COURS – TD 4: GESTIONNAIRE DE GÉOMETRIE

JComponent - composants et containers

Composants

- JLabel
- JButton
- JComboBox
- JList
- ...

Containers

- JPanel
- JScrollPane
- JTabbedPane
- ...

Les composants / containers

- Composants qui ont pour but principal de contenir d'autres composants

- Les principaux conteneurs :
  - JPanel
  - JScrollPane
  - JSplitPane
  - JTabbedPane

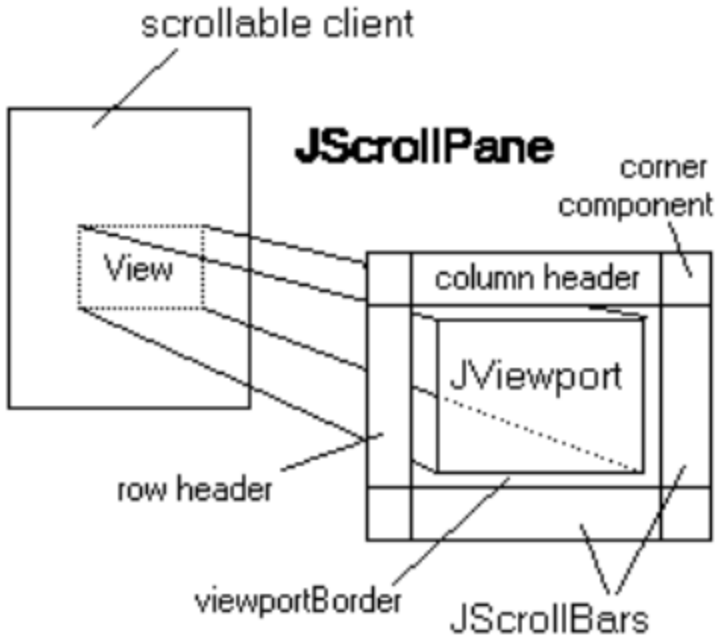
JPanel

- Contenant générique
- N’est pas visuellement détectable
- Permet de structurer une interface en rassemblant un ensemble de composants liés à une activité de l’utilisateur

JScrollPane

- Contenant pour 1 seul composant
- A utiliser pour afficher un composant dont la taille est telle qu'il ne peut être affiché intégralement dans l'emplacement prévu
  - Exemples: texte, liste,

    tableau, ...



JScrollPane

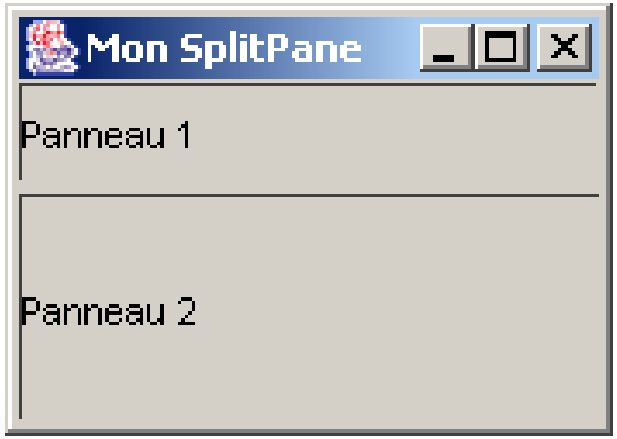
- Permet d'avoir des barres de défilement lorsque le composant qu'il contient est trop grand par rapport à la taille qui lui est allouée.


```
JLabel texte = new JLabel("un texte trop long par rapport à la taille du JLabel");

JScrollPane scroll = new JScrollPane(texte);
```

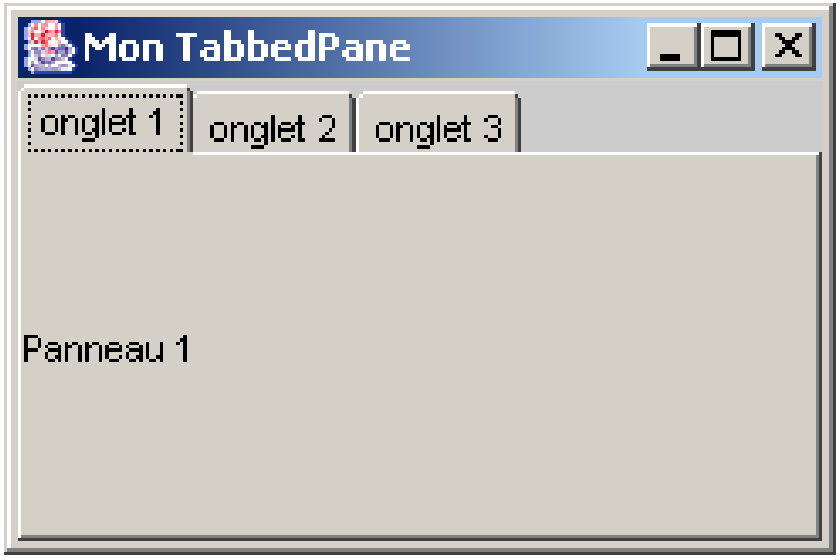
JSplitPane

- Permet de séparer un espace en 2 zones
- Ces 2 zones sont séparées par une barre (verticale ou horizontale) qui peut être déplacée dynamiquement.



JTabbedPane

- Reprend le système des onglets
- Permet d'avoir plusieurs panneaux sur la même surface



Container - Disposition

- Comment positionner les composants les uns par rapport aux autres?
- Comment un container agence-t-il les composants qu'il contient?
  - Gestion du redimensionnement dynamique
- Utilisation de différents types de layout

Layout

Une stratégie par type de layout

- Dispositions simples, peu d'éléments graphiques
  - BorderLayout (5 zones)
  - FlowLayout
  - BoxLayout
- Disposition matricielle
  - GridLayout
- Réalisation de formulaires, dispositions recherchées
  - FormLayout (n'appartient pas au JDK)
  - GridBagLayout
  - **GroupLayout (recommandé avec Netbeans)**

Positionner les composants

- Sur des containers
  - Fenêtre
  - Composants / containers

  add(Component comp)

- Au moyen d'un gestionnaire de géométrie : LayoutManager

  setLayout(LayoutManager mgr)

Positionnement sans LayoutManager

- setLayout(null)
- Positionnement et taille au pixel près
  - setLocation(x,y)
  - setSize(new Dimension(l,h))
  - setBounds(x,y,l,h)
- Problème : redimensionnement de la fenêtre

JComponent - taille

- Taille maximum maximumSize
- Taille minimum minimumSize
- Taille effective size
- Taille idéale preferredSize

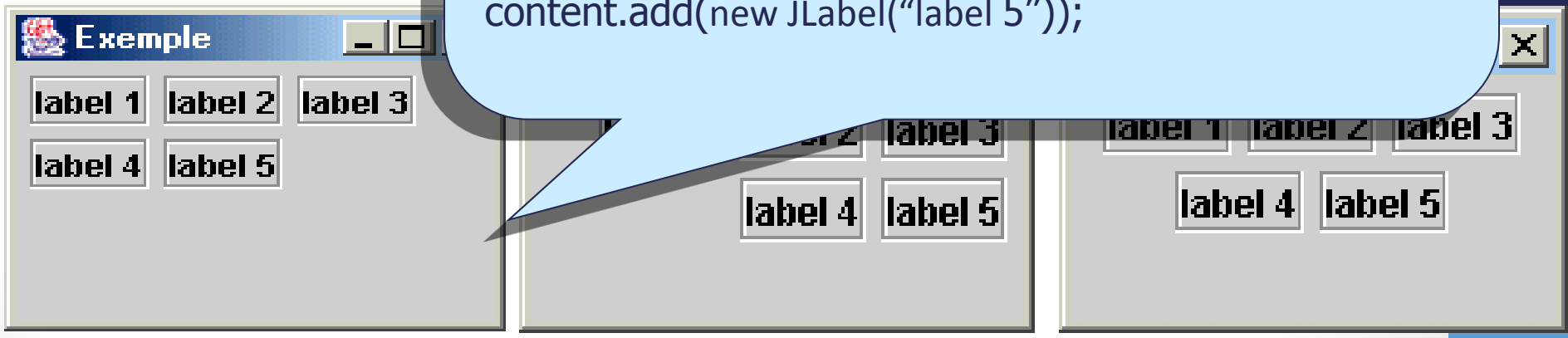
Les gestionnaires de géométrie

- Gestionnaire de géométrie simple
  - FlowLayout
  - GridLayout
  - BoxLayout
- Gestionnaire de géométrie avec contraintes
  - BorderLayout
  - GridBagLayout
- Possibilité de créer son propre Layout

Principe du FlowLayout

- Arrange
  droite.
- On peu
  à gauch

```
JFrame f = new JFrame("Exemple");
Container content = f.getContentPane();
content.setLayout(new FlowLayout(FlowLayout.LEFT));
content.add(new JLabel("label 1"));
content.add(new JLabel("label 2"));
content.add(new JLabel("label 3"));
content.add(new JLabel("label 4"));
content.add(new JLabel("label 5"));
```

