

# IHM avec JAVA

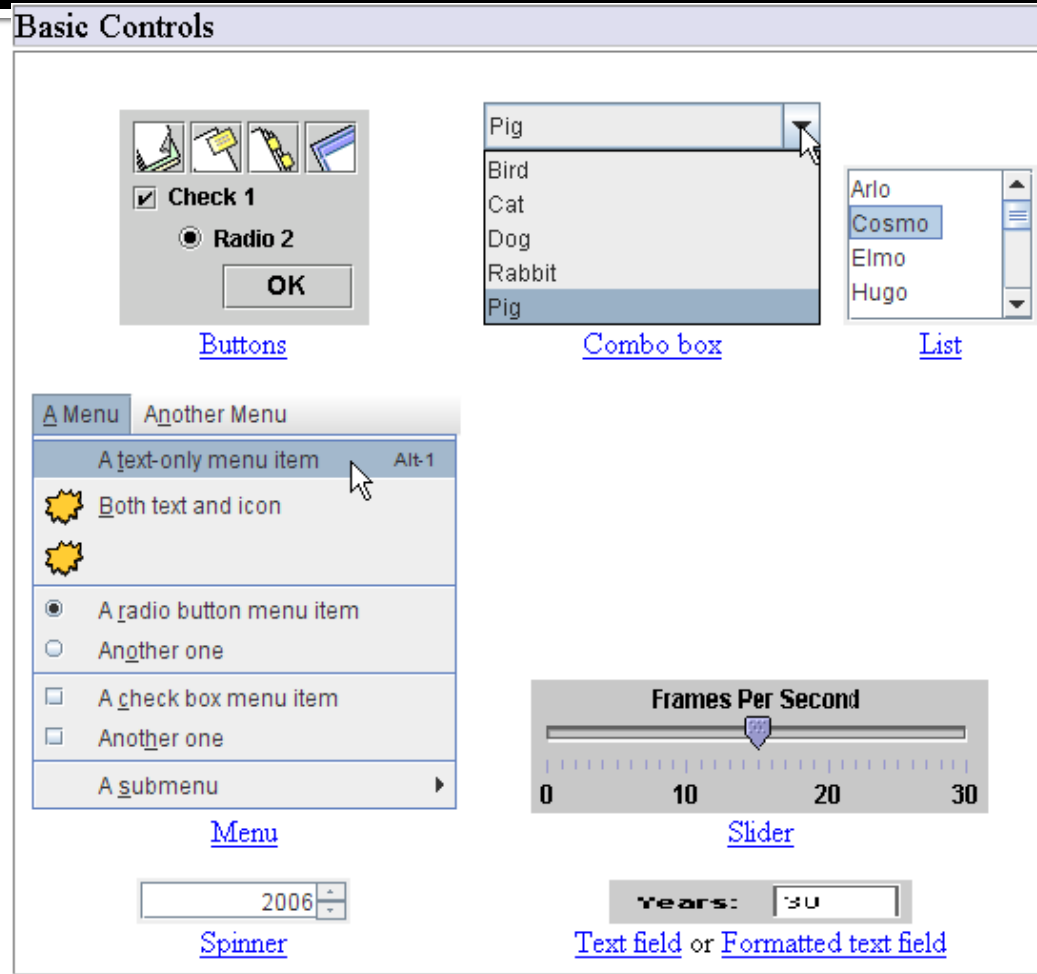
Isabelle Debled-Rennesson  
debled@loria.fr

# Programme du cours du 4 avril

---

- Quelques composants supplémentaires
  - Zones de saisie textuelles (JTextField)
  - Liste de choix (JComboBox)
- Gestionnaires de placement

# Composants Swing



- Pour visualiser tous les composants Swing et avoir des exemples d'utilisation :

