interfaces Graphiques

II- API AWT(Abstract Window Toolkit)

Interfaces Graphiques

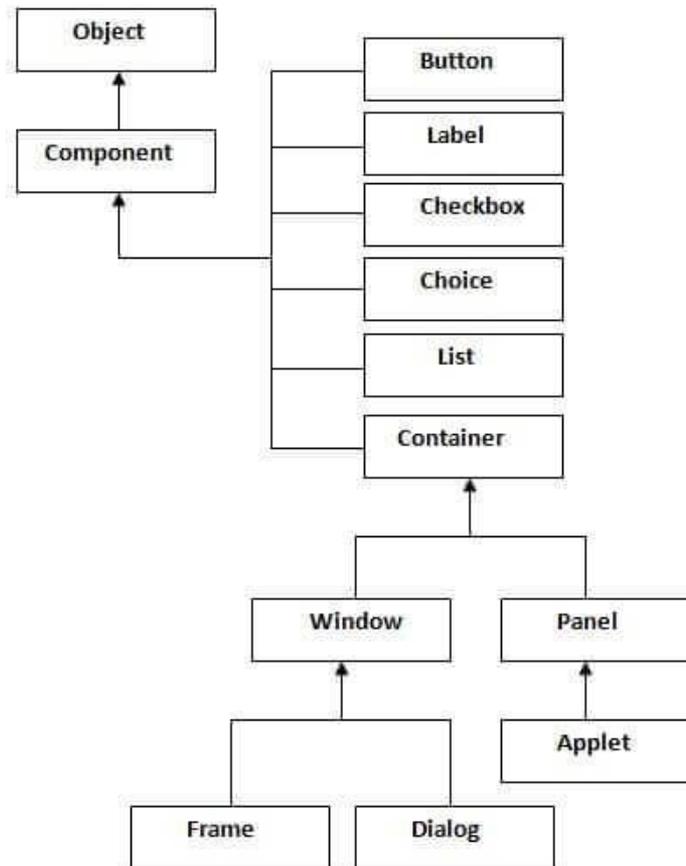
II- AWT

- Java AWT (Abstract Window Toolkit) est une API permettant de développer des applications GUI sous java.
- Les composants Java AWT dépendent de la plate-forme, c'est-à-dire que les composants sont affichés en fonction de la vue du système d'exploitation. AWT est **lourd**, par ce que ses composants utilisent les ressources du système d'exploitation.

Interfaces Graphiques

II- AWT

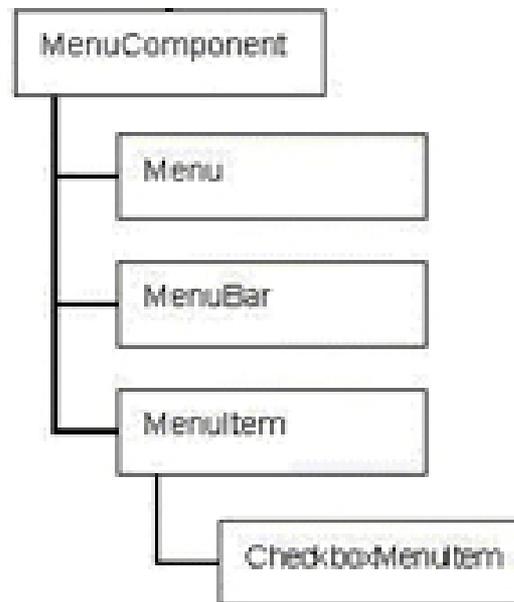
1- Hiérarchie des classes



Interfaces Graphiques

II- AWT

1- Hiérarchie des classes



Interfaces Graphiques

II- AWT

2- Composant de la fenêtre



Interfaces Graphiques

II- AWT

3- Ma première fenêtre

- Les Principales méthodes de la classe component

Method	Description
<code>public void add(Component c)</code>	Ajoute un composant dans un autre
<code>public void setSize(int larg,int haut)</code>	Change la largeur et la hauteur
<code>public void setLayout(LayoutManager m)</code>	Définie layout manager pour un composant
<code>public void setVisible(boolean status)</code>	Change la visibilité d'un composant (par défaut false)
<code>public void setBounds(int x, int y, int la, int haut)</code>	Change les position (x,y) , la largeur et la hauteur d'un composant

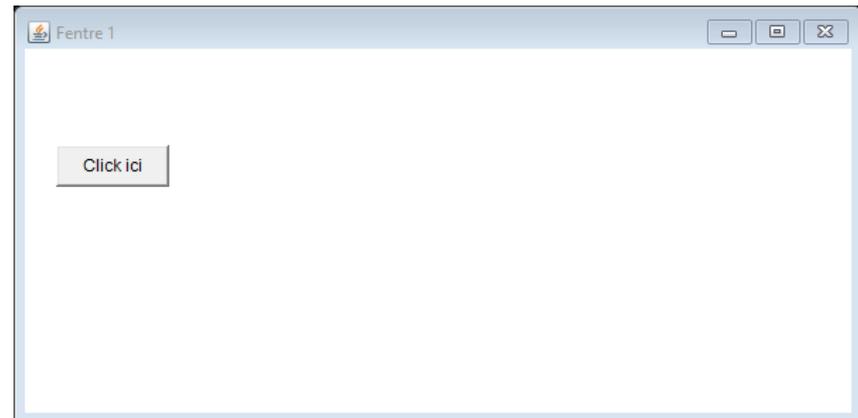
Interfaces Graphiques

II- AWT

3- Ma première fenêtre

- Héritage de la classe Frame

```
3
4 import java.awt.*;
5 class Fenetre1 extends Frame{
6
7     Fenetre1(){
8
9         setTitle("Fentre 1");
10        Button b=new Button("Click ici");
11        b.setBounds(30,100,80,30);
12        add(b);
13
14        setSize(600,300);
15        setLayout(null);
16        setVisible(true);
17    }
18
19    public static void main(String args[]){
20        Fenetre1 f=new Fenetre1();
21    }
22 }
```



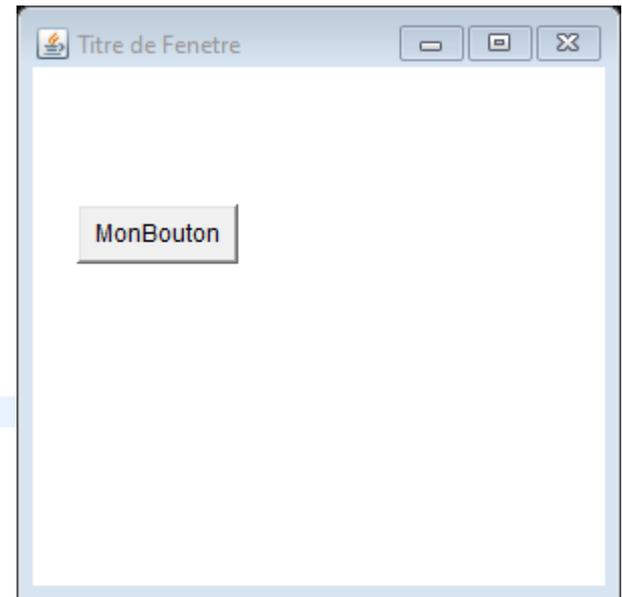
Interfaces Graphiques

II- AWT

3- Ma première fenêtre

- *Instanciation de la classe Frame*

```
3
4 import java.awt.*;
5 class Fenetre2 {
6
7     Fenetre2(){
8         Frame fen=new Frame("Titre de Fenetre");
9         Button b=new Button("MonBouton");
10        b.setBounds(30,100,80,30);
11
12        fen.add(b);
13
14
15        fen.setSize(300,300);
16        fen.setLayout(null);
17        fen.setVisible(true);
18    }
19
20    public static void main(String args[]){
21        Fenetre2 f=new Fenetre2();
22    }
23 }
```



Interfaces Graphiques

II- AWT

3- Ma première fenêtre

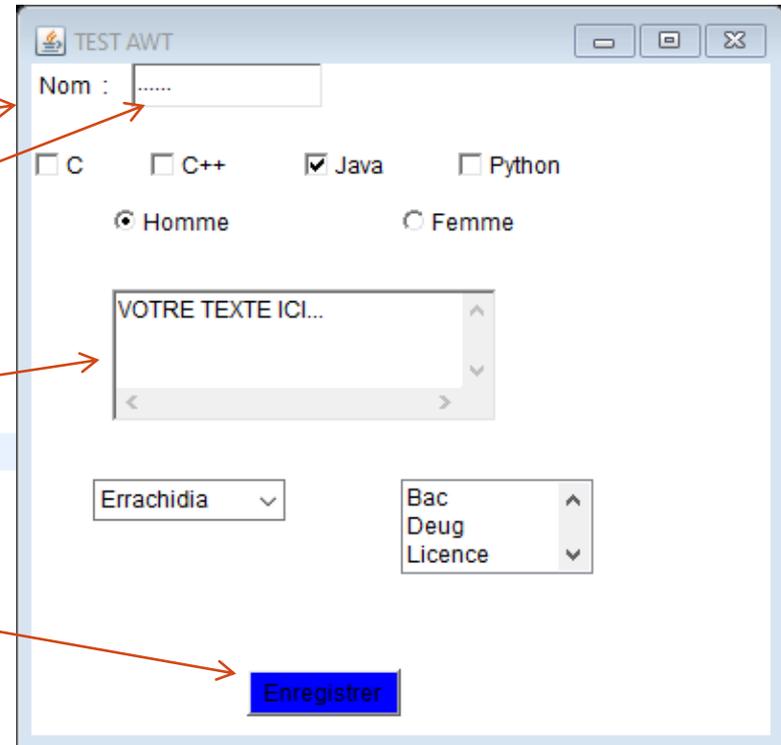
Création des composants

```
Label lab=new Label("Nom : ");  
lab.setBounds(10, 30, 50, 25);
```

```
TextField tf=new TextField(".....");  
tf.setBounds(60, 30, 100, 25);
```

```
TextArea zon=new TextArea("VOTRE TEXTE ICI...");  
zon.setBounds(50, 150, 200, 70);
```

```
Button b=new Button("Enregistrer");  
b.setBackground(Color.BLUE);  
b.setBounds(120,350,80,25);
```



