

## Programmation évènementielle

La programmation évènementielle est le développement ou l'implémentation des interfaces graphiques (IHM ou GUI). Dans ce type de programmation, votre programme ne réagira plus à des saisies au clavier mais à des événements provenant d'un composant graphique : un bouton, une liste, un menu...

Nous utiliserons essentiellement les bibliothèques Swing et AWT présentes d'office dans Java.

### Les fenêtres (JFrame)

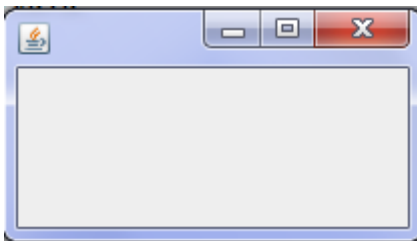
Pour utiliser une fenêtre de type JFrame, vous devez l'instancier, comme ceci :

```
JFrame fenetre = new JFrame();
```

Si vous exécutez ça, rien ne s'affiche sur l'écran, pour pouvoir afficher cette fenêtre, il faut la rendre visible comme suit :

```
fenetre.setVisible(true);
```

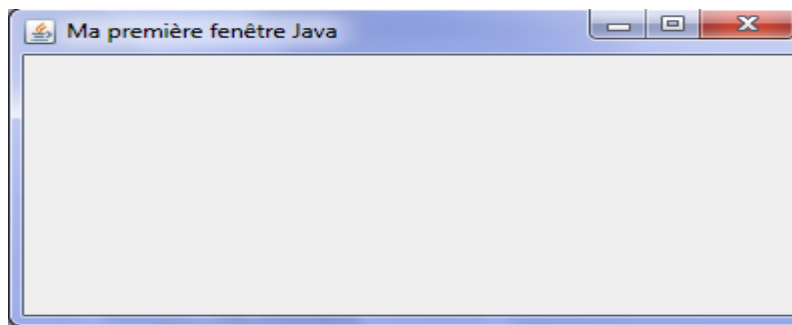
Là si vous exécutez, vous aurez une fenêtre qui s'affiche sur l'écran comme suit :



Pour obtenir une fenêtre plus conséquente, il faudrait donc:

- qu'elle soit plus grande ;
- qu'elle comporte un titre (ce ne serait pas du luxe !) ;
- qu'elle figure au centre de l'écran, ce serait parfait ;
- que notre programme s'arrête réellement lorsqu'on clique sur la croix rouge, car, pour ceux qui ne l'auraient pas remarqué, le processus NetBeans tourne encore même après la fermeture de la fenêtre.

```
//Définit un titre pour notre fenêtre
fenetre.setTitle("Ma première fenêtre Java");
//Définit sa taille : 400 pixels de large et 100 pixels de haut
fenetre.setSize(400, 200);
//Nous demandons maintenant à notre objet de se positionner au centre
fenetre.setLocationRelativeTo(null);
//Termine le processus lorsqu'on clique sur la croix rouge
fenetre.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



Pour pouvoir utiliser et réutiliser cette fenêtre sans refaire à chaque fois les dimensions...etc. on doit utiliser une hiérarchie de notre classe Fenetre avec la Classe JFrame, et ensuite de créer un constructeur Fenetre() et mettre les traitements précédents dedans, et enfin instancier un objet de la classe Fenetre().

```
public class Fenetre extends JFrame {  
  
    public Fenetre() {  
        this.setTitle("Ma première fenêtre Java");  
        this.setSize(400, 500);  
        this.setLocationRelativeTo(null);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setVisible(true);  
    }  
}
```

Ensuite dans le main on fait :

