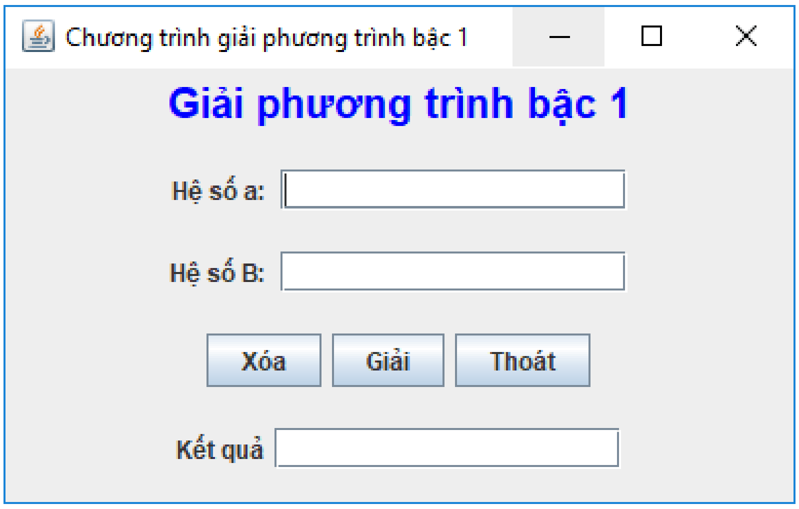
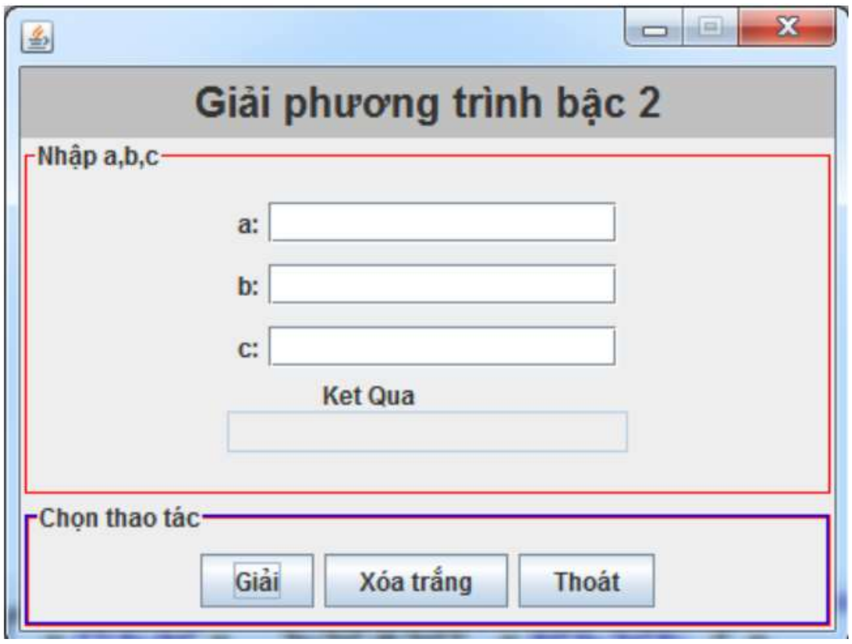
Phần 2. Xây dựng giao diện đồ họa với Swing

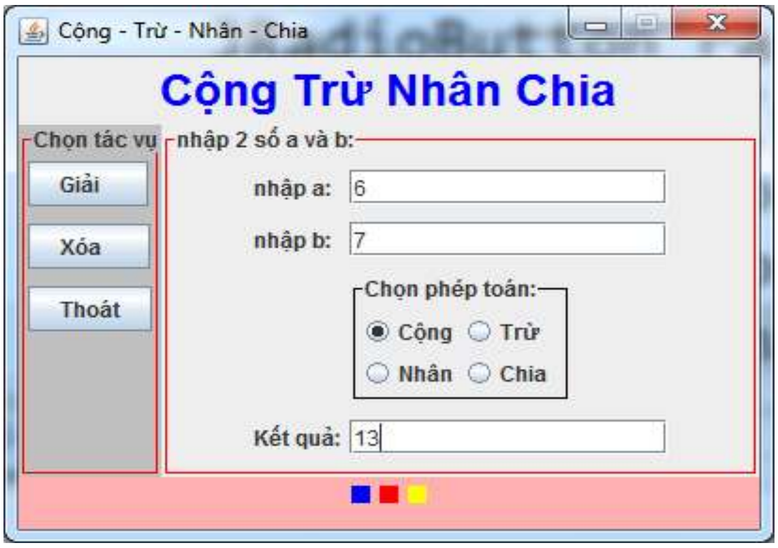
1. Xây dựng ứng dụng giải phương trình bậc 1:



