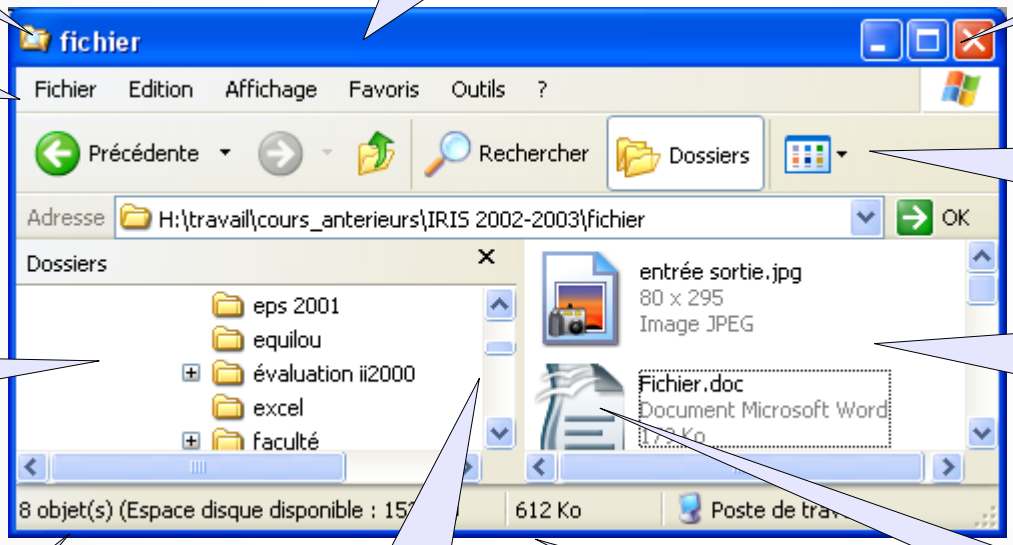
Interfaces graphiques en JAVA

Les environnements graphiques



Menu système

Fenêtre cadre

Icône système

Menu application

Barre d'outils

Zone client

Vue fractionnée

Barre d'état

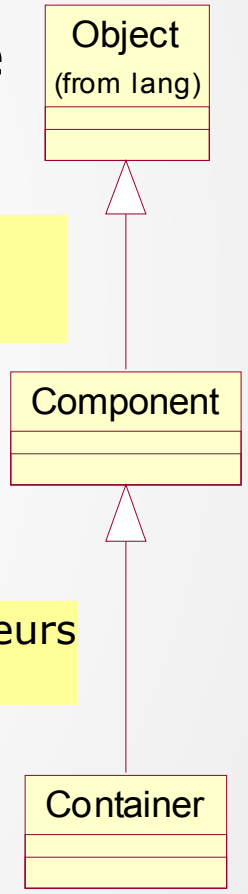
Vue avec ascenseur

conteneur

Composants graphiques

Conteneurs et composants (1) : généralités

- Une interface graphique en Java est un assemblage de
- conteneurs (Container) et de composants (Component).