```
Fenetre fen=new Fenetre();
```

On aura le même résultat qu'avant mais là avec un objet fen, on peut en créer autant d'objet qu'on veut.

Quelques fonctions utiles sur les JFrames :

**setResizable(boolean b)** : **false** empêche le redimensionnement tandis que **true** l'autorise.

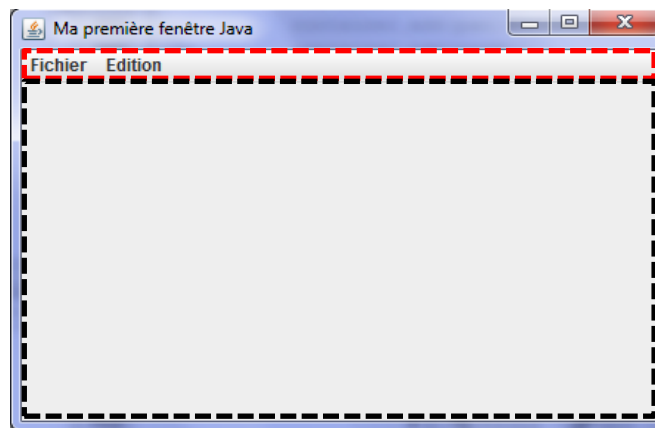
**setLocation(int x, int y)** : pour le positionnement de la fenêtre dans l'écran.

**setAlwaysOnTop(boolean b)** : afficher la fenêtre en premier plan sur l'écran si c'est true.

Une JFrame est découpée en plusieurs parties superposées, nous avons, dans l'ordre :

- la **fenêtre** ;
- le **RootPane** : le conteneur principal qui contient les autres composants ;
- le **LayeredPane** : qui forme juste un panneau composé du conteneur global et de la barre de menu(MenuBar) ;
- la **MenuBar** : la barre de menu, quand il y en a une ;
- le **content pane** : c'est dans celui-ci que nous placerons nos composants ;
- le **GlassPane** : couche utilisée pour intercepter les actions de l'utilisateur avant qu'elles ne parviennent aux composants.

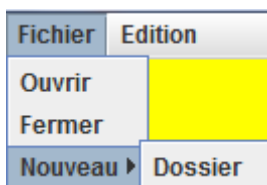
Ce qui nous intéresse dans cette partie du TP sont les couches **MenuBar** (pour mettre des menus sur nos fenêtres) et le **ContentPane** (pour les autres composants).



Sur la figure ci-dessus, le cadre en rouge représente la couche MenuBar de la fenêtre, tant dis que le cadre noir représente le ContentPane de la fenêtre.

## JMenuBar

Prenons l'exemple du menu suivant :



Pour mettre des menus sur une fenêtre, on doit d'abord définir un menu principal, ensuite les différents menus, et items. On compose nos menus en les ajoutant au menu principal et enfin on met notre menu principal dans la couche JMenuBar de la fenêtre comme suit :

```
JMenuBar menuPrincipal = new JMenuBar();

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

JMenuItem item1 = new JMenuItem("Ouvrir");
JMenuItem item2 = new JMenuItem("Fermer");
JMenuItem item3 = new JMenuItem("Enregistrer");
JMenuItem item4 = new JMenuItem("Imprimer");
JMenuItem item5 = new JMenuItem("Dossier");

menu1.add(item1);
menu1.add(item2);

menu1.add(menu3);

menu2.add(item3);
menu2.add(item4);
menu3.add(item5);

menuPrincipal.add(menu1);
menuPrincipal.add(menu2);
```

```
// On mets nos menu bar dans le MenuBar de notre Fenetre  
fen.setJMenuBar(menuPrincipal);
```

En exécutant, vous aurez le menu correspondant qui s'affiche sur la couche JMenuBar de la fenêtre fen.

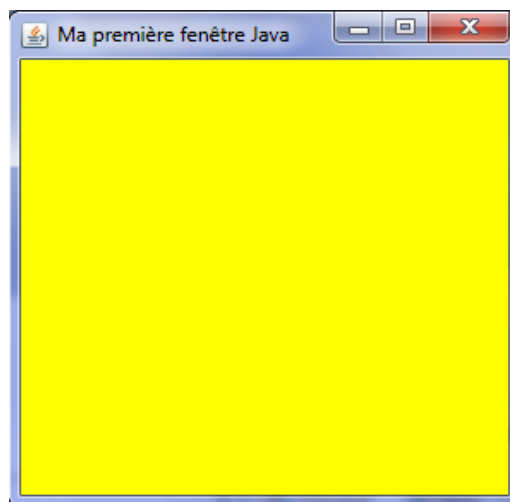
Pour mettre des composants (des champs de formulaires, des boutons...etc) dans une fenêtre, on va utiliser la couche ContentPane mais on ne va pas mettre les composants directement sur la fenêtre. On va d'abord créer un JPanel dans lequel nous insérerons nos composants.

## L'objet JPanel

Nous allons utiliser un JPanel, composant de type conteneur dont la vocation est d'accueillir d'autres objets de même type ou des objets de type composant (boutons, cases à cocher...).