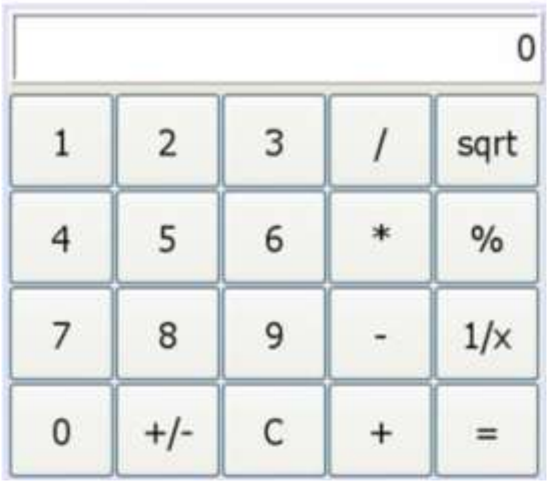
2. Xây dựng ứng dụng giải phương trình bậc 2



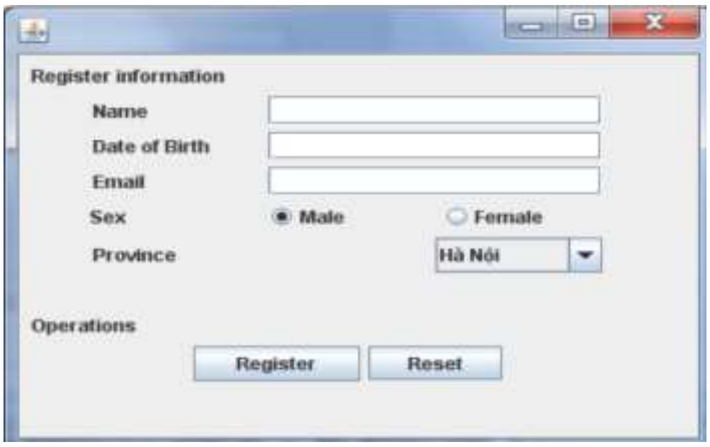
3. Xây dựng ứng dụng tính toán cộng, trừ, nhân, chia như sau:



4. Xây dựng máy tính đơn giản:



5. Thiết kế Frame theo mẫu sau:

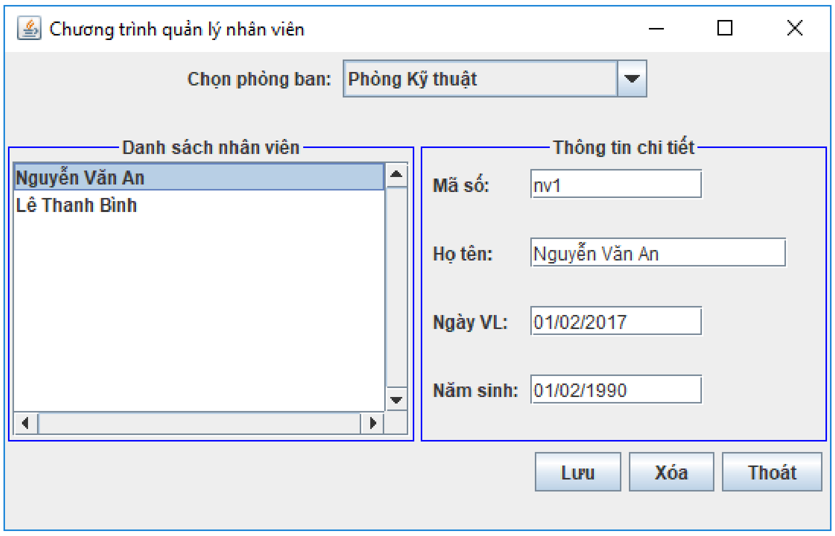


Viết code xử lý theo yêu cầu:

– Click nút **Register** thì các thông tin người dùng nhập trên Frame sẽ được ghi vào file "dulieu.txt" nếu thông tin đầy đủ và hợp lệ.

– Click nút **Reset** thì các thông tin người dùng nhập sẽ bị xóa bỏ.