<http://docs.oracle.com/javase/tutorial/uiswing/components/index.html>

# Quelques composants

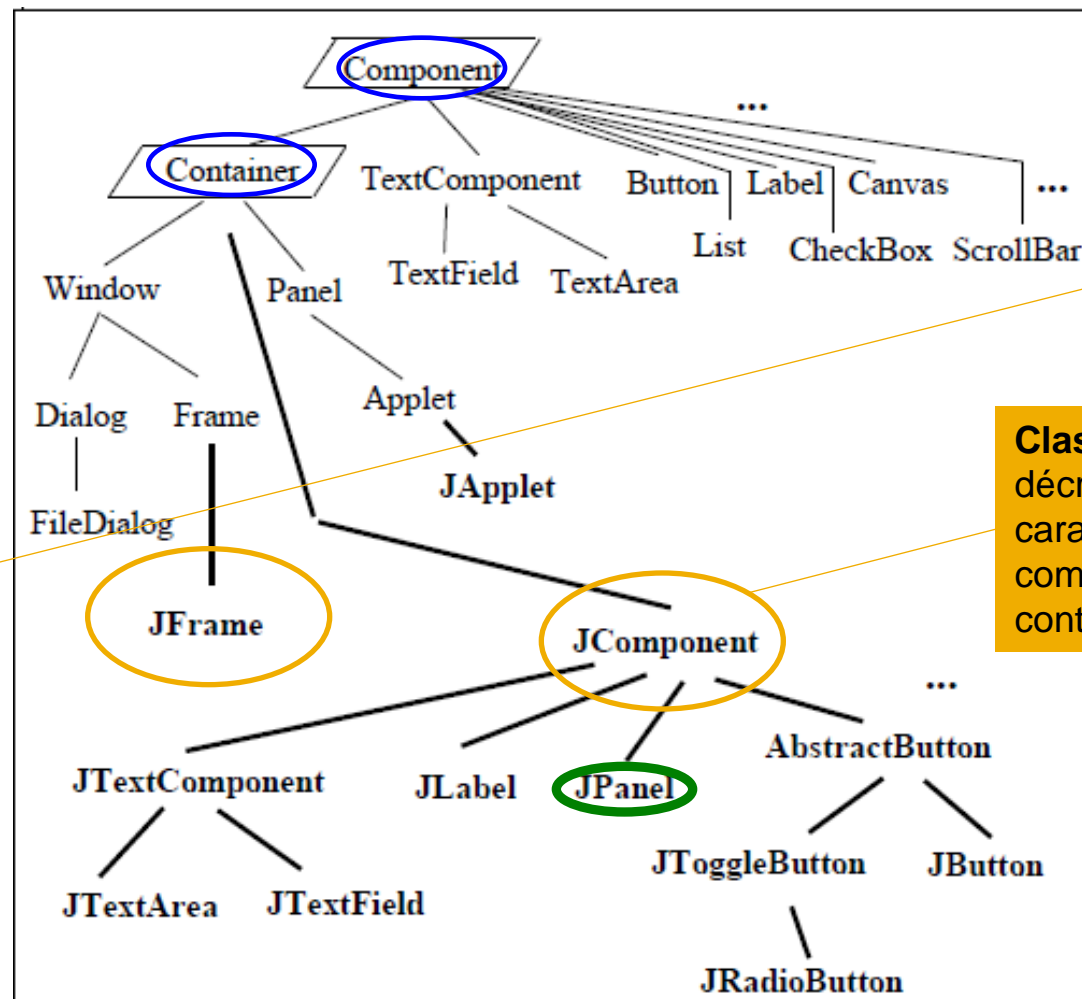
- **JPanel** : panneau destiné à contenir des composants chaque panneau doit être rempli avant d'être placé dans un panneau plus grand
- **JLabel** : étiquette contenant un texte et/ou une icône
- **JButton** : bouton
- **TextField** : zone d'affichage de texte
- **TextArea** : zone d'affichage et de saisie de texte
- **ScrollPane** : panneau avec barres de défilement si nécessaire
- **CheckBox** : cases à cocher
- **RadioButton** : groupes de cases à cocher
- **ComboBox** : liste de choix déroulante
- **OptionPane** : boîtes de dialogues standards
- **JMenu** : menu déroulant

