Lecture 02
Creating Graphical User Interface
Part 4

How to use Trees

Why should we study this lecture?

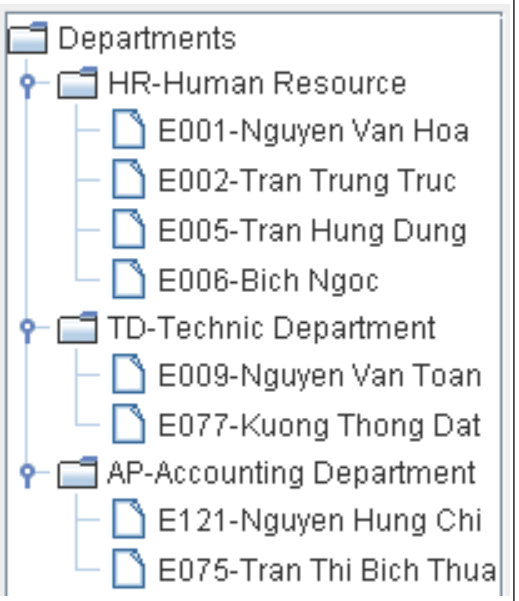
- How to present hierarchical data on a GUI?
- How to manage events on a tree?

Contents

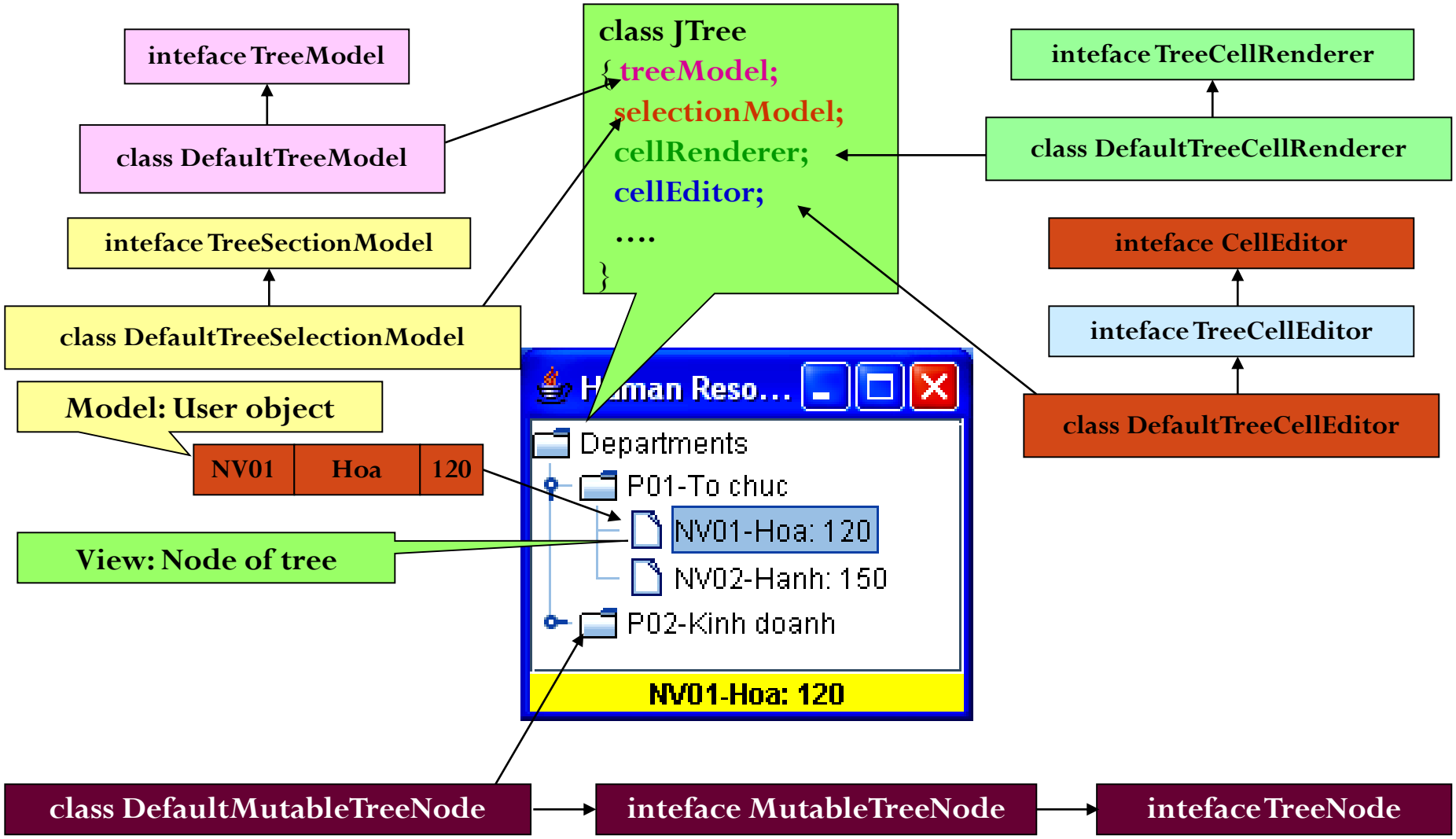
- The javax.swing.JTree class
- A Demonstration

