

Chapitre 11

Programmation graphique

Pour écrire des programmes graphiques, java dispose du package **Swing**. **Swing** dispose de plusieurs composantes qui commencent par **J**.

Pour utiliser les composantes **Swing**, il faut insérer au début de votre programme la déclaration :

```
import javax.swing.*;
```

La classe **JFrame**

Le package **javax.swing** dispose de la classe **JFrame** pour créer des fenêtres graphiques. Cette classe dispose des constructeurs :

- **JFrame()** : construit une nouvelle fenêtre invisible et sans titre.
- **JFrame(String titre)** : construit une nouvelle fenêtre invisible avec un titre.

Initialement, la fenêtre sera invisible et sans taille. Rien n'apparaîtra à l'écran.

Quelques méthodes de la classe **JFrame**

La classe **JFrame** dispose de plusieurs méthodes. Nous allons citer quelques unes de ces méthodes :

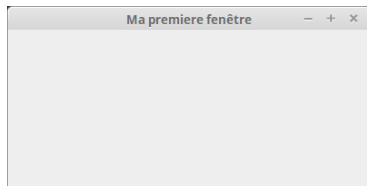
<code>void setTitle(String titre)</code>	Donne un titre à la fenêtre.
<code>void setSize(int largeur, int hauteur)</code>	met la taille de la fenêtre en pixels en largeur et hauteur.
<code>void setVisible(boolean)</code>	rend la fenêtre <ul style="list-style-type: none">• visible en utilisant l'argument true• invisible en utilisant l'argument false.
<code>String getTitle()</code>	retourne le titre de la fenêtre.
<code>void setResizable(boolean)</code>	l'argument true permet de rendre la fenêtre redimensionnable et false la rend non redimensionnable.

Quelques méthodes de la classe JFrame

boolean isResizable()	retourne true ou false pour indiquer si la fenêtre est redimensionnable ou non.
void setBounds(int posH, int posV, int largeur, int hauteur)	met la fenêtre dans la position horizontale et verticale et modifie sa taille.

Première fenêtre

```
import javax.swing.*;  
public class PremiereFenetre {  
    public static void main(String args[]) {  
        JFrame fen = new JFrame("Ma premiere fenetre"  
        );  
        fen.setSize(400, 200);  
        fen.setVisible(true);  
    }  
}
```



Remarque : fermeture de la fenêtre

Lorsqu'on ferme la fenêtre, celle est invisible seulement. Le programme reste actif.

Pour fermer la fenêtre et terminer le programme, il faut imposer le comportement avec la directive :

```
fen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

D'autres directives existent :

DO_NOTHING_ON_CLOSE	ne rien faire
HIDE_ON_CLOSE	cacher la fenêtre (comportement par défaut),
DISPOSE_ON_CLOSE	détruire l'objet fenêtre.

Création d'une classe JFrame personnalisée

Vous pouvez créer une nouvelle classe qui hérite de **JFrame**.

L'avantage de la création d'une nouvelle classe, c'est que vous spécifiez les propriétés de la fenêtre dans le constructeur.

Exemple

```
import javax.swing.*;

public class Mafenetre extends JFrame
{
    final int LARGEUR = 200;
    final int HAUTEUR = 120;
    public Mafenetre(String titre)
    {
        super(titre);
        setSize(LARGEUR, HAUTEUR);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Exemple

```
public class DeuxFenetres {  
    public static void main( String[] args) {  
        Mafenetre fen1 = new Mafenetre("Fenetre 1")  
        ;  
        Mafenetre fen2 = new Mafenetre("Fenetre 2")  
        ;  
    }  
}
```

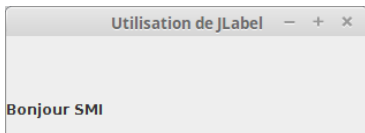
L'exécution du programme, crée deux fenêtres, l'une sur l'autre.