# Composants, conteneurs et gestionnaires

- **Composant** (élément d'une classe héritant de **Component** et **JComponent**) élément susceptible d'apparaître sur une interface utilisateur
  - bouton, liste à défilement, menu, case à cocher, champ de texte, fenêtre, curseur, etc.
- **Conteneur** (élément d'une classe héritant de **Container** ) élément susceptible de contenir plusieurs composants
- **Gestionnaire de présentation (Layout)** dispose et dimensionne les composants d'un conteneur.
  - appelé aussi gestionnaire de mise en page, gestionnaire de placement ou gestionnaire de géométrie

# Liens d'héritage (partiels) des composants et conteneurs

Awt



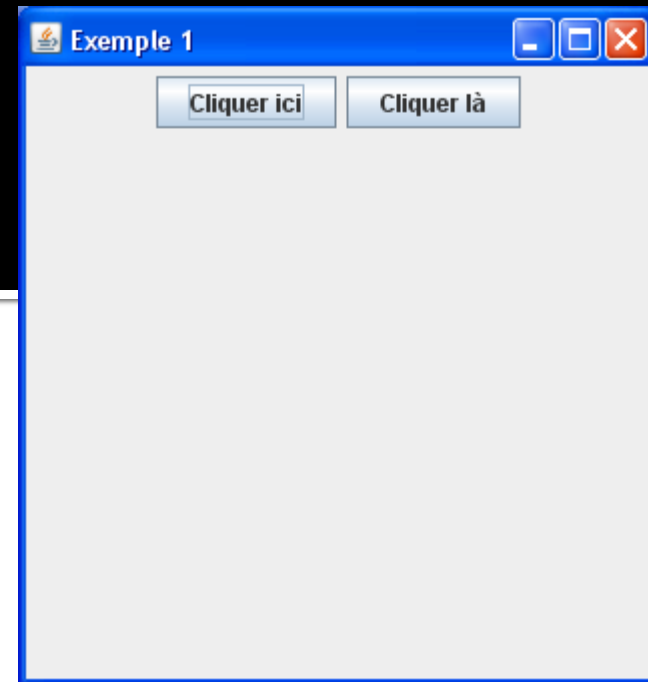
Swing

**Classe abstraite** décrivant les caractéristiques communes à tous les contrôles Swings

# Les conteneurs JPanel

- Ils peuvent regrouper plusieurs composants et être imbriqués
- Déclaration et construction d'un JPanel
  - *JPanel p= new JPanel();*
  - Un autre constructeur sera vu plus tard
- Pour poser un composant dans un JPanel p
  - *p.add(nomVariableComposant);*

# Exemple 1 -Ex1Bouton.java-



```
//...
```

```
class FrameEx1Bouton extends JFrame {
```

```
    private JButton bout1,bout2; //declarations
    private JPanel p;
```

```
    public FrameEx1Bouton(String titre){
```

```
        super(titre);
```

```
        bout1=new JButton("Cliquer là"); //créations des composants
```

```
        bout2= new JButton("Cliquer ici");
```

```
        p=new JPanel();
```

```
        p.add(bout2); //Ajout des boutons dans le JPanel
```

```
        p.add(bout1);
```

```
        setContentPane(p);// ou this.add(p);//Mise en relation du JPanel avec la
```