1- The JTree Class

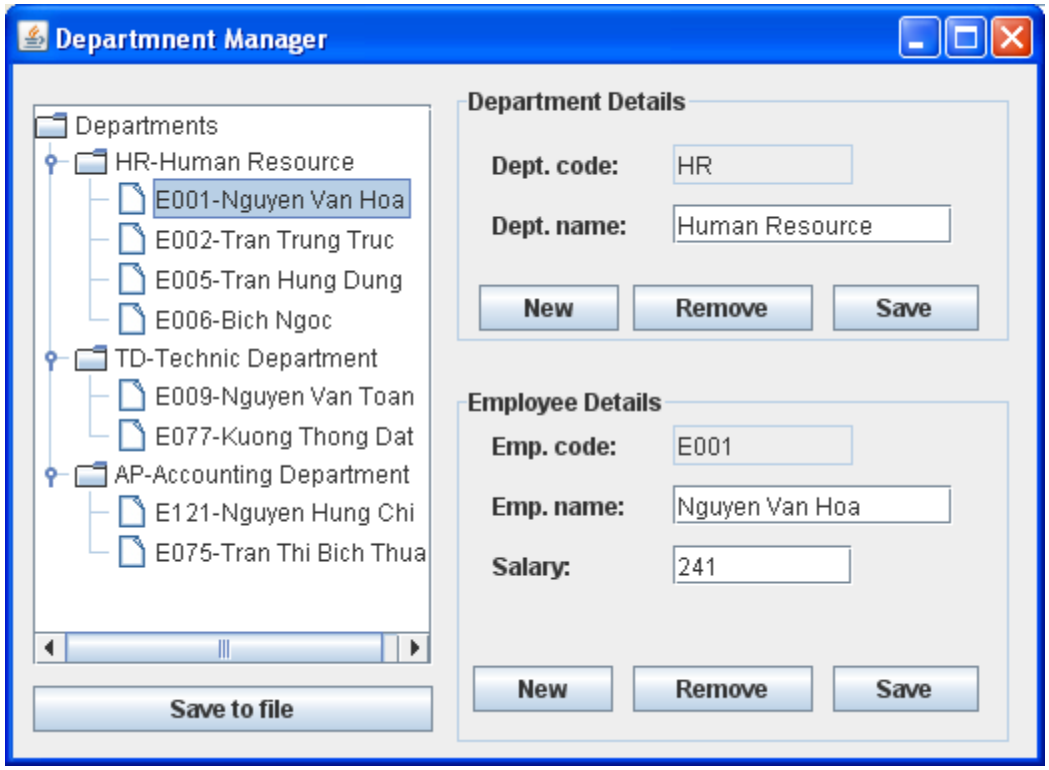
- A view for hierarchical data.
- Data in a tree are presented as an aligned list. So, we can identify a node when clicking the mouse.
- The model of JTree contains a group of nodes (the DefaultMutableTreeNode class) in which the root is the upmost node. Each node can contain its individual data.
- At a time, a branch of the tree is selected (TreePath). The JTree.getSelectionPath() will return the current selected path and the TreePath.getLastPathComponent() will return the last node in the path.