Interfaces Graphiques

II- AWT

3- Ma première fenêtre

Création des composants

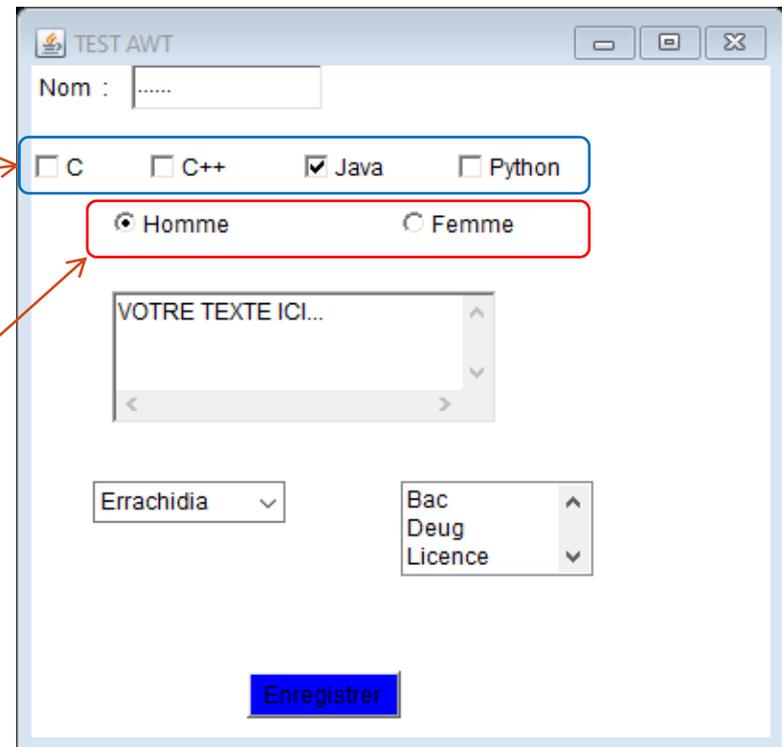
```
////Choix multiple
Checkbox chb1=new Checkbox("C");
Checkbox chb2=new Checkbox("C++");
Checkbox chb3=new Checkbox("Java",true);
Checkbox chb4=new Checkbox("Python");

chb1.setBounds(10, 70, 50, 30);
chb2.setBounds(70, 70, 50, 30);
chb3.setBounds(150, 70, 50, 30);
chb4.setBounds(230, 70, 60, 30);

//////////Choix unique
CheckboxGroup g1=new CheckboxGroup();

Checkbox choi1=new Checkbox("Homme",g1,true);
Checkbox choi2=new Checkbox("Femme",g1,false);

choi1.setBounds(50, 100, 100, 30);
choi2.setBounds(200, 100, 100, 30);
```



Interfaces Graphiques

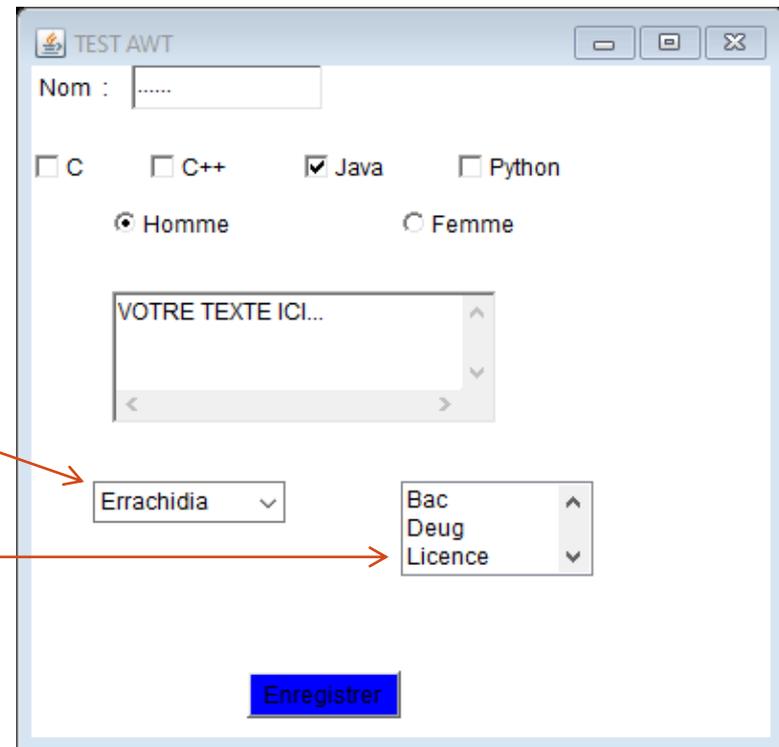
II- AWT

3- Ma première fenêtre

Création des composants

```
////////////////////liste deroulante
Choice c=new Choice();
c.setBounds(40, 250, 100, 50);
c.add("Errachidia");
c.add("Tinghir");
c.add("Rich");
c.add("Rabat");
c.add("Tanger");

////////////////////liste
List l1 =new List();
l1.setBounds(200, 250, 100, 50);
l1.add("Bac");
l1.add("Deug");
l1.add("Licence");
l1.add("Master");
l1.add("Doctorat");
```



Interfaces Graphiques

II- AWT

3- Ma première fenêtre

- Ajouter tous les composants créés dans la fenêtre

```
fen.add(l1);  
fen.add(c);  
fen.add(choi1);fen.add(choi2);  
fen.add(chb1);fen.add(chb2);fen.add(chb3);fen.add(chb4);  
fen.add(tf);  
fen.add(zon);  
fen.add(lab);  
fen.add(b);
```

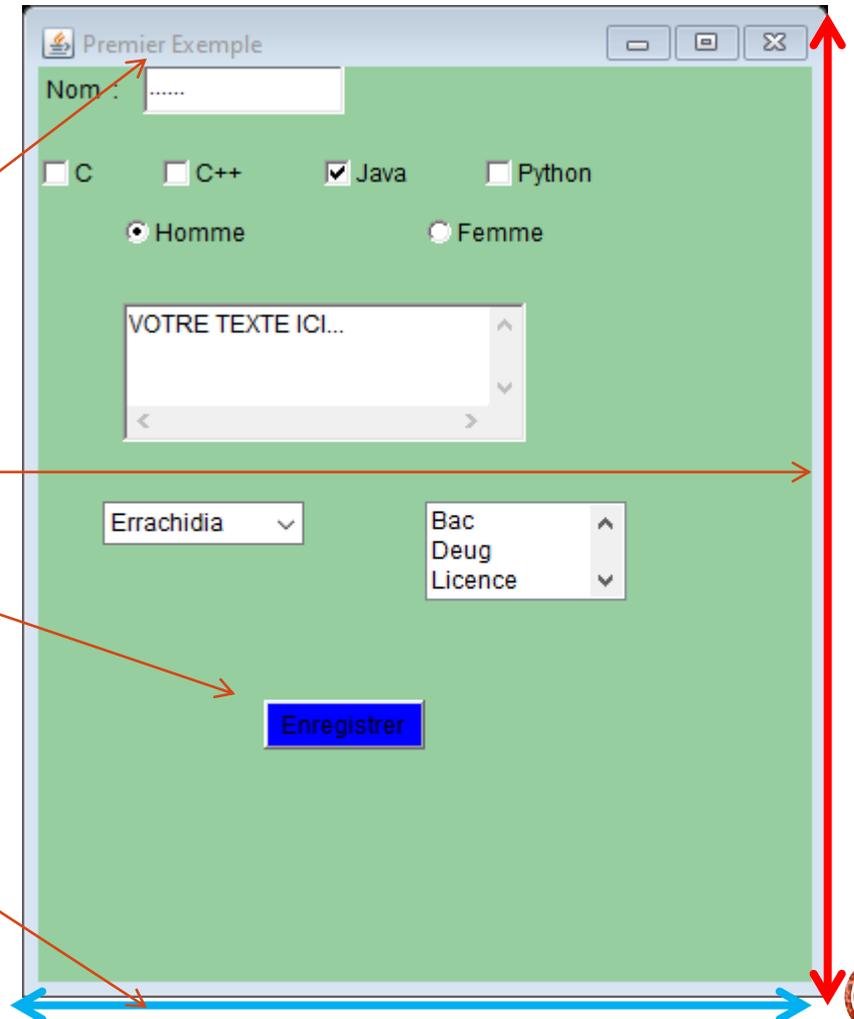
Interfaces Graphiques

II- AWT

3- Ma première fenêtre

- Personnaliser votre fenêtre

```
fen.setTitle("Premier Exemple");  
fen.setSize(400,500);  
fen.setBackground(new Color(150,206,159));  
fen.setLayout(null);  
fen.setVisible(true);
```



Interfaces Graphiques

II- AWT

4- Utilisation de Panel

- Le Panel est une classe de conteneur la plus simple. Il fournit un espace dans lequel une application peut attacher n'importe quel autre composant.

```
Frame f= new Frame("Exemple avec panel");
```

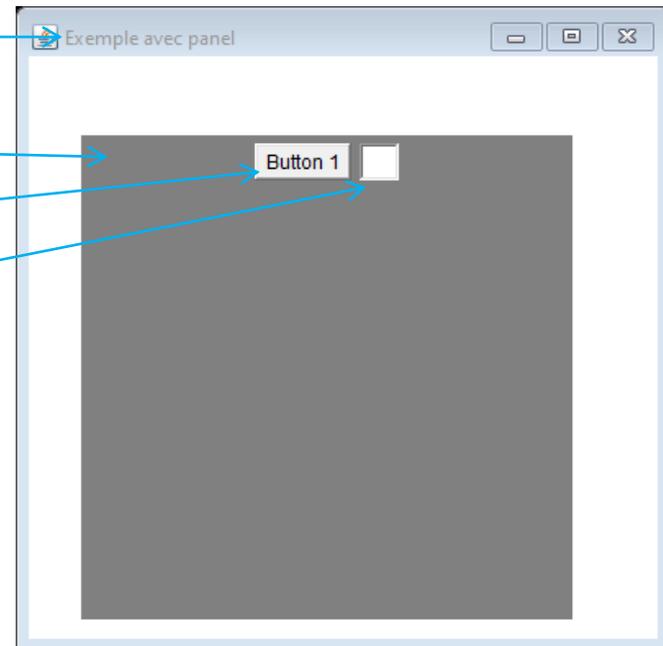
```
Panel panel=new Panel();  
panel.setBounds(40,80,300,300);  
panel.setBackground(Color.gray);
```

```
Button b1=new Button("Button 1");  
b1.setBounds(50,100,80,30);
```

```
TextField txt=new TextField();  
txt.setBounds(50, 200, 100, 30);
```

```
panel.add(b1);  
panel.add(txt);
```

```
f.add(panel);  
f.setSize(400,400);  
f.setLayout(null);  
f.setVisible(true);
```