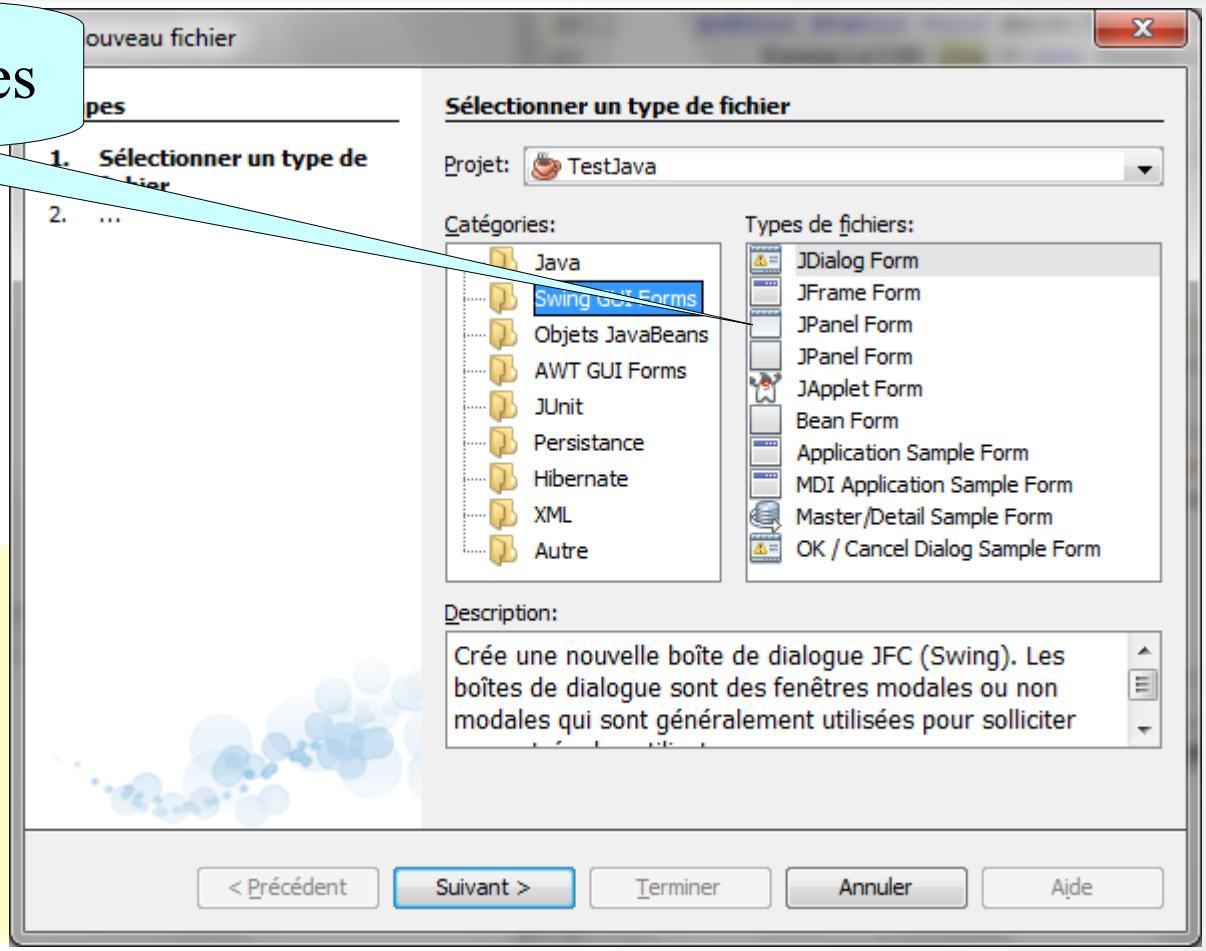


➔ Un composant est une partie "visible" de l'interface utilisateur Java.
- Exemple : les boutons, les zones de textes ou de dessin, etc.

➔ Un conteneur est un espace dans lequel on peut positionner plusieurs composants et donc des conteneurs

Les conteneurs

Conteneurs disponibles

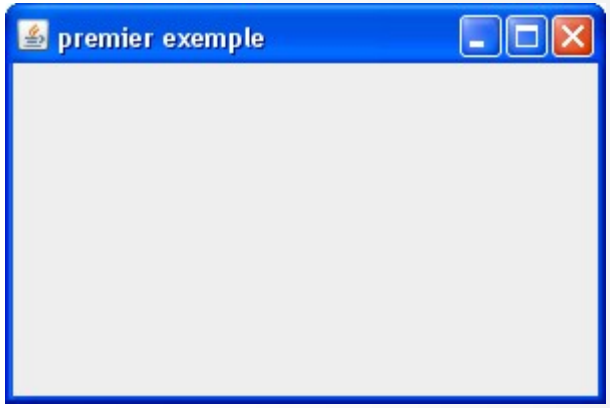


Parmi les conteneurs applicatifs, on peut citer :

- Les boites de dialogues
- Les fenêtres cadres
- Les applets

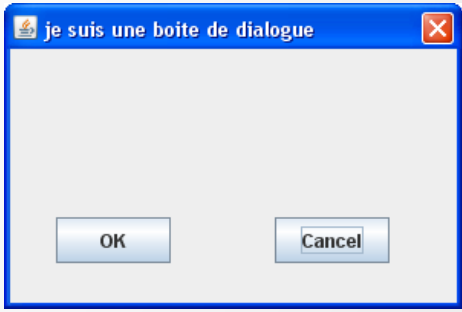
Les fenêtres d'application : JFrame

- C'est une fenêtre avec un cadre, utilisé de façon autonome,
- ➔C'est donc un conteneur général, qui dispose
  - d'un menu système,
  - de boutons d'agrandissement,
  - mise en icône
  - et de fermeture de l'application
- ➔Un JFrame peut contenir des composants ou d'autres conteneurs



Les boîtes de dialogue JDialog

- Un JDialog ressemble à un JFrame mais
  - Il n'a pas de bouton du gestionnaire des fenêtres permettant de l'iconiser.
  - On y associe habituellement un bouton de validation.
- Un JDialog dépend d'un frame parent(qui est passé comme premier argument au constructeur).
- Un JDialog n'est pas visible lors de sa création. -> Utiliser setVisible(true);