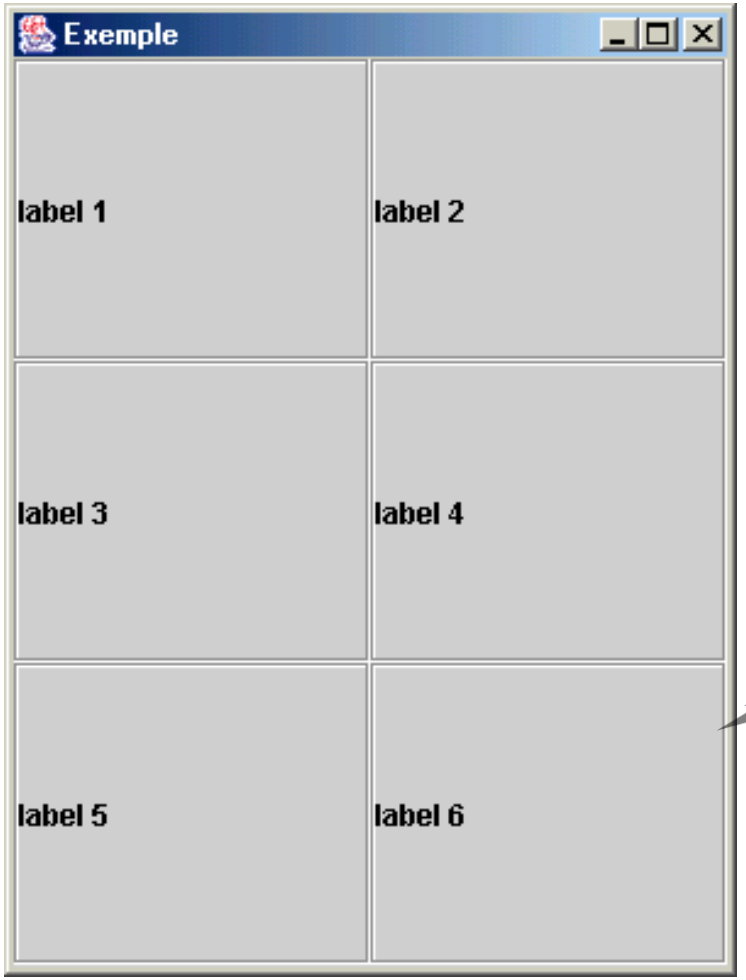


FlowLayout.LEFT FlowLayout.RIGHT FlowLayout.CENTER

Options du FlowLayout

- Alignement de chaque ligne : align
  - LEFT
  - CENTER (par défaut),
  - RIGHT,
  - LEADING,
  - TRAILING (selon l'orientation).
- Espaces entre composants : Hgap et Vgap
- Paramétrable
  - dans le constructeur : **FlowLayout(int align,int hgap,int vgap)**
  - Via les Getters et Setters associés
- Revalidate() pour prendre en compte les modifications

Principe du GridLayout




```
JFrame f = new JFrame("Exemple");
Container content = f.getContentPane();
content.setLayout(new GridLayout(3,2));
content.add(new JLabel("label 1"));
content.add(new JLabel("label 2"));
content.add(new JLabel("label 3"));
content.add(new JLabel("label 4"));
content.add(new JLabel("label 5"));
content.add(new JLabel("label 6"));
```

int vgap)

Principe du BoxLayout

- Arrange les composants
  - sur une ligne (X_AXIS)
  - sur une colonne (Y_AXIS)
  - En prenant la taille préférée des composants

- **BoxLayout(Container c, int axe)**
  - Le container doit être créé avant le BoxLayout

Le composant Box

- Composant transparent avec un BoxLayout associé

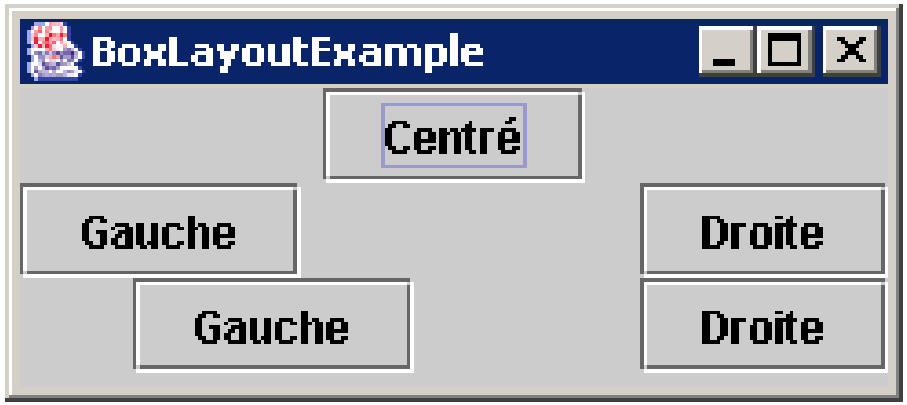
- Il peut contenir des glues pour « remplir » des espaces entre composants
  - Un composant entre deux glues sera centré
  - Une glue entre deux composants plaquera les composants sur les bords

static Component createHorizontalGlue()

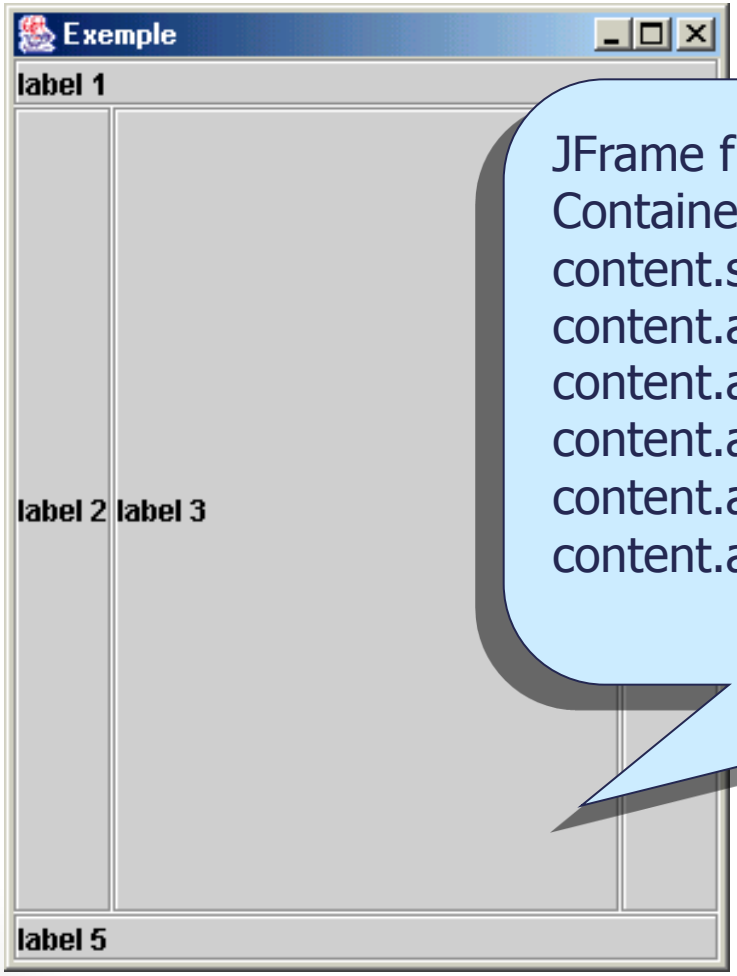
static Component createVerticalGlue()

Exercice

- En utilisant Le BoxLayout, le composant Box et les Glue, positionnez les composants de la manière suivante



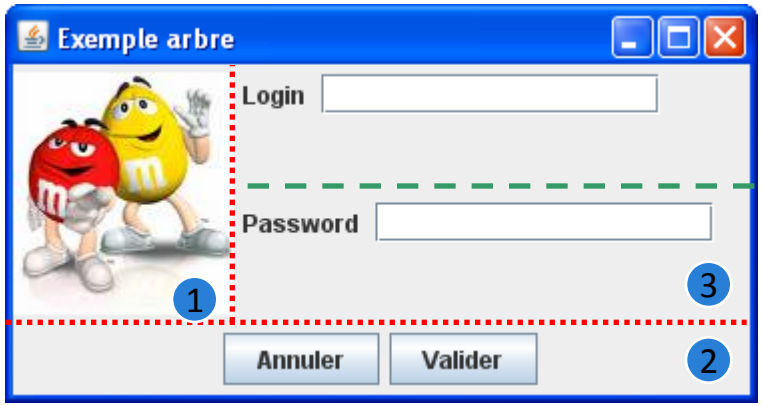
Principe du BorderLayout



- Peut contenir jusqu'à 5 éléments :

```
JFrame f = new JFrame("Exemple");
Container content = f.getContentPane();
content.setLayout(new BorderLayout());
content.add(new JLabel("label 1"),BorderLayout.NORTH);
content.add(new JLabel("label 2"),BorderLayout.WEST);
content.add(new JLabel("label 3"),BorderLayout.CENTER);
content.add(new JLabel("label 4"),BorderLayout.EAST);
content.add(new JLabel("label 5"),BorderLayout.SOUTH);
```

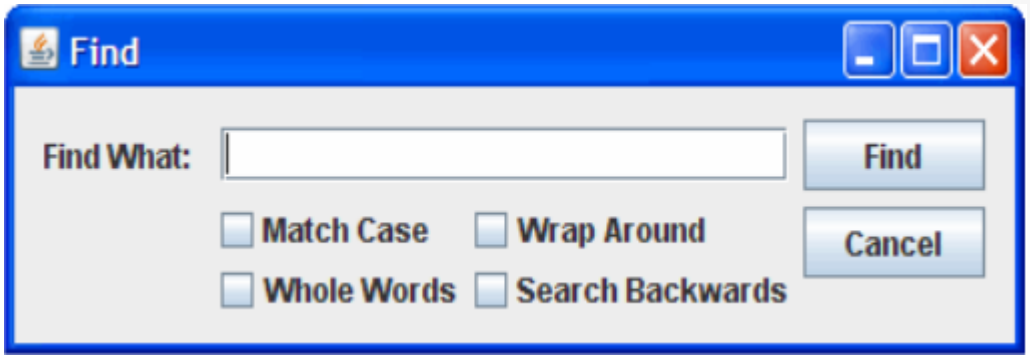
Exemple d'arbre de composants



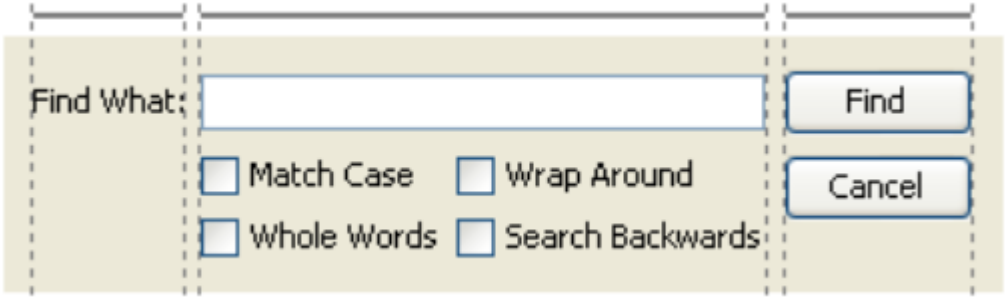
Principe du GroupLayout

- Hiérarchie de groupes
  - Séquentielle
  - Parallèle
- Positionnement
  - Horizontal
  - Vertical
- Layout utilisé par les éditeurs logiciels d’interface utilisateur

