Exercice1:Un exemple de fenêtre autonome

La classe de base d'une application est la classe JFrame. Son rôle est équivalent à la classe Frame de l'AWT et elle s'utilise de la même façon.

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

public class swing1 extends JFrame {
   public swing1() {
      super("titre de l'application");
      WindowListener l = new WindowAdapter() {
         public void windowClosing(WindowEvent e){
            System.exit(0);
         }
      };
      addWindowListener(l);
      setSize(200,100);
      setVisible(true);
   }
   public static void main(String [] args){
      JFrame frame = new swing1();
   }
}
```

Exercice 2 :Les composants Swing

Il existe des composants Swing équivalents pour chacun des composants AWT avec des constructeurs semblables. De nombreux constructeurs acceptent comme argument un objet de type Icon, qui représente une petite image généralement stockée au format Gif. Le constructeur d'un objet Icon admet comme seul paramètre le nom ou l'URL d'un fichier graphique

```
import javax.swing.*;
import java.awt.event.*;
public class swing3 extends JFrame {
  public swing3() {
     super("titre de l'application");
      WindowListener l = new WindowAdapter() {
         public void windowClosing(WindowEvent e){
            System.exit(0);
         }
      };
      addWindowListener(l);
      ImageIcon img = new ImageIcon("tips.gif");
      JButton bouton = new JButton("Mon bouton",img);
      JPanel panneau = new JPanel();
      panneau.add(bouton);
```

```
    setContentPane(panneau);
    setSize(200,100);
    setVisible(true);
  }
 public static void main(String [] args){
    JFrame frame = new swing3();
 }
}
```

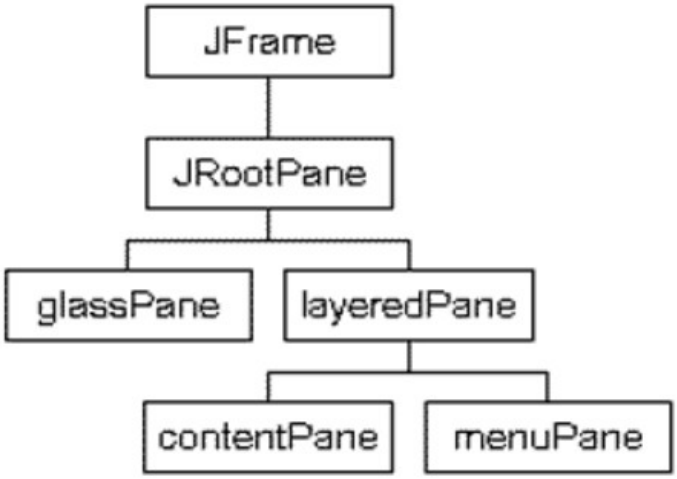
Exercice3 : La classe JFrame

JFrame est l'équivalent de la classe Frame de l'AWT : les principales différences sont l'utilisation du double buffering qui améliore les rafraichissements et l'utilisation d'un panneau de contenu (contentPane) pour insérer des composants (ils ne sont plus insérés sans le JFrame mais dans l'objet contentPane qui lui est associé). Elle représente une fenêtre principale qui possède un titre, une taille modifiable et éventuellement un menu.

```
import javax.swing.*;
public class TestJFrame1 {
  public static void main(String argv[]) {
    JFrame f = new JFrame("ma fenetre");
    f.setSize(300,100);
    f.setVisible(true);
  }
}
```

```
import javax.swing.*;
import java.awt.event.*;
public class swing2 extends JFrame {
   public swing2() {
     super("titre de l'application");
      WindowListener l = new WindowAdapter() {
        public void windowClosing(WindowEvent e){
          System.exit(0);
        }
      };
     addWindowListener(l);
     JButton bouton = new JButton("Mon bouton");
     JPanel panneau = new JPanel();
     panneau.add(bouton);
     setContentPane(panneau);
     setSize(200,100);
     setVisible(true);
   }
   public static void main(String [] args){
     JFrame frame = new swing2();
   }
}
```

Tous les composants associés à un objet JFrame sont gérés par un objet de la classe JRootPane. Un objet JRootPane contient plusieurs Panes. Tous les composants ajoutés au JFame doivent être ajoutés à un des Pane du JRootPane et non au JFrame directement. C'est aussi à un de ces Panes qu'il faut associer un layout manager si nécessaire.



Le Pane le plus utilisé est le ContentPane. Le Layout manager par défaut du contentPane est BorderLayout. Il est possible de le changer :

……..

```
f.getContentPane().setLayout(new FlowLayout());
```

……..

Exercice4

```
import javax.swing.*;

public class TestJFrame2 {

  public static void main(String argv[]) {

    JFrame f = new JFrame("ma fenetre");
    f.setSize(300,100);
    JButton b =new JButton("Mon bouton");
    f.getContentPane().add(b);
    f.setVisible(true);
  }
}
```

Le JRootPane se compose de plusieurs éléments :
glassPane : un JPanel par défaut
layeredPane qui se compose du contentPane (un JPanel par défaut) et du menuBar

(un objet de type JMenuBar)
Le glassPane est un JPanel transparent qui se situe au-dessus du layeredPane. Le glassPane peut être n'importe quel composant : pour le modifier il faut utiliser la méthode setGlassPane() en fournissant le composant en paramètre.
Le layeredPane regroupe le contentPane et le menuBar.
Le contentPane est par défaut un JPanel opaque dont le gestionnaire de présentation est un BorderLayout. Ce panel peut être remplacé par n'importe quel composant grâce à la méthode setContentPane().

Exercice5 La personnalisation de l'icône

La méthode setIconImage() permet de modifier l'icône de la JFrame.

```
import javax.swing.*;

public class TestJFrame4 {

  public static void main(String argv[]) {

    JFrame f = new JFrame("ma fenetre");
    f.setSize(300,100);
    JButton b =new JButton("Mon bouton");
    f.getContentPane().add(b);
    f.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);

    ImageIcon image = new ImageIcon("book.gif");
    f.setIconImage(image.getImage());
    f.setVisible(true);

  }
}
```