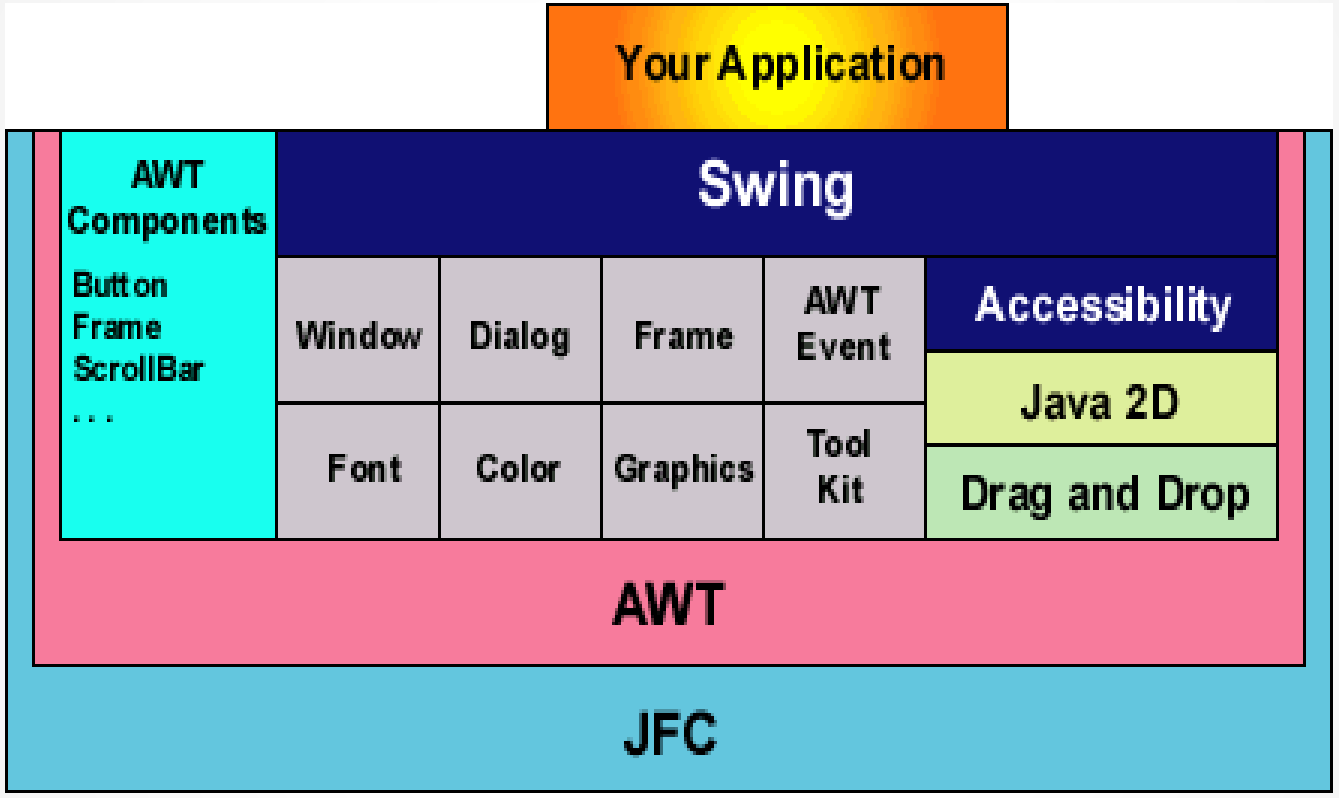


Conteneurs et composants (2) : Frame et Panel

- Les deux conteneurs les plus courants sont JFrame et JPanel.
- Un JFrame présente une fenêtre de haut niveau avec un titre, une bordure et des angles de redimensionnement.
  - Un JFrame est doté d'un BorderLayout par défaut.

- Un JPanel n'a pas une apparence propre et ne peut pas être utilisé comme fenêtre autonome.
  - Un JPanel est doté d'un FlowLayout par défaut.
  - Les JPanel sont créés et ajoutés aux autres conteneurs de la même façon que les composants tels que les boutons.
    - Les JPanel peuvent ensuite redéfinir une présentation qui leur soit propre pour contenir eux-mêmes d'autres composants.

La bibliothèque Swing



Conventions de nommage

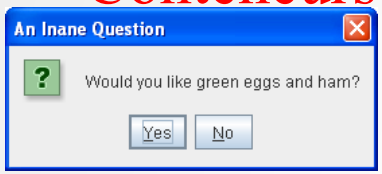
- Les composants Swing sont situés dans le paquetage javax.swing et ses sous paquetages.

- Ils portent des noms similaires à leurs correspondants de awt précédés d'un J.

  → JFrame, JPanel, JTextField, JButton, JCheckBox, JLabel, etc.

Aperçu des classes Swing (1)

- Conteneurs de haut niveau

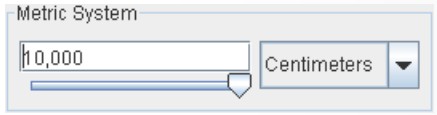


JDialog

JFrame

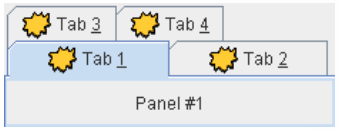
JApplet

- Conteneurs à usage général



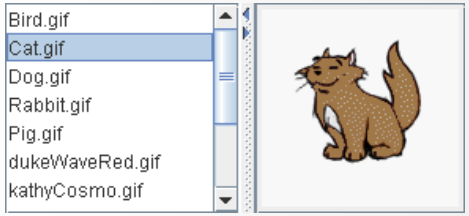
JPanel

JScrollPane



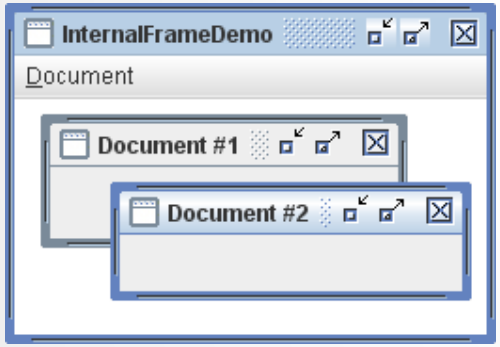
JTabbedPane

JtoolPane

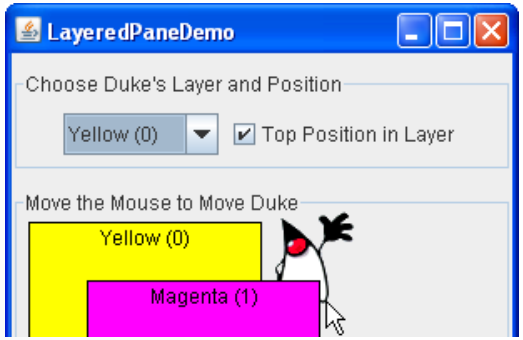


JSplitlPane

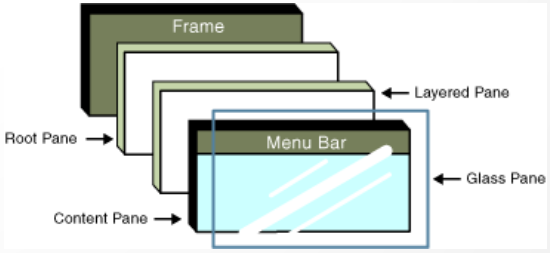
- Conteneurs particulier



JInternalPane



JLayeredPane



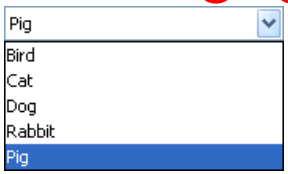
RootPane

Aperçu des classes Swing (2)

