

TP 4 JAVA – Layout Managers

Java offre trois types d'éléments graphiques

Les « Components » (composants ou contenus) : Constituent différents éléments de l'affichage (boutons, barres de menus, etc.). *Ex:* JButton, JLabel, JTextArea, JCheckbox, etc.

Les « Containers » (contenants, qui sont eux-mêmes des composants): destinés à accueillir des composants, gèrent l'affichage des composants (*Ex:* JFrame, JPanel, JScrollPane)

Les « LayoutManagers »: Gèrent la disposition des composants au sein d'un conteneur **JPanel** ou **Panel**, **JFrame** ou **Frame**, **Applet**, **JScrollPane** ...

Gestionnaire de Placement (Layout Manager)

Le gestionnaire de placement a pour rôle de gérer la position des composants d'une interface. Différents gestionnaires de placement existent.

BorderLayout gestionnaire de placement utilise les points cardinaux pour placer les composants

setLayout (new BorderLayout()) : divise le conteneur en 5 zones: N, S, E, W, C et permet d'ajouter un composant différent dans chaque zone (il n'est pas obligatoire d'utiliser les 5 zones)
add(Component c, "North") → Ajoute le composant « c » au « Nord »

Exemple : (on utilise JFrame, JPanel, ...)

```
JFrame fenetre = new JFrame("BorderLayoutExemple");
fenetre.setLayout(new BorderLayout());
fenetre.add(new JButton("Un"), BorderLayout.NORTH);
```

Les directions ou paramètres utilisées pour BorderLayout sont: EAST, SOUTH, NORTH, WEST, CENTER, PAGE_START, PAGE_END, LINE_START, LINE_END

FlowLayout permet de placer les composants horizontalement l'un à côté de l'autre, de gauche à droite par défaut

```
JFrame fenetre = new JFrame("JFrame");
fenetre.getContentPane().setLayout(new FlowLayout(FlowLayout.CENTER));
```

GridLayout découpe l'espace du conteneur en une grille (ou matrice de lignes et de colonnes) en remplissant les cases de gauche à droite et de haut en bas

setLayout (new GridLayout(x,y)) → Divise le conteneur en x lignes et y colonnes

Les composants ajoutés viennent remplir la grille (avec **GridLayout**) de gauche à droite et de haut en bas → **add(Component c)**: ajoute le composant « c » dans la première case disponible

```
JPanel controls = new JPanel(); // ou JFrame, ou Frame ou Panel ...
controls.setLayout(new GridLayout(2,3));
```

CardLayout permet à plusieurs composants d'occuper le même espace à l'écran.

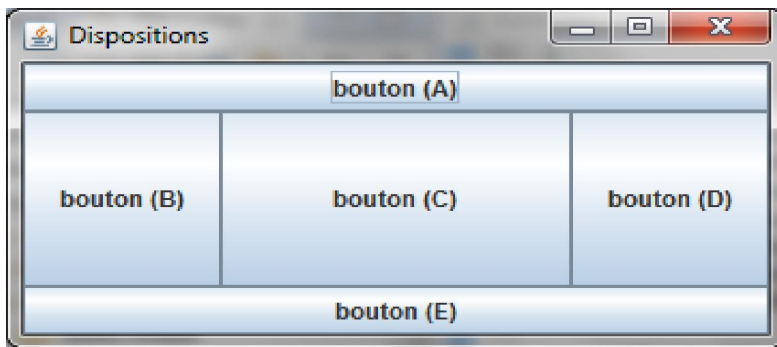
GridBagLayout place les composants sur une grille de lignes et colonnes de taille variable contrairement à GridLayout.

Disposition absolue un composant peut être disposé de façon absolue en ne fixant aucun gestionnaire au conteneur (ex: **JFrame f = new JFrame(); f.setLayout(null);**

bouton1.setBounds (x_debut, y_debut, largeur, hauteur);)

Remarque: dans un **Frame (JFrame)**, on peut ajouter plusieurs panels (donc, plusieurs **LayoutManagers**), et associer un **Layout Manager** avec chaque **Jpanel** (ou **Panel**).

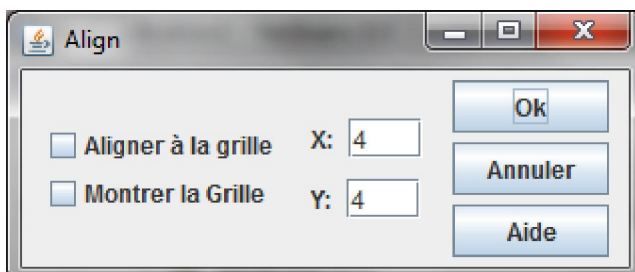
Exercice 1 : réalisez avec **BorderLayout** le programme Java qui affiche la fenêtre ci-dessous (sans NetBeans) :



Exemple de code à utiliser (dans le constructeur) :

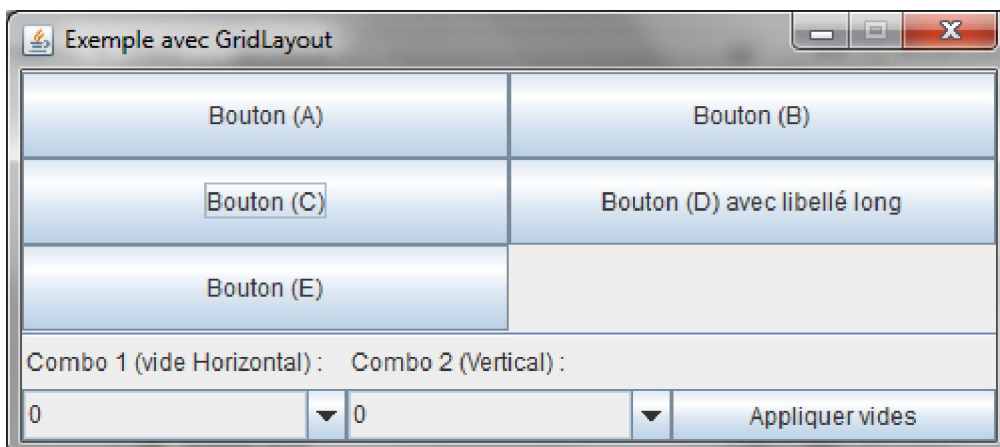
```
JButton bouton = new JButton("Bouton (A) debut de page");
frame.add(bouton, BorderLayout.PAGE_START);
```

Exercice 2 : se servir de JFrame, JPanel (on peut utiliser plusieurs JPanel dans un JFrame), avec BorderLayout, FlowLayout et éventuellement GridLayout pour réaliser l'interface suivante (utiliser JLabel, JCheckBox, JTextField et JButton):



Exemple: `JTextField t1 = new JTextField("04", 5); // 5: largeur du textfield`
`JCheckBox c1 = new JCheckBox ("Montrer la Grille");`

Exercice 3 : utiliser GridLayout (et éventuellement FlowLayout, BorderLayout) et les composants (JLabel, JComboBox) pour afficher l'interface suivante (on peut aussi utiliser plusieurs JPanel) :



Exercice 4 : utiliser **SetBounds** pour afficher une fenêtre de saisie (nom, prénom, date de naissance) avec les boutons 'OK' et 'Annuler'. Utilisez 3 labels, 3 zones de saisie (utiliser si possible JFormattedText) et 2 boutons. Utilisez des Panels pour améliorer la présentation de l'interface (couleurs de fond, d'affichage ...)