

Série TP 1 (2 séances)

IHM et Java Swing – JWindow, JDialog, et JFrame.

I. Introduction à Java Swing

...

II. Les containers dans Java Swing

Les conteneurs font partie intégrante des composants GUI SWING. Un conteneur fournit un espace où un composant peut être localisé.

Deux types:

- a) Heavyweight Containers (Top-Level): JDialog, JWindow, et JFrame.
- b) Lightweight Containers: Par défaut (ContentPane), JPanel, etc.

III. JWindow

C'est la fenêtre la plus basique. C'est juste un conteneur dont on peut afficher sur l'écran. Il n'a pas de barre de titre, pas de boutons de fermeture/redimensionnement et n'est pas redimensionnable par défaut. L'ajout de toutes ces fonctionnalités est évidemment possible. On utilise surtout les JWindow pour faire des SplashScreen, c'est-à-dire des interfaces d'attente qui se ferment automatiquement.¹

Pour créer une JWindow :

```
JWindow window = new JWindow();
```

A faire :

- Créer un nouveau projet Java sous le nom *IHMTP1*.
- Ajouter une classe main nommée *TestJWindow*. Créer dans cette dernière une instance de la classe *JWindow* nommée *window*, comme mentionné ci-dessus.
- Modifier la taille (largeur, longueur) de *window* en utilisant la méthode *setSize* :

```
window.setSize(300, 200);
```

- Afficher *window* en utilisant la méthode *setVisible* comme suit :

```
window.setVisible(true);
```

- Exécuter. Que remarquez-vous ?

¹ <http://baptiste-wicht.developpez.com/tutoriels/java/swing/>

- Fermer/Arrêter l'application.
- Reprendre le code et le modifier afin de permettre au programme de l'exécuter dans le thread réservé au graphique, appelé EDT (Event Dispatch Thread) :

```
SwingUtilities.invokeLater(new Runnable() {
    public void run() {
        //On crée une nouvelle instance de notre JWindow
        JWindow window = new JWindow();
        window.setSize(300, 200); //On lui donne une taille
        pour qu'on puisse la voir
        window.setVisible(true); //On la rend visible
    }
});
```

- Pour permettre à l'application de se fermer automatiquement après un certain temps prédéfini, on utilise la méthode *sleep* de la classe *Thread* et la méthode *exit* de la classe *System* comme suit :

```
public class TestJWindow {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                JWindow window = new JWindow();
                window.setSize(300, 200);
                window.setVisible(true);
            }
        });

        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {}

        System.exit(0);
    }
}
```

IV. JDialog

C'est une fenêtre destinée aux boîtes de dialogue. Ce type de fenêtre peut être modal, c'est-à-dire qu'elle bloque une autre fenêtre tant qu'elle est ouverte. Elles sont destinées à travailler de pair avec la fenêtre principale *JFrame*.

Pour créer une *JDialog* :

```
JDialog dialog = new JDialog();
```

A faire :

- Reprendre les mêmes étapes que pour *JWindow*, puis exécuter :

```

public class TestJDialog {
    public static void main(String[] args){
        SwingUtilities.invokeLater(new Runnable(){
            public void run(){
                JDialog dialog = new JDialog();
                dialog.setSize(300, 200);
                dialog.setTitle("Première fenêtre");
                dialog.setVisible(true);

                //Pour dire à l'application de se fermer lors du clic sur la croix
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            }
        });
    }
}

```

V. JFrame

C'est une fenêtre destinée à être la fenêtre principale dans une application. Elle n'est dépendante d'aucune autre fenêtre et ne peut pas être modale. Elle a une barre de titre et peut accueillir une barre de menu. Elle possède un bouton de fermeture, un bouton de redimensionnement et un bouton pour l'iconifier.

A faire :

- Créer un nouveau projet Java sous le nom *IHMTP1JFrame*.
- Pour créer une JFrame, ajouter une classe héritant de la classe *JFrame* comme suit :

```

import javax.swing.JFrame;

public class MyJFrame extends JFrame{

    public MyJFrame() {
        super();
    }

}

```

- Pour configurer l'état initial de la fenêtre créée, ajouter une méthode *initJFrame* à la classe *MyJFrame* et appeler celle-ci depuis le constructeur. Utiliser les méthodes suivantes pour l'initialisation :

```

public MyJFrame() {
    super();
    initJFrame();
}

private void initJFrame(){
    this.setTitle("Ma première fenêtre");
    this.setSize(320,240);
    this.setLocationRelativeTo(null);
    setResizable(false);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

```

Voici ce que font les différentes méthodes utilisées :

- **setTitle(String title)** : Cette méthode modifie le titre de la fenêtre. Ce titre va apparaître dans la barre de titre et également dans la barre des tâches.
 - **setSize(int width, int height)** : Cette méthode permet de modifier la taille de la fenêtre. **setLocationRelativeTo(Component c)** : Cette méthode permet de positionner la fenêtre par rapport à un composant. En indiquant un composant null, elle va se placer automatiquement au milieu de l'écran.
 - **setResizable(boolean resizable)** : Cette méthode permet ou non le redimensionnement de la fenêtre.
 - **setDefaultCloseOperation(int operation)** : Cette méthode permet de configurer l'action qui va être exécutée lors de la fermeture de la fenêtre. On peut choisir de faire quitter le programme, de fermer la fenêtre ou de la rendre invisible (ce qui n'est pas pareil)
- Ajouter une classe main nommée *TestJFrame*. Créer dans cette dernière une instance de la classe *MyJFrame* qui devrait s'exécuter dans l'Event Dispatch Thread (en utilisant *SwingUtilities*).

```
public class TestJFrame {  
    public static void main(String[] args){  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                MyJFrame fenetre = new MyJFrame ();  
                fenetre.setVisible(true);  
            }  
        });  
    }  
}
```

- Pour afficher cette instance, utiliser la méthode *setVisible(true)*.
- Exécuter et afficher la fenêtre.

Ajouter une barre de menu à une JFrame

En Swing, pour mettre un menu sur une JFrame, il faut employer une **JMenuBar**, qui sera composé de **JMenu** et de **JMenuItem**. Un *JMenu* étant un bouton que l'on trouve directement dans la barre de tâches et il possède des *JMenuItem* qui sont eux des éléments du menu.

- Reprendre la méthode *initJFrame* et créer une barre de menu **JMenuBar** comme suit :

```
JMenuBar menuBar = new JMenuBar();
```

- Créer un **JMenu** *compte* et l'ajouter au *menuBar* :

```
JMenu compte = new JMenu("mon compte");  
menuBar.add(compte) ;
```

- Ajouter au *JMenu compte* un *JMenuItem param* :

```
JMenuItem param = new JMenu("Paramètres");  
compte.add(param) ;
```

- Ajouter un trait séparateur séparant les différents items :

```
compte.addSeparator() ;
```

- Ajouter au *JMenu compte* un deuxième *JMenuItem exit* :

```
JMenuItem exit = new JMenu("Se Déconnecter");  
compte.add(exit) ;
```

- Définir *menuBar* comme barre de menu de la fenêtre *MyJFrame* :

```
this.setMenuBar(menuBar) ;
```

- Exécuter et afficher.

Exercice :

- Modifier *MyJFrame* en lui créant la barre de menu ci-dessous :

