

JTree

- Aperçu



- Rôle

- Afficher une structure arborescente

JTree

Le modèle TreeModel

- Structure
 - Object GetRoot() : la racine de l'arbre
 - Object getChild(Object obj, int n) :
 - Retourne le n^{ème} fils de obj
 - Int getChildCount(Object obj) :
 - Retourne le nombre de fils du noeud obj
 - Int getIndexOfChild(Object pere, Object fils)
 - Retourne l'indice du fils
 - Boolean isLeaf(Object obj)
 - Retourne vrai si obj est une feuille
 - Ecouteurs
 - void addTreeModelListener(TreeModelListener l)
 - void removeTreeModelListener(TreeModelListener l)
 - Prise en compte de modifications
 - void valueForPathChanged(TreePath path, Object nouveau)
-
-

JTree

La classe TreePath

- Rôle
 - Représente le chemin amenant à un noeud dans un arbre
 - Constructeurs
 - `TreePath(Object obj)`
 - `TreePath(Object [] obj)`
 - Quelques méthodes
 - `Object getLastPathComponent()`
 - `TreePath getParentPath()`
 - `Object[] getPath()`
 - `Object getPathComponent(int index)`
 - `int getPathCount()`
 - `boolean isDescendant(TreePath descendant)`
 - `TreePath pathByAddingChild(Object obj)`
-
-

JTree

La classe TreeModelListener

- Méthodes
 - Void treeNodeChanged(TreeModelEvent e)
 - Void treeNodeInserted(TreeModelEvent e)
 - Void treeNodeRemoved(TreeModelEvent e)
 - Void treeStructureChanged(TreeModelEvent e)
- Rmq :
 - Mis à part pour treeStructureChanged, les noeuds modifiés doivent avoir un même parent

JTree

La classe TreeModelEvent

- `Object[] getChildren()`
 - Les noeuds fils concernés
- `Int[] getChildIndices()`
 - Les indices des fils concernés
- `TreePath getTreePath()`
 - Chemin du noeud parent des noeuds concernés
- `Object[] getPath()`
 - raccourci de `getTreePath.getPath()`

JTree

Interface TreeNode

- Enumeration children()
 - boolean getAllowsChildren()
 - TreeNode getChildAt(int index)
 - int getChildCount()
 - int getIndex(TreeNode fils)
 - TreeNode getParent()
 - boolean isLeaf()
-
-

JTree

Interface MutableTreeNode (hérite de TreeNode)

- void insert(MutableTreeNode fils, int indice)
 - void remove(int indice)
 - void remove(MutableTreeNode fils)
 - void removeFromParent()
 - void setParent(MutableTreeNode nouveauParent)
 - void setUserObject(Object contenu)
-
-

JTree

La classe DefaultMutableTreeNode

- Constructeurs

- DefaultMutableTreeNode()
- DefaultMutableTreeNode(Object contenu)
- DefaultMutableTreeNode(Object contenu boolean filsAutorisés)

- Méthodes

- Void add(MutableTreeNode fils), void insert(MutableTreeNode fils, int indice)
 - Void setParent(MutableTreeNode parent). Attention : ne modifie pas les fils du parent
 - Void remove(int indice), void remove(MutableTreeNode fils)
 - Void removeAllChildren(), void removeFromParent()
 - Void setUserObject(Object contenu), Object getUserObject(), Object[] getUserObjectPath()
 - boolean getAllowsChildren()/void setAllowsChildren()
 - int getChildCount(), int getDepth(), int getLevel(), int getLeafCount()
 - int getIndex(TreeNode noeud)
 - boolean isRoot(), boolean isLeaf(), boolean isNodeChild(TreeNode n), boolean isNodeSibling(TreeNode n), boolean isNodeRelated(DefaultMutableTreeNode n), , boolean isNodeAncestor(TreeNode n), boolean isNodeDescendant(DefaultMutableTreeNode n),
-
-

JTree

La classe DefaultMutableTreeNode

- Méthodes (suite)

- Enumeration breadthFirstEnumeration(), Enumeration depthFirstEnumeration(), Enumeration postOrderEnumeration(), Enumeration preOrderEnumeration()
 - Enumeration children()
 - TreeNode getChildAfter(TreeNode fils), TreeNode getChildBefore(TreeNode fils), TreeNode getFirstChild(), TreeNode getLastChild(), TreeNode getChildAt(int indice)
 - DefaultMutableTreeNode getFirstLeaf(), DefaultMutableTreeNode getLastLeaf()
 - DefaultMutableTreeNode getNextLeaf()/getPreviousLeaf(), DefaultMutableTreeNode getNextNode()/getPreviousNode(), DefaultMutableTreeNode getNextSibling()/getPreviousSibling()
 - TreeNode getSharedAncestor(DefaultMutableTreeNode n)
 - TreeNode getParent(), TreeNode[] getPath(), TreeNode getRoot(), Enumeration pathFromAncestorEnumeration(TreeNode ancetre)
-
-

JTree

La classe DefaultTreeModel

- Constructeurs
 - DefaultTreeModel(TreeNode racine)
 - DefaultTreeModel(TreeNode racine, boolean askAllowsChildren)
- Feuilles d'un arbre
 - Si askAllowsChildren=true, feuille ssi noeud déclaré comme n'autorisant pas de fils. Un noeud autorisant des fils mais avec aucun fils ne sera pas considéré comme un noeud feuille

JTree

La classe *DefaultTreeModel*

Méthodes

- void addTreeModelListener(TreeModelListener l), void removeTreeModelListener(TreeModelListener l), TreeModelListener[] getTreeModelListeners()
 - boolean asksAllowsChildren(), void setAsksAllowsChildren(boolean newValue)
 - Object getChild(Object parent, int index), int getChildCount(Object parent), int getIndexOfChild(Object parent, Object child)
 - TreeNode[] getPathToRoot(TreeNode aNode)
 - Object getRoot(), void setRoot(TreeNode root)
 - void insertNodeInto(MutableTreeNode newChild, MutableTreeNode parent, int index), void removeNodeFromParent(MutableTreeNode node)
 - boolean isLeaf(Object node)
 - void nodeChanged(TreeNode node), void nodesChanged(TreeNode node, int[] childIndices), void nodeStructureChanged(TreeNode node), void nodesWereInserted(TreeNode node, int[] childIndices), void nodesWereRemoved(TreeNode node, int[] childIndices, Object[] removedChildren)
 - void valueForPathChanged(TreePath path, Object newValue)
 - void reload(), void reload(TreeNode node)
-
-