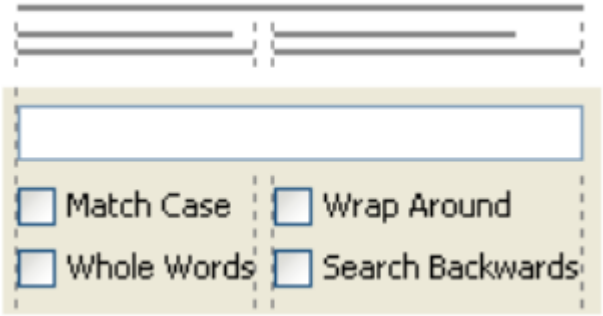
GroupLayout - Horizontal



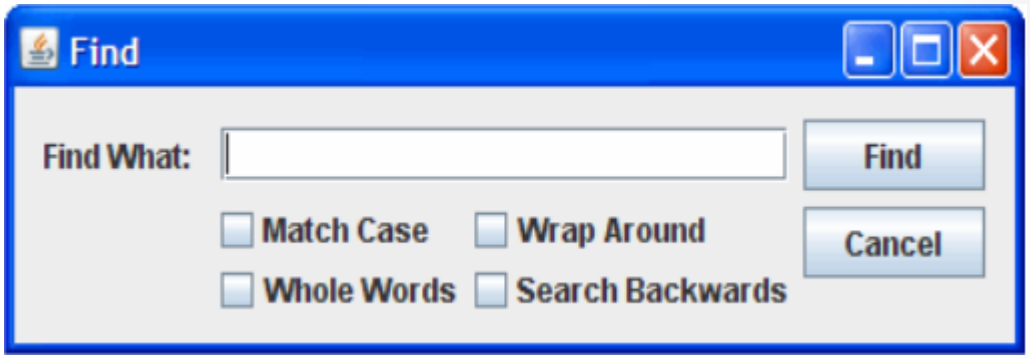
- Trois groupes en séquence



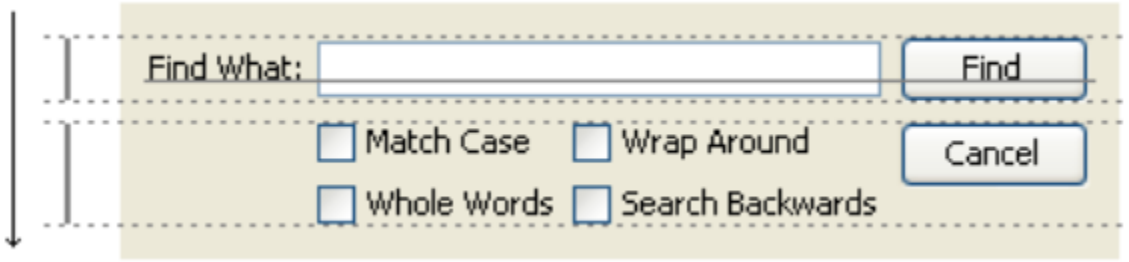
- Champ texte en parallèle avec une séquence de 2 groupes en parallèle



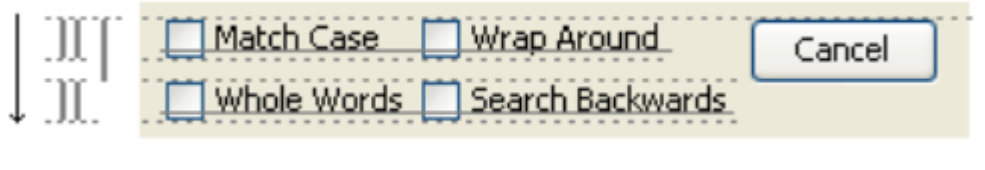
GroupLayout - Vertical



- Deux groupes en parallèle



- Bouton aligné sur le haut et séquence de deux groupes de checkbox en parallèles



Créer son propre LayoutManager

- Implementer l'interface LayoutManager
  - **public Dimension preferredLayoutSize(Container parent);**
  - **public Dimension minimumLayoutSize(Container parent);**

    → renvoie la taille du container

  - **void addLayoutComponent(String name, Component c);**
  - **void removeLayoutComponent(Component c);**

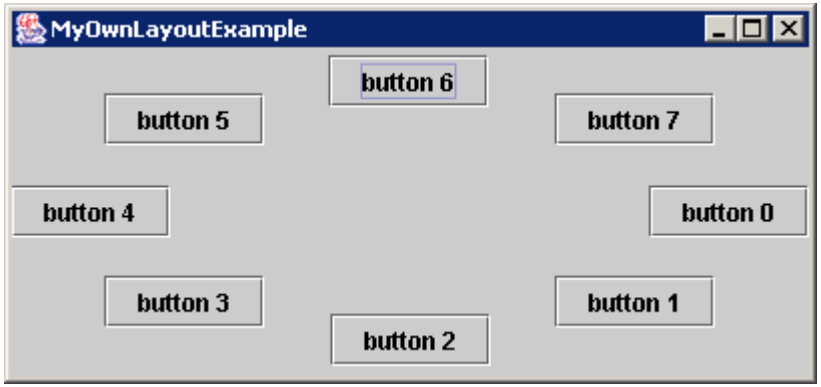
    → ajout / suppression de composant

  - **void layoutContainer(Container parent);**

    → placement des fils dans le container (avec **setBounds()** par ex.)

Exercice

- Créer le LayoutManager permettant de positionner les composants de la manière suivante



Développement d'une interface graphique avec Matisse

1. Création d'une nouvelle JFrame ou d'un nouveau Jpanel

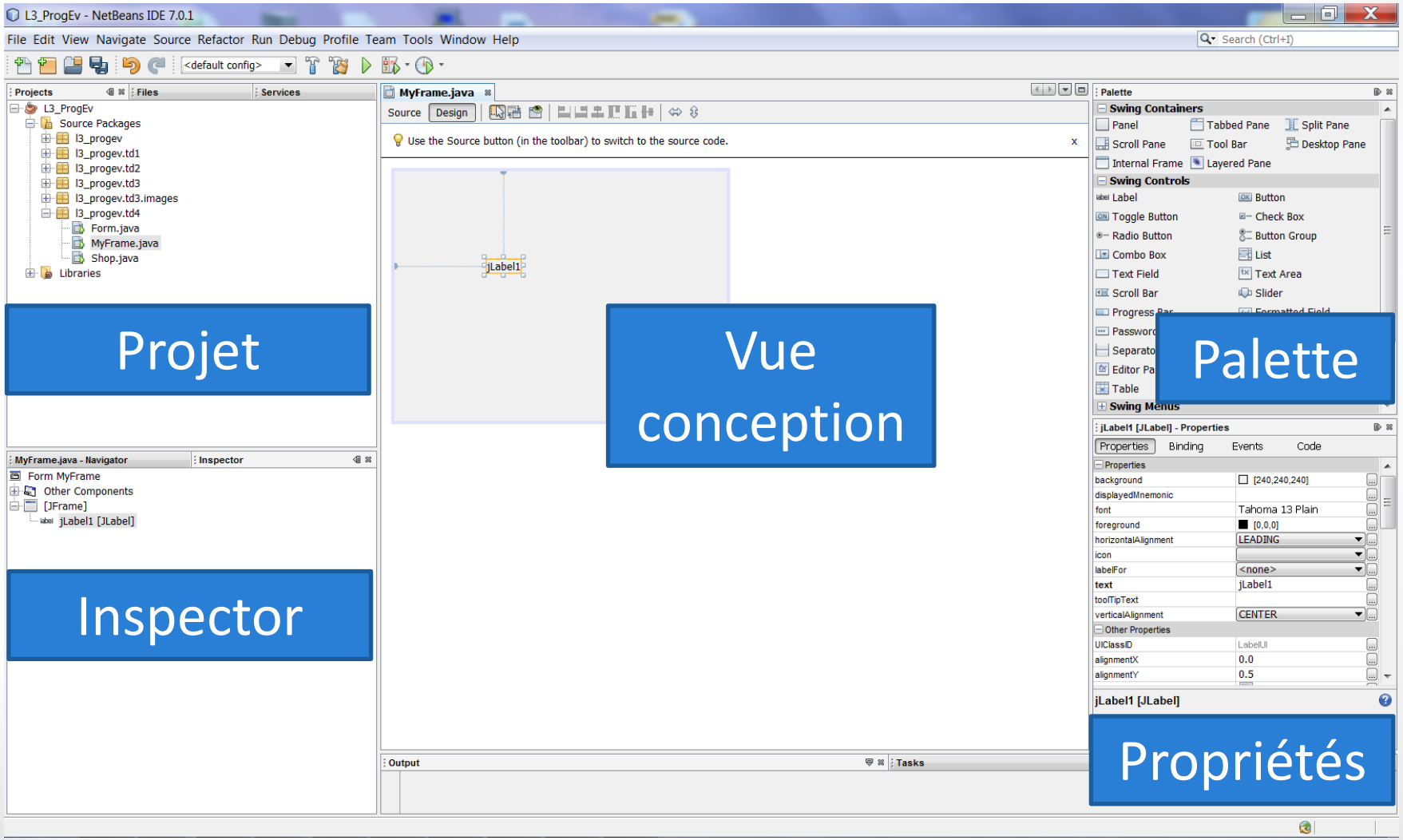
2. Ajout des composants graphiques dans la vue conception