6. Xây dựng ứng dụng quản lý nhân viên


Chương trình quản lý nhân viên

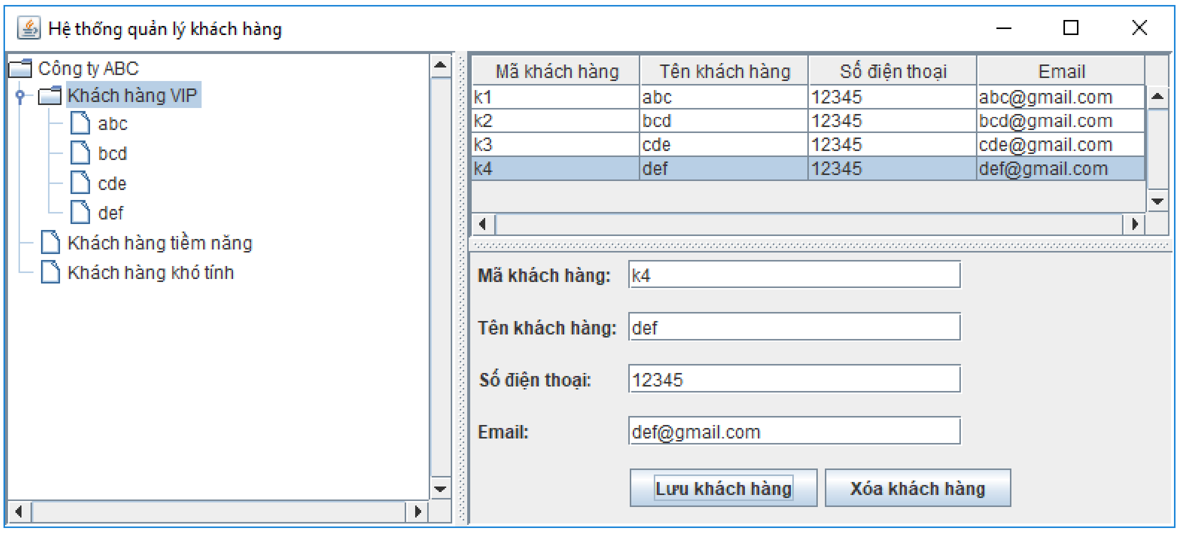
Chọn phòng ban: Phòng Kỹ thuật

Danh sách nhân viên
Nguyễn Văn An
Lê Thanh Bình

Thông tin chi tiết
Mã số: nv1
Họ tên: Nguyễn Văn An
Ngày VL: 01/02/2017
Năm sinh: 01/02/1990

Lưu ý: Thông tin phòng ban gồm Mã phòng, tên phòng; nhân viên gồm mã số, họ tên, ngày vào làm, ngày sinh.

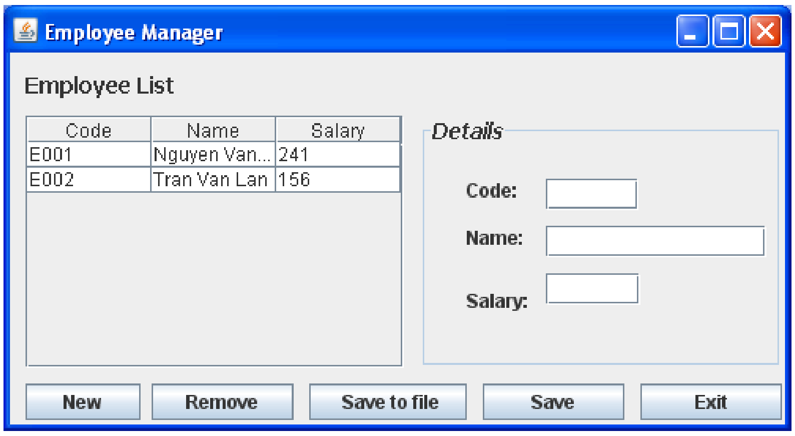
7. Xây dựng ứng dụng quản lý khách hàng



Mã khách hàng	Tên khách hàng	Số điện thoại	Email
k1	abc	12345	abc@gmail.com
k2	bcd	12345	bcd@gmail.com
k3	cde	12345	cde@gmail.com
k4	def	12345	def@gmail.com

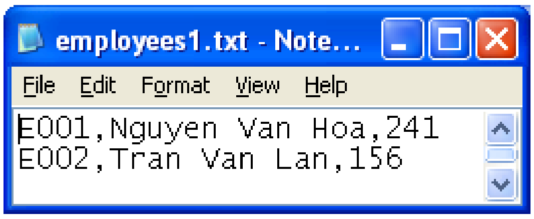
Mã khách hàng:	k4
Tên khách hàng:	def
Số điện thoại:	12345
Email:	def@gmail.com