Utilisation de la classe JLabel

JLabel permet l'affichage d'un texte dans une fenêtre.

```
import javax.swing.JLabel;

public class TestJLabel1 {
    public static void main(String[] args) {
        Mafenetre fen = new Mafenetre("Utilisation de
        JLabel");
        JLabel bonjour = new JLabel("Bonjour SMI");
        fen.add(bonjour);
    }
}
```



Changer la fonte (police) d'un objet de type JLabel

- Pour modifier la fonte, on aura besoin d'un objet de type **Font**. Pour cela, il faut importer la classe **Font** :

`import java.awt.Font;`

- Pour construire un objet de type **Font**, on aura besoin de 3 arguments

① Type : nom de la fonte.

② Style : style de de la fonte

- plain : `Font.PLAIN`
- gras : `Font.BOLD`
- italique : `Font.ITALIC`

③ Taille : taille de la fonte en points ($1pt \simeq \frac{2,5}{72} cm \simeq 0,034cm$).

Exemple

```
import java.awt.Font;
import javax.swing.JLabel;

public class TestJlabelFont {
    public static void main(String[] args) {
        Mafenetre fen = new Mafenetre("Utilisation de
        JLabel");
        JLabel bonjour = new JLabel("Bonjour SMI");
        Font entete = new Font("Times", Font.BOLD, 36)
        ;
        bonjour.setFont(entete);
        fen.add(bonjour);
    }
}
```

Gestionnaire de mise en forme (Layout Manager)

```
import javax.swing.JLabel;

public class DeuxLabels {
    public static void main(String[] args) {
        Mafenetre fen = new Mafenetre("Tests");
        JLabel bonjour = new JLabel("Bonjour ");
        JLabel smi = new JLabel("SMI");
        fen.add(bonjour);
        fen.add(smi);
    }
}
```

L'exécution du programme précédent affichera seulement le dernier label. Le deuxième label a été placé au dessus du premier.

Gestionnaire de mise en forme (Layout Manager))

- La disposition des composants dans une fenêtre est gérée par un gestionnaire de mise en forme.
- Par défaut, Java utilise le gestionnaire **BorderLayout**, qui met les composantes l'une sur l'autre et chaque composante occupe toute la fenêtre.
- Le gestionnaire **FlowLayout** met les composantes l'une après l'autre (en flot).

Gestionnaire de mise en forme

```
import javax.swing.JLabel;  
import java.awt.FlowLayout;  
  
public class DeuxLabelsLayout {  
    public static void main(String[] args) {  
        Mafenetre fen = new Mafenetre("Tests");  
        JLabel bonjour = new JLabel("Bonjour ");  
        JLabel smi = new JLabel("SMI");  
        fen.setLayout(new FlowLayout());  
        fen.add(bonjour);  
        fen.add(smi);  
    }  
}
```


Champ de texte (**TextField**)

C'est un champ qui peut contenir une seule ligne de texte (chaîne de caractères).

Un objet de type **TextField** est construit en utilisant l'un des constructeurs suivants :

- **TextField()** : champ vide et de sans taille.
- **TextField(int N)** : champ de taille N (vide).
- **TextField(String texte)** : champ initialisé avec le texte spécifié.
- **public TextField(String texte, int N)** : champ de taille N et initialisé avec le texte spécifié.

Quelques méthodes de **JTextField** :

- **setText()** : permet de modifier le texte dans un champ JTextField.
- **getText()** : permet de récupérer le texte d'un champ JTextField.
String nom=reponse.getText();
- **setEditable()** : permet de changer le statut d'édition
reponse.setEditable(**false**) : désactive l'édition du champ "reponse".

Un bouton **JButton**

Pour créer le bouton « test » qui contient le texte « Test », on utilise :

```
JButton test = new JButton("Test");
```

- Pour utiliser un raccourci clavier (Att+caractère) : caractère souligné aussi appelé caractère mnémonique, on utilise la méthode **setMnemonic()**
bouton.setMnemonic('A');
- Un bouton peut être activé/désactivé avec la méthode **setEnabled()**.