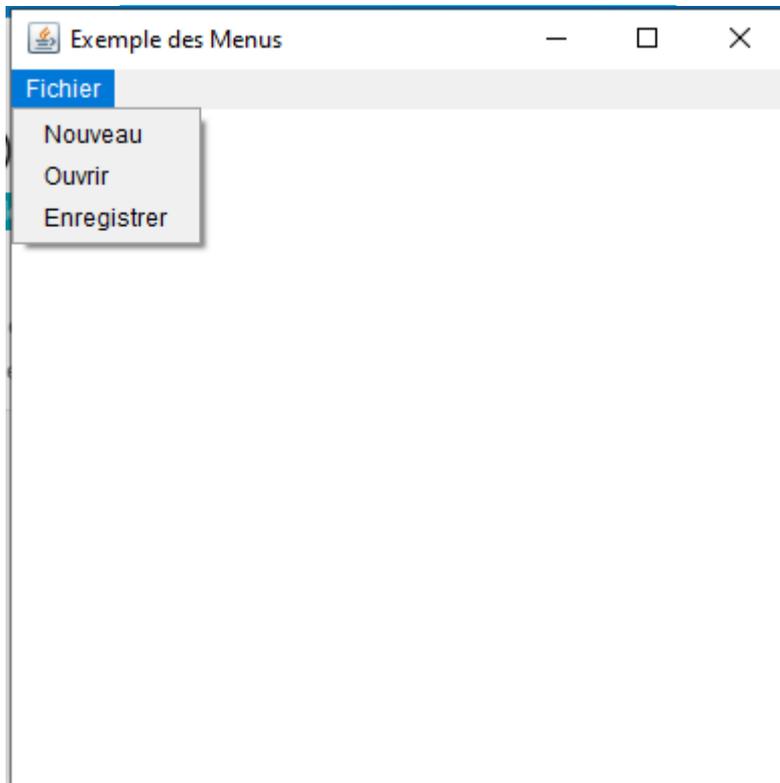


Interfaces Graphiques

II- AWT

5- Les Menus

- Les étapes pour créer un Menu



```
MenuBar barre=new MenuBar();
```

```
Menu menu1=new Menu("Fichier");
```

```
MenuItem i1=new MenuItem("Nouveau");  
MenuItem i2=new MenuItem("Ouvrir");  
MenuItem i3=new MenuItem("Enregistrer");
```

```
menu1.add(i1);  
menu1.add(i2);  
menu1.add(i3);
```

```
barre.add(menu1);
```

```
fen.setMenuBar(barre);
```

Interfaces Graphiques

II- AWT

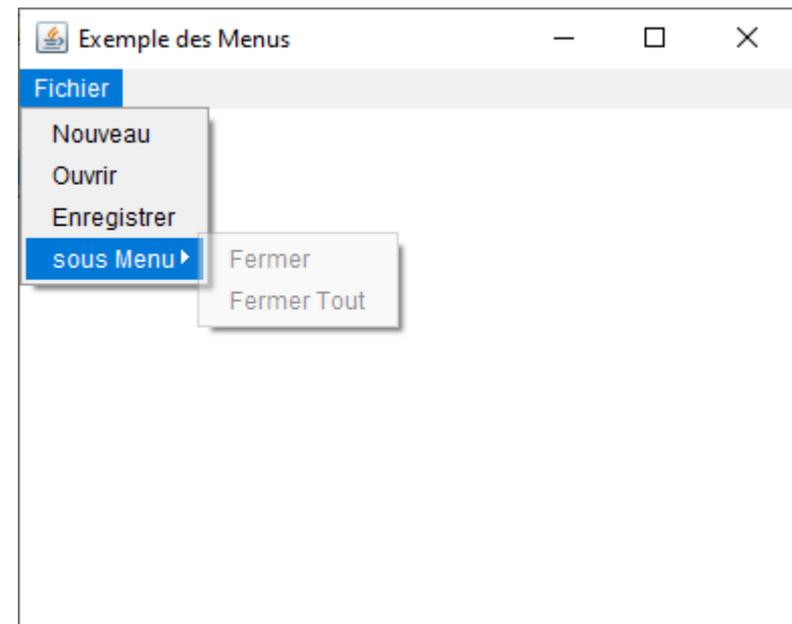
5- Les Menus

- Sous-menu

```
//////////creation d'un sous-menu
//1- Créer un Menu (sous-menu)
Menu sousmenu=new Menu("sous Menu");

//2- Créer les éléments de ce- sous-menu
MenuItem is1=new MenuItem("Fermer");
MenuItem is2=new MenuItem("Fermer Tout");
//3- Ajouter ces éléments au sous-menu
sousmenu.add(is1);
sousmenu.add(is2);

//4-Ajouter ce sous-menu dans un menu Existant
menu1.add(sousmenu);
/////Fin de sous-menu
```



III- Gestions des événements

Interfaces Graphiques

III- Gestions des événements

1- Interfaces des écouteurs « Listeners »

- Pour gérer les événements et les actions Java dispose de plusieurs interface nommées 'Listeners' (écouteurs) :

- ActionListener → `public void actionPerformed(ActionEvent e);`
- MouseListener → `public void mouseClicked(MouseEvent e);`
- MouseMotionListener → `public void mouseEntered(MouseEvent e);`
`public void mouseExited(MouseEvent e);`
- ItemListener → `public abstract void mouseDragged(MouseEvent e);`
`public abstract void mouseMoved(MouseEvent e);`
- KeyListener → `public void windowActivated(WindowEvent e);`
`public void itemStateChanged(ItemEvent e);`
- TextListener → `public void windowClosed(WindowEvent e);`
`public void keyPressed(KeyEvent e);`
- WindowListener → `public void windowDeactivated(WindowEvent e);`
`public void windowDeiconified(WindowEvent e);`
`public Text valueChanged(TextEvent e);`
`public void windowIconified(WindowEvent e);`
`public void windowOpened(WindowEvent e);`

Interfaces Graphiques

III- Gestions des événements

2- Trois étapes à suivre

- Pour gérer les événements et les actions Java, il faut suivre 3 étapes
- 1) Implémenter l'interface selon l'événement
 - Exemple : `public class Test implements ActionListener { ... }`
- 2) Enregistrez le composant auprès de Listener (Ecouleur)
 - Exemple : `bouton1.addActionListener(this)`
- 3) Ecrire le code de la méthode déclenchée par l'événement
 - Exemple : `public void actionPerformed(ActionEvent e){`
 - `System.out.print("Vous avez cliquer sur Bouton1");`
 - `}`

Interfaces Graphiques

III- Gestions des événements

3- Exemple 1 :

```
import java.awt.event.*;
import java.awt.*;

public class TestEvenement implements ActionListener{

    ///////////////////////////////////////////////////Constructeur//////////////////////////////////////
    public TestEvenement() {

        Frame fen=new Frame("Azrou");

        Button bt=new Button("Click");
        bt.setBounds(100, 50,100 , 25);
        bt.addActionListener(this);

        fen.add(bt);
        fen.setLayout(null);
        fen.setSize(500, 400);
        fen.setVisible(true);
    }

    ///////////////////////////////////////////////////Gestion des événements//////////////////////////////////////
    public void actionPerformed(ActionEvent ev) {
        System.out.println("Evenement Capturé...");
    }

    ///////////////////////////////////////////////////MATH//////////////////////////////////////
}
```

Interfaces Graphiques

III- Gestions des événements

3- Exemple 2 :

```
public class TestEvenement implements MouseListener{
    ////////////////////////////////////Constructeur////////////////////////////////////
    public TestEvenement() {
        Frame fen=new Frame("Azrou");
        Button bt=new Button("Click");
        bt.setBounds(100, 50,100 , 25);
        bt.addMouseListener(this);
        fen.add(bt);
        fen.setLayout(null);
        fen.setSize(500, 400);
        fen.setVisible(true);
    }
    ////////////////////////////////////Gestion des événements////////////////////////////////////
    @Override
    public void mouseClicked(MouseEvent e) {
        // TODO Auto-generated method stub
        System.out.println("tu as cliqué");
    }
    @Override
    public void mouseEntered(MouseEvent e) {
        // TODO Auto-generated method stub
        System.out.println("Entrer");
    }
    @Override
    public void mouseExited(MouseEvent e) {
        // TODO Auto-generated method stub
        System.out.println("quitter");
    }
}
```

Interfaces Graphiques

III- Gestions des événements

3- Exemple 3 :

```
Button bt_en=new Button("Enregistrer");
bt_en.setBounds(25, 50,100 , 25);
bt_en.addActionListener(this);

Button bt_ou=new Button("Ouvrir");
bt_ou.setBounds(150, 50,100 , 25);
bt_ou.addActionListener(this);
```

```
////////////////////////////////////Gestion des événements////////////////////////////////////
public void actionPerformed(ActionEvent ev) {
    System.out.println("Evenement Capturé...");

    if(ev.getActionCommand().equals("Enregistrer")) {
        txt.setText("tu as cliqué sur Enregistrer ");
    }else if(ev.getActionCommand().equals("Ouvrir")) {
        txt.setText("tu as cliqué sur Ouvrir");
    }
}
.....
```

Interfaces Graphiques

- IV- API Swing

Interfaces Graphiques

IV- Swing

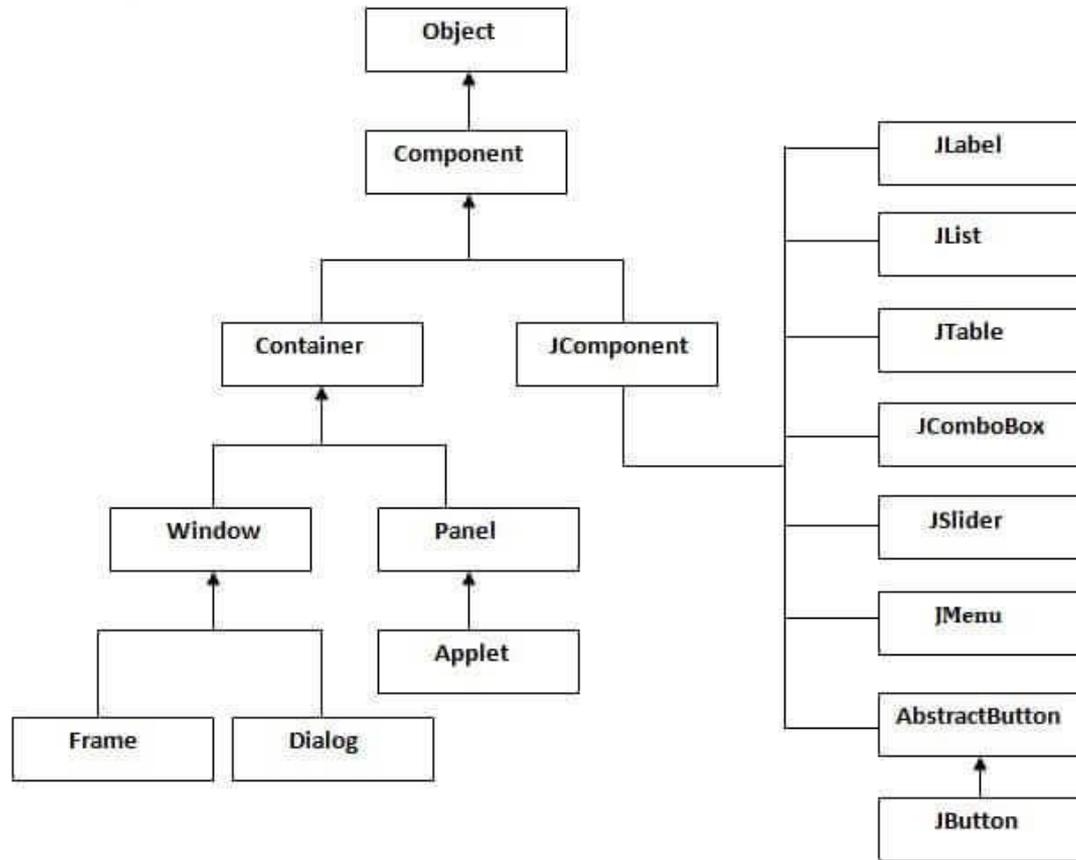
L'API **Swing** est utilisée pour créer des applications basées sur des fenêtres. Il est construit basant sur de l'API AWT. Elle est entièrement écrite en java.