JTree

La classe JTree : constructeurs

- JTree()
 - JTree(Hashtable<?,?> value)
 - JTree(Object[] value)
 - JTree(TreeModel newModel)
 - JTree(TreeNode root)
 - JTree(TreeNode root, boolean asksAllowsChildren)
 - JTree(Vector<?> value)
-
-

JTree

La classe JTree : la sélection (1)

- Général
 - Void clearSelection()
 - Object getLastSelectedPathComponent()
 - TreeSelectionModel getSelectionModel()
 - Void setSelectionModel(TreeSelectionModel m)
 - Boolean isSelectionEmpty()
 - Écouteurs de la sélection
 - Void addTreeSelectionListener(SelectionListener l)
 - Void removeTreeSelectionListener(SelectionListener l)
 - TreeSelectionListener[] getTreeSelectionListeners()
 - Gestion à partir des chemins
 - void addSelectionPath(TreePath chem), void addSelectionPath(TreePath[] chem)
 - void setSelectionPath(TreePath chemin), void setSelectionPaths(TreePath[] chemins)
 - removeSelectionPath(TreePath chemin), removeSelectionPaths(TreePath[] chemins)
 - TreePath getAnchorSelectionPath(), void setAnchorSelectionPath(TreePath chem)
 - TreePath getLeadSelectionPath(), void setLeadSelectionPath(TreePath chemin)
 - TreePath getSelectionPath(), TreePath[] getSelectionPaths()
 - Boolean isPathSelected(TreePath chemin)
-
-

JTree

La classe JTree : la sélection (2)

- Gestion à partir des lignes
 - `void addSelectionInterval(int indice0, indice1), setSelectionInterval(int indice0, int indice1), removeSelectionInterval(int indice0, int indice1)`
 - `void addSelectionRow(int indice), void setSelectionRow(int indice), removeSelectionRow(int indice)`
 - `void addSelectionRows(int[] indices), void setSelectionRows(int[] indices), removeSelectionRows(int[] indices)`
 - `int getLeadSelectionRow(), int getMaxSelectionRow(), int getMinSelectionRow(), int[] getSelectionRows()`
 - `Boolean isRowSelected(int ligne)`
 - Conversion lignes/path
 - `TreePath getPathForRow(int ligne)`
 - `int getRowForPath(TreePath path)`
-
-

JTree

Options d'affichage

- `TreeCellRenderer getCellRenderer()`
 - `Void setCellRenderer(TreeCellRenderer t)`
 - `Int getRowHeight()`, `void setRowHeight (int h)`, `boolean isFixedRowHeight()`
 - `Boolean getScrollsOnExpand()`, `void setScrollsOnExpand(boolean b)`
 - `Boolean getShowsRootHandles()`, `void setShowsRootHandles(boolean b)`
 - `Boolean isRootVisible()`, `void setRootVisible(boolean b)`
 - `Void setVisibleRowCount (int nb)`, `int getVisibleRowCount()`
-
-

JTree

expansion/compression

- void expandPath(TreePath chemin)
- void expandRow(int ligne)
- boolean hasBeenExpanded(TreePath chemin)
- boolean isCollapsed(int row)
- boolean isCollapsed(TreePath path)
- boolean isExpanded(int row)
- boolean isExpanded(TreePath path)
- boolean isVisible(TreePath path)
- void makeVisible(TreePath path)
- void scrollPathToVisible(TreePath chemin)
- void scrollRowToVisible(int ligne)
- void setExpandsSelectedPath(boolean b)
- void setScrollsOnExpand(boolean b)
- void setToggleClickCount(int n)

JTree

Edition

- `Void cancelEditing()`
- `TreeCellEditor getCellEditor()`
- `TreePath getEditingPath()`
- `Boolean getInvokesStopCellEditing()`
- `Boolean isEditable()`
- `Boolean isEditing()`
- `Boolean isPathEditable()`
- `Void setCellEditor(TreeCellEditor t)`
- `Void setEditable(boolean b)`
- `Boolean stopEditing()`
- `Void setInvokesStopCellEditing(boolean b)`
- `Void startEditingAtPath(TreePath chemin)`

JTree

coordonnées

- `TreePath` `getClosestPathForLocation(int x, int y)`
 - `int` `getClosestRowForLocation(int x, int y)`
 - `Rectangle` `getPathBounds(TreePath path)`
 - `Rectangle` `getRowBounds(int row)`
 - `TreePath` `getPathForLocation(int x, int y)`
 - `int` `getRowForLocation(int x, int y)`
-
-

JTree : interface TreeSelectionModel et classe DefaultTreeSelectionModel

- void addPropertyChangeListener(PropertyChangeListener listener), void removePropertyChangeListener(PropertyChangeListener listener)
 - void addTreeSelectionListener(TreeSelectionListener x), void removeTreeSelectionListener(TreeSelectionListener x)
 - void addSelectionPath(TreePath path), void addSelectionPaths(TreePath[] paths)
 - void clearSelection()
 - TreePath getLeadSelectionPath(), int getLeadSelectionRow()
 - int getMaxSelectionRow(), int getMinSelectionRow(), int getSelectionCount()
 - RowMapper getRowMapper(), void setRowMapper(RowMapper newMapper), void resetRowSelection()
 - int getSelectionMode(), void setSelectionMode(int mode) : 3 modes
 - SINGLE_TREE_SELECTION,
 - CONTIGUOUS_TREE_SELECTION
 - DISCONTIGUOUS_TREE_SELECTION.
 - TreePath getSelectionPath(), TreePath[] getSelectionPaths(), int[] getSelectionRows()
 - boolean isPathSelected(TreePath path), boolean isRowSelected(int row)
 - boolean isSelectionEmpty()
 - void removeSelectionPath(TreePath path), void removeSelectionPaths(TreePath[] paths)
 - void setSelectionPath(TreePath path), void setSelectionPaths(TreePath[] paths)
-
-

