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 treeNodesChanged(TreeModelEvent e)
  - Void treeNodesInserted(TreeModelEvent e)
  - Void treeNodesRemoved(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 askAllowsChidren=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 setSelectoinPaths(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
Evénements de sélection

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

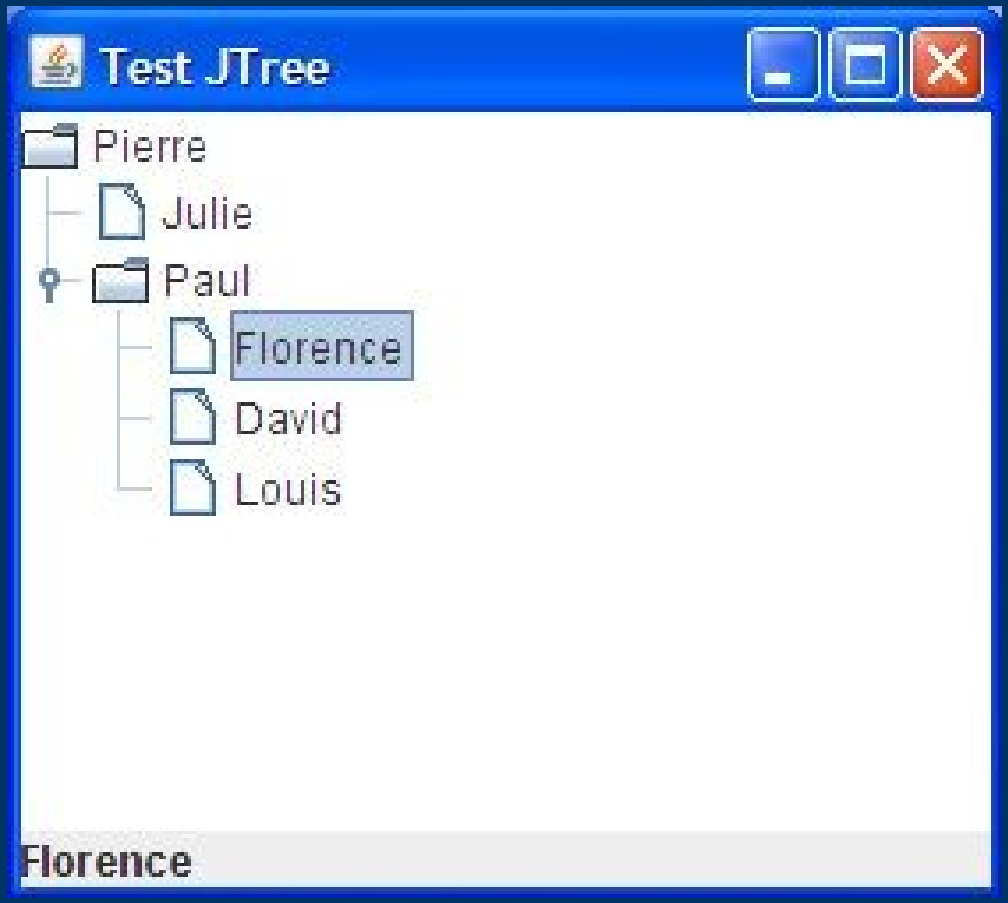
JTree : Evénements développement/réduction

- 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;
    ancetre = racine; noeudCourant = new DefaultMutableTreeNode("Julie"); ancetre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("Paul"); ancetre.add(noeudCourant);
    ancetre = noeudCourant; noeudCourant = new DefaultMutableTreeNode("Florence");
   ancetre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("David"); ancetre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("Louis"); ancetre.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

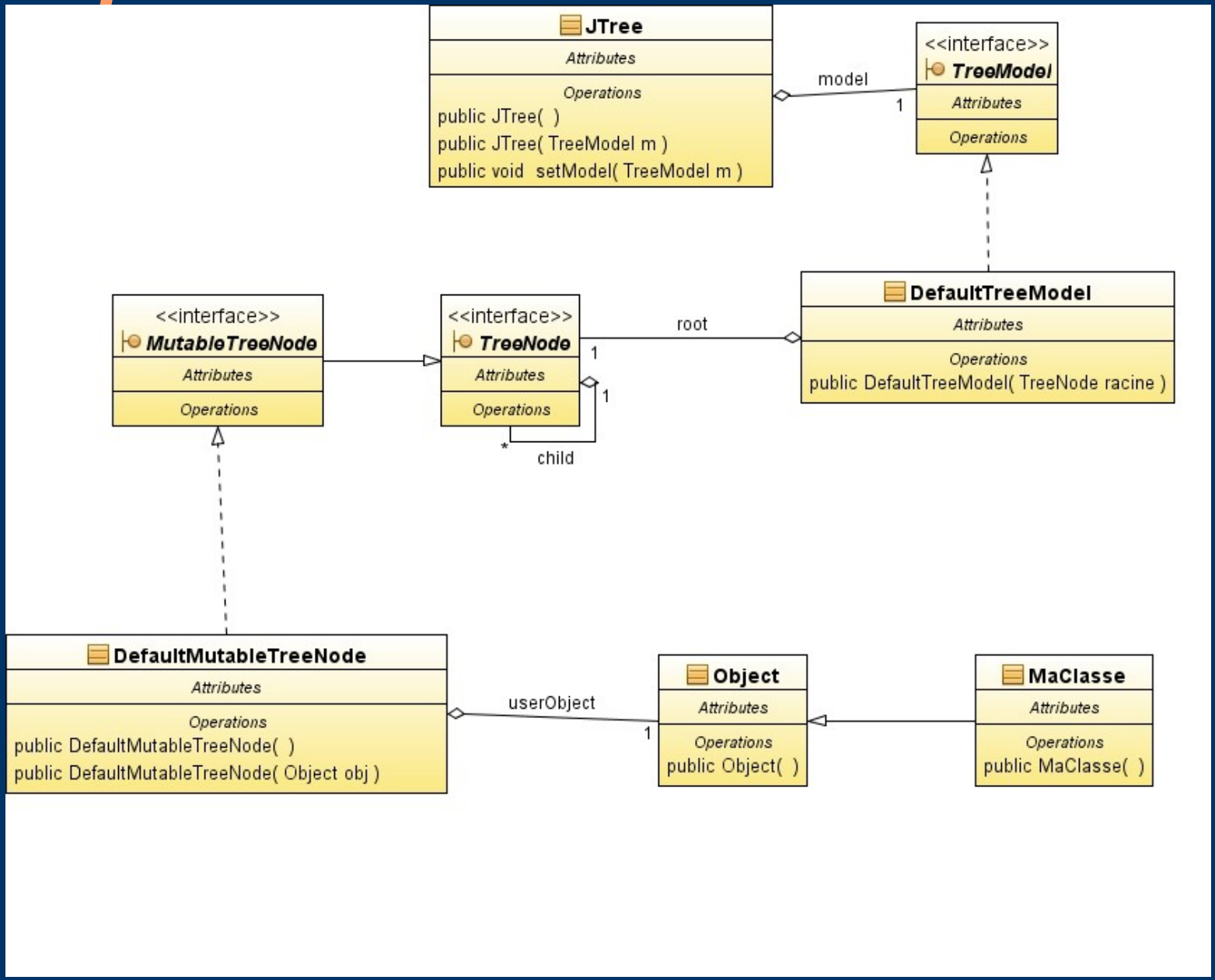
```
public class TestJTree2 extends JFrame {
  private JTree Jarbre; private DefaultTreeModel arbre; private JTextField etiquette; private JButton valider;
  public TestJTree2() {super("Test JTree"); JPanel panneau = new JPanel(); panneau.setLayout(new BorderLayout());
    setContentPane(panneau); JPanel 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; ancetre = racine;
    noeudCourant = new DefaultMutableTreeNode("Julie"); ancetre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("Paul"); ancetre.add(noeudCourant);
    ancetre = noeudCourant; noeudCourant = new DefaultMutableTreeNode("Florence"); ancetre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("David"); ancetre.add(noeudCourant);
    noeudCourant = new DefaultMutableTreeNode("Louis"); ancetre.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*

JTable : exemple

JTable : ajout dans un conteneur

- Pour un JScrollPane

```
JScrollPane scrollPane = new JScrollPane(table);
table.setFillsViewportHeight(true);
```

- Pour un autre conteneur

```
container.setLayout(new BorderLayout());
container.add(table.getTableHeader(), BorderLayout.PAGE_START);
container.add(table, BorderLayout.CENTER);
```

- Ainsi, les titres de colonne resteront toujours visible

JTable : cas de base

- Contraintes d'utilisation du cas de base
  - Tout est éditable sous la forme de chaîne de caractère
- Constructeurs :
  - JTable(Object[][] données, Object[] nomsColonnes)