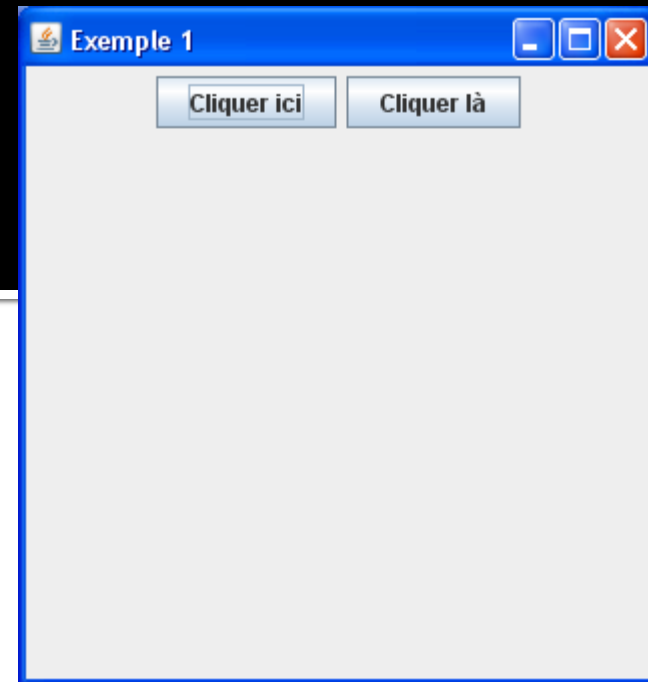
```
        JFrame
```

```
    }
```

```
}
```



# Exemple 1 -Ex1Bouton.java-



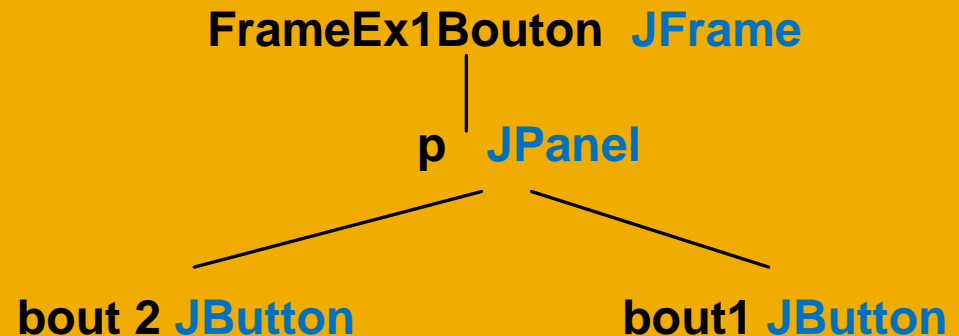
```
//...
```

```
class FrameEx1Bouton extends JFrame {
```

```
    private JButton bout1,bout2; //declarations  
    private JPanel p;
```

```
    public FrameEx1Bouton(String titre)  
    {  
        super(titre);  
        bout1=new JButton("Cliquer ici");  
        bout2= new JButton("Cliquer là");  
        p=new JPanel();  
        p.add(bout2); //Ajout des boutons  
        p.add(bout1);  
        setContentPane(p); // ou add(p);  
    }  
}
```

## Arborescence des composants



# Exemple 3 –Ex3Bouton.java-



//...

```
class FrameEx3Bouton extends JFrame {
    JButton bout1,bout2,bout3,bout4; //declarations
    JPanel p1,p2;

    public FrameEx3Bouton(String titre){
        super(titre);
        bout1=new JButton("Cliquer là"); //créations des composants
        bout2= new JButton("Cliquer ici");
        bout3= new JButton("Info");
        bout4= new JButton("Attention");
        p1=new JPanel();
        p1.setBackground(Color.blue);
        p2=new JPanel();
        p2.setBackground(Color.red);

        p1.add(bout2); //Ajout des boutons dans le JPanel p1
        p1.add(bout1);
        p2.add(bout3); //Ajout des boutons dans le JPanel p2
        p2.add(bout4);
        p1.add(p2); //Ajout du JPanel p2 dans le JPanel p1
        setContentPane(p1); // ou this.add(p1);
    }
}
```

# Autres composants

---

- Zone de saisie de texte sur une ligne
  - JTextField et JPasswordField
- Zone de saisie sur plusieurs lignes
  - JTextArea
- Label
  - JLabel