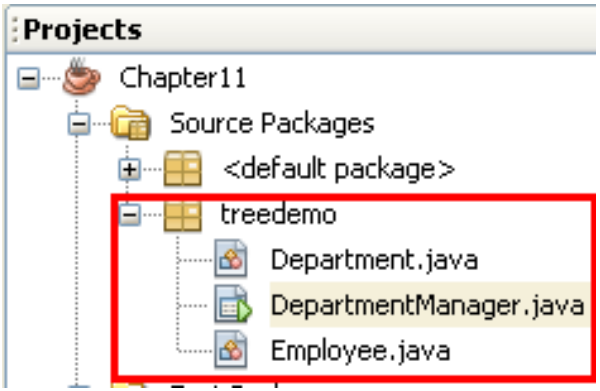


JTree: Architecture

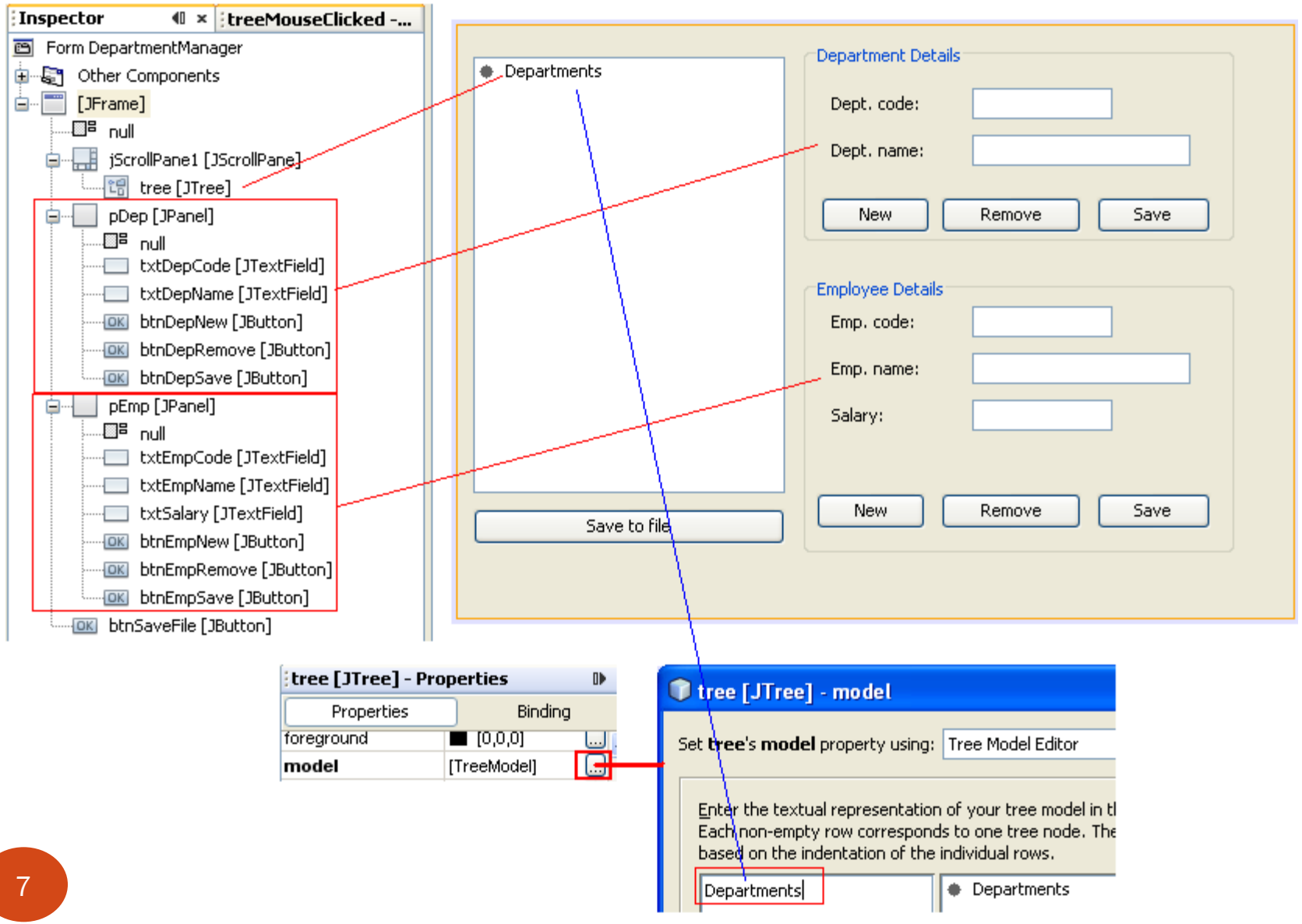


Demo 9: Using JTree





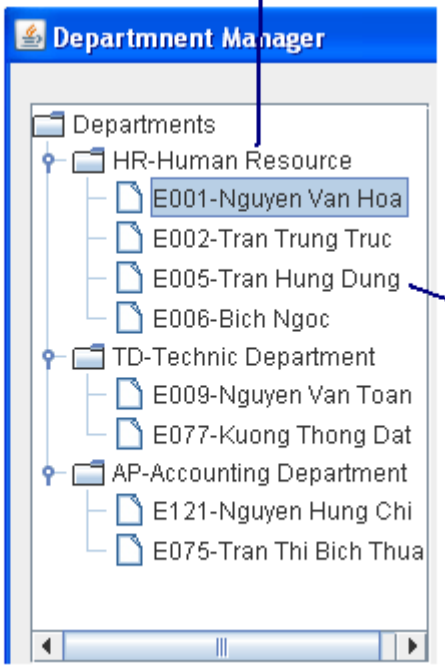
Demo 9: Using JTree…



Demo 9: Using JTree…

```
/* Class for a department */
package treedemo;
public class Department {
    String depCode, depName;

    public Department(String depCode, String depName) {...}
    public String getDepCode() {...}
    public void setDepCode(String depCode) {...}
    public String getDepName() {...}
    public void setDepName(String depName) {...}
    public String toString(){
        return depCode + "-" + depName;
    }
}
```




```
/* Class for an employee */
package treedemo;
public class Employee {
    String empCode, empName; int salary;
    public Employee(String empCode, String empName, int salary){
        this.empCode = empCode;
        this.empName = empName;
        this.salary = salary;
    }
    public String getEmpCode() {...}
    public void setEmpCode(String empCode) {...}
    public String getEmpName() {...}
    public void setEmpName(String empName) {...}
    public int getSalary() {...}
    public void setSalary(int salary) {...}
    public String toString() {
        return empCode + "-" + empName;
    }
}
```

