JTREE: JAVA SWING COMPONENT

SUKHMEET KHALAR

UMGC - SWEN 661

MAY 20TH, 2025

INTRODUCTION TO JTREE

- **JTree** is a Swing component for displaying data in a hierarchical, tree-like format.
- Commonly used in:
  - File explorers (e.g., Windows Explorer, IDE project views)
  - Organization charts
  - Category browsers

PURPOSE OF JTREE

- **Displays hierarchical data** where items can have sub-items (nodes and child nodes).
- Allows users to **expand or collapse nodes** to show or hide details.
- Supports **single or multiple selection** of nodes for further actions.
- Useful for navigating complex, nested structures.

COMMON PROPERTIES AND METHODS

KEY PROPERTIES

- **TreeModel:** The underlying data structure. Default is `DefaultTreeModel`.
- **Root Node:** The top-most node, often a `DefaultMutableTreeNode`.
- **Selection Model:** Controls how nodes are selected (single, contiguous, or discontiguous).
- **Editable:** Allows users to edit node names directly (`setEditable(true)`).

KEY METHODS

- `getSelectionPath()`/`getSelectionPaths()`: Returns the selected node(s) as TreePath objects.
- `setSelectionPath(TreePath path)`: Selects a specific node.
- `expandPath(TreePath path)`/ `collapsePath(TreePath path)`: Expands or collapses a node.
- `addTreeSelectionListener(TreeSelectionListener l)`: Registers a listener for selection changes.
- `setShowsRootHandles(boolean)`: Shows or hides expand/collapse handles for the root node.

BASIC JTREE EXAMPLE

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

public class SimpleJTreeExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("JTree Example");

        // Create the root node and child nodes
        DefaultMutableTreeNode root = new DefaultMutableTreeNo
        DefaultMutableTreeNode child1 = new DefaultMutableTree
        DefaultMutableTreeNode child2 = new DefaultMutableTree
        DefaultMutableTreeNode grandchild = new DefaultMutable
        child1.add(grandchild);
        root.add(child1);
```
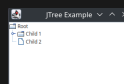
LISTENING FOR SELECTION EVENTS

```java
tree.addTreeSelectionListener(e -> {
    DefaultMutableTreeNode selectedNode =
        (DefaultMutableTreeNode) tree.getLastSelectedPathCompo
    if (selectedNode != null) {
        System.out.println("Selected: " + selectedNode.getUser
    }
});
```

```
/home/skhalar/.jdks/openjdk-24.0.1/bin/java -javaagent:/storage/JetBrains/apps
Selected: Child 1
```

JTree Example ∨ ∧ ×

```
Root
  Child 1
  Child 2
```
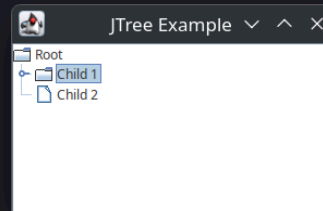
EXPANDING AND COLLAPSING NODES PROGRAMMATICALLY

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

public class SimpleJTreeExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("JTree Example");

        // Create the root node and child nodes
        DefaultMutableTreeNode root = new DefaultMutableTre
        DefaultMutableTreeNode child1 = new DefaultMutableT
        DefaultMutableTreeNode child2 = new DefaultMutableT
        DefaultMutableTreeNode grandchild = new DefaultMuta
        child1.add(grandchild);
        root.add(child1);
```

CUSTOMIZING NODE ICONS

```java
DefaultTreeCellRenderer renderer = (DefaultTreeCellRenderer) t
final Icon openIcon = UIManager.getIcon("OptionPane.questionIc
final Icon closeIcon = UIManager.getIcon("OptionPane.errorIcon
final Icon leafIcon = UIManager.getIcon("OptionPane.informatio
renderer.setLeafIcon(leafIcon);
renderer.setOpenIcon(openIcon);
renderer.setClosedIcon(closeIcon);
```
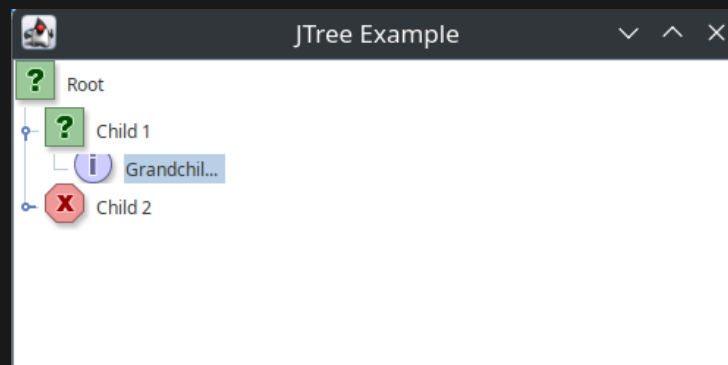
ALTERNATE USES FOR JTREE

- **File system browsers:** Display folders and files.
- **Organization charts:** Show company hierarchy.
- **Menu navigation:** Represent nested menus or categories.
- **Custom data:** Any nested data structure (e.g., XML, JSON).

SUMMARY

- **JTree** is ideal for displaying and interacting with hierarchical data.
- Focus on:
  - **Nodes** (`DefaultMutableTreeNode`)
  - **TreeModel**
  - **Selection and event handling**
  - **Expand/collapse functionality**
- **Customizable:** Change icons, respond to events, and display any nested data.

QUESTIONS?