JTree

Événements de sélection

- Interface `TreeSelectionListener`
 - 1 méthode : `void valueChanged(TreeSelectionEvent e)`
 - Classe `TreeSelectionEvent`
 - `TreePath` `getNewLeadSelectionPath()`
 - `TreePath` `getOldLeadSelectionPath()`
 - `TreePath` `getPath()`
 - `TreePath[]` `getPaths()`
 - `Boolean` `isAddedPath()`
 - `Boolean` `isAddedPath(int indice)`
 - `Boolean` `isAddedPath(TreePath path)`
-
-

JTree : Événements développement/réduction

- Interface TreeExpansionListener
 - void treeCollapsed(TreeExpansionEvent e)
 - void treeExpanded(TreeExpansionEvent e)
- Interface TreeWillExpandListener
 - void treeWillCollapse(TreeExpansionEvent e)
 - void treeWillExpand(TreeExpansionEvent e)
- Classe TreeExpansionEvent
 - TreePath getPath()

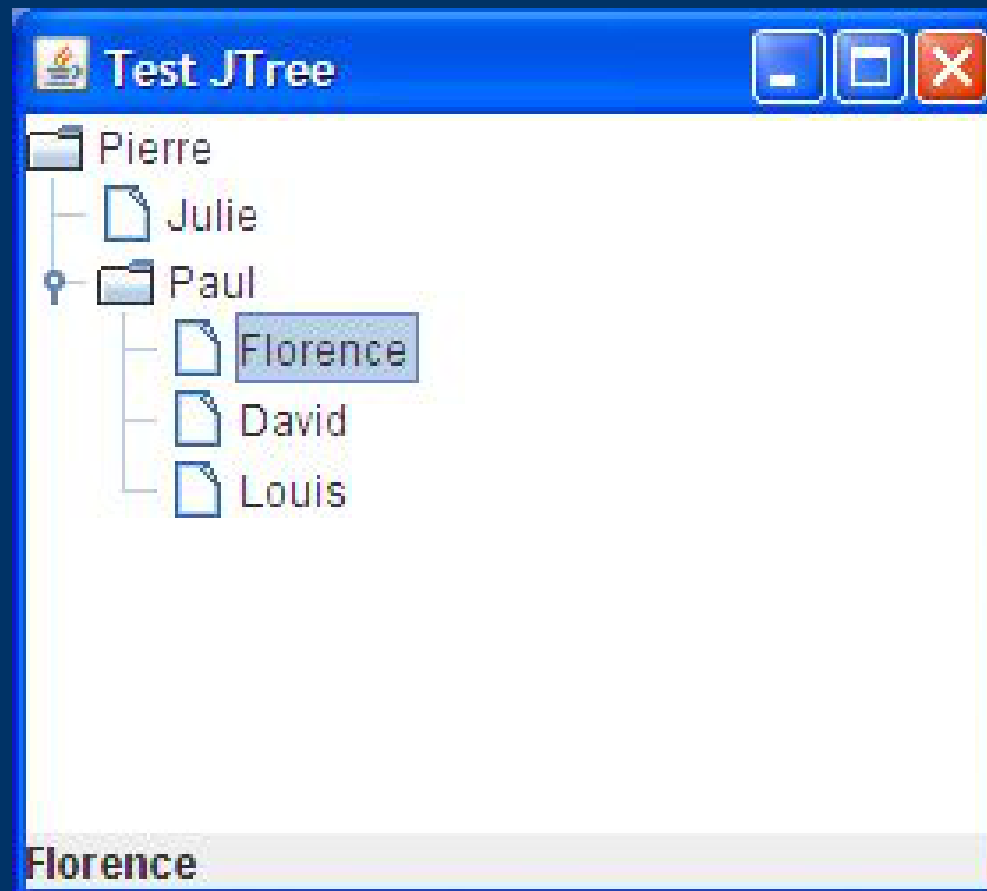
Exemple de JTree - code

```
public TestJTree() {
    super("Test JTree");
    JPanel panneau = new JPanel(); panneau.setLayout(new BorderLayout()); setContentPane(panneau);
    Jarbre = new JTree(); panneau.add(Jarbre,BorderLayout.CENTER);
    etiquette = new JLabel("    "); panneau.add(etiquette,BorderLayout.SOUTH);
    DefaultMutableTreeNode racine = new DefaultMutableTreeNode("Pierre");
    TreeModel arbre = new DefaultTreeModel(racine); DefaultMutableTreeNode noeudCourant,ancetre;
    anctre = racine; noeudCourant = new DefaultMutableTreeNode("Julie"); anctre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("Paul"); anctre.add(noeudCourant);
    anctre = noeudCourant; noeudCourant = new DefaultMutableTreeNode("Florence");
    anctre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("David"); anctre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("Louis"); anctre.add(noeudCourant);
    Jarbre.setModel(arbre); TreeSelectionModel modeleSelection = new DefaultTreeSelectionModel();
    modeleSelection.setSelectionMode(TreeSelectionModel.SINGLE_TREE_SELECTION);
    Jarbre.setSelectionModel(modeleSelection); Jarbre.addTreeSelectionListener(new gestionSelection());
    pack(); setVisible(true);
}

class gestionSelection implements TreeSelectionListener {
    public void valueChanged(TreeSelectionEvent e) {
        TreeNode noeud = (TreeNode) e.getPath().getLastPathComponent();
        etiquette.setText(noeud.toString());
    }
}

public static void main(String[] args) {
    JFrame fenetre = new TestJTree(); fenetre.setDefaultCloseOperation(EXIT_ON_CLOSE);
}}
```

Exemple de JTree - aperçu



JTree : mise à jour après modification d'un noeud

- Pb
 - Un noeud est modifié sans passer par le Jtree
 - Mise à jour de l'affichage ?
- Solution : 2 méthodes au choix
 - Appeler la méthode `updateUI()` sur le Jtree
 - Si le modèle sous-jacent est un `DefaultTreeModel`, appeler sur le modèle `reload(noeudModifié)`