Demo 9: Using JTree…

DepartmentManager.java *

Source | Design

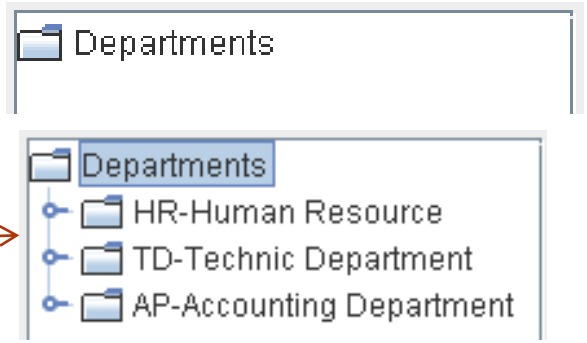
```
13  import javax.swing.tree.DefaultMutableTreeNode; // a node on a tree
14  import javax.swing.tree.TreePath; // get a path from the root of the tree
15  import javax.swing.JOptionPane; // common dialog
16  import java.io.*;                // File processing
17  import java.util.StringTokenizer; // for string splitting
18  import java.util.Enumeration; // interface for enumeration
    import java.util.Iterator;  // for traversing an enumeration
20
21  public class DepartmentManager extends javax.swing.JFrame {
22      String filename= "employees2.txt";
23      DefaultMutableTreeNode root= null; // root of the tree
24      DefaultMutableTreeNode curDepNode= null; // current department
25      DefaultMutableTreeNode curEmpNode= null; // current employee
26      boolean addNewDep= false;
27      boolean addNewEmp= false;
```

employees2.txt - Notepad

File Edit Format View Help

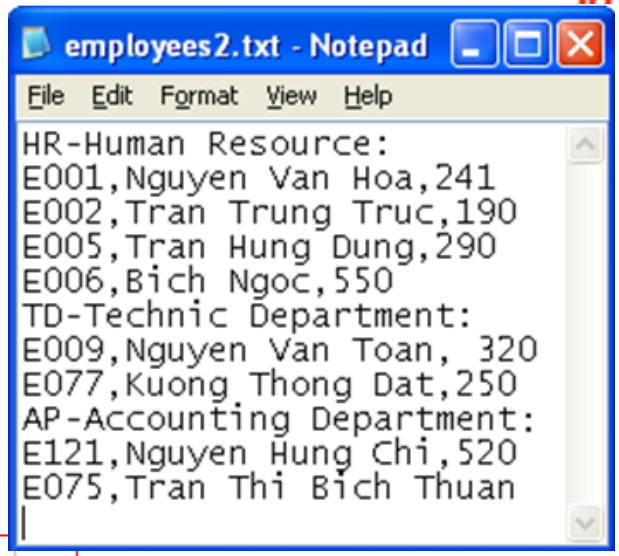
```
HR-Human Resource:
E001,Nguyen Van Hoa,241
E002,Tran Trung Truc,190
E005,Tran Hung Dung,290
E006,Bich Ngoc,550
TD-Technic Department:
E009,Nguyen Van Toan, 320
E077,Kuong Thong Dat,250
AP-Accounting Department:
E121,Nguyen Hung Chi,520
E075,Tran Thi Bich Thuan
```

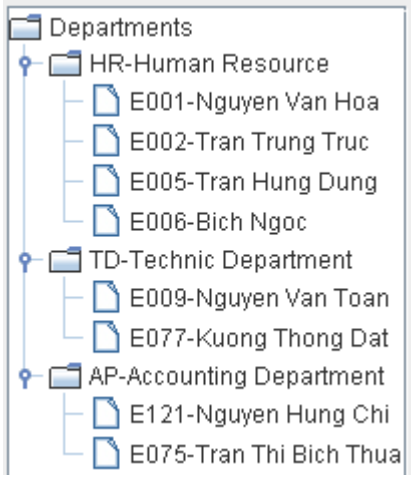
```
/** Creates new form DepartmentManager */
public DepartmentManager() {
    initComponents();
    this.setSize(520,380);
    root= (DefaultMutableTreeNode)(this.tree.getModel().getRoot());
    loadData(); // Loading initial data from file
    TreePath path=new TreePath(root);// expansing the tree
    tree.expandPath(path);
}
```



Demo 9: Using JTree…

```
// Load initial data from ehe employee2.txt file
private void loadData(){
    String S=""; StringTokenizer stk;
    try {
       FileReader f= new FileReader(filename);
       BufferedReader bf= new BufferedReader (f);
       while ((S=bf.readLine())!=null){
              S= S.trim();
              boolean isDept = (S.charAt(S.length()-1)==':');
              stk= new StringTokenizer(S, "-:,");
              String code= stk.nextToken().trim();
              String name= stk.nextToken().trim();
              if (isDept) {  // department details
                curDepNode=new DefaultMutableTreeNode(new Department(code,name));
                root.add(curDepNode); // add the node root
              }
              else { // employee details
                int salary= Integer.parseInt(stk.nextToken().trim());
                curEmpNode=new DefaultMutableTreeNode(new Employee(code,name,salary));
                curDepNode.add(curEmpNode); // add to the department
              }
       }
       bf.close(); f.close();
    }
    catch (Exception e){
       e.printStackTrace();
    }
}
```