# TextField

## ■ Constructeurs

- `public JTextField()`
- `public JTextField(String text)`
  - Construit une zone de texte initialisée avec le texte passé en paramètre
- `public JTextField(int columns)`
  - Construit une zone de texte vide du nombre de caractères indiqué en paramètre
- `public JTextField(String text, int columns)`
  - Construit une zone de texte contenant le texte passé en paramètre, sa taille correspond au nombre de caractères indiqué en paramètre

Idem pour **JPasswordField**

# JTextArea

## ■ Constructeurs

- `public JTextArea(int rows, int columns)`
  - Construit une zone de texte sur plusieurs lignes et colonnes indiqués en paramètre
- `public JTextArea(String text, int rows, int columns)`
  - Construit une zone de texte sur plusieurs lignes et colonnes indiqués en paramètre avec un texte initial

# JLabel

## ■ Constructeurs

- `public JLabel(String text)`
  - Crée un JLabel avec le texte spécifié en paramètre. Le label est centré verticalement.
- `public JLabel(String text, int horizontalAlignment)`
  - Crée un JLabel avec le texte spécifié en paramètre et l'alignement horizontale spécifié en paramètre. Le label est centré verticalement.
  - **Paramètres:**
    - `horizontalAlignment` : LEFT, CENTER, RIGHT (de la classe `SwingConstants`)

# Les composants textuels

- Lecture et écriture dans les composants JLabel, JTextField, JTextArea, JPasswordField, ...
  - setText et getText méthodes héritées de JTextComponent

```
String s;...
```

```
nomComp.setText(s); //place la chaîne s dans le  
                    //composant nomComp
```

```
s=nomComp.getText(); // récupère la chaîne située  
                    //dans le composant nomComp et la place dans s
```

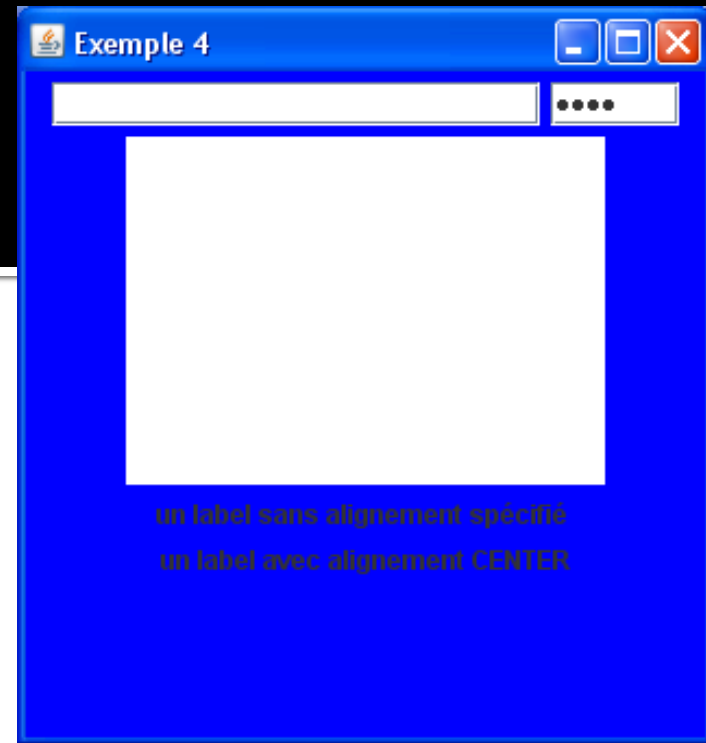
# Exemple 4

//...

```
class FrameEx4Bouton extends JFrame {
    JPanel p1;          //declarations
    JTextField jt;
    JPasswordField jp;
    JLabel l1,l2;
    JTextArea jta;

    public FrameEx4Bouton(String titre){
        super(titre);
        p1=new JPanel(); //créations des composants
        p1.setBackground(Color.blue);
        jt= new JTextField(20);
        jp=new JPasswordField("toto",5);
        l1=new JLabel("un label sans alignement spécifié");
        l2=new JLabel("un label avec alignement CENTER",SwingConstants.CENTER);
        jta=new JTextArea(10,20);

        p1.add(jt); //Ajout des composants dans le JPanel
        p1.add(jp);
        p1.add(jta);
        p1.add(l1);
        p1.add(l2);
        this.add(p1); // ou setContentPane(p1);
    }
}
```