Modification d'un nœud de JTree - code

```
public class TestJTree2 extends JFrame {
    private JTree Jarbre; private DefaultTreeModel arbre; private JTextField etiquette; private JButton valider;
    public TestJTree2() {super("Test JTree"); JPanel panneau = new JPanel(); panneau.setLayout(new BorderLayout());
        setContentPane(panneau); JPanel panneauBas = new JPanel(); Jarbre = new JTree(); Jarbre.setEditable(true);
        etiquette = new JTextField(""); etiquette.setPreferredSize(new Dimension(etiquette.getPreferredSize().width+150,
        etiquette.getPreferredSize().height));
        valider = new JButton("Valider"); valider.addActionListener(new gestionValider());
        panneauBas.add(etiquette); panneauBas.add(valider);
        panneau.add(Jarbre,BorderLayout.CENTER); panneau.add(panneauBas,BorderLayout.SOUTH);
        DefaultMutableTreeNode racine = new DefaultMutableTreeNode("Pierre");
        arbre = new DefaultTreeModel(racine); DefaultMutableTreeNode noeudCourant,ancetre; ancre = racine;
        noeudCourant = new DefaultMutableTreeNode("Julie"); ancre.add(noeudCourant);
        noeudCourant = new DefaultMutableTreeNode("Paul"); ancre.add(noeudCourant);
        ancre = noeudCourant; noeudCourant = new DefaultMutableTreeNode("Florence"); ancre.add(noeudCourant);
        noeudCourant = new DefaultMutableTreeNode("David"); ancre.add(noeudCourant);
        noeudCourant = new DefaultMutableTreeNode("Louis"); ancre.add(noeudCourant);
        Jarbre.setModel(arbre); TreeSelectionModel modeleSelection = new DefaultTreeSelectionModel();
        modeleSelection.setSelectionMode(TreeSelectionModel.SINGLE_TREE_SELECTION);
        Jarbre.setSelectionModel(modeleSelection); Jarbre.addTreeSelectionListener(new gestionSelection());
        pack(); setVisible(true);}
    class gestionSelection implements TreeSelectionListener { public void valueChanged(TreeSelectionEvent e) {
        TreeNode noeud = (TreeNode) e.getPath().getLastPathComponent(); etiquette.setText(noeud.toString());}}
    class gestionValider implements ActionListener {public void actionPerformed(ActionEvent e) {
        MutableTreeNode noeud = (MutableTreeNode) Jarbre.getSelectionPath().getLastPathComponent();
        noeud.setUserObject(etiquette.getText()); Jarbre.updateUI();}}
    public static void main(String[] args) {JFrame fenetre = new TestJTree2();
        fenetre.setDefaultCloseOperation(EXIT_ON_CLOSE);
    }}
```