3. Développement du comportement de l'interface

Outil logiciel

- Netbeans
- GUI builder: Matisse
  - WYSIWYG: “What You See Is What You Get”
  - Support au développement d’interfaces , au positionnement relatif de plusieurs éléments
  - Disposition basée sur GroupLayout

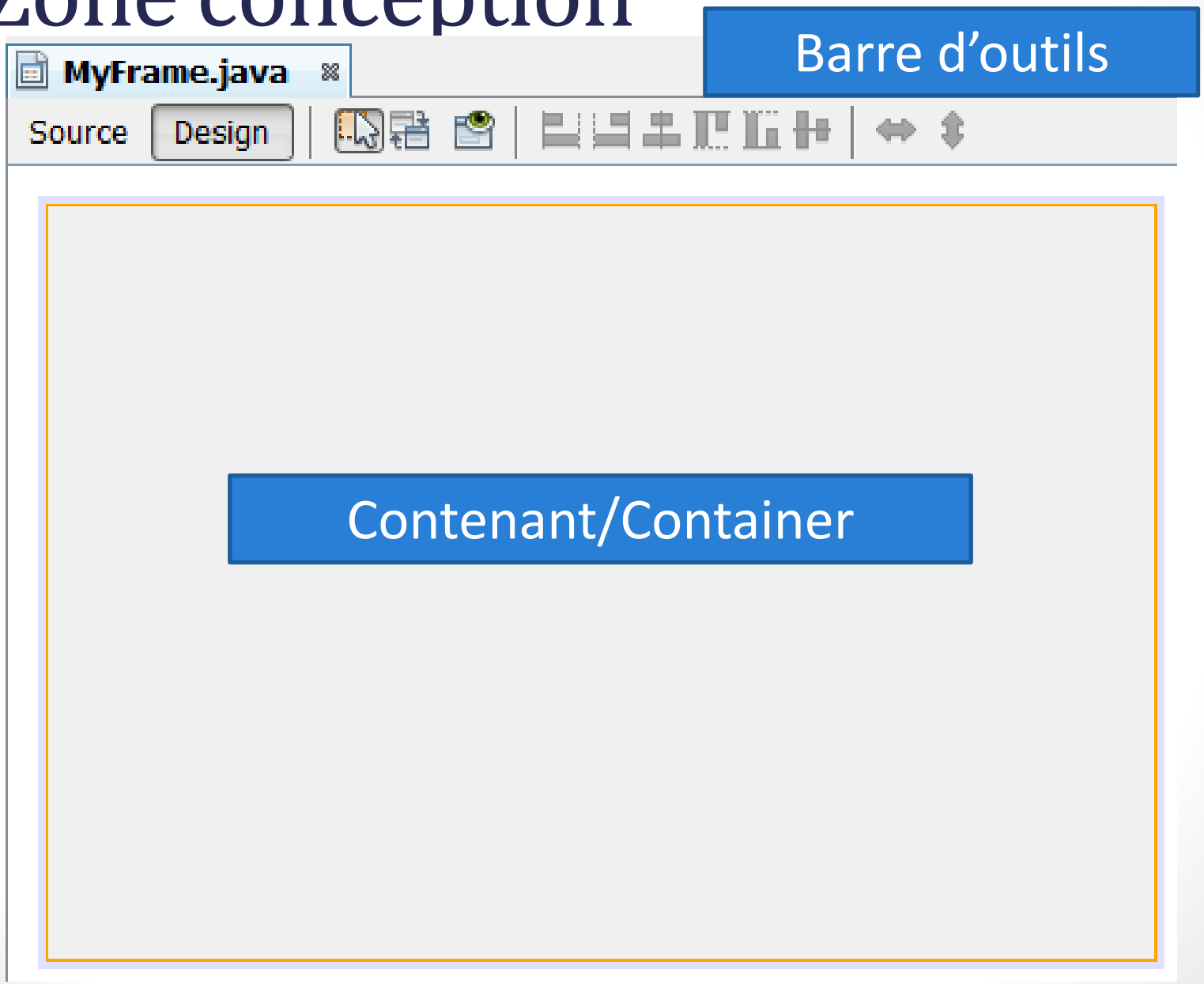
Matisse - GUI builder



Zones et fenêtres

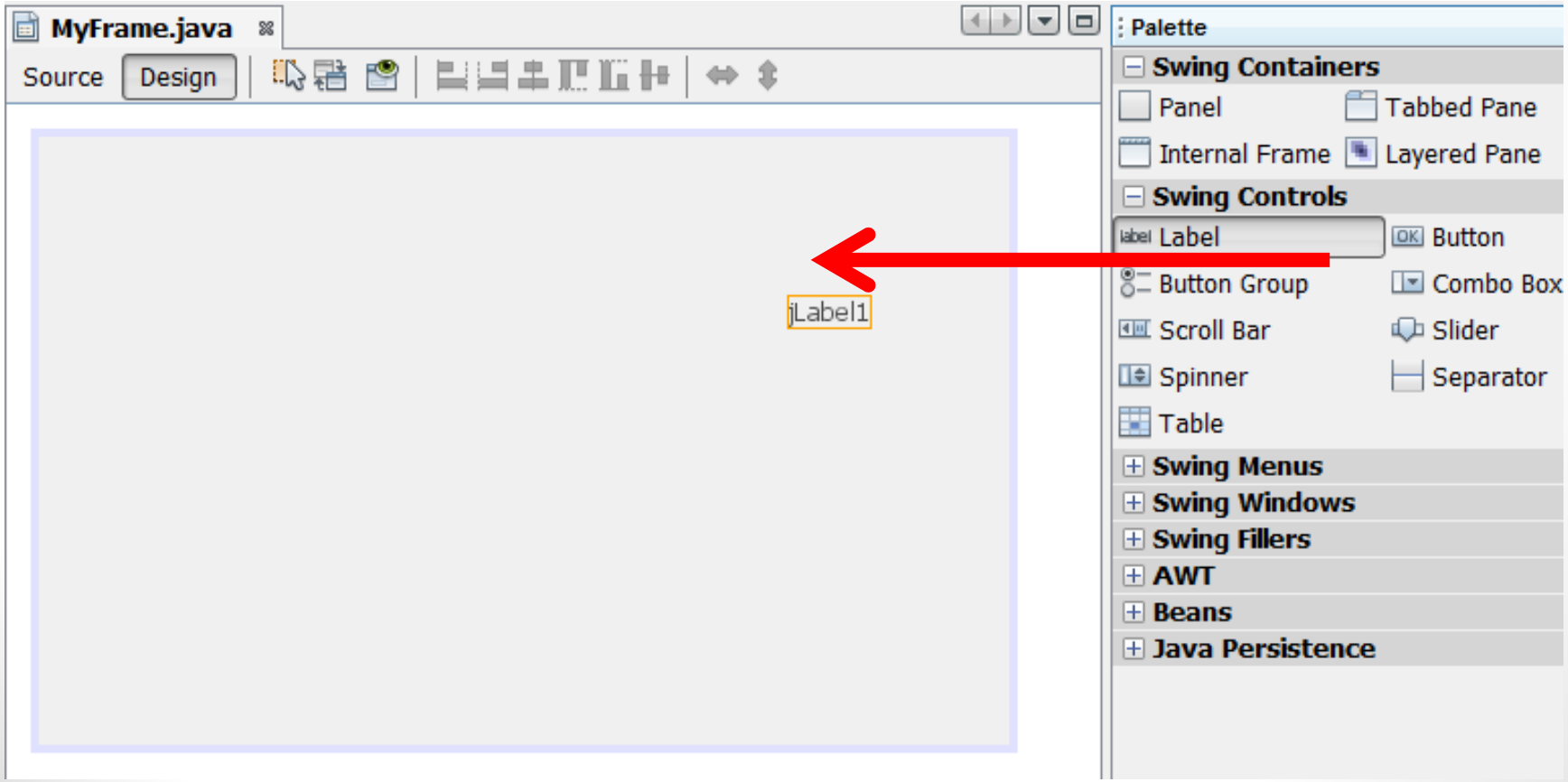
- Zone conception
- Palette
- Propriétés
  - De l'élément sélectionné dans la vue conception
- Inspector
  - Hiérarchie d'éléments visuels et non visuels
  - Organisation des composants dans les containers