```
JPanel container = new JPanel();  
container.setBackground(Color.yellow);  
fen.setContentPane(container);
```

En exécutant, on aura :



Là vous avez une fenêtre qui contient un panel, et c'est dans ce panel que nous allons ajouter nos composants (boutons, combobox, ...etc).

Dans ce TP, nous n'allons pas voir tous les composants Swing mais juste quelques composants qui sont les plus utilisés dans le développement des interfaces graphiques (les champs de formulaires).

Pour chaque type de composant, on doit d'abord le déclarer, lui donner certaines paramètres s'il le faut, ensuite on le met dans le panel correspondant (dans notre exemple on va le mettre dans le panel container) et enfin on met ce panel dans la couche ContentPane de la fenêtre.

## JLabel

```
JLabel labNom = new JLabel("Nom");
```

Nom

## JTextArea

```
final JTextArea txta = new JTextArea();
txta.setColumns(20);
txta.setRows(5);
```

un cahmp de texte multiligne  
ligne2  
lign3  
...  
etc

## JTextField

```
final JTextField jtf = new JTextField();
jtf.setColumns(20);
```

un exemple de champs de texte

## JPasswordField

```
final JPasswordField jpsw = new JPasswordField();
jpsw.setColumns(20);
```

.....

## JCombobox

```
// Définir une liste de choix
final JComboBox comb = new JComboBox();

comb.addItem("Choix 1");
comb.addItem("Choix 2");
comb.addItem("Choix 3");
```

Choix 1 ▼  
Choix 1  
Choix 2  
Choix 3

## JRadioButton

```
final JRadioButton radHomme = new JRadioButton("Homme");
final JRadioButton radFemme = new JRadioButton("Femme");

//On met nos radios dans un ButtonGroup
final ButtonGroup bg = new ButtonGroup();
bg.add(radHomme);
bg.add(radFemme);
```

```
//On présélectionne la première radio
radHomme.setSelected(true);
```

☒ Homme ☐ Femme

## JCheckBox

```
final JCheckBox jcheck1 = new JCheckBox("Java");
final JCheckBox jcheck2 = new JCheckBox("PHP");
```

☒ Java ☒ PHP

## JButton

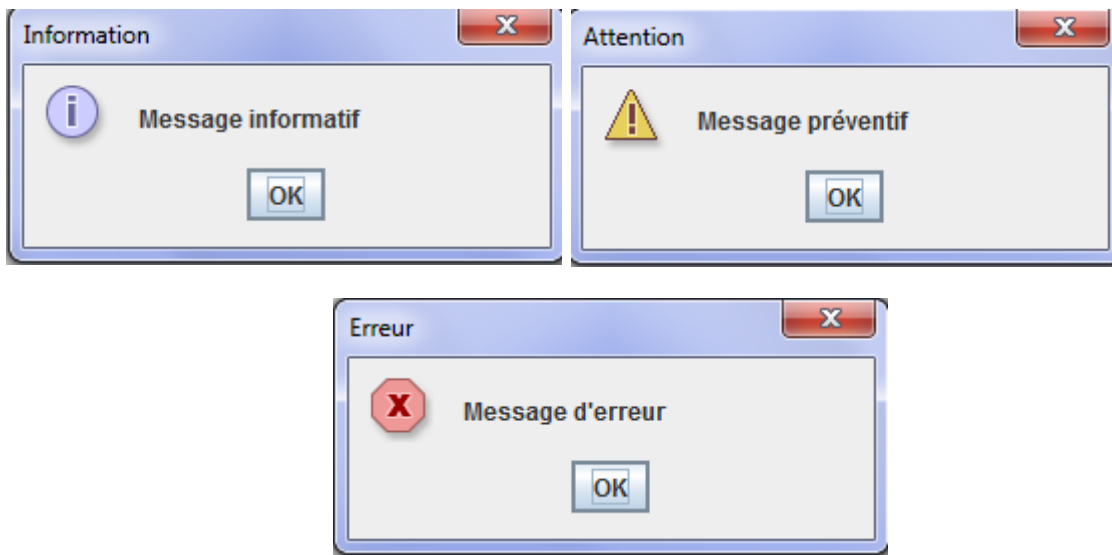
```
// Le button
JButton btn = new JButton("OK");
```

OK

## Les Boites de Dialogues

Les boîtes de dialogues sont des fenêtres spéciales qu'on affiche à l'utilisateur pour transmettre un message bien défini. Il existe trois types de boîtes de dialogues prédéfinies en Java, il s'agit des : boîtes de dialogues d'information, boîte de dialogues de prévention et boîtes de dialogues d'erreur.