Modification d'un nœud de JTree

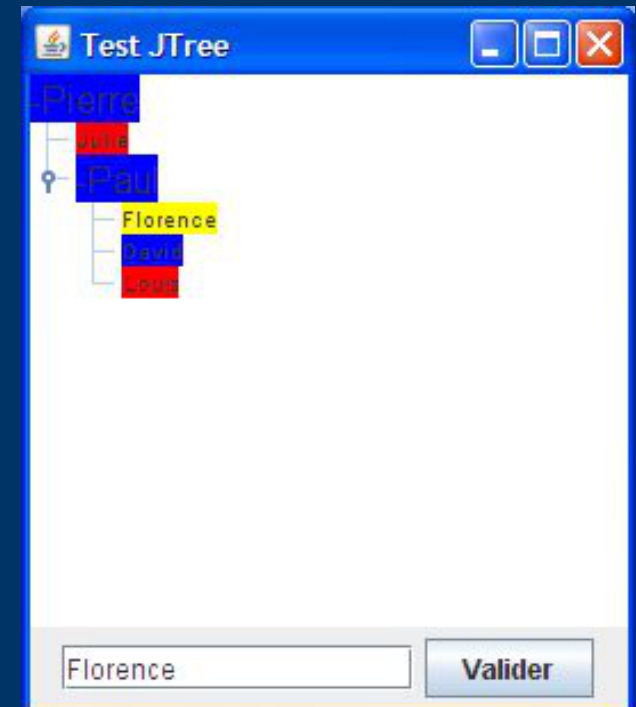
aperçu



JTree

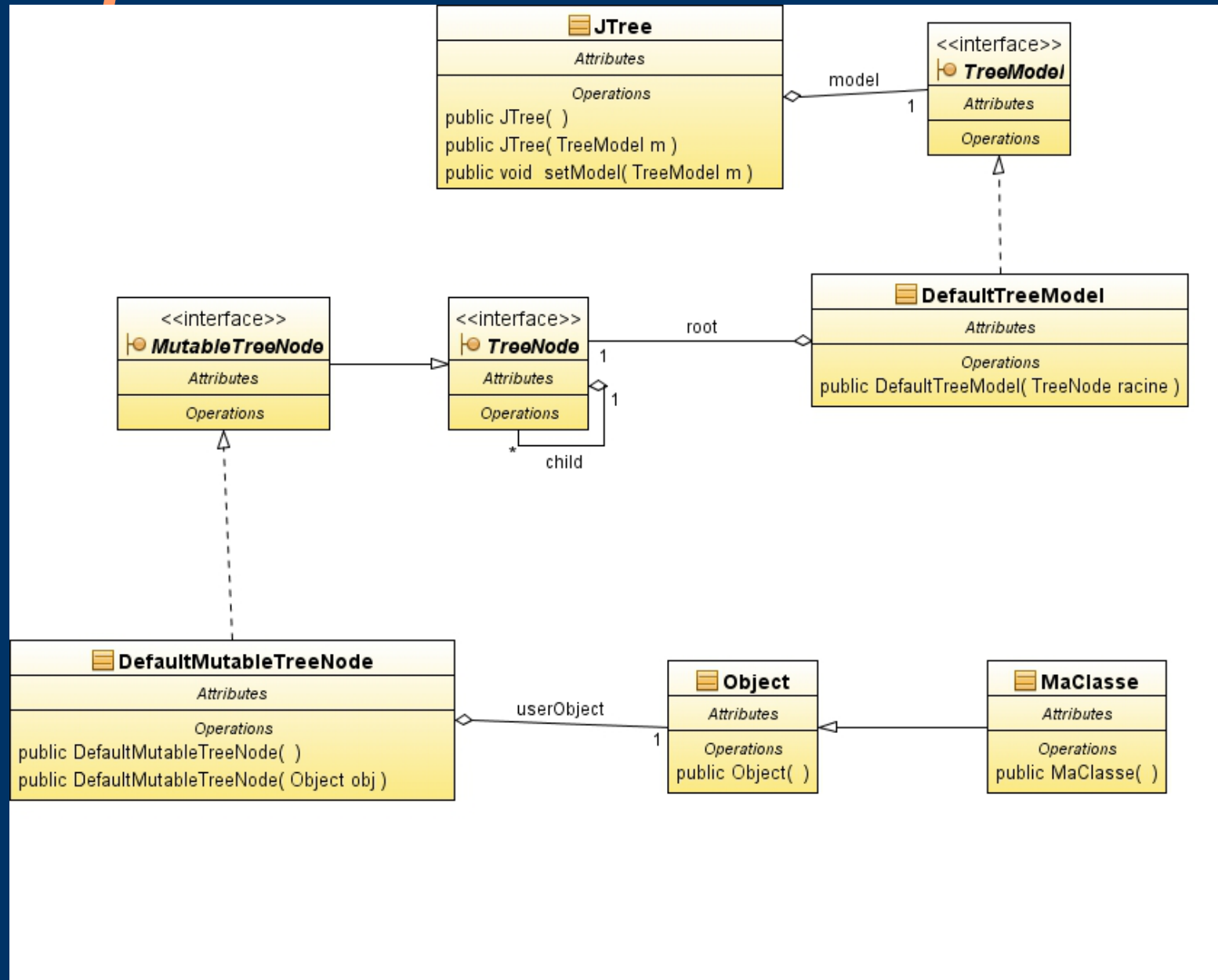
Modification de l'affichage

```
class affichage implements TreeCellRenderer {
    public Component getTreeCellRendererComponent(JTree tree, Object value, boolean selected,
        boolean expanded, boolean leaf, int row, boolean hasFocus) {
        //On ne va pas afficher d'icone mais mettre un "+" pour un noeud non-feuille non développé
        // et un "-" pour un noeud non-feuille développé. Les noeuds non-feuilles seront affichés en 16 points, les
        noeuds feuilles en 10 points. De plus, on va alterner les couleurs de fond.
        DefaultMutableTreeNode noeud = (DefaultMutableTreeNode) value;
        String contenu = (String) noeud.getUserObject();
        JLabel resultat = new JLabel();
        Font fonte1 = new Font("SansSerif", 0, 16); Font fonte2 = fonte1.deriveFont(0, 10);
        if (!leaf) {
            if (expanded) { contenu = "-" + contenu; } else { contenu = "+" + contenu; }
            resultat.setFont(fonte1);
        }
        else { resultat.setFont(fonte2); }
        if (selected) { resultat.setBackground(Color.YELLOW); }
        else {
            if (row % 2 == 0) { resultat.setBackground(Color.BLUE); }
            else { resultat.setBackground(Color.RED); }
        }
        resultat.setOpaque(true); resultat.setText(contenu);
        return resultat;
    }
}
```



JTree – Bilan

Classes pour une utilisation de base



JTree – Bilan

Obtension de l'objet sélectionné

Soit *arbre* le JTree :

```
TreePath tp = arbre.getSelectionPath();
```

```
DefaultMutableTreeNode noeud = (DefaultMutableTreeNode)  
    tp.getLastPathComponent();
```

```
MaClasse donnee = (MaClasse) noeud.getUserObject();
```

Remarques :

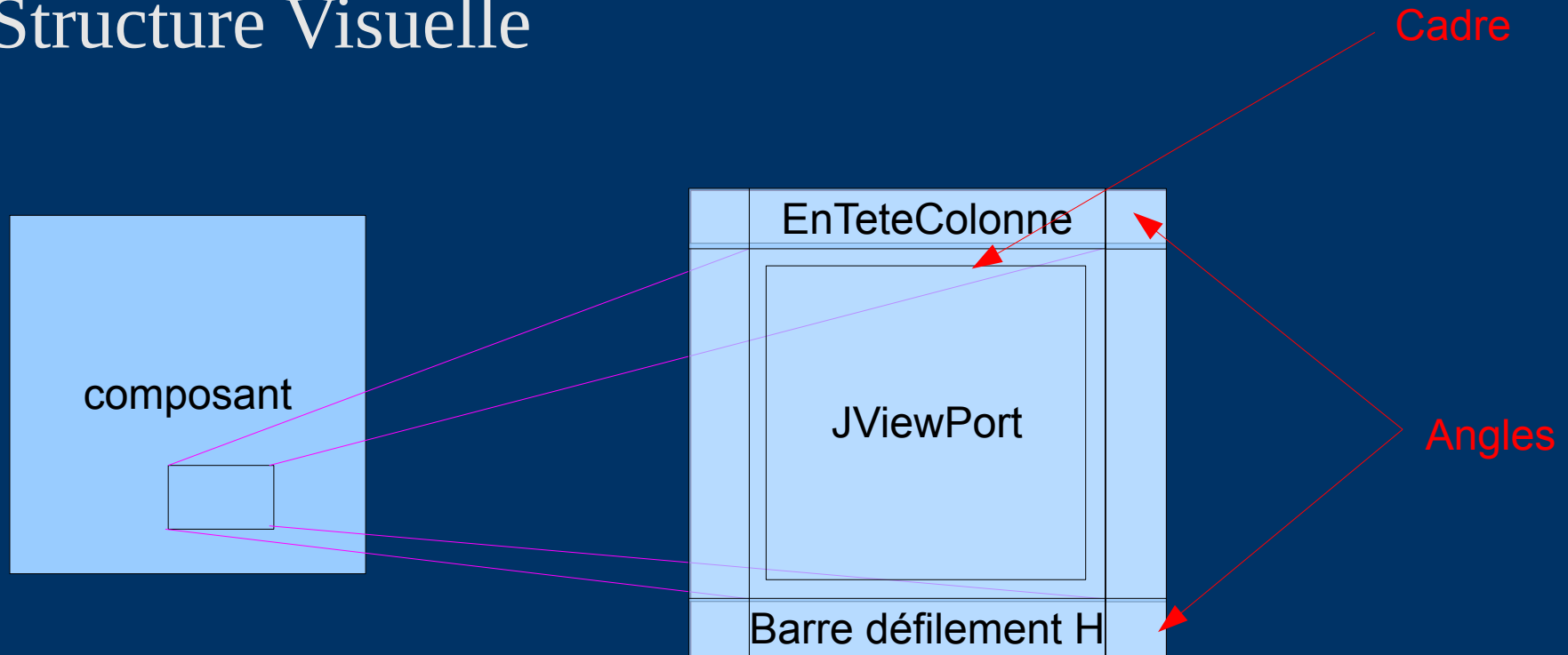
- `getLastPathComponent()` retourne un *Object*
 - `getUserObject()` retourne un *Object*
-
-