Contrairement à AWT, Java Swing fournit des composants légers et indépendants de la plate-forme.

Le package **javax.swing** fournit des classes pour l'API Swing telles que:
JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser ...

Interfaces Graphiques

IV- Swing



Interfaces Graphiques

IV- Swing

1- Méthodes plus Utilisées :

```
public void add(Component c)
```

```
public void setSize(int width,int height)
```

```
public void setLayout(LayoutManager m)
```

```
public void setVisible(boolean b)
```

```
public void setBounds(int x,int y, int width,int height)
```

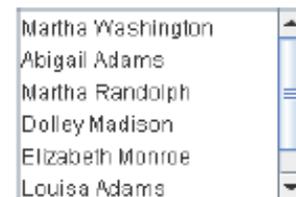
Interfaces Graphiques

IV- Swing

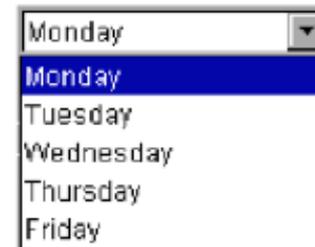
2- Les Composants standards:



[JLabel](#)



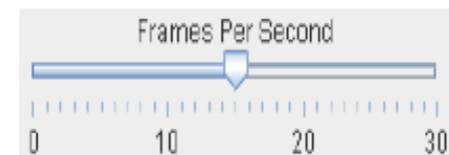
[JList](#)



[JComboBox](#)



[JScrollBar](#)



[JSlider](#)



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

a- Etiquette

JLabel()
JLabel(String s)
JLabel(Icon i)
JLabel(String s, Icon i, int horizontalAlignment)

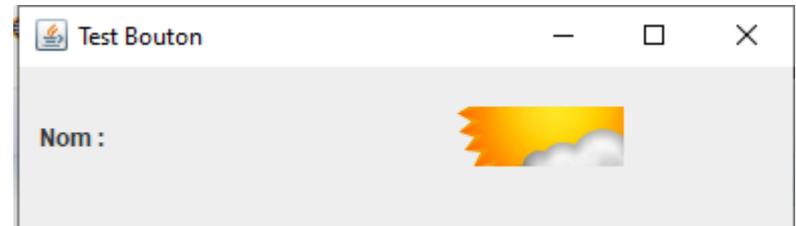
Exemple

```
JFrame f=new JFrame("Test Bouton");

JLabel lab1=new JLabel("Nom :");
lab1.setBounds(10, 20, 100, 30);

JLabel lab2=new JLabel("Label2",new ImageIcon("D:\\figures\\44.png"),2);
lab2.setBounds(200, 20, 100, 30);

f.add(lab1);
f.add(lab2);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

b- Bouton

JButton()

JButton(String)

JButton(Icon i)

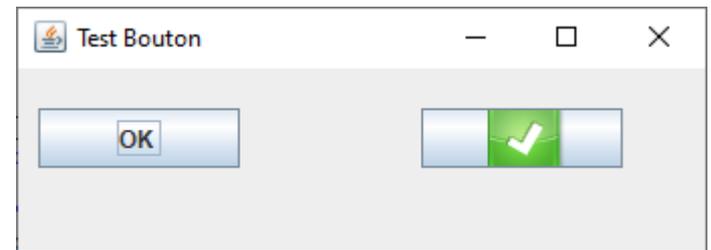
Exemple

```
JFrame f=new JFrame("Test Bouton");

JButton b1=new JButton("OK ");
b1.setBounds(10, 20, 100, 30);

JButton b2=new JButton(new ImageIcon("D:\\figures\\Checkmark.png"));
b2.setBounds(200, 20, 100, 30);

f.add(b1);
f.add(b2);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

c- Zones de texte :

```
JFrame f=new JFrame("Zone des textes");

JLabel lab1=new JLabel("Login");
JLabel lab2=new JLabel("Mot de passe");
JLabel lab3=new JLabel("Adresse :");

lab1.setBounds(10, 20, 100, 30);
lab2.setBounds(10, 70, 100, 30);
lab3.setBounds(10, 120, 100, 30);

JTextField nm=new JTextField();
JPasswordField pass=new JPasswordField("");
JTextArea adr=new JTextArea();

nm.setBounds(150, 20, 100, 30);
pass.setBounds(150, 70, 100, 30);
adr.setBounds(150, 120, 200, 200);
```



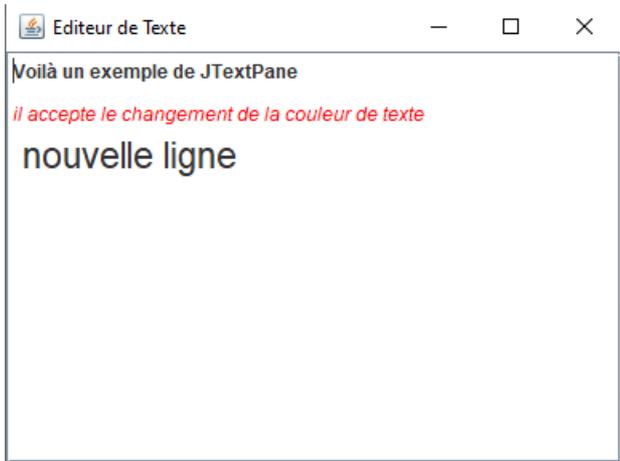
Interfaces Graphiques

IV- Swing

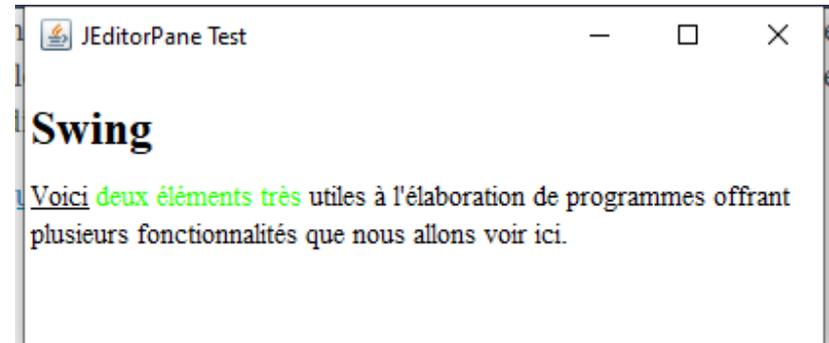
2- Les Composants standards:

c- Zones de texte :

JTextPane



JEditorPane



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

d- Case à cocher

JCheckBox()

JCheckBox(String s)

JCheckBox(String text, boolean selected)

Exemple

```
JCheckBox checkBox1 = new JCheckBox("C++");  
JCheckBox checkBox2 = new JCheckBox("Java", false);  
JCheckBox checkBox3 = new JCheckBox("Python", true);
```

```
checkBox1.setBounds(10,30, 50,50);  
checkBox2.setBounds(150,30, 100,50);  
checkBox3.setBounds(300,30, 120,50);
```

```
f.add(checkBox1);  
f.add(checkBox2);  
f.add(checkBox3);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

e- Bouton radio

```
JFrame f= new JFrame("CheckBox Example");

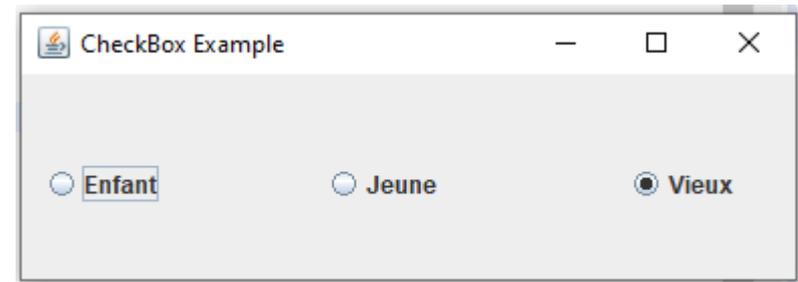
JRadioButton ch1 = new JRadioButton("Enfant");
JRadioButton ch2 = new JRadioButton("Jeune", false);
JRadioButton ch3 = new JRadioButton("Vieux", true);

ch1.setBounds(10,30, 120,50);
ch2.setBounds(150,30, 100,50);
ch3.setBounds(300,30, 120,50);

ButtonGroup gp=new ButtonGroup();
gp.add(ch1);
gp.add(ch2);
gp.add(ch3);

f.add(ch1);
f.add(ch2);
f.add(ch3);

f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

f- Les listes : JComboBox

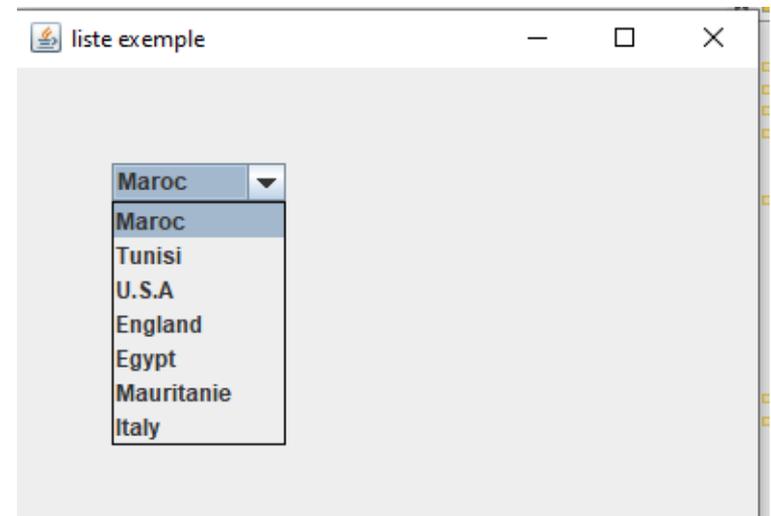
JComboBox()

JComboBox(Object[] items)

JComboBox(Vector<?> items)

Exemple

```
f=new JFrame("liste exemple");  
String py[]={ "Maroc", "Tunisi", "U.S.A", "England", "Egypt", "Mauritanie"};  
JComboBox cb=new JComboBox(py);  
cb.addItem("Italy");  
cb.setBounds(50, 50, 90, 20);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

g- Les listes: JList

JList()

JList(ary[] listData)

JList(ListModel<ary> dataModel)

Exemple

```
f=new JFrame("liste ");  
String ville[]={ "Rabat", "Errachidia", "Ouarzazate", "Tanger", "Agadir", "Rich", "Oujda"};  
JList lst=new JList(ville);  
lst.setBounds(50, 50,100,100);
```