Exemple avec deux boutons (un activé et l'autre désactivé)

```
public class DeuxBoutons extends JFrame {
    JButton bouton1 = new JButton("Active");
    JButton bouton2 = new JButton("Desactive");
    public DeuxBoutons() {
        bouton1.setMnemonic('A'); // Alt+A
        bouton2.setMnemonic('D'); // Alt+D
        add(bouton1);
        add(bouton2);
        bouton2.setEnabled(false);
        setLayout(new FlowLayout());
    }
    public static void main(String[] args) {
        DeuxBoutons f = new DeuxBoutons();
        f.setSize(150, 150);    f.setVisible(true);
    }
}
```

Exemple avec plusieurs composantes

```
import javax.swing.JButton;  
import javax.swing.JLabel;  
import javax.swing.JTextField;  
  
public class PlusieursComposantes {  
    public static void main(String[] args) {  
        Mafenetre fen = new Mafenetre("Plusieurs  
composantes");  
        JLabel titre = new JLabel("Fenetre avec  
plusieurs composantes");  
        Font fonte = new Font("Times", Font.BOLD, 24);  
        titre.setFont(fonte);  
  
        JLabel nomLabel = new JLabel("Nom :");  
        JTextField nomTexte = new JTextField(10);  
        JButton bouton = new JButton("Valider");  
    }  
}
```

Exemple avec plusieurs composantes (suite)

```
fen.setLayout(new FlowLayout());  
fen.add(titre);  
fen.add(nomLabel);  
fen.add(nomTexte);  
fen.add(bouton);  
}  
}
```

Écouteur (Listner)

Le bouton défini dans l'exemple précédent, ne fourni aucun résultat. Pour fournir un résultat, il faut lui associer un événement.

Un événement est une action de l'utilisateur (clic de la souris, appui sur la barre d'espace, ...).

Pour gérer les événements de l'utilisateur, vous devez suivre les étapes suivantes :

- ❶ Préparer la classe pour accepter les événements :
 - importer le package **java.awt.event**
 - implémenter l'interface **ActionListener**
- ❷ La classe doit s'attendre à recevoir des événements :
Appel de la méthode **addActionListener()**.
- ❸ Comment réagir aux événements :
l'interface **ActionListener** contient la méthode **actionPerformed(ActionEvent e)** qui doit contenir les actions à exécuter.

Exemple

```
import java.awt.*;  
import javax.swing.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Fenetre extends JFrame  
    implements ActionListener {  
  
    private JLabel nomSaisi = new JLabel();  
    private JTextField nom = new JTextField(10);  
    Font entete = new Font("Times", Font.BOLD, 24);  
  
    public void actionPerformed(ActionEvent e) {  
        nomSaisi.setForeground(java.awt.Color.red);  
        nomSaisi.setText("Bonjour "+nom.getText());  
    }  
}
```

```
public Fenetre(String titreFenetre) {  
    super(titreFenetre);  
    setSize(350, 150);  
  
    setVisible(true);  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    JLabel titre = new JLabel("Saisi le nom :");  
    titre.setFont(entete);  
    JButton bouton = new JButton("Valider");  
    setLayout(new FlowLayout());  
  
    add(titre);  
    add(nom);  
    add(bouton);  
    add(nomSaisi);  
    bouton.addActionListener(this);  
}
```

Remarque

Vous pouvez associer la même action à la zone de texte :

```
nomTexte.addActionListener(this);
```

Deux boutons avec actions

```
import ...;
public class DeuxBoutonsAction extends JFrame
    implements ActionListener {
    JButton bouton1 = new JButton("Active");
    JButton bouton2 = new JButton("Desactive");
    JLabel lab = new JLabel();
    public DeuxBoutonsAction() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        add(bouton1);    add(bouton2);    add(lab);
        bouton1.setActionCommand("b1");
        bouton2.setActionCommand("b2");
        bouton1.addActionListener(this);
        bouton2.addActionListener(this);
        bouton2.setEnabled(false);
        setLayout(new FlowLayout());
    }
```

```
public static void main(String[] args) {  
    DeuxBoutonsAction f = new DeuxBoutonsAction();  
    f.setSize(150, 150);  
    f.setVisible(true);  
}  
  
public void actionPerformed(ActionEvent e) {  
    if (e.getActionCommand().equals("b1")) {  
        lab.setText("Bouton1 est desactive");  
        bouton1.setEnabled(false);  
        bouton2.setEnabled(true);  
    }  
    else {  
        lab.setText("Bouton1 est active");  
        bouton1.setEnabled(true);  
        bouton2.setEnabled(false);  
    }  
}
```