- Composants à usage général

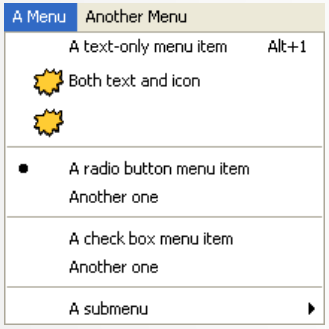
JButton

JCheckBox



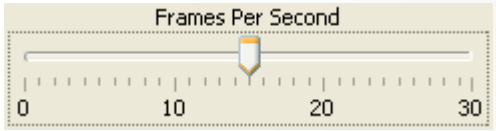
JComboBox

JList



JMenu

JRadioButton



JSlider

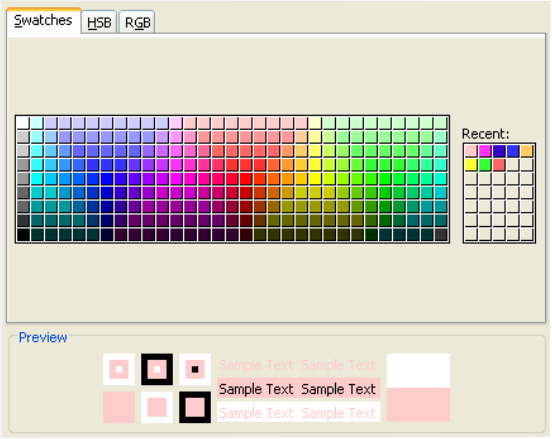
JSpinner

JTextField

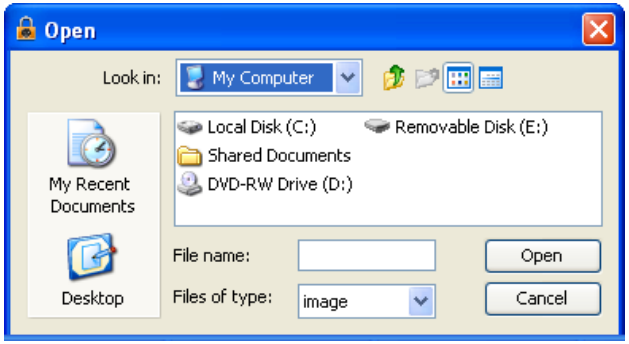
JPasswordField

Aperçu des classes Swing (3)

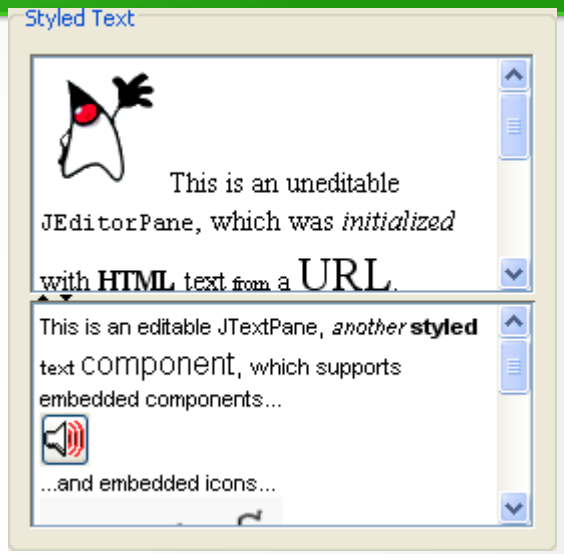
- Composants interactifs



JColorChooser



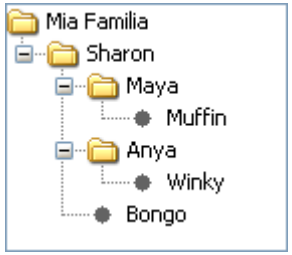
JFileChooser



JEditorPane ou JTextPane

Host	User	Password	Last Modified
Biocca Games	Freddy	!#asf6Awwzb	Mar 16, 2006
zabble	ichabod	Tazb!34$fZ	Mar 6, 2006
Sun Developer	fraz@hotmail.com	AasW541!fbZ	Feb 22, 2006
Heirloom Seeds	shams@gmail.com	bkz[ADF78!	Jul 29, 2005
Pacific Zoo Shop	seal@hotmail.com	vbAf124%z	Feb 22, 2006

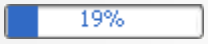
JTable



JTree

Aperçu des classes Swing (3)