```
f.add(lst);  
f.setLayout(null);  
f.setSize(400,500);  
f.setVisible(true);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

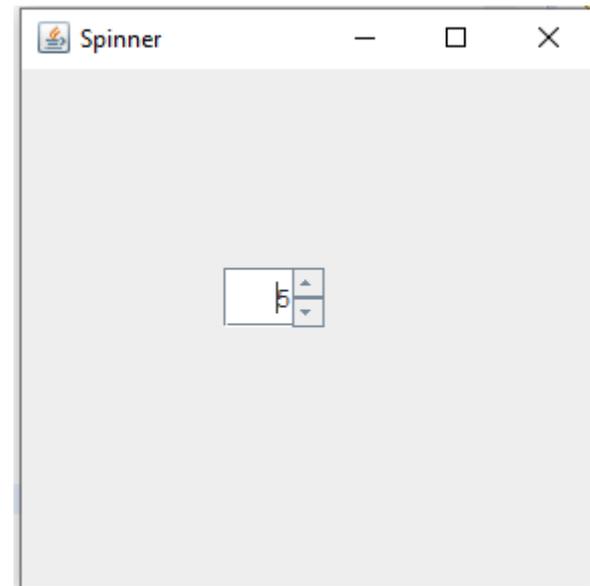
h- JSpinner

JSpinner()

JSpinner(SpinnerModel model)

Exemple

```
public static void main(String[] args) {  
    JFrame f=new JFrame("Spinner ");  
  
    SpinnerModel mod = new SpinnerNumberModel(5, 0, 20,1);  
    JSpinner spinner = new JSpinner(mod);  
    spinner.setBounds(100,100,50,30);  
    f.add(spinner);  
    f.setSize(300,300);  
    f.setLayout(null);  
    f.setVisible(true);  
}
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

i- Barre de défilement

JScrollBar()

JScrollBar(int orientation)

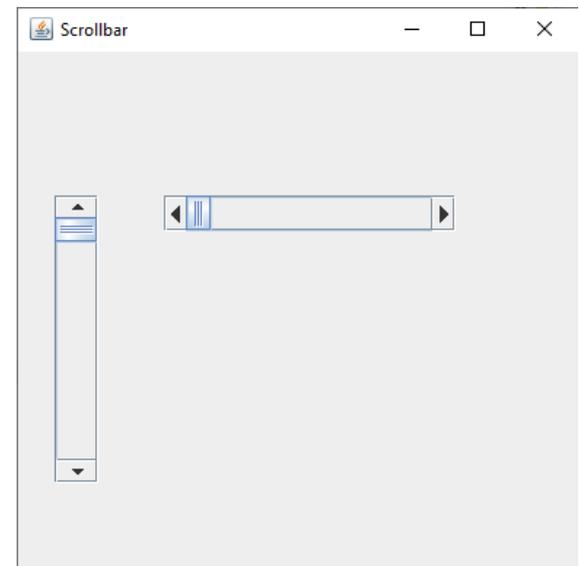
JScrollBar(int orientation, int value, int extent, int min, int max)

Exemple

```
JFrame f= new JFrame("Scrollbar");  
JScrollBar barh=new JScrollBar(JScrollBar.HORIZONTAL);  
barh.setBounds(100,100, 200,25);
```

```
JScrollBar barv=new JScrollBar(JScrollBar.VERTICAL);  
barv.setBounds(25,100, 30,200);
```

```
f.add(barh);  
f.add(barv);  
f.setSize(400,400);  
f.setLayout(null);  
f.setVisible(true);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

j- Glissière

JSlider()

JSlider(int orientation)

JSlider(int min, int max)

JSlider(int min, int max, int value)

JSlider(int orientation, int min, int max, int value)

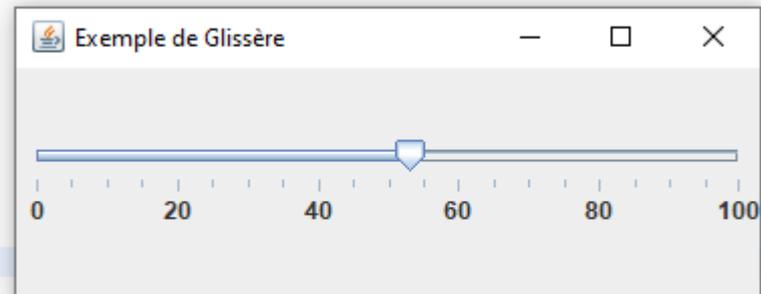
Exemple

```
JFrame fen = new JFrame("Exemple de Glissère");
```

```
JSlider slider = new JSlider();  
slider.setBounds(10, 100, 200, 25);  
slider.setMajorTickSpacing(20);  
slider.setMinorTickSpacing(5);  
slider.setPaintTicks(true);  
slider.setPaintLabels(true);
```

```
fen.add(slider);
```

```
fen.setSize(500, 500);  
fen.setVisible(true);
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

k- Barre de progression

JProgressBar()

JProgressBar(int min, int max)

JProgressBar(int orient)

JProgressBar(int orient, int min, int max)

Exemple

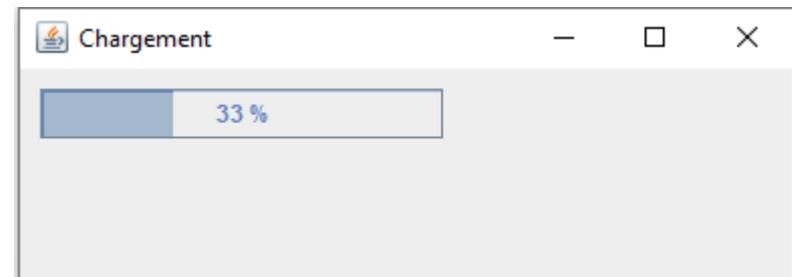
```
JFrame f= new JFrame("Chargement");

JProgressBar progbar=new JProgressBar(JProgressBar.HORIZONTAL,0,5000);
progbar.setBounds(10, 10, 200, 25);
progbar.setStringPainted(true);

f.add(progbar);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);

///pour voir le chargement
for(int i=0; i<=5000;i++) {
    progbar.setValue(i);

    try{Thread.sleep(10);}catch(Exception e){}
}
```



Interfaces Graphiques

IV- Swing

2- Les Composants standards:

1- Tableau

JTable()

JTable(Object[][] rows, Object[] columns)

Exemple

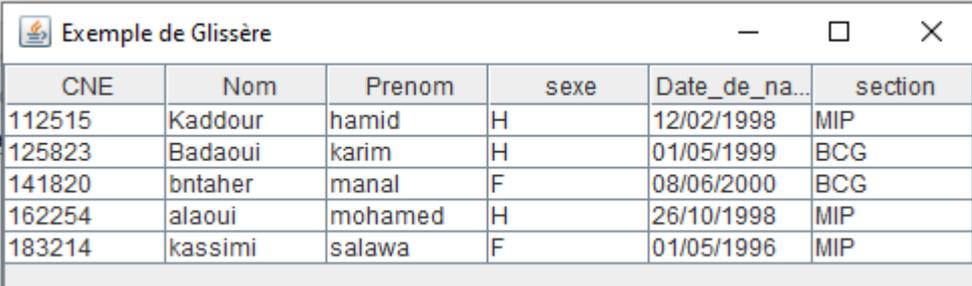
```
JFrame fen = new JFrame("Exemple de Glissère");
```

```
String titre[]= {"CNE","Nom","Prenom","sexe","Date_de_naiss","section"};  
String donnee[][]= { {"112515","Kaddour","hamid","H","12/02/1998","MIP"},  
                      {"125823","Badaoui","karim","H","01/05/1999","BCG"},  
                      {"141820","bntaher","manal","F","08/06/2000","BCG"},  
                      {"162254","alaoui","mohamed","H","26/10/1998","MIP"},  
                      {"183214","kassimi","salawa","F","01/05/1996","MIP"}  
};
```

```
JTable tab=new JTable(donnee, titre);  
tab.setBounds(10, 50, 400, 200);
```

```
JScrollPane pan=new JScrollPane(tab);  
pan.setBounds(10, 50, 400, 200);
```

```
fen.add(pan);  
fen.setSize(500, 500);  
fen.setVisible(true);
```



CNE	Nom	Prenom	sexe	Date_de_na...	section
112515	Kaddour	hamid	H	12/02/1998	MIP
125823	Badaoui	karim	H	01/05/1999	BCG
141820	bntaher	manal	F	08/06/2000	BCG
162254	alaoui	mohamed	H	26/10/1998	MIP
183214	kassimi	salawa	F	01/05/1996	MIP

Interfaces Graphiques

IV- Swing

3- Les conteneurs(panels):

a- JPanel

```
JFrame.setDefaultLookAndFeelDecorated(true); //changer le décor de la fenêtre

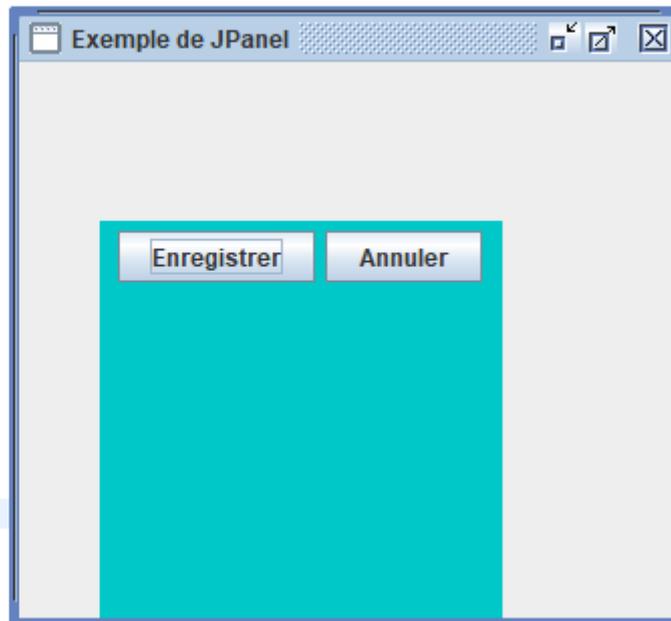
JFrame f= new JFrame("Exemple de JPanel");

JPanel pan=new JPanel();
pan.setBounds(40,80,200,200);
pan.setBackground(new Color(0,200,200));