Récupérer un mot de passe

```
import javax.swing.JPasswordField;  
  
JPasswordField mp = new JPasswordField(10);  
mp.setEchoChar( '*' );
```

Le mot de passe doit être récupéré dans un tableau de char :

```
char[] mpSaisi = mp.getPassword();
```

Mot de passe (Exemple)

```
import java.awt.FlowLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.Arrays;  
  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPasswordField;
```

```

public class MotPasse extends JFrame implements
    ActionListener {
    private JLabel motPasse = new JLabel();
    JPasswordField mp = new JPasswordField(10);

    public static void main(String[] args) {
        MotPasse fen = new MotPasse("Test Mot Passe");
    }

    public void actionPerformed(ActionEvent e) {
        motPasse.setForeground(java.awt.Color.red);
        char[] mpSaisi = mp.getPassword();

        //On peut utiliser du HTML dans un JLabel,
        JButton, ...
        motPasse.setText("<html>Mot de Passe saisi: <p
            > <h1>" + Arrays.toString(mpSaisi) + "</h1>
            ></html>");
    }
}

```


// Constructeur

```
public MotPasse(String titreFenetre) {  
    super(titreFenetre);  
    setSize(350, 150);  
    setVisible(true);  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    JLabel titre = new JLabel("Mot de Passe :");  
  
    mp.setEchoChar('*');  
    mp.addActionListener(this);  
  
    JButton bouton = new JButton("Valider");  
    setLayout(new FlowLayout());  
  
    add(titre); add(mp); add(bouton); add(motPasse);  
    bouton.addActionListener(this);  
    mp.addActionListener(this);  
}
```

Zone de texte (JTextArea)

Constructeurs :

```
JTextArea textArea = new JTextArea("Ligne1\nLigne2")  
;  
  
// nombre de lignes et de colonnes en parametres  
JTextArea textArea = new JTextArea(10, 20);
```

Le plus souvent on met la zone de texte dans un ScrollPane :

```
JScrollPane sc = new JScrollPane(textArea);
```

- **getText()** : récupère le texte du JTextArea
- **setText(String)** : donne la valeur du texte
- **append(String)** : ajoute du texte à la fin

```
import java.awt.FlowLayout;
import javax.swing.*;

public class TestJTextArea {
    public static void main(String[] args) {
        Mafenetre fen = new Mafenetre("Test JTextArea"
        );
        JTextArea textArea = new JTextArea(10, 20);
        JScrollPane sc = new JScrollPane(textArea);
        fen.setLayout(new FlowLayout());
        fen.add(sc);
    }
}
```

Les cases à cocher (**JCheckBox**)

Une case à cocher permet à l'utilisateur d'effectuer un choix de type oui/non.

```
import java.awt.*;  
import javax.swing.*;
```

```
public class DemoCaseAcocher extends JFrame {  
    FlowLayout flow = new FlowLayout();  
    JLabel label = new JLabel("Choix ?");  
  
    JCheckBox cafe = new JCheckBox("Cafe");  
    JCheckBox the = new JCheckBox("The", false);  
    JCheckBox eau = new JCheckBox("Eau", true);  
  
    public DemoCaseAcocher() {  
        super("Demonstration de JCheckBox");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setLayout(new FlowLayout());  
        add(label);        add(cafe);  
        add(the);          add(eau);  
    }  
}
```

```
public static void main(String[] arguments) {  
    DemoCaseAcocher fen = new DemoCaseAcocher();  
    fen.setSize(350, 150);  
    fen.setVisible(true);  
}  
}
```

BONNE CHANCE