- Composants non éditables



JProgressBar

JSeparator

JLabel

JToolTip

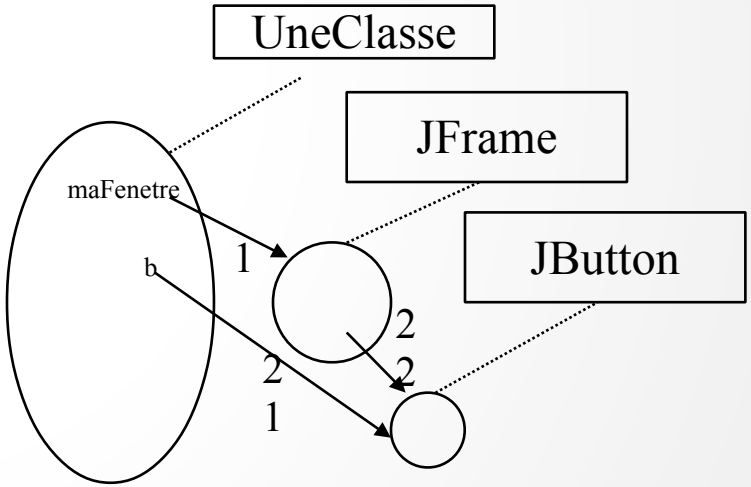
Composants graphiques principaux

Classe	Description
Jframe	Fenêtre graphique.
Jpanel	Zone graphique, container
Jbutton	Objet bouton
JtextField	Zone de texte à saisir
Jlabel	Texte non modifiable
JcomboBox	Choix d'éléments dans une liste avec sélecteur
JscrollBar	Barre de défilement
Jtable	Tableau
Jlist	Choix d'éléments dans une liste sans sélecteur
JcheckBox	Objet pouvant être coché ou décoché
JradioButton	Choix exclusifs pour les options

Organisation d'un écran

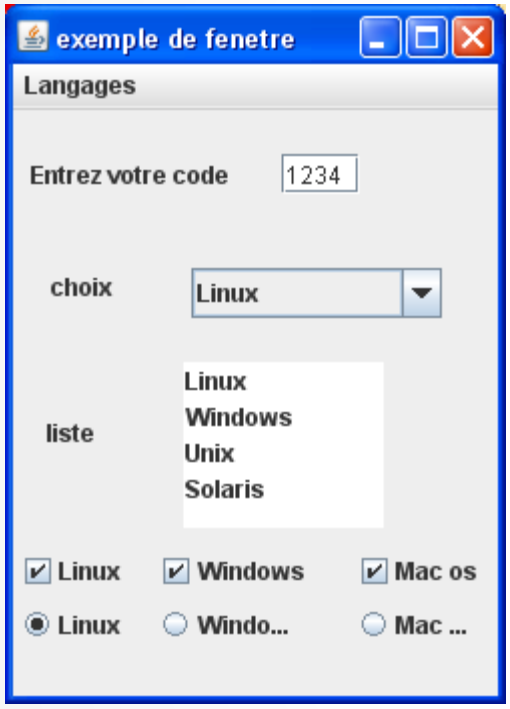
- Composition d'objets graphique dans des containers
- 1-déclaration du container
- 2-insertion des composants dans le container
    - 1) Déclaration du composant
    - 2) Insertion du composant dans son container

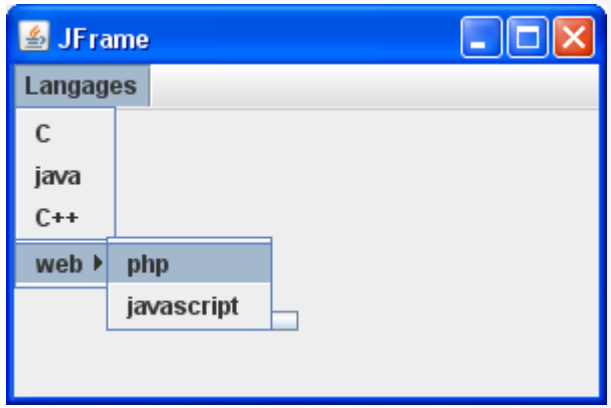
```
import javax.swing.JButton;
import javax.swing.JFrame;

public class TestIHM {

    public static void main(String[] args) {

        JFrame maFenetre = new JFrame("ma fenetre");
        JButton b=new JButton("mon bouton");

        maFenetre.getContentPane().add(b);
        maFenetre.pack();
        maFenetre.setVisible(true);
    }

}
```



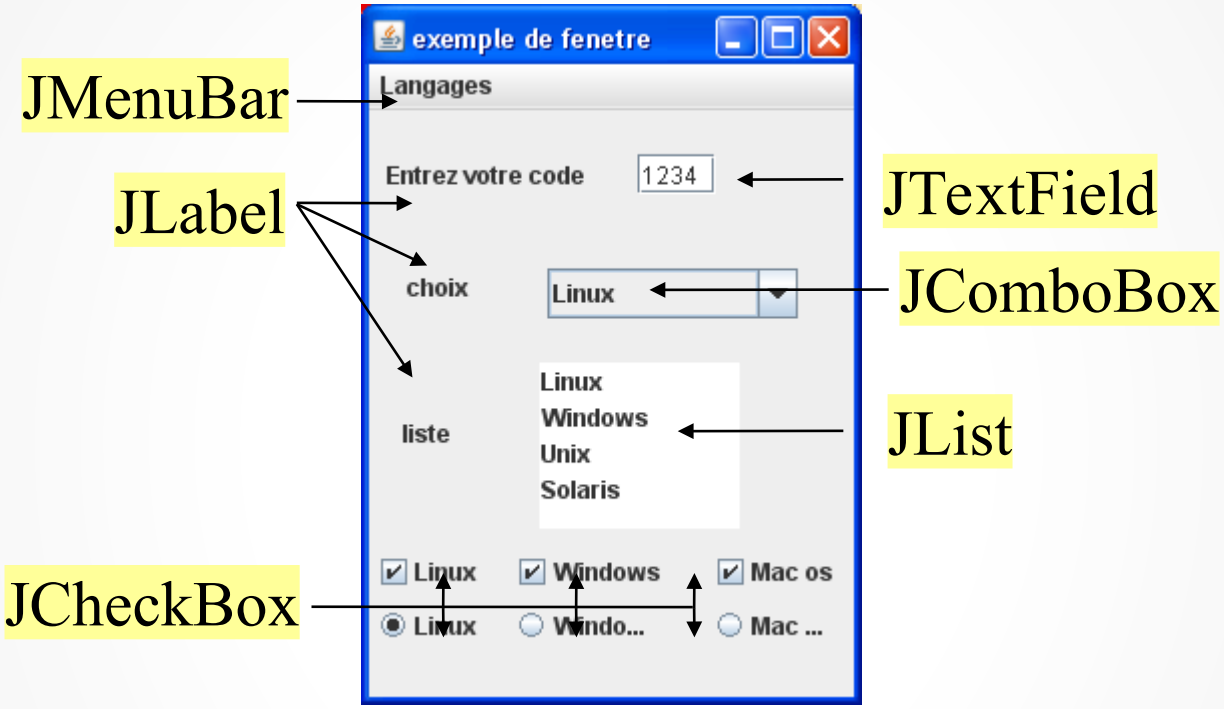
Remarque : Un container peut être inséré dans un autre container