Demo 9: Using JTree…

```
// Show details of current department and employee
private void viewDeptAndEmp() {
    Department curDep = null;
    Employee curEmp = null;
    if (curDepNode != null) {
        curDep = (Department) (curDepNode.getUserObject());
    }
    if (curEmpNode != null) {
        curEmp = (Employee) (curEmpNode.getUserObject());
    }
    this.txtDepCode.setText(curDep != null ? curDep.getDepCode() : "");
    this.txtDepName.setText(curDep != null ? curDep.getDepName() : "");
    this.txtEmpCode.setText(curEmp != null ? curEmp.getEmpCode() : "");
    this.txtEmpName.setText(curEmp != null ? curEmp.getEmpName() : "");
    this.txtSalary.setText("" + (curEmp != null ? curEmp.getSalary() : ""));
    this.txtDepCode.setEditable(false);
    this.txtEmpCode.setEditable(false);
}
```

Demo 9: Using JTree…

```
private void treeMouseClicked(java.awt.event.MouseEvent evt) {
    // turn of the on-tree ediiting mode
    tree.cancelEditing();
    // Get the clicked node at the last component of the path
    TreePath path= tree.getSelectionPath();
    if (path==null) return;
    DefaultMutableTreeNode selectedNode = null;
    selectedNode= (DefaultMutableTreeNode)(path.getLastPathComponent());
    // Get the selected object in the model
    Object selectedObj =selectedNode.getUserObject();
    // Checking what is the selected object
    if (selectedNode==root)
      this.curDepNode= this.curEmpNode=null;
    else {
       if (selectedObj instanceof Department) {
           this.curDepNode= selectedNode;
           this.curEmpNode=null;
       }
       else if (selectedObj instanceof Employee){
         curEmpNode= selectedNode;
         curDepNode= (DefaultMutableTreeNode)(selectedNode.getParent());
       }
    }
    viewDeptAndEmp();
}
```

addNewDept=addNewEmp=false;

Demo 9: Using JTree…

```
private void btnDepNewActionPerformed(java.awt.event.ActionEvent evt) {
    // Make the GUI ready for a new department detals entered
    this.addNewDep=true;
    this.txtDepCode.setText("");
    this.txtDepName.setText("");
    this.txtEmpCode.setText("");
    this.txtEmpName.setText("");
    this.txtSalary.setText("");
    this.txtDepCode.setEditable(true);
    this.txtDepCode.requestFocus();
}
```

```
private void btnDepRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    // Removing a department
    if (this.curDepNode.getChildCount()>0) {
        String msg = "Remove all employees before deleting a department.";
        JOptionPane.showMessageDialog(this,msg);
    }
    else {
        int response= JOptionPane.showConfirmDialog(this,"Delete this department- OK?");
        if (response==JOptionPane.OK_OPTION)  {
            root.remove(this.curDepNode);
            tree.updateUI();
        }
```

curDepNode=curEmpNode=null;
(the department was removed)

Demo 9: Using JTree…

```
// checking details of the department is valid or not
private boolean validDepDetails(){
    // your code here
    return true;
}
```

```
private void btnDepSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // Save department details
    String code= this.txtDepCode.getText().trim().toUpperCase();
    txtDepCode.setText(code);
    String name= this.txtDepName.getText().trim();
    txtDepName.setText(name);
    if (! validDepDetails()) return;
    if (addNewDep==true){
        Department newDep= new Department(code, name);
        root.add(new DefaultMutableTreeNode (newDep));
    }
    else {
        ((Department)curDepNode.getUserObject()).setDepName(name);
    }
    this.tree.updateUI();
    this.addNewDep=false;
}
```

curDepNode=curEmpNode=null;