  - JTable(Vector données, Vector nomsSolonnes)

    Données est un vecteur de vecteurs d'objets

JTable : exemple du cas de base

```
import java.util.Vector;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;

public class Base extends JFrame {
    public Base() {
        super("table de base");
        // Intitulés des colonnes
        Vector<String> colonnes = new Vector<String>();
        colonnes.add("Nom"); colonnes.add("Prénom"); colonnes.add("Age");
        // Contenu de la table
        Vector<Vector<String>> donnees = new Vector<Vector<String>>();
        Vector<String> donnee1 = new Vector<String>();
        donnee1.add("fournier"); donnee1.add("domique"); donnee1.add("35"); donnees.add(donnee1);
        Vector<String> donnee2 = new Vector<String>();
        donnee2.add("amanton");donnee2.add("laurent"); donnee2.add("20"); donnees.add(donnee2);
        // Création de la table
        JTable table = new JTable(donnees, colonnes);
        JScrollPane panneau; panneau = new JScrollPane(table);
        table.setFillsViewportHeight(true);
        add(panneau);
        setDefaultCloseOperation(EXIT_ON_CLOSE); pack(); setVisible(true);
    }
    public static void main(String [] args) {
        Base fenetre = new Base();
    }}
```

JTable : modèles sous-jacents

- Modèle des colonnes (TableColumnModel)
  - Interface TableColumnModel
  - Implantation de base : DefaultTableColumnModel
- Modèle des données (TableModel)
  - Interface TableModel
  - Implantation de base : DefaultTableModel (sous-classe de AbstractTableModel)
- Modèle de sélection (ListSelectionModel)

JTable : interface TableColumnModel

- Modification des colonnes
  - void addColumn(TableColumn col)
  - void removeColumn(TableColumn col)
  - void moveColumn(int columnIndex, int newIndex)
  - void setColumnMargin(int newMargin)
- Infos sur les colonnes
  - int getColumnMargin()
  - Enumeration<TableColumn> getColumns()
  - int getColumnCount()
  - int getColumnIndex(Object columnIdentifier)
  - int getColumnIndexAtX(int xPosition)
  - int getTotalColumnWidth()
  - TableColumn getColumn(int columnIndex)
- Sélection
  - boolean getColumnSelectionAllowed()
  - int getSelectedColumnCount()
  - int[] getSelectedColumns()
  - ListSelectionModel getSelectionModel()
  - void setColumnSelectionAllowed(boolean flag)
  - void setSelectionModel(ListSelectionModel newModel)
- Ecouteurs
  - void addColumnModelListener(TableColumnModelListener x)
  - void removeColumnModelListener(TableColumnModelListener x)

JTable : la classe TableColumn (1)

