



## TP 4

# Conteneurs intermédiaires

Swing propose plusieurs conteneurs intermédiaires, le but d'un conteneur intermédiaire est de faciliter le positionnement du composant que contient l'application. Il existe plusieurs types de conteneurs intermédiaires, comme le panel (Classe JPanel), les panneaux de défilements (classe JScrollPane), les panneaux à onglets (classe JTabbedPane), etc...

### Conteneur - JPanel

JPanel est un conteneur générique qui permet de contenir d'autres composants dont d'autres JPanel. Son gestionnaire de placement par défaut est FlowLayout. L'exemple suivant permet de voir comment utiliser un conteneur intermédiaire de type JPanel.

```
class Adresse extends JPanel {

    public Adresse() {
        super();
        JLabel eNom = new JLabel("Nom : ");
        JLabel ePrenom = new JLabel("Prénom : ");
        JLabel eAdresse = new JLabel("Adresse : ");
        JTextField tNom = new JTextField(8);
        JTextField tPrenom = new JTextField(8);
        JTextArea tAdresse = new JTextArea("Lahoucine", 4, 20);
        add(eNom);
        add(tNom);
        add(ePrenom);
        add(tPrenom);
        add(eAdresse);
        add(tAdresse);
    };
}

public class JPanelExemple {

    public static void main(String[] args) {
        JFrame cadre = new JFrame();
        Adresse identite = new Adresse();
        cadre.setContentPane(identite); // ou cadre1.add(identite);
        cadre.pack();
        cadre.setDefaultCloseOperation(3);
        cadre.setTitle("Identité");
        cadre.setVisible(true);
    }
}
```

- Compilez puis exécutez ce programme.



## Conteneur - JScrollPane

Un JScrollPane est un conteneur qui offre des ascenseurs, il permet de visionner un composant plus grand que lui comme par exemple insertion d'une photo à haute résolution dans une étiquette (problème que nous avons rencontré dans le TP précédent). L'exemple suivant permet de voir comment utiliser un conteneur intermédiaire de type JScrollPane.

```
public class JScrollPaneExemple {
    public JScrollPaneExemple() {
        JFrame cadre = new JFrame();
        JLabel eImage = new JLabel(new ImageIcon("./img/nImage.JPG"));
        JScrollPane pDefilement = new JScrollPane(eImage);
        cadre.setContentPane(pDefilement); // ou cadre.add(pDefilement);
        cadre.setTitle("Le chemin par Google Map !");
        cadre.setSize(400, 400);
        cadre.setDefaultCloseOperation(3);
        cadre.setResizable(false); // true laisse la possibilité de
                                   // redimensionnement
        cadre.setVisible(true);
    }

    public static void main(String[] args) {
        new JScrollPaneExemple();
    }
}
```

- Compilez puis exécutez ce programme.

## Conteneur - JSplitPane

Un JSplitPane est un panel coupé en deux par une barre de séparation, cette séparation peut être horizontale ou verticale. Un JSplitPane accueille deux composants. L'exemple suivant illustre comment utiliser un conteneur intermédiaire de type JSplitPane.

```
public class JSplitPaneExemple {
    public JSplitPaneExemple() {
        JFrame cadre = new JFrame();
        JLabel eImage = new JLabel(new ImageIcon("./img/nImage.JPG"));
        JTextArea ta = new JTextArea(texte, 20, 10);
        ta.setLineWrap(true); // passage automatique à la ligne
        JSplitPane pSeparation = new
            JSplitPane(JSplitPane.HORIZONTAL_SPLIT, ta, eImage);
                                   // ou VERTICAL_SPLIT
        cadre.setContentPane(pSeparation); // ou cadre.add(pSeparation);
        cadre.setTitle("Le chemin par Google Map !");
        cadre.pack();
        cadre.setDefaultCloseOperation(3);
        cadre.setResizable(true); // autoriser le redimensionnement
        cadre.setVisible(true);
    }

    static String texte = "Il ne faut pas se fier aveuglément à la
```



technologie. Un chauffeur qui a été guidé par la géolocalisation de Google !";

```
public static void main(String[] args) {  
    new JSplitPaneExemple();  
}  
}
```

- Compilez puis exécutez ce programme.