JButton b1=new JButton("Enregistrer");
b1.setBounds(50,100,80,30);
JButton b2=new JButton("Annuler");
b2.setBounds(100,100,80,30);

pan.add(b1);
pan.add(b2);
f.add(pan);

f.setLayout(null);
f.setSize(400,400);
f.setVisible(true);
```



Interfaces Graphiques

IV- Swing

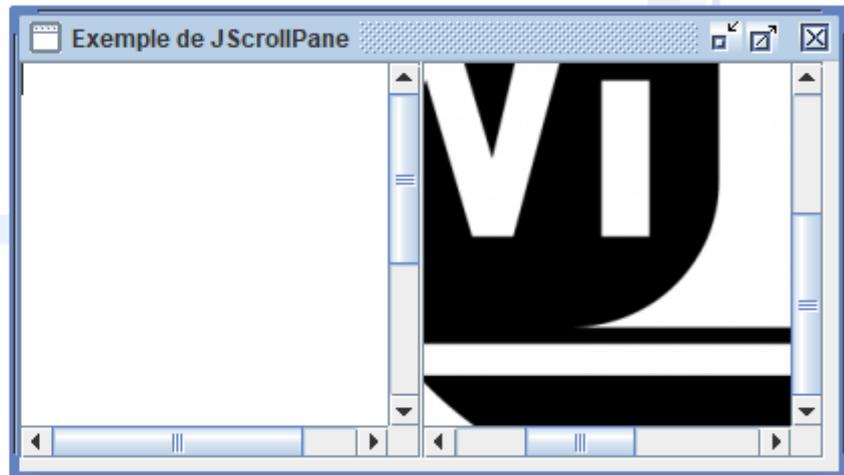
3- Les conteneurs(panels):

b- JScrollPane

Un JScrollPane est utilisé pour créer une vue défilante d'un composant. Lorsque la taille de l'écran est limitée, nous utilisons un volet de défilement pour afficher un grand composant ou un composant dont la taille peut changer dynamiquement.

Exemple

```
JFrame f= new JFrame("Exemple de JScrollPane");  
JTextArea txt = new JTextArea(20, 20);  
JScrollPane pan1=new JScrollPane(txt);  
pan1.setBounds(0,00,200,200);  
Icon img = new ImageIcon("D:\\figures\\WIFI.png");  
JLabel lab = new JLabel(img);  
JScrollPane pan2=new JScrollPane(lab);  
pan2.setBounds(200,0,200,200);  
  
f.add(pan1);  
f.add(pan2);
```



Interfaces Graphiques

IV- Swing

3- Les conteneurs(panels):

c- JTabbedPane

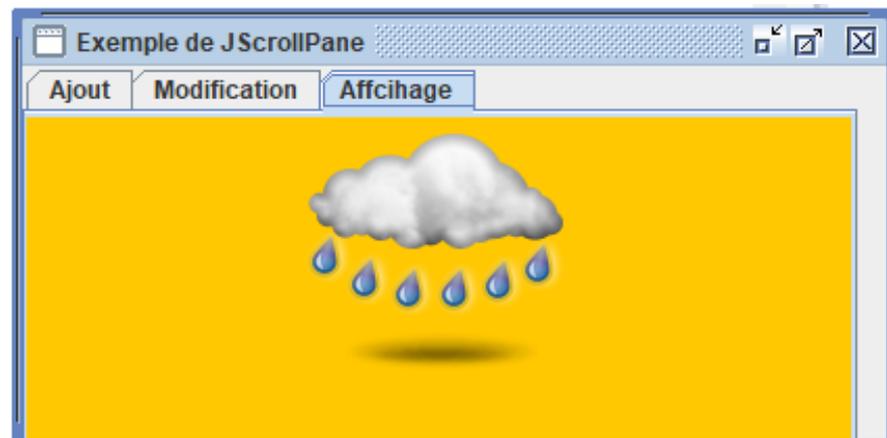
```
//creation des composants simple
JTextArea txt = new JTextArea(20, 20);
JButton bt=new JButton("Enregistrer");
Icon img = new ImageIcon("D:\\figures\\40.png");
JLabel lab = new JLabel(img);
```

```
//creation des JPanells simple
JPanel pan1=new JPanel();
pan1.setBackground(Color.BLUE);
JPanel pan2=new JPanel();
pan2.setBackground(Color.cyan);
JPanel pan3=new JPanel();
pan3.setBackground(Color.orange);
```

```
//Ajout des composants dans les JPanell
pan1.add(bt);
pan2.add(txt);
pan3.add(lab);
```

```
///creation de JTabbedPane
JTabbedPane jt=new JTabbedPane();
jt.setBounds(0, 0, 400, 400);
//ajout des JPanell dans JTabbedPane
jt.add("Ajout", pan1);
jt.add("Modification", pan2);
jt.add("Affcihage", pan3);
```

```
//ajout de JTabbedPane dans la fenetre
f.add(jt);
```



Interfaces Graphiques

IV- Swing

3- Les conteneurs(panels):

d- JSplitPane

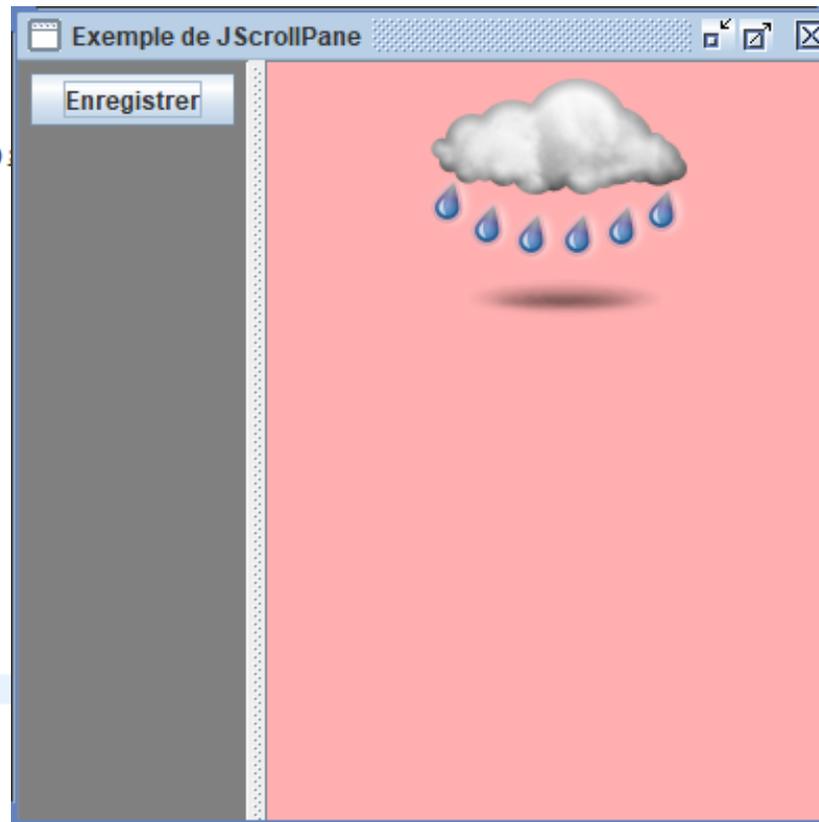
```
//creation des composants simple
JTextArea txt = new JTextArea(20, 20);
JButton bt=new JButton("Enregistrer");
Icon img = new ImageIcon("D:\\figures\\40.png");
JLabel lab = new JLabel(img);

//creation des JPanels simple
JPanel pan1=new JPanel();
pan1.setBackground(Color.GRAY);
JPanel pan2=new JPanel();
pan2.setBackground(Color.pink);

//Ajout des composants dans les JPanel
pan1.add(bt);
pan2.add(lab);

///creation de JSplitPane
JSplitPane jsp=new JSplitPane();
jsp.setBounds(00, 0, 400, 400);
//ajout des JPanel dans JSplitPane
jsp.setLeftComponent(pan1);
jsp.setRightComponent(pan2);

//ajout de JSplitPane dans la fenetre
f.add(jsp);
```



Interfaces Graphiques

IV- Swing

3- Les conteneurs(panels):

e- JEditorPane

```
String type,txt;

JFrame.setDefaultLookAndFeelDecorated(true);
JFrame fen = new JFrame("Mon Editeur");

fen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JEditorPane editeur = new JTextPane();

type="text/html"; //type a trois valeur "text/html" "text/plain" "text/rtf"
editeur.setContentType(type);

txt="<h1>JAVA </h1> "
  + "<p>JAVA est un langage de programmation </p> "
  + "<img src=\"http://inforce.rs/wp-content/uploads/2018/06/jw4dnFtA_400x400.jpg\" width=200 height=200 >"
  ;

editeur.setText(txt);

fen.setContentPane(editeur);
fen.setSize(400, 300);
fen.setVisible(true);
```



Interfaces Graphiques

IV- Swing

3- Les conteneurs(panels):

f- JTextPane

```
JFrame.setDefaultLookAndFeelDecorated(true);
JFrame fen = new JFrame("JTextPane");

fen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JTextPane pane = new JTextPane();

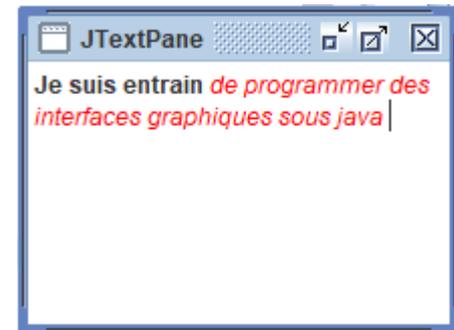
SimpleAttributeSet parametres = new SimpleAttributeSet();

//modification des parametres
StyleConstants.setBold(parametres, true); //Gras=OK
pane.setCharacterAttributes(parametres, true);

// Set the attributes before adding text
pane.setText("Je suis entrain");

//modification des parametres
parametres = new SimpleAttributeSet(); //reinitialiser tout les paramètres
StyleConstants.setItalic(parametres, true); //Italic=OK
StyleConstants.setForeground(parametres, Color.red); //couleur=rouge
//ajoute de texte
Document doc = pane.getStyledDocument();
doc.insertString(doc.getLength(), " de programmer des interfaces"
    + " graphiques sous java ", parametres);

fen.add(pane);
fen.setSize(400, 300);
fen.setVisible(true);
```



Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

Il s'agit d'une Petite fenêtre pouvant servir à plusieurs choses :

- Afficher une information (message d'erreur, d'avertissement...)
- Demander une validation, une réfutation ou une annulation ;
- Demander à l'utilisateur de saisir une information dont le système a besoin ;
- Etc.

Le nom de la classe est : **JOptionPane**

Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

Méthodes pour préciser le type de message

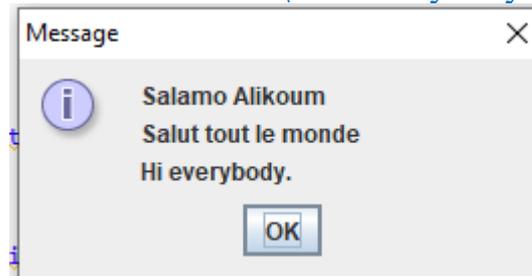
```
static void showMessageDialog(Component parentComponent, Object message)
```

```
static int showConfirmDialog(Component parentComponent, Object message)
```

```
static String showInputDialog(Component parentComponent, Object message)
```

Exemple:

```
JOptionPane mess1=new JOptionPane();  
mess1.showMessageDialog(null,"Salamo Alikoum\nSalut tout le monde \nHi everybody.");
```



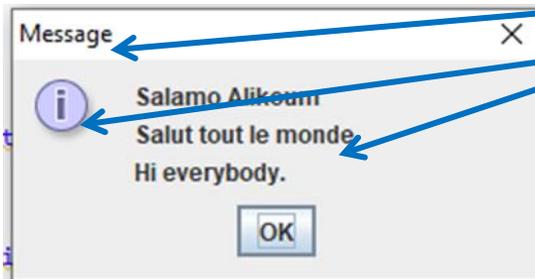
Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

a- Les boîtes d'information

static void showMessageDialog(Component parentComponent, Object message, String title, **int messageType**)



Type	Valeur int	icone
JOptionPane.PLAIN_MESSAGE	-1	Sans icone
JOptionPane.ERROR_MESSAGE	0	
JOptionPane.INFORMATION_MESSAGE	1	
JOptionPane.WARNING_MESSAGE	2	
JOptionPane.QUESTION_MESSAGE	3	

Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

a- Les boîtes d'information

static void showMessageDialog(Component parentComponent, Object message, String title, int messageType)



```
JOptionPane boit_1, boit_2, boit_3;  
boit_1 = new JOptionPane();  
boit_2 = new JOptionPane();  
boit_3 = new JOptionPane();  
  
/*Boîte du message d'information  
boit_1.showMessageDialog(null, "Message informatif", "Information", JOptionPane.INFORMATION_MESSAGE);  
  
/*Boîte du message préventif  
boit_2.showMessageDialog(null, "Message préventif", "Attention", JOptionPane.WARNING_MESSAGE);  
  
/*Boîte du message d'erreur  
boit_3.showMessageDialog(null, "Message d'erreur", "Erreur", JOptionPane.ERROR_MESSAGE);
```

Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

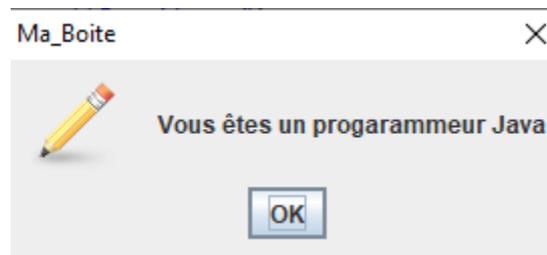
a- Les boîtes d'information

Pour personnaliser l'icône utiliser cette méthode :

```
static void showMessageDialog(Component parentComponent, Object message, String title, int messageType, Icon icone )
```

Exemple

```
ImageIcon icon1=new ImageIcon("D:\\figures\\Pencil.png");  
boit_1.showMessageDialog(null, "Vous êtes un progarammeur Java", "Ma_Boite", JOptionPane.INFORMATION_MESSAGE, icon1);
```



Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

b- Les boîtes de confirmation

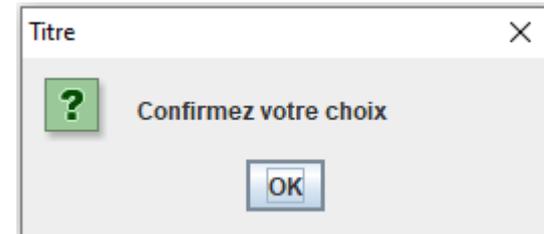
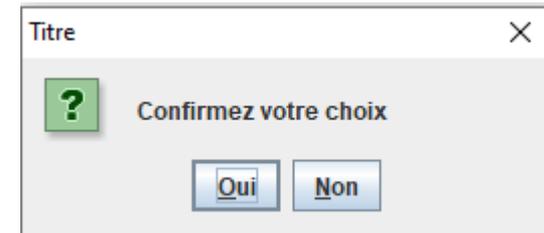
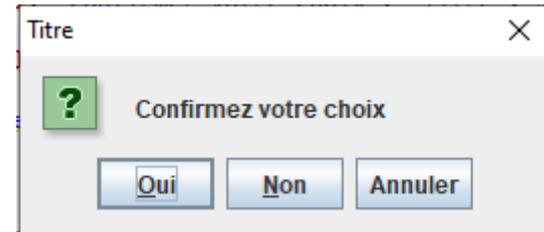
Static int showConfirmDialog(parentComponent, message, title, **int optionType**, int messageType)

optionType peut être :

JOptionPane.YES_NO_CANCEL_OPTION

JOptionPane.YES_NO_OPTION

JOptionPane.DEFAULT_OPTION



Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

b- Les boîtes de confirmation

Les valeurs renvoyées sont :

Valeur	Valeur int
JOptionPane.CLOSED_OPTION	-1
JOptionPane.OK_OPTION	0
JOptionPane.NO_OPTION	1
JOptionPane.CANCEL_OPTION	2

```
int repl;  
repl=boit_1.showConfirmDialog(null, "Confirmez votre choix", "Titre", JOptionPane.YES_NO_CANCEL_OPTION,3);  
  
if(repl== -1) {  
    System.out.print("Boite Fermée");  
}else if(repl==0) {  
    System.out.print("Vous avez choisit 'oui'");  
}else if(repl==1) {  
    System.out.print("Vous avez choisit 'non'");  
}else if(repl==2) {  
    System.out.print("Vous avez choisit 'annuler'");  
}
```

Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

c- Les boîtes de saisie(avec une champ de texte)

static String showInputDialog(Component parentComponent, Object message)

```
String rep1;  
rep1=boit_1.showInputDialog(null, "Entrer votre nom");
```



Chapitre III: Interfaces Graphiques

IV- Swing

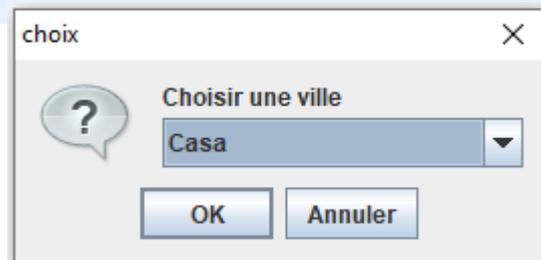
4- Les Boites de dialogue:

c- Les boîtes de saisie (avec une liste de choix)

```
static String showInputDialog(parentComponent, message, title, messageType, icon, selectionValues [],  
initialSelectionValue)
```

Exemple

```
String rep1;  
String [] ville= {"Errachidia", "Rabat", "Oujda", "Casa", "Tanger", "Rich", "Tinghir", "Ouarzazate"};  
ImageIcon icone=new ImageIcon("D:\\figures\\Questions.png");  
rep1=(String) JOptionPane.showInputDialog(null, "Choisir une ville", "choix",  
JOptionPane.QUESTION_MESSAGE, icone, ville, ville[3]);  
  
System.out.print(" nom :"+rep1);
```



Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

d- les boîtes de dialogue personnalisées

Utiliser pour cela la classe : **JDialog**

Exemple

The screenshot shows a custom Java Swing dialog box titled "Ma_Boite". The dialog is divided into two sections. The top section, titled "Nom du personnage", contains two text input fields: "Nom :" and "Prenom :". The bottom section, titled "Formations", is highlighted with a cyan background and contains three controls: a dropdown menu for "Type de Bac" with the value "Science Math A", a spin box for "Année de Bac" with the value "2 020", and another dropdown menu for "Dernier diplome" with the value "Deug". At the bottom of the dialog are two buttons: "OK" and "Annuler".

Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

e- Choisir une couleur (JColorChooser)

Constructeur

JColorChooser()

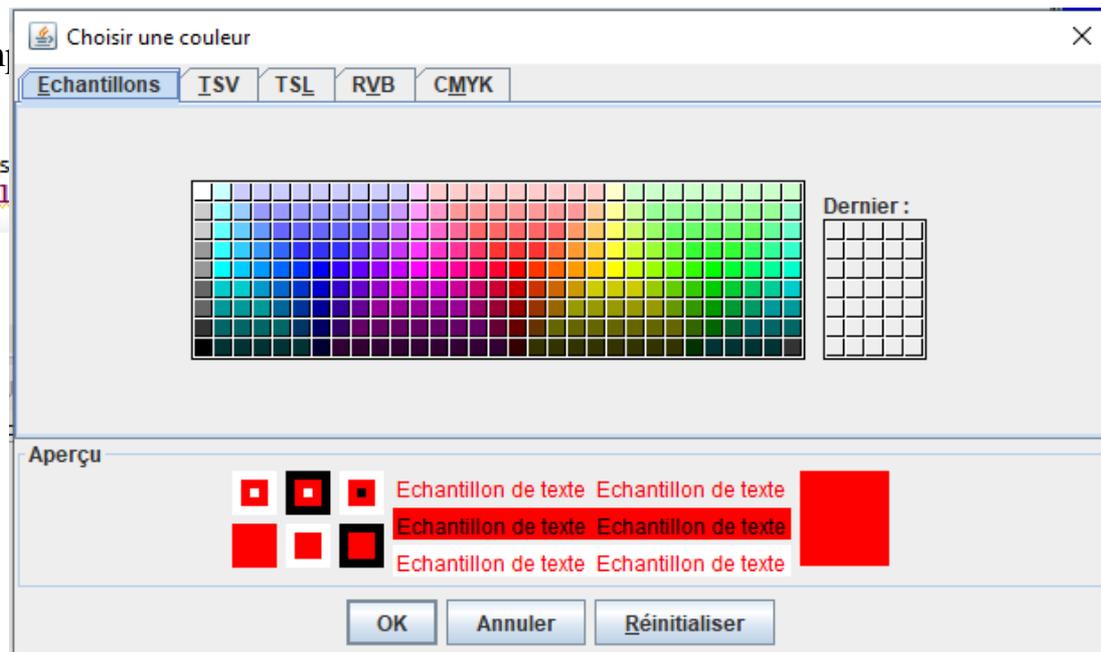
JColorChooser(color initialcolor)

Methode

static Color showDialog(Component)

Exemple

```
JColorChooser choosecolor=new JColorChoos  
Color couleur=choosecolor.showDialog(null
```



Interfaces Graphiques

IV- Swing

4- Les Boites de dialogue:

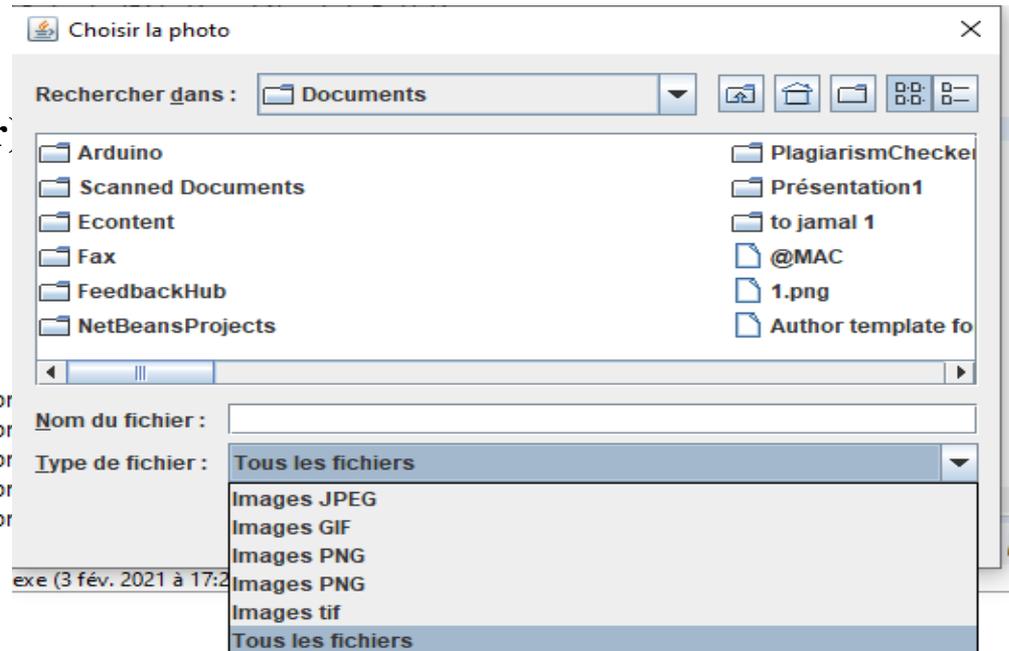
f- Choisir un fichier(JFileChooser)

Constructeur