Exemple composants swing





Exemple de composants swing



Conteneurs et composants (4) : utilisation

- On ajoute un composant dans un conteneur, avec la méthode add() :

```
JPanel p = new JPanel();
JButton b = new JButton();
p.add(b);
```

- De manière similaire, un composant est retiré de son conteneur par la méthode remove() :

```
p.remove(b);
```

- Un composant a (notamment) :
  - une taille préférée que l'on obtient/change avec getPreferredSize() et setPreferredSize()
  - une taille minimum que l'on obtient/change avec getMinimunSize() et setMinimunSize()
  - une taille maximum que l'on obtient/change avec getMaximunSize() et setMaximunSize()

Gestionnaire de placement (1)

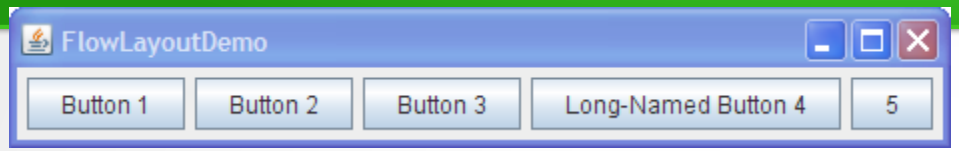
- Les conteneurs définissent la position des composants
  - ➔Soit en **position absolue** (x, y, largeur, hauteur)
  - ➔Soit selon le gestionnaire de placement associé au conteneur (**LayoutManager**)
    - Le gestionnaire calcule la position des composants

Gestionnaire de placement (2)

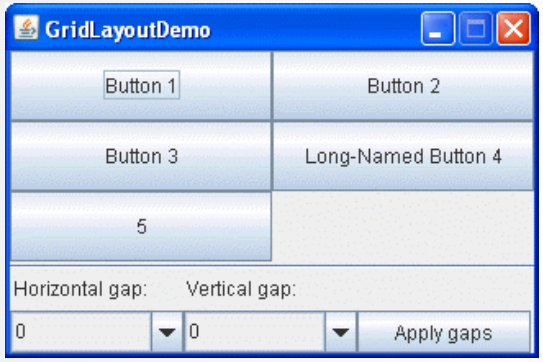
- La méthode setLayout(layoutManager) définie dans les conteneurs permet de changer le gestionnaire par défaut
- Un gestionnaire par défaut est défini pour chaque conteneur
- Il est possible de choisir un autre gestionnaire de présentation pour un conteneur donné

- La méthode pack() déclenche le calcul du placement
- La méthode invalidate() rend le placement courant invalide et donc le redimensionnement des composants à l'intérieur de celui-ci

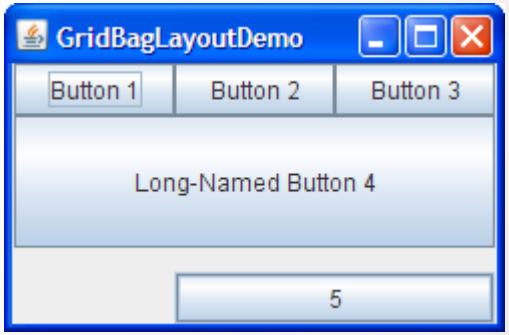
Gestionnaire de placement (3)



- FlowLayout
  - Place les composants de gauche à droite
- CardLayout
  - Superpose les composants



- GridLayout
  - Découpe en une grille régulière sur laquelle les composants sont placés