Zone conception



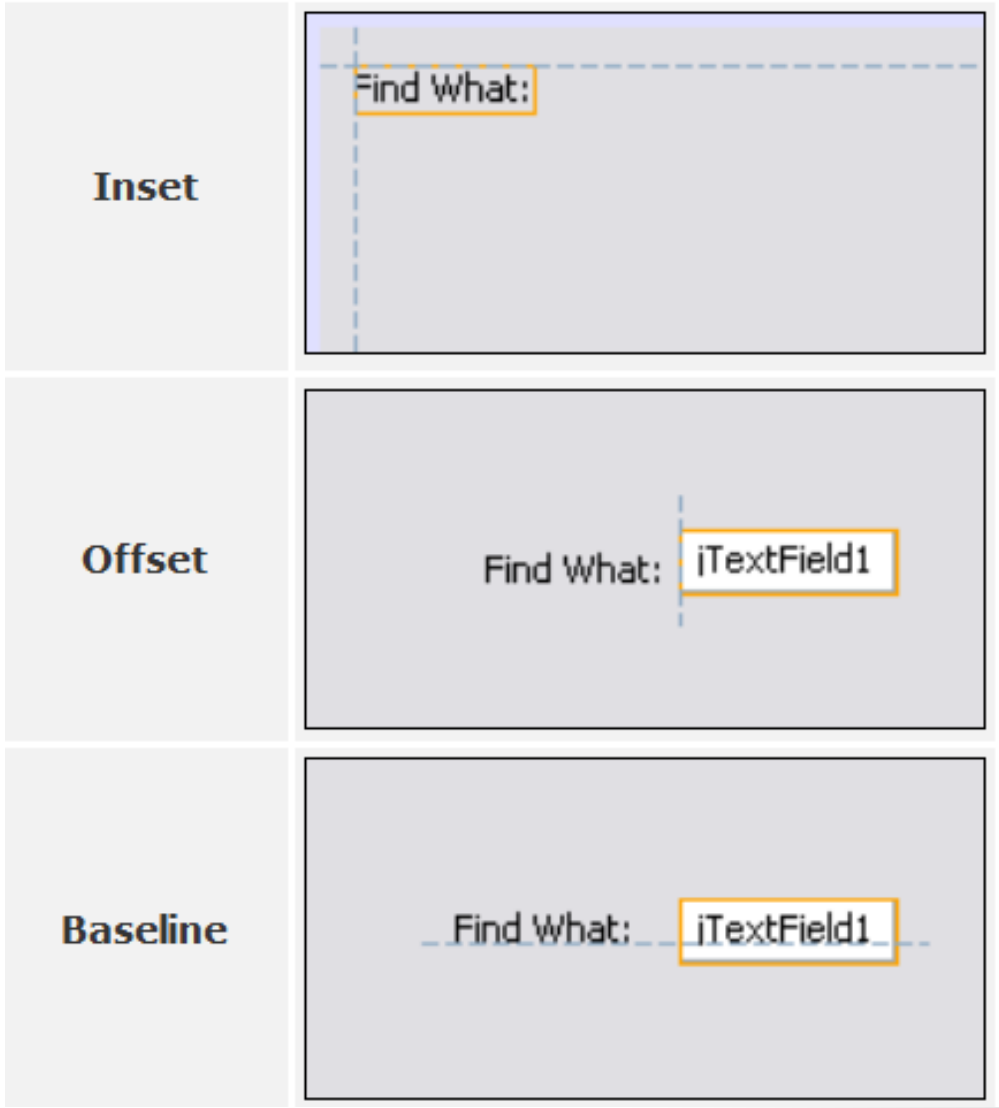
Palette

- Ajout de composant vers la zone de conception par glisser-déposer



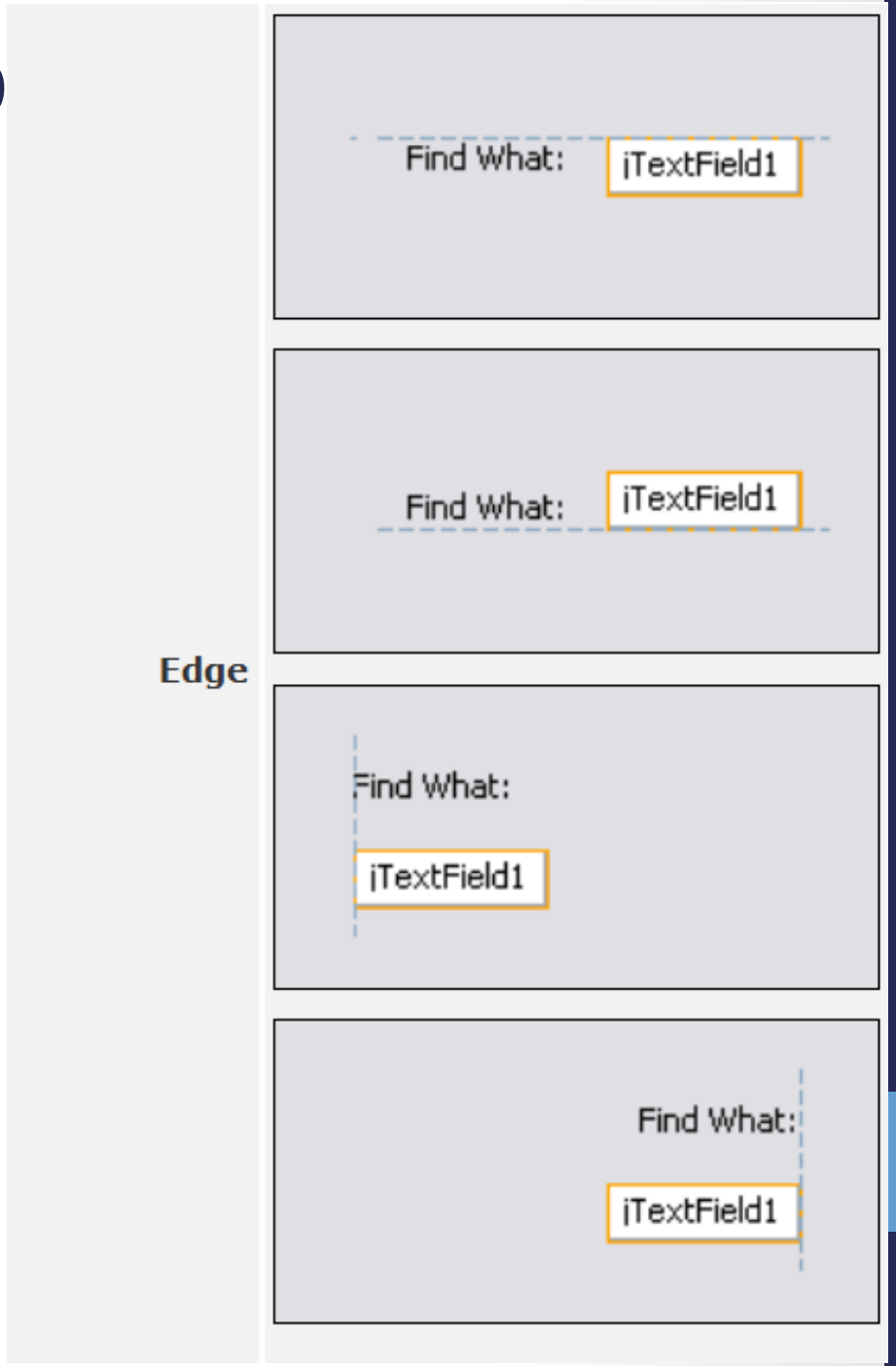
Zone conception Alignement (1)

- Alignement par rapport au contenant/container
- Alignement par rapport au composant adjacent
- Alignement des lignes de texte de composants adjacents

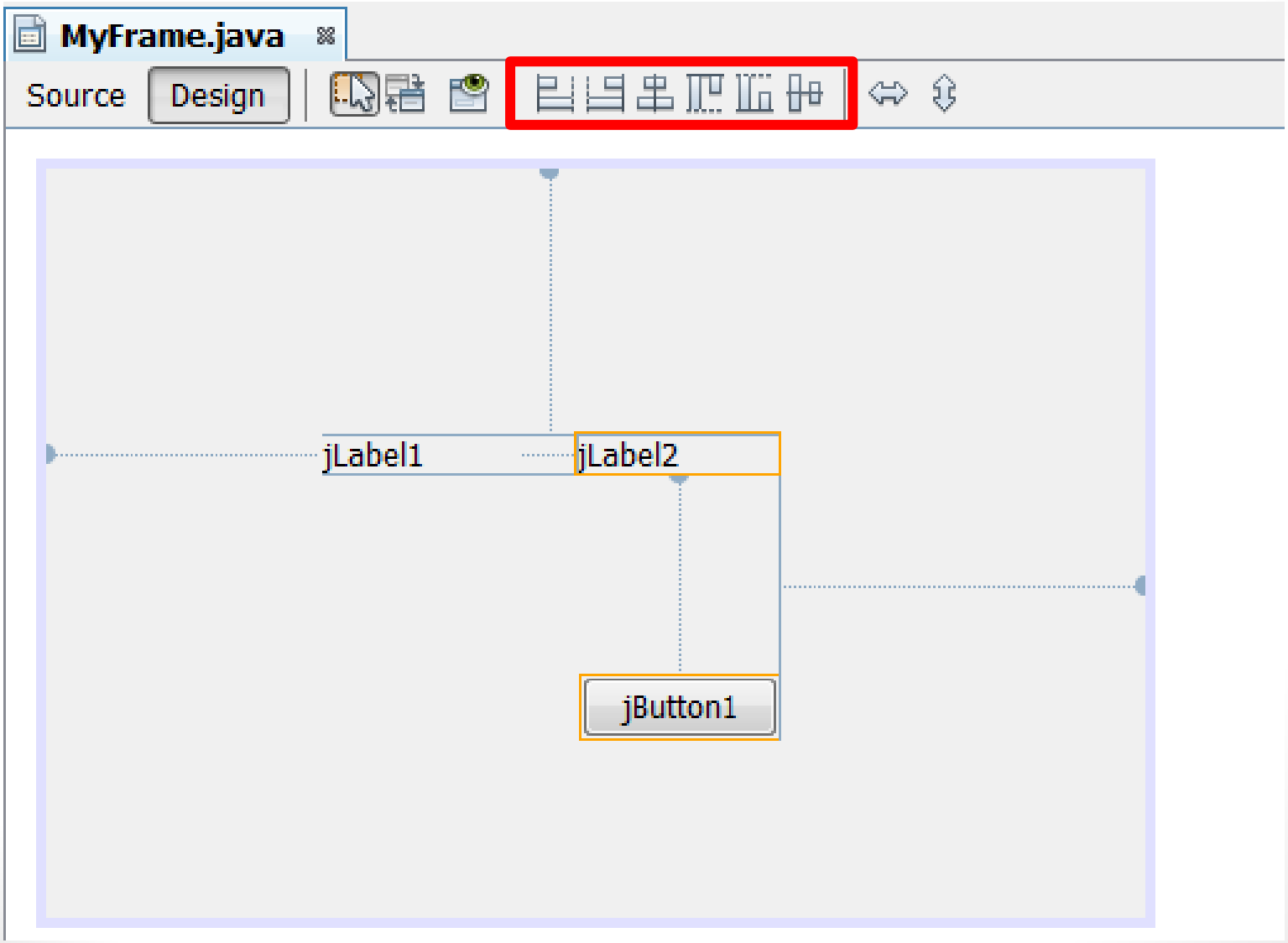


Zone conceptio Alignement (2)