# JComboBox

---

- P 42-44 du poly

# Les Gestionnaires de placement (Layout)

package java.awt

P 49 à 56 du poly de cours

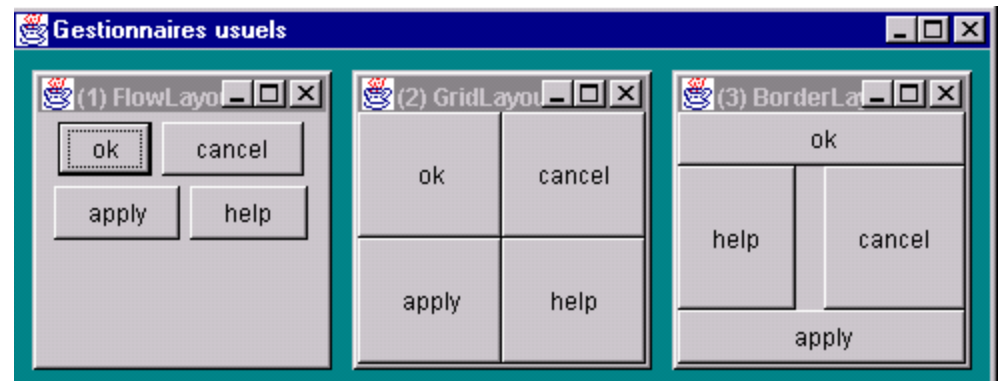
# Gestionnaires de présentation Layout

## ■ Gestionnaires les plus courants:

- `FlowLayout`
- `BorderLayout`
- `GridLayout`

Plus complexe :

- `GridBagLayout`



## ■ Choix par

```
conteneur.setLayout(new  
XyzLayout())
```

Ou à la construction d'un  
`JPanel`:

```
Public JPanel(LayoutManager layout)
```

# FlowLayout

- **FlowLayout** est le gestionnaire par défaut des composants **JPanel**.
- affiche les composants **de la gauche vers la droite**, et passe à la ligne s'il n'y a plus de place.
- après retaille, la disposition est recalculée.

## Constructeurs :

```
FlowLayout(int align, int hgap, int vgap)  
FlowLayout(int align)  
FlowLayout()
```

- **align** vaut par exemple **LEFT**, **CENTER** (défaut) ou **RIGHT** et indique comment chaque ligne est remplie.
- **hgap** (= 5) et **vgap** (=5) sont les espaces entre composants.