```
// création de la boîte de dialogue
JFileChooser dialogue = new JFileChooser();
//chnager le titre
dialogue.setDialogTitle("Choisir la photo");

dialogue.addChoosableFileFilter(new FileNameExtensionFilter(""));
dialogue.addChoosableFileFilter(new FileNameExtensionFilter(""));
dialogue.addChoosableFileFilter(new FileNameExtensionFilter(""));
dialogue.addChoosableFileFilter(new FileNameExtensionFilter(""));
dialogue.addChoosableFileFilter(new FileNameExtensionFilter(""));

dialogue.setAcceptAllFileFilterUsed(true);
//changer la couleur
dialogue.setBackground(Color.ORANGE);
//lancer la boîte de dialogue
dialogue.showOpenDialog(null);
// récupération du fichier sélectionné et son dossier
File pht=dialogue.getSelectedFile();
String la_photo=dialogue.getSelectedFile().getAbsolutePath();
//ou bien
//String la_photo =pht.getAbsolutePath();
```



Interfaces Graphiques

IV- Swing

5- Les Menus:

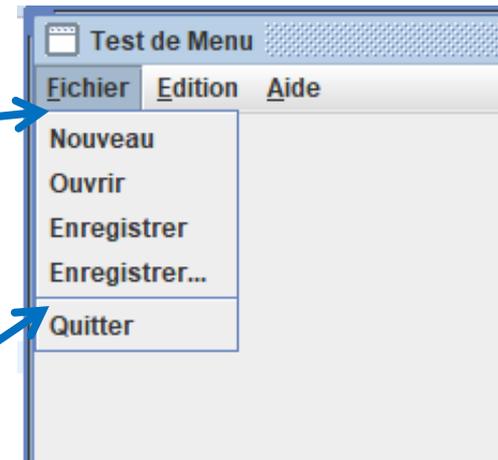
```
barreMenu=new JMenuBar();

menu1=new JMenu("Fichier");
menu2=new JMenu("Edition");
menu3=new JMenu("Aide");

//créer les raccourcis
menu1.setMnemonic('F');
menu2.setMnemonic('E');
menu3.setMnemonic('A');

//les options de Menu1
mlopt1=new JMenuItem("Nouveau");
mlopt2=new JMenuItem("Ouvrir");
mlopt3=new JMenuItem("Enregistrer");
mlopt4=new JMenuItem("Enregistrer...");
mlopt5=new JMenuItem("Quitter");

menu1.add(mlopt1);
menu1.add(mlopt2);
menu1.add(mlopt3);
menu1.add(mlopt4);
menu1.addSeparator();//Ajouter un separateur
menu1.add(mlopt5);
/////fin des options de Menu1
//"
```



Interfaces Graphiques

IV- Swing

7- L'arbre:JTree

```
DefaultMutableTreeNode style=new DefaultMutableTreeNode("Style");
DefaultMutableTreeNode color=new DefaultMutableTreeNode("Couleur");
DefaultMutableTreeNode font=new DefaultMutableTreeNode("Font");

style.add(color); style.add(font);

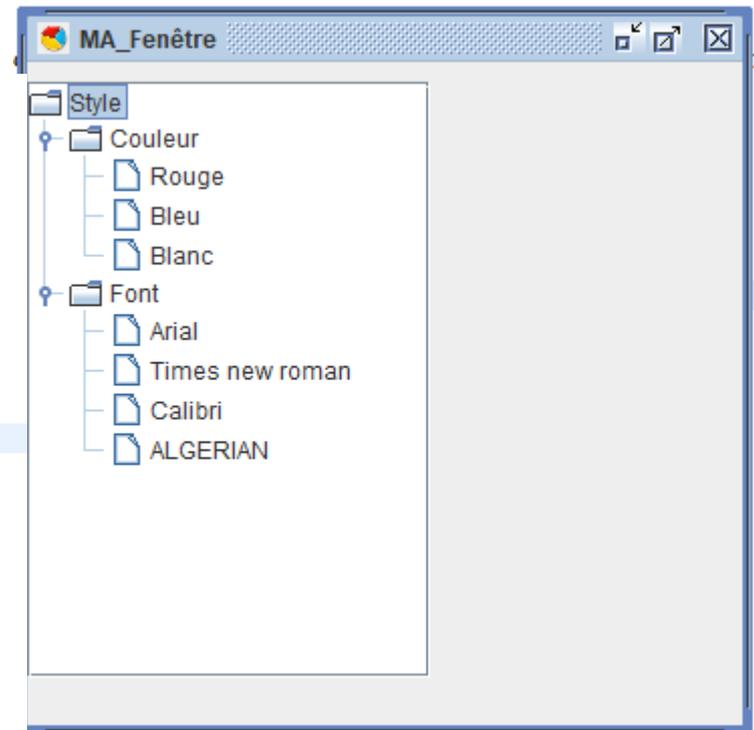
DefaultMutableTreeNode rouge=new DefaultMutableTreeNode("Rouge");
DefaultMutableTreeNode bleu=new DefaultMutableTreeNode("Bleu");
DefaultMutableTreeNode blanc=new DefaultMutableTreeNode("Blanc");

color.add(rouge); color.add(bleu); color.add(blanc);

DefaultMutableTreeNode f1=new DefaultMutableTreeNode("Arial");
DefaultMutableTreeNode f2=new DefaultMutableTreeNode("Times new roman");
DefaultMutableTreeNode f3=new DefaultMutableTreeNode("Calibri");
DefaultMutableTreeNode f4=new DefaultMutableTreeNode("ALGERIAN");

font.add(f1);font.add(f2);font.add(f3);font.add(f4);

JTree jt=new JTree(style);
jt.setBounds(100, 25, 100, 200);
```



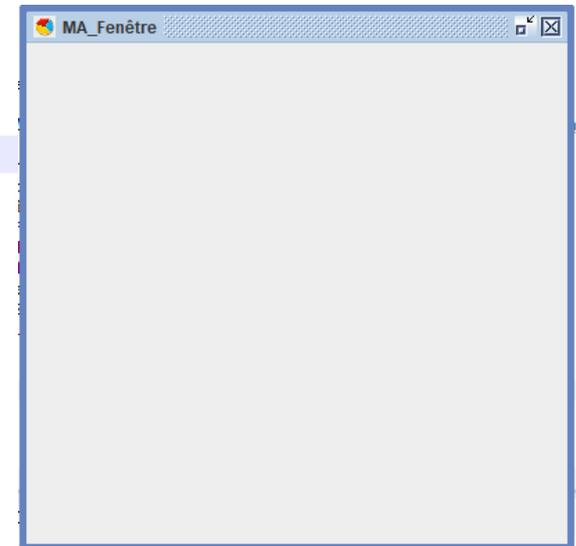
Interfaces Graphiques

IV- Swing

10- Quelques trucs utiles:

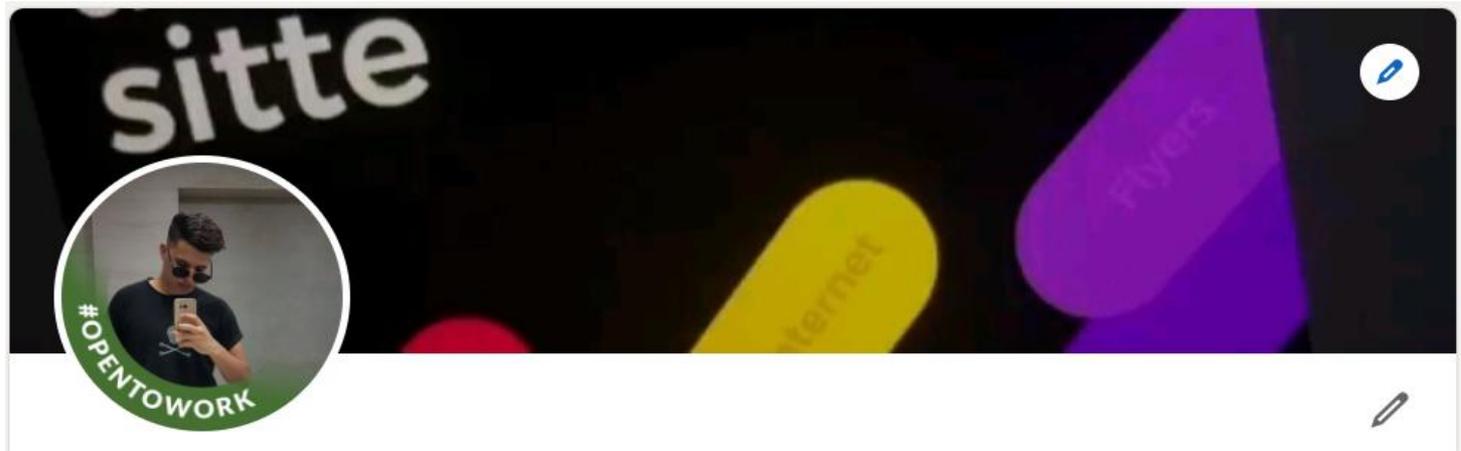
```
JFrame.setDefaultLookAndFeelDecorated(true); //changer le décor de la fenêtre
```

```
JFrame f=new JFrame();  
Image icon = Toolkit.getDefaultToolkit().getImage("D:\\figures\\Piechart.png");  
f.setIconImage(icon); //changer l'icone de la fenêtre  
f.setTitle("MA_Fenêtre") ;//changer le titre de la fenêtre  
f.setLayout(null);  
f.setResizable(false); //Fixer la taille de la fenêtre  
f.setCursor(Cursor.HAND_CURSOR); //changer le curseur  
f.setSize(400,400);  
f.setVisible(true);
```



RÉALISER PAR :

67



ELHOUSSAINE E-CHAJIAI

[Gmail: chajiaihou@gmail.com](mailto:chajiaihou@gmail.com)

LinkedIn : <https://www.linkedin.com/in/elhoussaine-e-14a9a3218/>