- Alignement des arrêtes de composants adjacents
  - Haut/Top
  - Bas/Bottom
  - Gauche/Left
  - Droite/Right

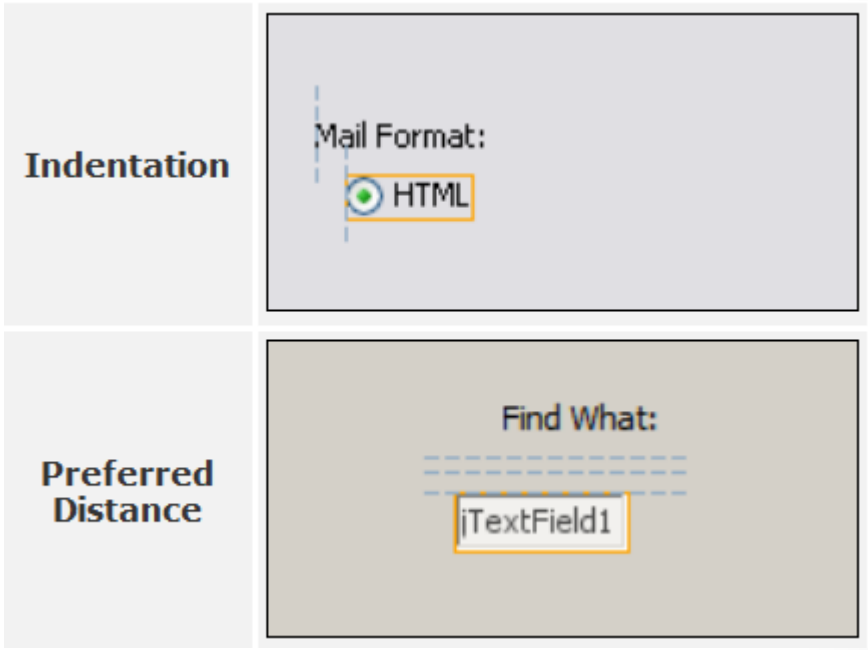


Alignement – barre d'outils



Zone conception Alignement (3)

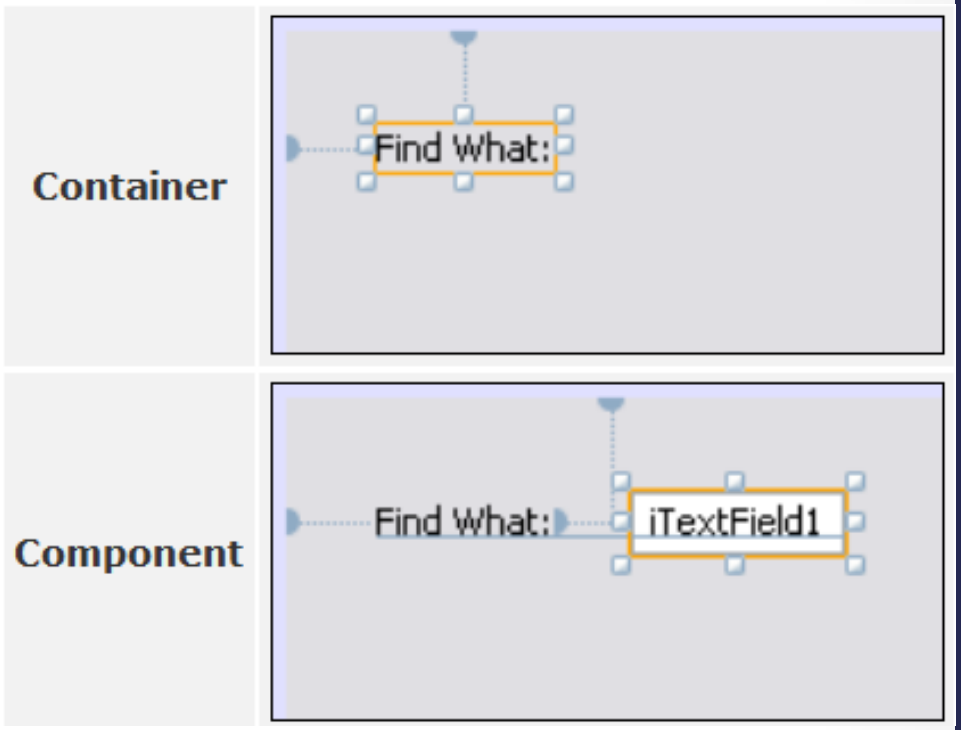
- Indentation



- Distance préférée entre composants
  - Grande/Large
  - Moyenne/Medium
  - Petite/Small

Zone conception - Ancrage

- Au contenant/container
- Au composant adjacent



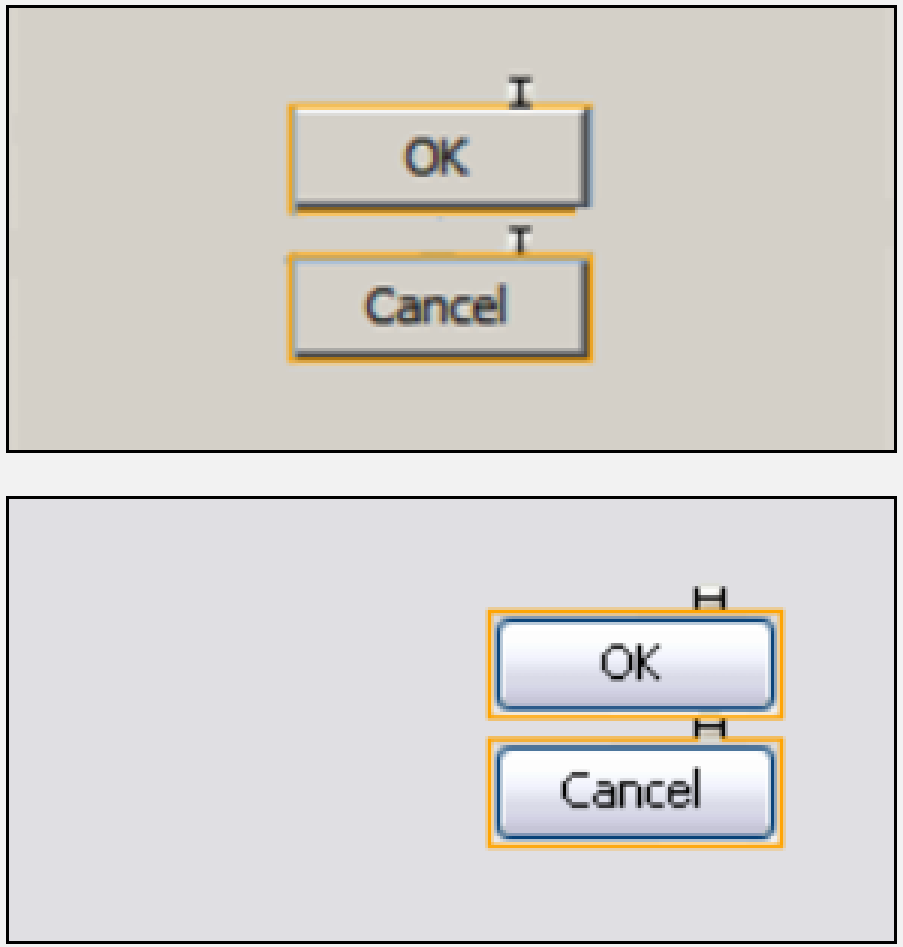
Zone conception
Indicateurs de taille (1)