Conteneurs spécifiques



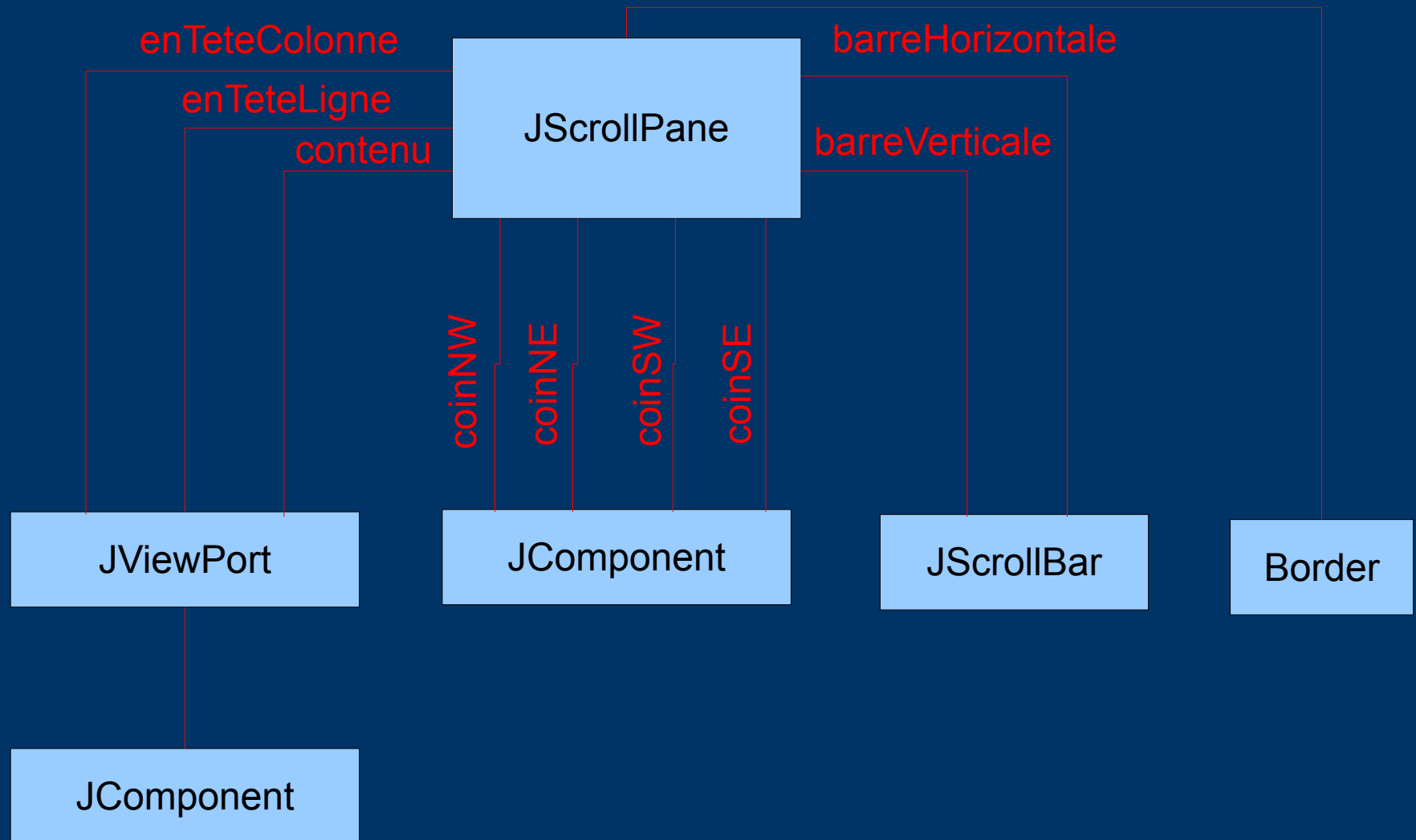
JScrollPane

- Rôle
Ajouter des ascenseurs à n'importe quel composant
- Structure Visuelle



JScrollPane

Structure interne



JScrollPane

Affichage du composant principal

- Taille du ViewPort
 - Déterminée par le JScrollPaneLayout
 - Affichage des barres de défilement
 - Méthodes
 - Void setHorizontalScrollBarPolicy(int politique)
 - Void setVerticalScrollBarPolicy(int politique)
 - Valeurs ($XXX \in \{\text{HORIZONTAL}, \text{VERTICAL}\}$)
 - XXX_SCROLLBAR_AS_NEEDED
 - XXX_SCROLLBAR_NEVER
 - XXX_SCROLLBAR_ALWAYS
-
-