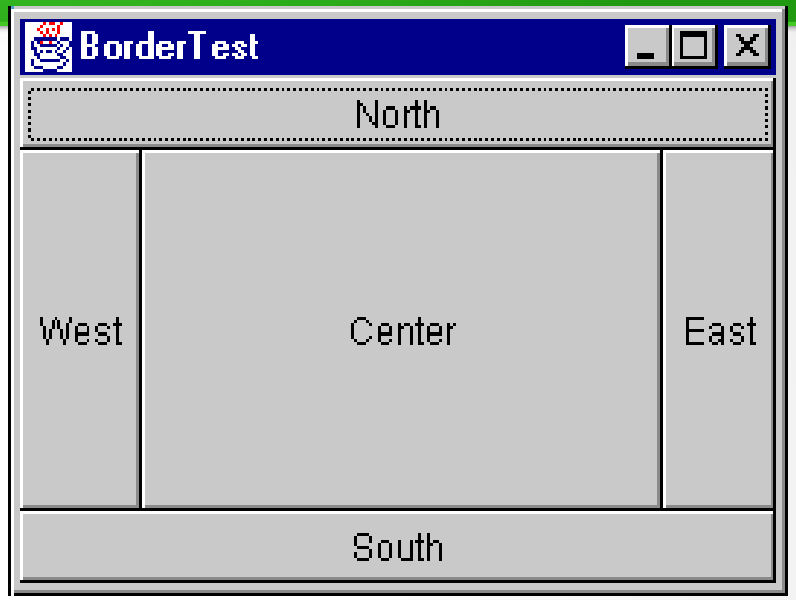


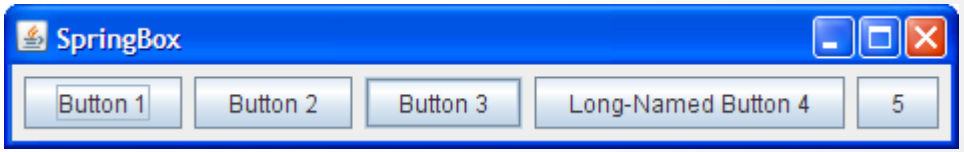
- GridBagLayout
  - Découpe en une grille et place les composants sur une ou plusieurs cases

Gestionnaire de placement (4)

- BorderLayout

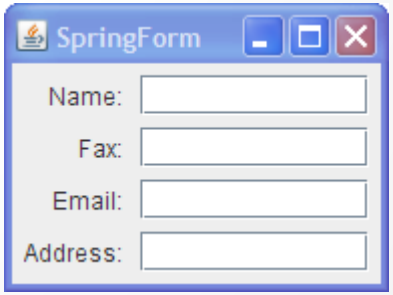
Découpe l'espace en 5 régions : centre, est, ouest, sud et nord





- SpringLayout

  Permet de définir de manière souple les distances entre les composants



Gestionnaire de placement (5)

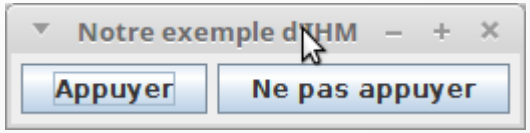
- Positionnement absolu
  - Null
  - On place les composants sur le conteneur en indiquant la position absolue avec la méthode **setBounds(x, y, h, l)**

Gestionnaire de placement (6)

- Tout conteneur possède un gestionnaire de présentation par défaut.
  - Toute instance de Container référence une instance de LayoutManager.
  - Il est possible d'en changer grâce à setLayout().

Premier exemple : ExempleIHM

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


public class ExempleIHM extends JFrame{
    private JButton b1;
    private JButton b2;

    public static void main(String args[]) {
        ExempleIHM that = new ExempleIHM();
        that.pack();  //change taille du Frame pour englober boutons
        that.setVisible(true);
    }

    public ExempleIHM() {
        super("Notre exemple d'IHM");  // lance le constructeur de la super classe, ici Frame
        setLayout(new FlowLayout());   // nouveau gestionnaire pres.
        b1 = new JButton("Appuyer");
        b2 = new JButton("Ne pas appuyer");
        Container content = getContentPane();
        content.add(b1);
        content.add(b2);
    }