- Pour un ensemble de composants

Hauteur identique

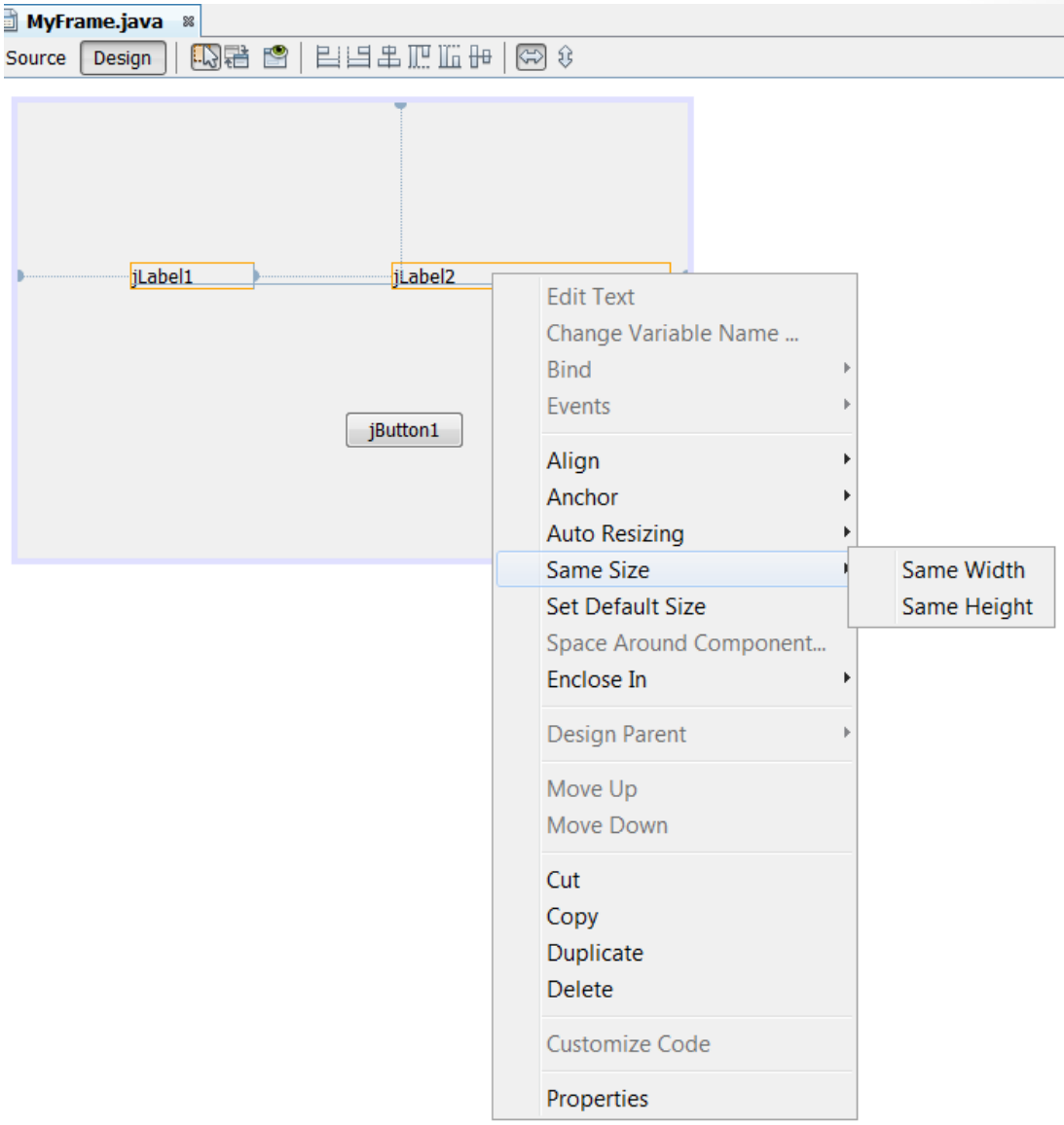
Largeur identique

Same Size

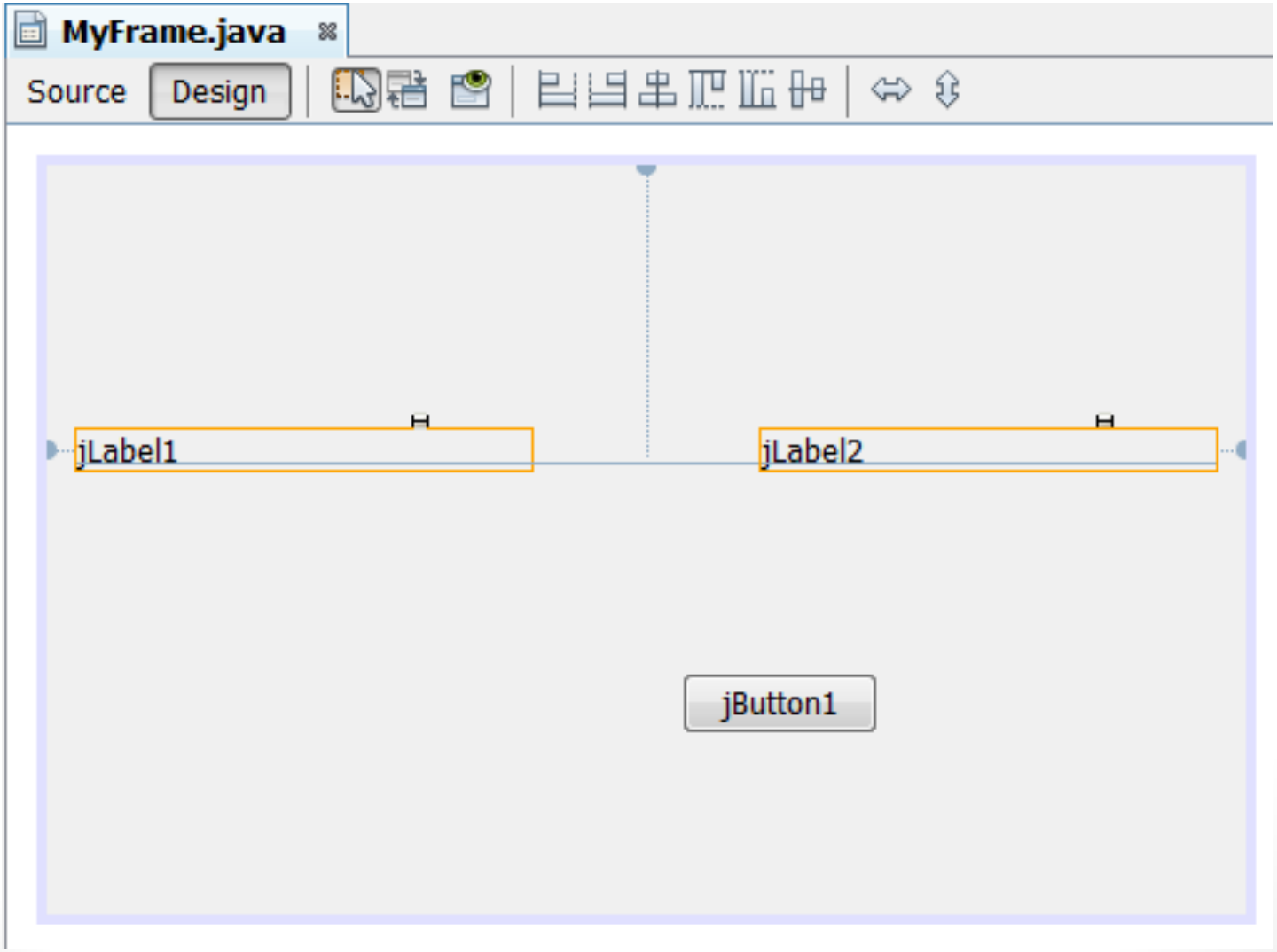


Taille identique

1. Sélectionner les composants
2. Clic droit

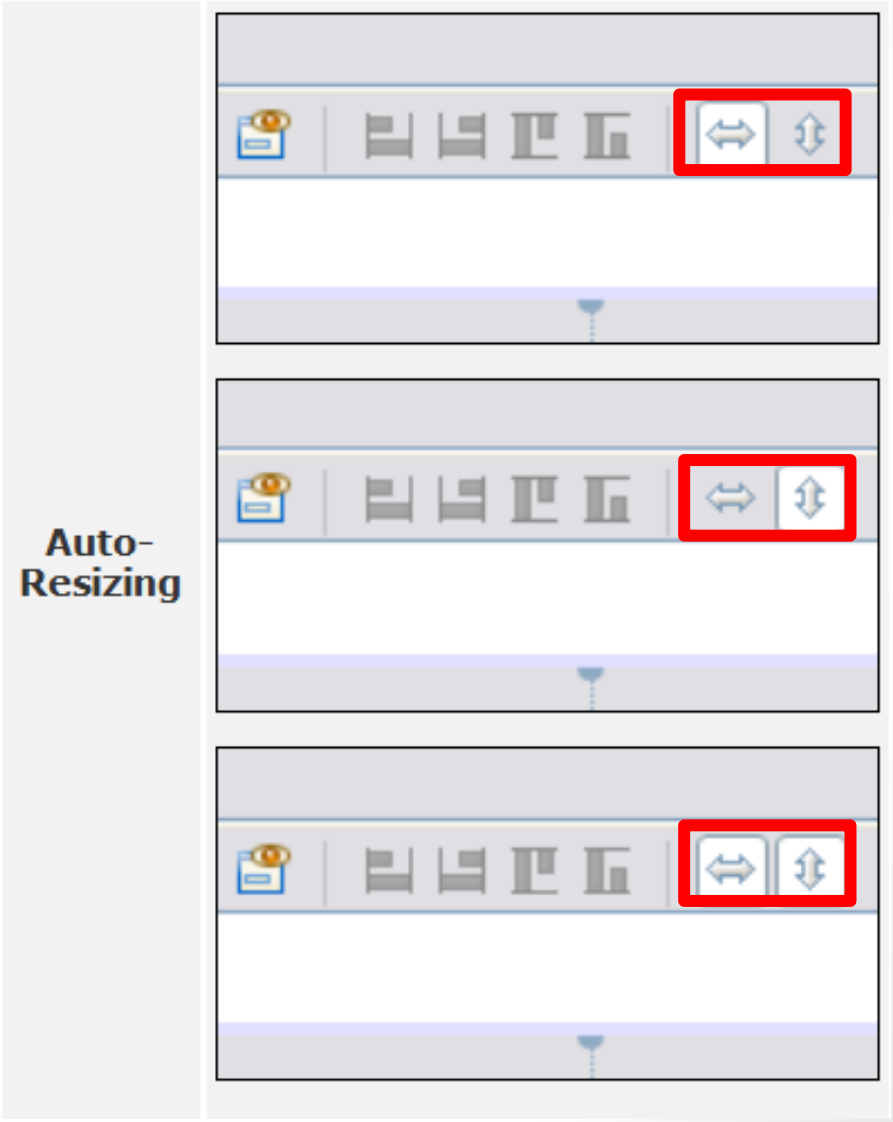


Largeur identique



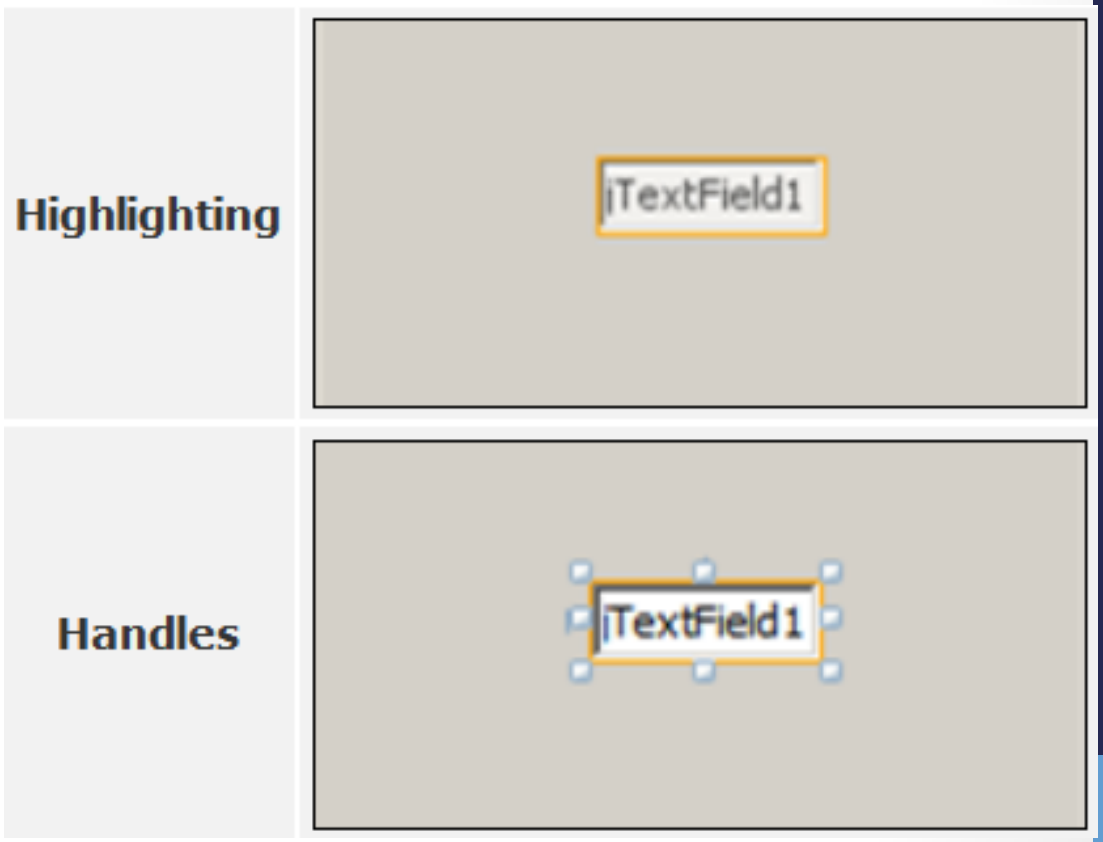
Zone conception
Indicateurs de taille (2)

- Redimensionnement automatique
  - Vertical
  - Horizontal
  - Vertical et horizontal



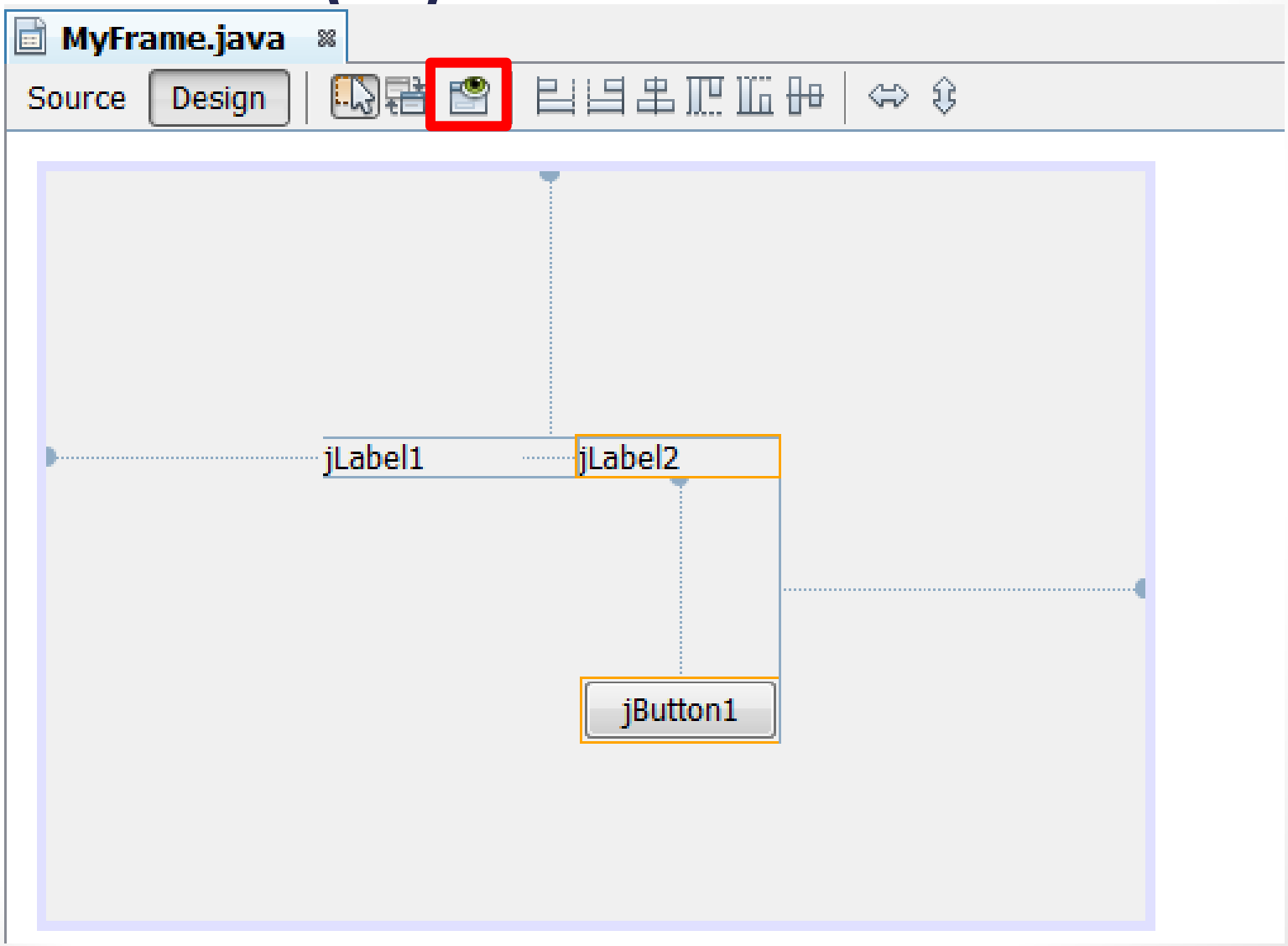
Zone conception - manipulation

- Surlignage orange
  - Permet de visualiser où un composant va être placé

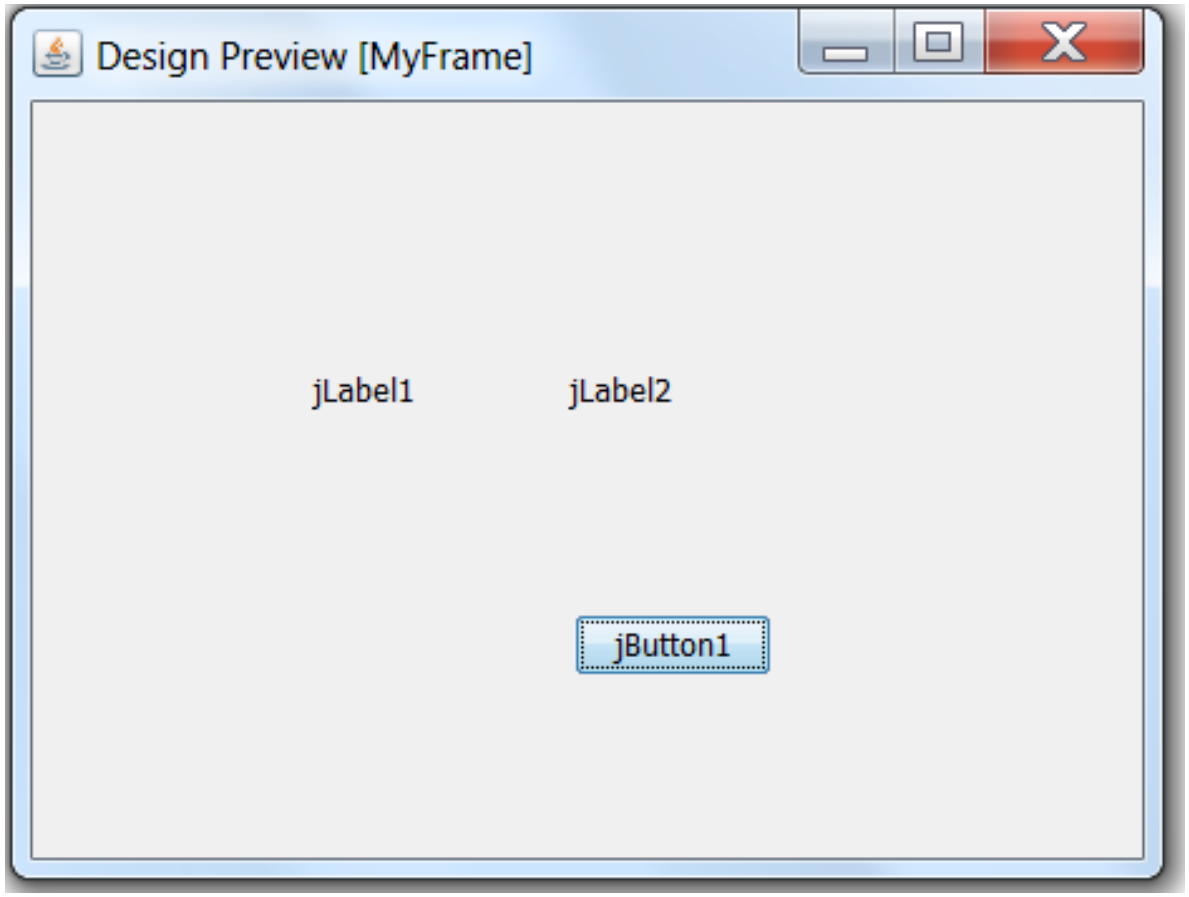


- Surlignage orange avec carrés blancs
  - Permet de redimensionner le composant

Pre-visualisation Preview (1)



Pre-visualisation
Preview (2)



Exercice

- Afin de préparer le développement d'un formulaire de saisie des coordonnées, vous devez tout d'abord avoir une idée de la disposition. Ce formulaire devra contenir:
  - Des informations sur l'identité de la personne (nom, prénom, sexe, date de naissance,...)
  - Des informations sur son lieu de résidence (rue, n°, ville, CP, pays,...)
  - Des information sur la manière de la joindre rapidement (email, tél., portable,...)

1. Dessiner le formulaire et ses éléments graphiques
2. Indiquer les composants SWING correspondants aux éléments retenus