JScrollPane

Affichage d'autres composants

- En-têtes
 - void setColumnHeaderView(Component c)
 - void setRowHeaderView(Component c)
 - Coins
 - void setCorner(String position, Component c)
 - Avec *position* pouvant valoir :
 - LOWER_LEFT_CORNER, LOWER_RIGHT_CORNER
 - UPPER_LEFT_CORNER, UPPER_RIGHT_CORNER
 - LOWER_LEADING_CORNER, LOWER_TRAILING_CORNER
 - UPPER_LEADING_CORNER, UPPER_TRAILING_CORNER
-
-

JScrollPane

Gestion du déplacement

- BlockIncrement
 - Déplacement lorsque l'on clique avant/après un ascenseur
- UnitIncrement
 - Déplacement lorsque l'on clique sur les flèches de l'ascenseur
- Molette de la souris
 - Géré par le système
 - Activé par `setWheelScrollingEnabled(true)`

JScrollPane

Détermination des incréments

- Si le composant n'implante pas Scrollable
 - Incréments déterminés par JScrollPane
- Si le composant implante Scrollable
 - Par le composant lui-même

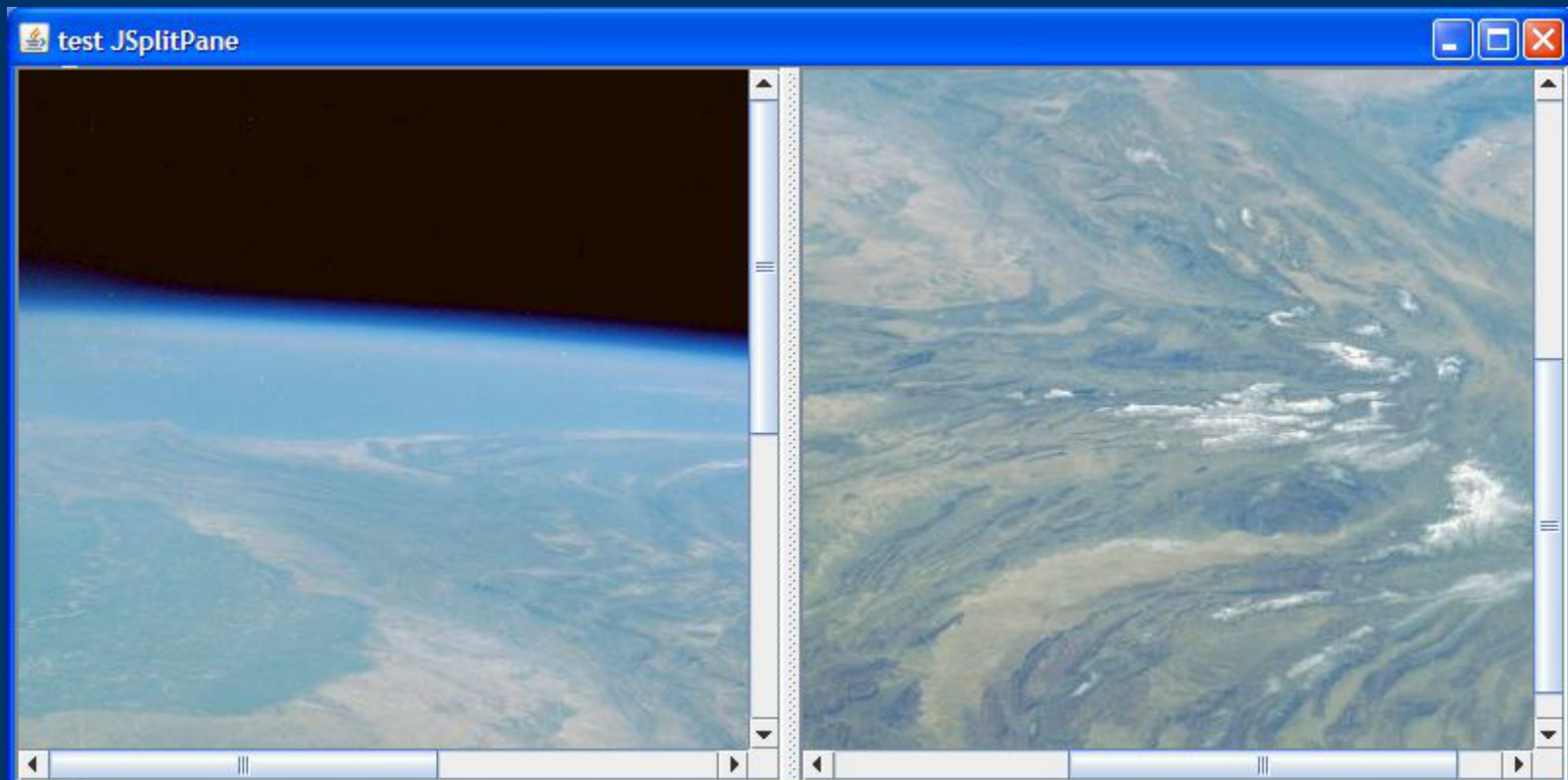
JScrollPane

L'interface Scrollable

- Taille préférée du ViewPort
Dimension `getPreferredScrollableViewportSize()`
- Détermination des incréments
 - Méthodes
 - `Int getScrollableBlockIncrement(Rectangle recVisible, int orientation, int direction)`
 - `Int getScrollableUnitIncrement(Rectangle recVisible, int orientation, int direction)`
 - Paramètres
 - Orientation : VERTICAL/HORIZONTAL
 - Direction : $<0 \rightarrow$ gauche/haut, $>0 \rightarrow$ droite/bas
- Forcer la taille du composant à être la taille du viewPort
 - `Boolean getScrollableTrackViewportHeight()`
 - `Boolean getScrollableTrackViewportWidth()`

JSplitPane

- But
 - Partager une fenêtre en 2 parties, verticalement ou horizontalement, avec barre de séparation déplaçable
- Aperçu



JSplitPane

Constructeurs

- Liste
 - JSplitPane()
 - JSplitPane(int newOrientation)
 - JSplitPane(int newOrientation, boolean newContinuousLayout)
 - JSplitPane(int newOrientation, boolean newContinuousLayout, Component newLeftComponent, Component newRightComponent)
 - JSplitPane(int newOrientation, Component newLeftComponent, Component newRightComponent)
- Paramètres
 - NewOrientation : HORIZONTAL_SPLIT / VERTICAL_SPLIT
 - NewContinuousLayout : mise à jour en temps réel ou seulement après le positionnement
 - NewLeftComponent : composant de gauche/du haut
 - NewRightComponent : composant de droite/du bas