- Gère ce qui est propre à une colonne :
  - Sa largeur
  - Son intitulé
  - Les afficheurs (*renderer*) et éditeurs (*editor*) de ses données
- Constructeurs
  - TableColumn()
  - TableColumn(int modelIndex)

    m*odelIndex* : indice du champ des données associé à la colonne (quelle que soit la position de la colonne affichée)
  - TableColumn(int modelIndex, int width)
  - TableColumn(int modelIndex, int width, TableCellRenderer cellRenderer, TableCellEditor cellEditor)

JTable : la classe TableColumn (2)

Méthodes

- Void addPropertyChangeListener(PropertyChangeListener listener)
- TableCellEditor getCellEditor()
- TableCellRenderer getCellRenderer()
- TableCellRenderer getHeaderRenderer()
- Object getHeaderValue() / void setHeaderValue(Object headerValue)
- Object getIdentifier() / void setIdentifier(Object identifier)
- int getMaxWidth() / void setMaxWidth(int maxWidth)
- int getMinWidth() / void setMinWidth(int minWidth)
- int getModelIndex() / void setModelIndex(int modelIndex)
- int getPreferredWidth() / void setPreferredWidth(int preferredWidth)
- PropertyChangeListener[] getPropertyChangeListeners()
- boolean getResizable()
- int getWidth() / void setResizable(boolean isResizable)
- void removePropertyChangeListener(PropertyChangeListener listener)
- void setCellEditor(TableCellEditor cellEditor)
- void setCellRenderer(TableCellRenderer cellRenderer)
- void setHeaderRenderer(TableCellRenderer headerRenderer)
- void sizeWidthToFit()

JTable : la classe TableColumnModelListener

- 5 méthodes
  - void columnAdded(TableColumnModelEvent e)
  - void columnMarginChanged(ChangeEvent e)
  - void columnMoved(TableColumnModelEvent e)
  - void columnRemoved(TableColumnModelEvent e)
  - void columnSelectionChanged(ListSelectionEvent e)

- La classe TableColumnModelEvent
  - void columnAdded(TableColumnModelEvent e)
  - void columnMarginChanged(ChangeEvent e)
  - void columnMoved(TableColumnModelEvent e)
  - void columnRemoved(TableColumnModelEvent e)
  - void columnSelectionChanged(ListSelectionEvent e)

JTable : modèle des données
L'interface TableModel

- void addTableModelListener(TableModelListener l)
- Class<?> getColumnClass(int columnIndex)
- int getColumnCount()
- String getColumnName(int columnIndex)
- int getRowCount()
- Object getValueAt(int rowIndex, int columnIndex)
- boolean isCellEditable(int rowIndex, int columnIndex)
- void removeTableModelListener(TableModelListener l)
- void setValueAt(Object aValue, int rowIndex, int columnIndex)

JTable : la classe DefaultTableModel (1)