## Conteneur - JTabbedPane

Le JTabbedPane est un conteneur qui contient d'autres conteneurs; la navigation entre ces différents conteneurs se faisant par des onglets. Les onglets de navigation peuvent être positionnés à gauche, en haut, à droite ou en bas. Voici un exemple de JTabbedPane avec deux onglets identité et études.

```
public class JTabbedPaneExemple1 {  
    public JTabbedPaneExemple1() {  
        JFrame cadre = new JFrame();  
        JTabbedPane pOnglets = new JTabbedPane(JTabbedPane.TOP);  
                                                                    //par défaut  
  
        JLabel nom = new JLabel(" Nom : ");  
        JTextField tNom = new JTextField(10);  
        JLabel prenom = new JLabel(" Prénom : ");  
        JTextField tPrenom = new JTextField(10);  
        JPanel identite = new JPanel();  
        identite.add(nom);  
        identite.add(tNom);  
        identite.add(prenom);  
        identite.add(tPrenom);  
        pOnglets.addTab(" Identité ", identite);  
                                                                    // ajout d'un 1er panneau  
  
        JLabel niveau = new JLabel(" Niveau : ");  
        JTextField tNiveau = new JTextField(10);  
        JLabel annee = new JLabel(" Année : ");  
        JTextField tAnnee = new JTextField(10);  
        JPanel etudes = new JPanel();  
        etudes.add(niveau);  
        etudes.add(tNiveau);  
        etudes.add(annee);  
        etudes.add(tAnnee);  
        pOnglets.addTab(" Etudes ", etudes);  
                                                                    // ajout d'un 2ème panneau  
  
        cadre.add(pOnglets);  
        cadre.setTitle("Saisie d'informations");  
        cadre.pack(); // la taille depend du plus grand onglet  
        cadre.setDefaultCloseOperation(3);  
        cadre.setResizable(false);  
                                                                    // bloquer le changement de la taille  
        cadre.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        new JTabbedPaneExemple1();  
    }  
}
```



```
}  
}
```

- Compilez puis exécutez ce programme.
- Voici un autre exemple de JTabbedPane avec quatre onglets dont chaque onglet contient une image.

```
public class JTabbedPaneExemple2 {  
    public JTabbedPaneExemple2() {  
  
        JFrame cadre = new JFrame();  
        JTabbedPane pOnglets = new JTabbedPane();  
        JLabel eImage1 = new JLabel(new ImageIcon("./img/1.JPG "));  
        JPanel image1 = new JPanel();  
        image1.add(eImage1);  
        pOnglets.addTab(" Photo 1 ", image1);  
        JLabel eImage2 = new JLabel(new ImageIcon("./img/2.JPG "));  
        JPanel image2 = new JPanel();  
        image2.add(eImage2);  
        pOnglets.addTab(" Photo 2 ", image2);  
        JLabel eImage3 = new JLabel(new ImageIcon("./img/3.JPG "));  
        JPanel image3 = new JPanel();  
        image3.add(eImage3);  
        pOnglets.addTab(" Photo 3 ", image3);  
        JLabel eImage4 = new JLabel(new ImageIcon("./img/4.JPG "));  
        JPanel image4 = new JPanel();  
        image4.add(eImage4);  
        pOnglets.addTab(" Photo 4 ", image4);  
        cadre.setContentPane(pOnglets);  
        cadre.setTitle("Album photos");  
        cadre.pack();  
        cadre.setDefaultCloseOperation(3);  
        cadre.setResizable(true);  
        cadre.setVisible(true);  
    }  
};
```

- Compilez puis exécutez ce programme.

## Conteneur – JToolBar

JToolBar est un conteneur intermédiaire qui permet de réaliser une barre d'outils contenant les actions les plus utilisées d'une application. L'utilisateur peut déplacer ce composant dans une fenêtre séparée.

```
public class JToolBarExemple1 {  
    public JToolBarExemple1() {  
  
        JButton bOuvrir = new JButton(new ImageIcon("1.JPG"));  
        JButton bEnregistrer = new JButton(new ImageIcon("2.JPG"));  
        JButton bCouper = new JButton(new ImageIcon("3.JPG"));  
        JButton bCopier = new JButton(new ImageIcon("4.JPG"));  
        JButton bColler = new JButton(new ImageIcon("5.JPG"));  
        bOuvrir.setToolTipText("Ouvrir un fichier");  