Demo 9: Using JTree…

```
private void btnEmpNewActionPerformed(java.awt.event.ActionEvent evt) {
    // Make the GUI ready for a new employee detals entered
    this.addNewEmp=true;
    this.txtEmpCode.setText("");
    this.txtEmpName.setText("");
    this.txtSalary.setText("");
    this.txtEmpCode.setEditable(true);
    this.txtEmpCode.requestFocus();
}
```

```
private void btnEmpRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    // Removing an employee
    if (this.curEmpNode !=null) {
        int response= JOptionPane.showConfirmDialog(this,"Delete this employee- OK?");
        if (response==JOptionPane.OK_OPTION)  {
            curDepNode.remove(this.curEmpNode);
            tree.updateUI();
        }
    }
}
```

curEmpNode=null;

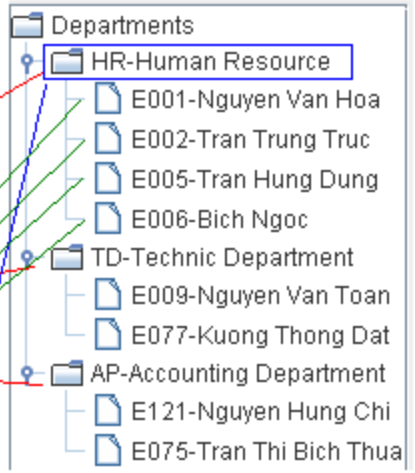
Demo 9: Using JTree…


```
private void btnEmpSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // Save/ update new employee/ employee details
    String code= this.txtEmpCode.getText().trim().toUpperCase();
    txtEmpCode.setText(code);
    String name= this.txtEmpName.getText().trim();
    txtEmpName.setText(name);
    String salaryStr= this.txtSalary.getText().trim();
    txtSalary.setText(salaryStr);
    int sal= Integer.parseInt(salaryStr);
    if (! validEmpDetails()) return;
    if (addNewEmp==true){
        Employee newEmp= new Employee(code, name, sal);
        curDepNode.add(new DefaultMutableTreeNode (newEmp));
    }
    else {
        Employee emp=(Employee)(curEmpNode.getUserObject());
        emp.setEmpName(name);
        emp.setSalary(sal);
    }
    this.tree.updateUI();
    this.addNewEmp=false;
}
```

```
// checking details of the employee is valid or not
private boolean validEmpDetails(){
    // your code here
    return true;
}
```

Demo 9: Using JTree…

```
private void btnSaveFileActionPerformed(java.awt.event.ActionEvent evt) {
    // Saving details to the file
    if (root.getChildCount()==0) return;
    String S;
    try {
        FileWriter f= new FileWriter (filename);
        PrintWriter pf= new PrintWriter(f);
        Enumeration depts= root.children();// get departments
        while (depts.hasMoreElements()) {
           DefaultMutableTreeNode depNode= (DefaultMutableTreeNode)depts.nextElement();
           Department dept= (Department)(depNode.getUserObject());
           S = dept.getDepCode() + "-" + dept.getDepName() + ":" ;
           pf.println(S);
           Enumeration emps= depNode.children(); // get employees
           while (emps.hasMoreElements()){
              DefaultMutableTreeNode empNode= (DefaultMutableTreeNode)emps.nextElement();
              Employee emp= (Employee)(empNode.getUserObject());
              S = emp.getEmpCode() + "," + emp.getEmpName() + "," + emp.getSalary();
              pf.println(S);
           }
         }
        pf.close();f.close();
        JOptionPane.showMessageDialog(this, "Data are saved to the file " + filename);
    }
    catch (Exception e){
       JOptionPane.showMessageDialog(this, e);
}
```



Summary

- The javax.swing.JTree class
- A Demonstration

Thank You