JSplitPane

Quelques méthodes

- `Void setDividerLocation(double pos)`
 - `Void setRightComponent(Component c) / void setBottomComponent`
 - `Void setLeftComponent(Component c) / void setTopComponent`
 - `Void setOrientation(int orientation)`
 - `Void setOneTouchExpandable(boolean b)`
 - `Void setResizeWeight(double val)`
 - `Void setDividerSize(int taille)`
 - `Void setContinuousLayout(boolean b)`
-
-

JTabbedPane

- Rôle

Gérer un affichage de composants sous forme d'onglets

- Aperçu



JTabbedPane

Affichage des onglets

- Position
 - Spécification
 - Constructeurs
 - `setTabPlacement(int place)`
 - Valeurs possibles
 - TOP, BOTTOM
 - LEFT, RIGHT
 - Politique
 - Spécification
 - Constructeurs
 - `setTabLayoutPolicy(int politique)`
 - Valeurs possibles
 - WRAP_TAB_LAYOUT
 - SCROLL_TAB_LAYOUT
 - Nombre de lignes/colonnes utilisées : `int getTabRunCount()`
-
-

JTabbedPane

Ajout/Suppression d'une page

- Ajout
 - Component add(String titre, Component c)
 - Component add(Component c) (titre = c.getName())
 - Component add(Component c, int index) (=insertTab)
 - void add(Component c, Object contrainte)
 Si contrainte : String ou Icon, utilisé pour le titre
 - void add(Component c, Object contrainte, int index)
 - void addTab(String nom, Component c)
 - void addTab(String nom, Icon icone, Component c)
 - void addTab(String nom, Icon icone, Component c, String tip)
 - void insertTab(String titre, Icon icone, Component c, String tip, int index)
- Suppression
 - Void remove(Component c)
 - Void remove(int index)
 - Void removeAll()
 - Void removeTabAt(int index)

JTabbedPane

Gestion de la sélection

- Modèle sous-jacent
 - SingleSelectionModel (getModel/setModel)
- Méthodes
 - Component GetSelectedComponent()
 - Int GetSelectedIndex()
 - Void setSelectedComponent(Component c)
 - Void setSelectedIndex(int i)
- Détection de changements de sélection
 - addChangeListener/removeChangeListener()

JTabbedPane

Paramètres des pages

- Icône/Titre
 - `setIconAt(int index, Icon icône)/setDisabledIconAt...`
 - `setTitleAt(int index, String titre)`
- Bulle d'aide
 - `setToolTipTextAt(int index, String tip)`
- Couleurs
 - `setForegroundAt(int index, Color c)`
 - `setBackgroundAt(int index, Color c)`
- Validation
 - `setEnabledAt(int index, boolean b)`
- Code mnémonique
 - `setMnemonicAt(int index, int m)`

JTabbedPane

Conversion composant/index

- Méthodes
 - Component `getComponentAt(int index)`
 - Int `indexOfComponent(Component c)`
 - Int `indexOfTab(Icon icône)`
 - Int `indexOfTab(String titre)`
 - Int `indexOfTabComponent(Component c)`
- Attention !
 - Component `getTabComponentAt(int index)`
 - Ne renvoie pas le composant mis dans la page, mais le composant affichant le titre !
 - Void `setTabComponentAt(int index)`
 - Spécifie le composant qui affichera le titre de la page

JDesktopPane et JInternalFrame

- But
 - Gérer des sous-fenêtres dans une fenêtre (bureau virtuel)
- Aperçu



JDesktopPane et JInternalFrame

Exemple

```
public class TestJDesktopPane extends JFrame {
    public TestJDesktopPane() {
        super("test JDesktopPane"); JDesktopPane contenu = new JDesktopPane(); setContentPane(contenu);
        JInternalFrame f1 = new JInternalFrame("Fenêtre 1");
        JLabel p1 = new JLabel("contenu de la fenêtre 1");
        f1.add(p1); f1.pack(); f1.setVisible(true);
        f1.setResizable(true); f1.setClosable(true);
        JInternalFrame f2 = new JInternalFrame("Fenêtre 2");
        JMenuBar barre = new JMenuBar();
        JMenu fichier = new JMenu("Fichier");
        fichier.add("Ouvrir"); fichier.add("Sauver"); fichier.add("Fermer"); barre.add(fichier);
        f2.setJMenuBar(barre);
        f2.add(new JLabel(new ImageIcon("puits.gif")));
        f2.pack(); f2.setVisible(true); f2.setResizable(true);
        contenu.add(f1); contenu.add(f2);
        setSize(400,200); setVisible(true);
    }
    public static void main(String[] args) {
        JFrame fenetre = new TestJDesktopPane();
        fenetre.setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

JToolBar

Principe

- But :
 - Implanter des palettes d'outils, en général via des boutons
- Principe
 - Les composants sont disposés en utilisant un BorderLayout
 - La palette doit être disposée dans un composant dont la mise en page est géré par un BorderLayout
 - Le contenu du composant est dans la zone CENTER
 - La palette peut être déplacé entre les 4 zones périphériques, voire transformée en fenêtre

JToolBar Exemple



```
JLabel photo1 = new JLabel(new  
ImageIcon(url));  
add(photo1, BorderLayout.CENTER);  
JToolBar palette = new JToolBar("Palette");  
JButton b1 = new JButton("b1");  
JButton b2 = new JButton("b2");  
palette.add(b1);  
palette.add(b2);  
add(palette, BorderLayout.NORTH);
```

JToolBar

Paramétrage

- `setFloatable(false)`
la palette ne peut pas être déplacée
 - `setRollOver(false)`
le bouton sous la souris n'est pas mis en évidence
 - `AddSeparator()`
Ajoute un séparateur
 - `setBorderPainted(true)`
Si un cadre a été défini via `setBorder(Border b)`, le dessine ;
semble incompatible avec `Floatable`
-
-