```



```
// ajout de bulles d'aide
bEnregistrer.setToolTipText("Enregistrer le fichier");
bCouper.setToolTipText("Couper vers le presse papier");
bCopier.setToolTipText("Copier vers le presse papier");
bColler.setToolTipText("Coller depuis le presse papier");
JToolBar tb = new JToolBar();
tb.add(bOuvrir);
tb.add(bEnregistrer);
tb.addSeparator(); // ajout d'une barre de séparation
tb.add(bCouper);
tb.add(bCopier);
tb.add(bColler);
JPanel panneau = new JPanel();
panneau.setLayout(new BorderLayout());
panneau.add(tb, BorderLayout.NORTH);
JFrame cadre = new JFrame();
cadre.add(panneau);
cadre.setTitle("Barre d'outils");
cadre.setSize(600, 400);
cadre.setDefaultCloseOperation(3);
cadre.setVisible(true);
};

public static void main(String[] args) {
    new JToolBarExemple1();
}
}
```

- Compilez puis exécutez ce programme.
- Pour déplacer cette barre de menu, utilisez la méthode `setFloatable(boolean)`.
- On peut construire une barre de menu avec des composants de différents types. Soit le programme suivant :

```
public class JToolBarExemple2 {
    public JToolBarExemple2() {
        JToolBar tb = new JToolBar();
        tb.setFloatable(true); // true par défaut
        tb.add(new JButton("Ouvrir"));
        tb.add(new JButton("Enregistrer"));
        tb.addSeparator();
        Icon ic = new ImageIcon("./img/1.JPG");
        tb.add(new JButton("Java", ic));
        tb.add(new JRadioButtonMenuItem("ON/OFF"));
        JFrame cadre = new JFrame();
        cadre.add(tb, BorderLayout.PAGE_END);
        // en bas de page ou par défaut en haut : PAGE_START
        cadre.setTitle("Barre d'outils (2)");
        cadre.setSize(600, 200);
        cadre.setDefaultCloseOperation(3);
        cadre.setVisible(true);
    };
}
```

- Compilez puis exécutez ce programme.



## Conteneur – JDesktopPane

JDesktopPane est un conteneur qui est fait pour gérer les JInternalFrame. Ces composants peuvent être déplacés, redimensionnés, minimisés, etc. Il dérive de JLayeredPane (voir plus tard) et il n'utilise pas de gestionnaire de disposition. Soit le programme suivant :

```
public class JDesktopPaneExemple1 {
    public static void main(String[] args) {
        boolean resizable = true;
        boolean closeable = true;
        boolean maximizable = true;
        boolean iconifiable = true;
        String title = "Cadre interne";
        JInternalFrame icadre = new JInternalFrame(title,
            resizable, closeable, maximizable, iconifiable);
        icadre.setSize(300, 300);
        icadre.setVisible(true);
        icadre.add(new JTextArea());
        JDesktopPane desktop = new JDesktopPane();
        desktop.add(icadre);
        JFrame f = new JFrame("Cadre externe");
        f.add(desktop);
        f.setSize(400, 400);
        f.setVisible(true);
    }
}
```

- Compilez puis exécutez ce programme.
- Notez qu'on peut aussi créer plusieurs JInternalFrame. Soit le programme suivant :

```
class CadreMDI extends JInternalFrame {

    public CadreMDI() {
        super("", true, true, true, true);
    }

    public void setImage(String nomImage) {
        JPanel p = new JPanel();
        JLabel uneImage = new JLabel(new ImageIcon(nomImage));
        p.add(uneImage);
        this.add(p);
        this.setTitle(nomImage);
    }
}

public class JDesktopPaneExemple2 extends JDesktopPane {
    CadreMDI[] images;
    String[] nomsFichier = { "../img/1.JPG", "../img/2.JPG",
        "../img/3.JPG", "../img/4.JPG",
        "../img/5.JPG" };

    public JDesktopPaneExemple2() {
        super();
        images = new CadreMDI[nomsFichier.length];
    }
}
```



```
        for (int i = 0; i < nomsFichier.length; i++) {
            images[i] = new CadreMDI();
            images[i].setSize(200, 200);
            images[i].setLocation(100 * i, 50 * i); //effet 3D!
            images[i].setImage(nomsFichier[i]);
            images[i].setVisible(true);
            this.add(images[i]);
        }
    }
    public static void main(String[] args) {
        JFrame f = new JFrame();
        f.add(new JDesktopPaneExemple2());
        f.setDefaultCloseOperation(3);
        f.setVisible(true);
        f.setSize(500, 500);
        f.setTitle("Afficheur d'images");
    }
}
```

- Compilez puis exécutez ce programme.