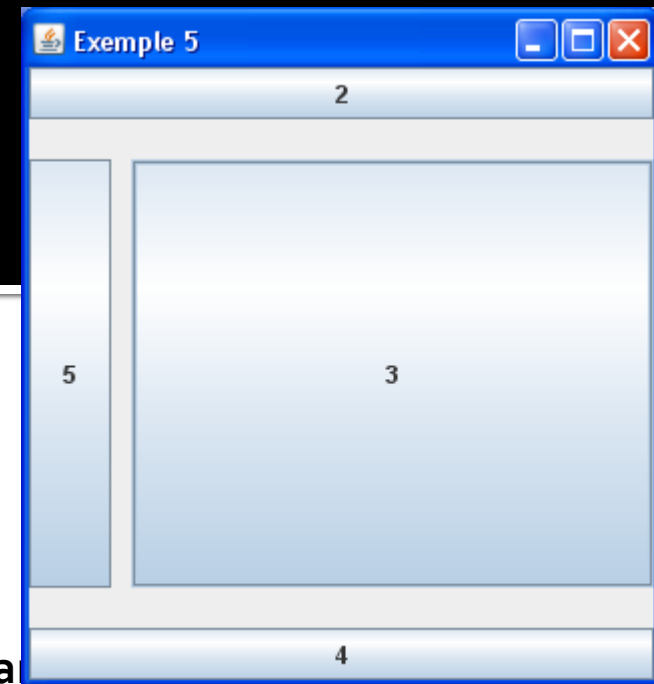
# BorderLayout

- **BorderLayout** est le gestionnaire de placement par défaut des **JFrames**
- divise ses composants en 5 régions : **nord, sud, ouest, est, et centre.**
- "nord" et "sud" occupent *toute la largeur*,
- "ouest" et "est" occupent *la hauteur qui reste*,
- "centre" occupe la *place restante*.
- On utilise la constante correspondante de **BorderLayout**.

## Constructeurs :

```
BorderLayout(int hgap, int vgap)  
BorderLayout()
```

# Exemple 5

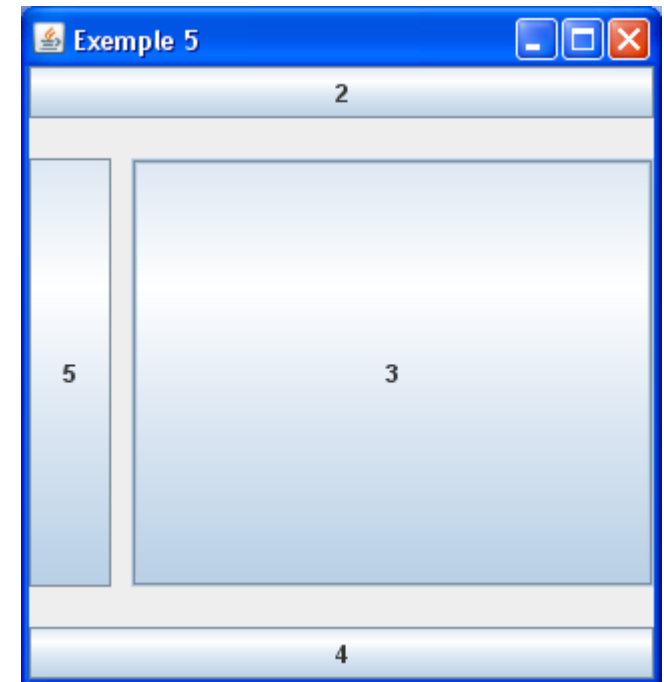


```
class FrameEx5Bouton extends JFrame {  
  
    private JButton b1,b2,b3,b4,b5; //declarations  
    private JPanel p;  
  
    public FrameEx5Bouton(String titre){  
        super(titre);  
        b1=new JButton("1"); //créations des compos  
        b2= new JButton("2");  
        b3= new JButton("3");  
        b4= new JButton("4");  
        b5= new JButton("5");  
        p=new JPanel(new BorderLayout(10,20));  
        //Utilisation du constructeur imposant un gestionnaire de placement  
  
        p.add(b1,BorderLayout.NORTH); //Ajout des boutons dans le JPanel  
        p.add(b2,BorderLayout.NORTH);  
        p.add(b3,BorderLayout.CENTER);  
        p.add(b4,BorderLayout.SOUTH);  
        p.add(b5,BorderLayout.WEST);  
        this.add(p); // ou setContentPane(p);  
  
    }  
}
```

# Exercice

Faire l'arborescence des composants de cette interface.

Comment faire pour que les 2 boutons b1 et b2 apparaissent dans la partie Nord ?