```
JOptionPane jop1, jop2, jop3;
//Boîte du message d'information
jop1 = new JOptionPane();
jop1.showMessageDialog(null, "Message informatif", "Information", JOptionPane.INFORMATION_MESSAGE);
//Boîte du message préventif
jop2 = new JOptionPane();
jop2.showMessageDialog(null, "Message préventif", "Attention", JOptionPane.WARNING_MESSAGE);
//Boîte du message d'erreur
jop3 = new JOptionPane();
jop3.showMessageDialog(null, "Message d'erreur", "Erreur", JOptionPane.ERROR_MESSAGE);
```



## Les Actions Listner

Ce sont les actions qu'on déclenche lors d'un événement sur un composant (un click, un double click, un click droit, un survol avec la souris...etc).

Dans cette partie du TP, nous allons voir juste les actions qui se déclenchent en cliquant sur un composant, donc on peut faire ces actions aux Menus et MenuItem, aux buttons, aux ComboBox, aux CheckBox et aux JRadioButton.

Prenons l'exemple du bouton btn qu'on a déclaré en haut, son action listner en cliquant est la suivante :

```
btn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        // L'action qui se déclenche en cliquant sur le bouton
    }
});
```

Pour faire cet Actions Listner aux autres composants, il suffit de remplacer le nom btn avec le nom de votre composant.

Une fois que vos composants sont déclarés, il faut les ajouter au panel sur lequel vous voulez les mettre.

Dans cet exemple, on va les mettre dans le panel container.

```
container.add(labNom);  
  
container.add(txta);  
  
container.add(jtf);  
  
container.add(jpsw);  
  
  
container.add(comb);  
  
container.add(radHomme);  
container.add(radFemme);  
  
container.add(jcheck1);  
container.add(jcheck2);  
  
  
container.add(btn);
```

Après que vous ajoutez vos composants au panel global (dans notre exemple c'est le panel container) il faut mettre ce panel dans le ContentPane de la fenêtre comme suit :

```
// On met notre panel global dans le content pane de notre Fenetre  
fen.setContentPane(container);
```

Et enfin, on affiche la fenêtre :

```
// On affiche notre fenetre  
fen.setVisible(true);
```

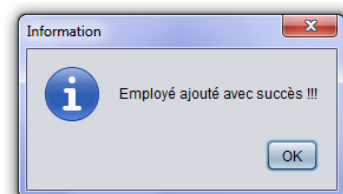
On va se contenter de ces composants dans ce TP, si vous voulez en découvrir plus, faites une autoformation sur Java Swing.

## Exercice :

Soit l'exemple du projet de gestion du personnel de la 2<sup>ème</sup> partie du TP GL.

- Réalisez à l'aide de Java Swing une **seule** interface graphique qui permet d'ajouter un employé ou un chef de service comme suit :


```
run:
l'employé s'appelle Ben Ali Karim et il a 40 ans et il habite à Cité 300 Ighil Ouazoug
Béjaia
```





Ajout d'un employé

## Ajout d'un Employé



Type: Chef Service


Nom: Benhemadi

Prénom: Mouloud

Age: 58

Adresse: Rue Laarbi Touati  
Béjaia

Service: Personnel

 Enregistrer

```

run:
1'employé s'appelle Ben Ali Karim et il a 40 ans et il habite à Cité 300 Ighil Ouazoug
Béjaia
1'employé s'appelle Benhemadi Mouloud et il a 58 ans et il habite à Rue Laarbi Touati
Béjaia
et il gère le servicePersonnel
    
```