## Conteneur – JLayeredPane

JLayeredPane est un conteneur qui permet de gérer des composants disposés sur plusieurs couches. À chaque composant de ce conteneur sont associés deux entiers appelés niveau (layer) et position (position). Ces deux entiers définissent l'ordre d'empilement des composants contenus dans le panneau. Soit l'exemple suivant :

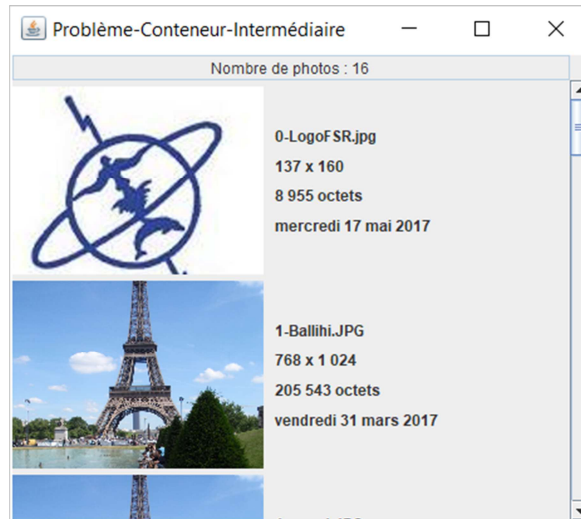
```
public class JLayeredPaneExemple extends JFrame {
    public JLayeredPaneExemple() {
        super("Test JLayeredPane");
        setSize(300, 200);
        setDefaultCloseOperation(3);
        JLayeredPane lp = getLayeredPane();
        JButton b3 = new JButton("3");
        b3.setBackground(Color.white);
        b3.setBounds(60, 100, 60, 60);
        JButton b2 = new JButton("2");
        b2.setBackground(Color.yellow);
        b2.setBounds(40, 60, 60, 60);
        JButton b1 = new JButton("1");
        b1.setBackground(Color.cyan);
        b1.setBounds(20, 20, 60, 60);
        // Place les boutons dans les différentes couches
        lp.add(b1, new Integer(1));
        lp.add(b2, new Integer(2));
        lp.add(b3, new Integer(3));
    }
}
```

- Compilez puis exécutez ce programme.



## Application – Galerie d'images

On souhaite faire une application java qui va lister verticalement la vignette ainsi les propriétés d'un ensemble d'images, qui se trouve dans un dossier donné, comme illustré par l'interface suivante.



Pour faire cette application, tous d'abord, vous représentez les propriétés de chaque image sous forme d'une table html, vous pouvez donc déclarer **tabHtml** comme attribut de la classe de type **String** initialisé comme suit ;

```
tabHtml = "<html><table>" + "<tr><td>{0}</td></tr>" +  
        "<tr><td>{1} x {2}</td></tr>" + "<tr><td>{3} octets</td></tr>" +  
        "<tr><td>{4, date, full}</td></tr>" + "</table></html>";
```

Puis, vous formatez les propriétés de chaque image sous forme d'une chaîne de caractère :

```
description = MessageFormat.format(tabHtml, fichier.getName(),  
                                   largeur, hauteur, fichier.length(), fichier.lastModified());
```

Ensuite, pour récupérer la vignette de chaque image en utilisant la fonction suivante :

```
private Icon vignette(File fichier) throws IOException {  
    BufferedImage source = ImageIO.read(fichier);  
    largeur = source.getWidth();  
    hauteur = source.getHeight();  
    BufferedImage image = new BufferedImage(200, 150,  
                                             source.getType());  
  
    double ratio = 200 / (double) largeur;  
    AffineTransform retailler =  
        AffineTransform.getScaleInstance(ratio, ratio);  
    int interpolation = AffineTransformOp.TYPE_BICUBIC;  
    AffineTransformOp retaillerImage = new  
        AffineTransformOp(retailler, interpolation);  
    retaillerImage.filter(source, image);  
    return new ImageIcon(image);  
}
```

Enfin, vous rassemblez la description et la vignette dans une étiquette :

```
JLabel etiquette = new JLabel(description, icone, JLabel.LEFT);
```