# Exemple 5 - variante

```
class FrameEx5Bouton extends JFrame {  
  
    private JButton b2,b3,b4,b5; //declarations  
    private JPanel p;  
  
    public FrameEx5Bouton(String titre){  
        super(titre);  
        //créations des composants  
        b2= new JButton("2");  
        b3= new JButton("3");  
        b4= new JButton("4");  
        b5= new JButton("5");  
  
        //Ajout des boutons dans le JPanel  
        this.add(b2, BorderLayout.NORTH);  
        this.add(b3, BorderLayout.CENTER);  
        this.add(b4, BorderLayout.SOUTH);  
        this.add(b5, BorderLayout.WEST);  
  
    }  
}
```



# GridLayout

## Constructeurs :

```
GridLayout(int lignes, int cols, int hgap, int vgap)
```

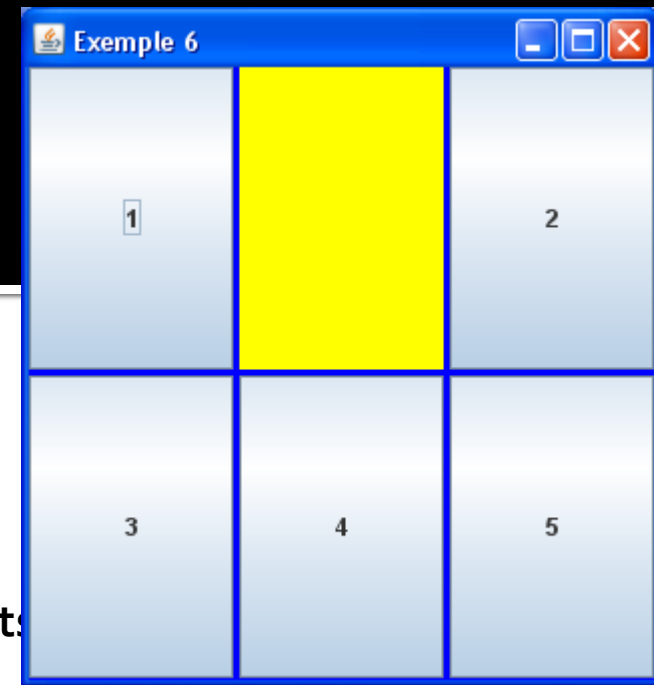
```
GridLayout(int lignes, int cols)
```

- composants sur une *grille*, ligne par ligne (dans l'ordre d'adjonction),
- les cellules ont la *même taille*,
- `hgap` et `vgap` sont nuls par défaut.
- On fixe le nombre de lignes *ou* de colonnes,
  - *si* `lignes > 0`, *alors* `cols` *est ignoré*,
  - *si* `lignes = 0`, *alors* `lignes` *est ignoré*.

# Exemple 6

```
//...
class FrameEx6Bouton extends JFrame{
    private JButton b1,b2,b3,b4,b5; //declarations
    private JPanel p,pan;

    public FrameEx6Bouton(String titre){
        super(titre);
        b1=new JButton("1"); //créations des composants
        b2= new JButton("2");
        b3= new JButton("3");
        b4= new JButton("4");
        b5= new JButton("5");
        pan=new JPanel();
        pan.setBackground(Color.YELLOW);
        p=new JPanel(new GridLayout(2,0,3,3)); //Utilisation du constructeur
        imposant un gestionnaire de placement
        p.setBackground(Color.BLUE);
        p.add(b1); //Ajout des boutons dans le JPanel
        p.add(pan);
        p.add(b2);
        p.add(b3);
        p.add(b4);
        p.add(b5);
        this.add(p); // ou setContentPane(p);
```



# Exemple P52-55

---

# Exercices

- Pour les IG ci-contre,
    - Dessiner l'**arborescence des composants** en indiquant les algorithmes de placement utilisés
  - Faire le TP 7 (IG a.) puis le TP8 (IG c.)
  - Construire les IG c. et d
- b.

Exercice 2

Login

Mot de passe

OK Annuler

Traceur Forme

Effacer noir

a.

Calculette

0.0

7	8	9	+
4	5	6	-
1	2	3	*
0	Clr	x2	/
		sqrt	=

c.

d.

Mon dialogue

Nom

Prenom

Adresse

☐ Tennis

☐ Squash

☐ Natation

☐ Athlétisme

☐ Randonnée

☐ Foot

☐ Basket

☐ Volley

☐ Petanque

Sexe ☒ Homme ☐ Femme

OK Annuler