- Constructeurs
  - DefaultTableModel() / DefaultTableModel(int rowCount, int columnCount)
  - DefaultTableModel(Object[][] data, Object[] columnNames)
  - DefaultTableModel(Object[] columnNames, int rowCount)
  - DefaultTableModel(Vector columnNames, int rowCount)
  - DefaultTableModel(Vector data, Vector columnNames)
- Méthodes de la classe AbstractTableModel
  - void addTableModelListener(TableModelListener l)
  - int findColumn(String columnName)
  - void fireTableCellUpdated(int row, int column)
  - void fireTableChanged(TableModelEvent e)
  - void fireTableDataChanged()
  - void fireTableRowsDeleted(int firstRow, int lastRow)
  - void fireTableRowsInserted(int firstRow, int lastRow)
  - void fireTableRowsUpdated(int firstRow, int lastRow)
  - void fireTableStructureChanged()
  - Class<?> getColumnClass(int columnIndex)
  - String getColumnName(int column)
  - <T extends EventListener> T[] getListeners(Class<T> listenerType)
  - TableModelListener[] getTableModelListeners()
  - void removeTableModelListener(TableModelListener l)
  - void setValueAt(Object aValue, int rowIndex, int columnIndex)

JTable : la classe DefaultTableModel (2)

- Méthodes
  - void addColumn(Object columnName) / void addColumn(Object columnName, Object[] columnData) / void addColumn(Object columnName, Vector columnData)
  - void addRow(Object[] rowData) / void addRow(Vector rowData)
  - void insertRow(int row, Object[] rowData) / void insertRow(int row, Vector rowData)
  - void removeRow(int row)
  - void moveRow(int start, int end, int to)
  - int getColumnCount() / void setColumnCount(int columnCount)
  - int getRowCount() / void setNumRows(int rowCount) / void setRowCount(int rowCount)
  - String getColumnName(int column)
  - Vector getDataVector() / void setDataVector(Object[][] dataVector, Object[] columnIdentifiers) / void setDataVector(Vector dataVector, Vector columnIdentifiers)
  - Object getValueAt(int row, int column) / void setValueAt(Object aValue, int row, int column)
  - boolean isCellEditable(int row, int column)
  - void newDataAvailable(TableModelEvent event) / void newRowsAdded(TableModelEvent e) / void rowsRemoved(TableModelEvent event)
  - void setColumnIdentifiers(Object[] newIdentifiers) / void setColumnIdentifiers(Vector columnIdentifiers)

JTable : la classe TableModelListener

- 1 méthode

  void tableChanged(TableModelEvent e)

- La classe TableModelEvent
  - int getColumn()
  - int getFirstRow()
  - int getLastRow()
  - int getType()

    type d'événement : INSERT, UPDATE and DELETE.