}
```



Les étiquettes d'aide en ligne

- La classe ToolTipText permet de créer des aides en lignes qui apparaissent lorsque la souris passe sur un composant.

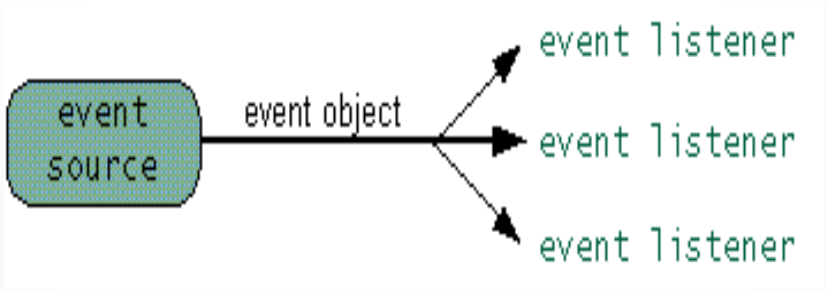
```
JButton monBouton = new JButton ("Un bouton");
monBouton.setToolTipText ("Aide de mon bouton");
```

- Ces quelques éléments ne sont qu'un aperçu de ce que propose Swing.

  - Il y a beaucoup de composants, de nouveaux gestionnaires de présentation, de nouveaux événements graphiques que l ’on ne peut présenter dans le cadre de ce cours...

Les événements générés

- Chaque fois qu'un utilisateur frappe un caractère du clavier, appuie sur un bouton de la souris ou la déplace, un événement est généré.
- Ces événements sont envoyés à un composant appelé écouteur, c'est lui qui appelera une routine de traitement d'événement pour recevoir l'événement.
- Le traitement de l'événement est ainsi délégué à une classe séparée.



Les événements générés (2)

- La gestion des événements passe par l'utilisation d'objets ***Listener*** et d'objets sources d'événements.
  - Un objet écouteur est l'instance d'une classe implémentant l'interface ***XXXXListener***.
  - Une source d'événements est un objet pouvant recenser des objets écouteurs et leur envoyer des objets événements.

Les événements générés (4)

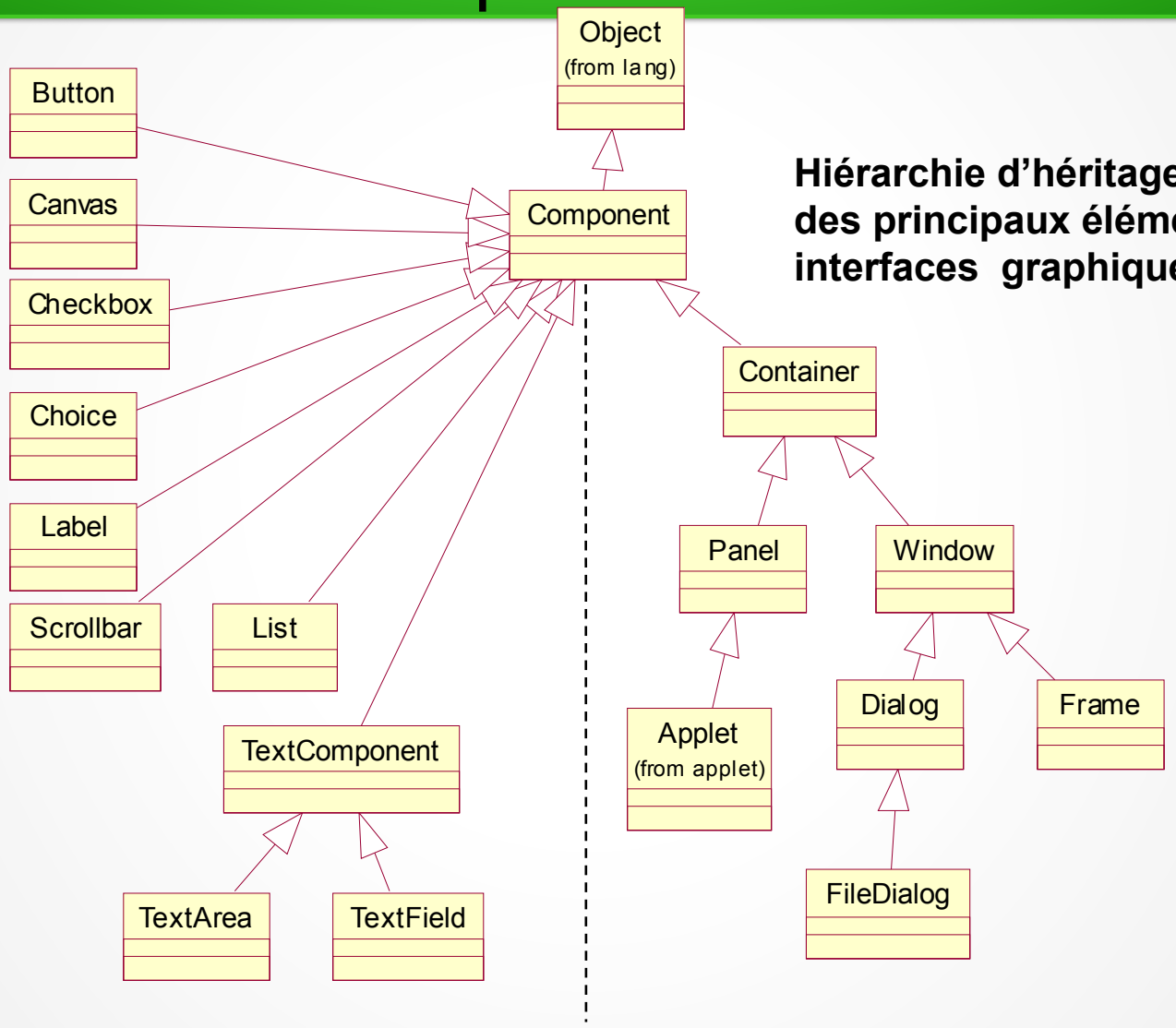
Signifie que la classe peut gérer des événements

Méthode invoquée lors d'un événement

```
import java.awt.*;
import java.awt.event.*;
import ButtonHandler;
class TestButton
{
  JFrame f;
 JButton b;
  public TestButton()
  {
     super(); f = new JFrame ("TestButton");
    Container panneauContenu;
      panneauContenu = f.getContentPane();
     b = new JButton ("Pressez Moi");
     panneauContenu.add ("Center", b) ;
     f.pack (); f.setVisible (true) ;
     b.addActionListener (new GestionEvenement ());
  }
  public static void main(String args[])
  {
    TestButton that = new TestButton();
 }}
```

```
import java.awt.event.*;
public class GestionEvenement implements ActionListener {
public void actionPerformed (ActionEvent e)
   {
     // recuperation de l'objet à l'origine de l'evenement
          Object src = e.getSource();
          // si l'objet est un bouton
          if (src instanceof JButton) {
             // recuperer le texte du bouton
             String texteDeLaTouche = ((JButton) src).getText();
           System.out.println("touche : "+texteDeLaTouche);

          }
   }
}
```

Conteneurs et composants : hiérarchie



Hiérarchie d'héritage des principaux éléments des interfaces graphiques en Java

Conclusion

- Beaucoup de widgets.
- Beaucoup de gestionnaires de placement.
- Les IDEs proposent des interfaces wysiwyg

  (What You See Is What You Get) facilitant le codage des événements et le placement des widgets.
- Le placement et l'adaptabilité des widgets sont des problématiques essentielles sur les interfaces graphiques.