La classe JTable : Constructeurs

- JTable()
- JTable(int numRows, int numColumns)
- JTable(Object[][] rowData, Object[] columnNames)
- JTable(TableModel dm)
- JTable(TableModel dm, TableColumnModel cm)
- JTable(TableModel dm, TableColumnModel cm, ListSelectionModel sm)
- JTable(Vector rowData, Vector columnNames)

Méthodes de JTable : données

- Modèle des données
  - TableModel getModel()
  - void setModel(TableModel dataModel)
- Données
  - Object getValueAt(int row, int column)
  - void setValueAt(Object aValue, int row, int column)
  - int getRowCount()
- Colonnes
  - void removeColumn(TableColumn aColumn)
  - void setAutoCreateColumnsFromModel(boolean autoCreateColumnsFromModel)
  - void moveColumn(int column, int targetColumn)
  - void addColumn(TableColumn aColumn)
  - void createDefaultColumnsFromModel()
  - boolean getAutoCreateColumnsFromModel()
  - TableColumn getColumn(Object identifier)
  - int getColumnCount()
  - String getColumnName(int column)
  - Class<?> getColumnClass(int column)
- Modèle des colonnes
  - TableColumnModel getColumnModel()
  - void setColumnModel(TableColumnModel columnModel)

Méthodes de JTable : mise en forme (1)

- Conversion des indices affichage/données
  - int convertColumnIndexToModel(int viewColumnIndex)
  - int convertColumnIndexToView(int modelColumnIndex)
  - int convertRowIndexToModel(int viewRowIndex)
  - int convertRowIndexToView(int modelRowIndex)

- Grille
  - Color getGridColor() / void setGridColor(Color gridColor)
  - boolean getShowHorizontalLines() / void setShowHorizontalLines(bolean b)
  - boolean getShowVerticalLines() / void setShowVerticalLines(boolean b)
  - void setShowGrid(boolean showGrid)

- Tailles variées
  - int getRowHeight() / void setRowHeight(int rowHeight)
  - int getRowHeight(int row) / void setRowHeight(int row, int rowHeight)
  - int getRowMargin() / void setRowMargin(int rowMargin)
  - Dimension getIntercellSpacing()
  - void setIntercellSpacing(Dimension intercellSpacing)

Méthodes de JTable : mise en forme (2)

- Défilement
  - int getScrollableBlockIncrement(Rectangle visibleRect, int orientation, int direction)
  - boolean getScrollableTracksViewportHeight()
  - boolean getScrollableTracksViewportWidth()
  - Dimension getPreferredScrollableViewportSize()
  - void setPreferredScrollableViewportSize(Dimension size)
  - int getScrollableUnitIncrement(Rectangle visibleRect, int orientation, int direction)
- UI
  - TableUI getUI() / void setUI(TableUI ui) / void updateUI() / String getUIClassID()
- En-tête
  - JTableHeader getTableHeader() / void setTableHeader(JTableHeader tableHeader)
- Autre
  - int getAutoResizeMode() / void setAutoResizeMode(int mode)
  - boolean getFillsViewportHeight() / void setFillsViewportHeight(boolean b)
  - boolean getSurrendersFocusOnKeystroke()
  - void setSurrendersFocusOnKeystroke(boolean b)
  - String getToolTipText(MouseEvent event)

Méthodes de JTable : divers

- Edition
  - boolean editCellAt(int row, int column)
  - boolean editCellAt(int row, int column, EventObject e)
  - int getEditingColumn() / void setEditingColumn(int aColumn)
  - int getEditingRow() / void setEditingRow(int aRow)
  - boolean isCellEditable(int row, int column)
  - boolean isEditing()

- Tri
  - boolean getAutoCreateRowSorter()
  - RowSorter<? extends TableModel> getRowSorter()
  - void setAutoCreateRowSorter(boolean autoCreateRowSorter)
  - void setRowSorter(RowSorter<? extends TableModel> sorter)

- Coordonnées
  - Rectangle getCellRect(int row, int column, boolean includeSpacing)
  - int columnAtPoint(Point point) / int rowAtPoint(Point point)

Méthodes de JTable : éditeurs/afficheurs

- Editeurs
  - TableCellEditor getCellEditor()
  - TableCellEditor getCellEditor(int row, int column)
  - TableCellEditor getDefaultEditor(Class<?> columnClass)
  - Component prepareEditor(TableCellEditor editor, int row, int column)
  - void setDefaultEditor(Class<?> columnClass, TableCellEditor editor)
  - void setCellEditor(TableCellEditor anEditor)
  - void removeEditor()
  - Component getEditorComponent()

- Afficheurs
  - void createDefaultRenderers()
  - TableCellRenderer getCellRenderer(int row, int column)
  - TableCellRenderer getDefaultRenderer(Class<?> columnClass)
  - Component prepareRenderer(TableCellRenderer renderer, int row, int column)
  - void setDefaultRenderer(Class<?> columnClass, TableCellRenderer renderer)

Méthodes de JTable : sélection (1)

- Général
  - void clearSelection() / void selectAll()
  - boolean getCellSelectionEnabled() / void setCellSelectionEnabled(boolean b)
  - Color getSelectionBackground() / void setSelectionBackground(Color col)
  - Color getSelectionForeground() / void setSelectionForeground(Color col)
  - boolean isCellSelected(int row, int column)
  - void setSelectionMode(int selectionMode)

- Colonnes
  - void addColumnSelectionInterval(int index0, int index1)
  - boolean getColumnSelectionAllowed() / void setColumnSelectionAllowed(boolean b)
  - int getSelectedColumn() / int[] getSelectedColumns()
  - boolean isColumnSelected(int col)
  - int getSelectedColumnCount()
  - void removeColumnSelectionInterval(int index0, int index1)
  - void setColumnSelectionInterval(int index0, int index1)

Méthodes de JTable : sélection (2)

- Lignes
  - void addRowSelectionInterval(int index0, int index1)
  - void removeRowSelectionInterval(int index0, int index1)
  - void setRowSelectionInterval(int index0, int index1)
  - void changeSelection(int rowIndex, int columnIndex, boolean toggle, boolean extend)
  - int getSelectedRow() / int[] getSelectedRows()
  - boolean isRowSelected(int row)
  - int getSelectedRowCount()
  - ListSelectionModel getSelectionModel() /void setSelectionModel(ListSelectionModel M)
  - boolean getRowSelectionAllowed() / void setRowSelectionAllowed(boolean b)
  - void setUpdateSelectionOnSort(boolean b) / boolean getUpdateSelectionOnSort()

Méthodes de JTable : événements

- Interfaces notamment implantées par JTable :
  - TableModelListener
  - TableColumnModelListener
  - ListSelectionListener
  - CellEditorListener
  - RowSorterListener

- Méthodes associées :
  - void columnAdded(TableColumnModelEvent e)
  - void columnMarginChanged(ChangeEvent e)
  - void columnMoved(TableColumnModelEvent e)
  - void columnRemoved(TableColumnModelEvent e)
  - void columnSelectionChanged(ListSelectionEvent e)
  - void editingCanceled(ChangeEvent e)
  - void editingStopped(ChangeEvent e)
  - void tableChanged(TableModelEvent e)
  - void valueChanged(ListSelectionEvent e)
  - void sorterChanged(RowSorterEvent e)

Tri de lignes