8. Xây dựng ứng dụng quản lý nhân viên như sau:



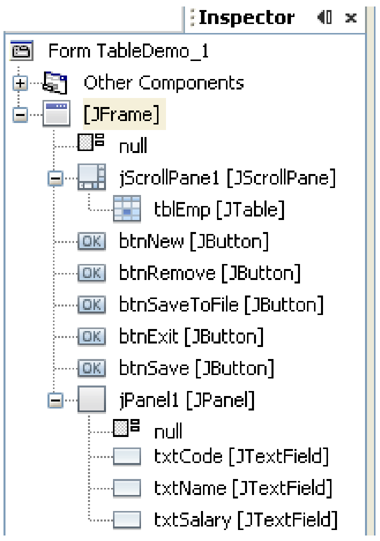
Code	Name	Salary
E001	Nguyen Van...	241
E002	Tran Van Lan	156

Cấu trúc file text lưu thông tin Employee:

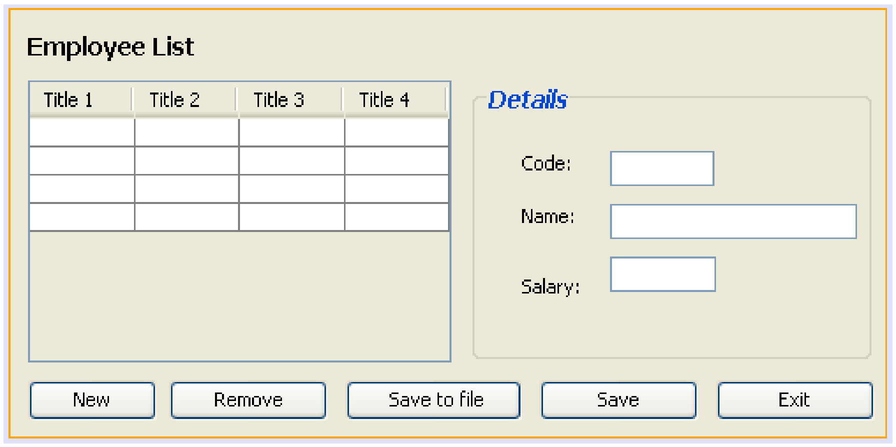


```
E001,Nguyen Van Hoa,241
E002,Tran Van Lan,156
```

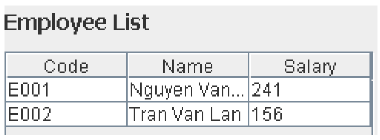
Hướng dẫn:



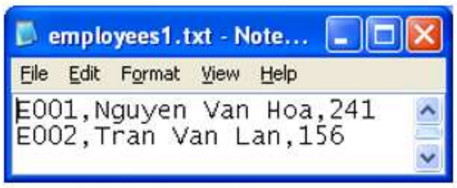
```
Form TableDemo_1
  Other Components
  [JFrame]
    null
    jScrollPane1 [JScrollPane]
      tblEmp [JTable]
    btnNew [JButton]
    btnRemove [JButton]
    btnSaveToFile [JButton]
    btnExit [JButton]
    btnSave [JButton]
    jPanel1 [JPanel]
      null
      txtCode [JTextField]
      txtName [JTextField]
      txtSalary [JTextField]
```



```java
import java.util.Vector; // header and data of the table
import java.util.StringTokenizer; // for splitting string
import java.io.File;
import java.io.FileReader;    // reading data from text file
import java.io.BufferedReader;
import java.io.PrintWriter;    // write data to text file
import javax.swing.table.DefaultTableModel; // control table
import javax.swing.JOptionPane;  // for build-in dialog
public class TableDemo_1 extends javax.swing.JFrame {
    String filename="employees1.txt";
    Vector<String> header= new Vector<String>();
    Vector data= new Vector();
    boolean addNew=false;
    boolean changed=false; // Are data changed?
    /** Creates new form TableDemo_1 */

private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (changed==true)
    { if (JOptionPane.showConfirmDialog(this,"Data changed. Save Y/N?")==
              JOptionPane.OK_OPTION)
      btnSaveToFileActionPerformed(null);
    }
    System.exit(0);
}

 /** Creates new form TableDemo_1 */
 public TableDemo_1() {
     initComponents();
     this.setSize(500,270);
     // setup header of the table
     header.add("Code"); header.add("Name");header.add("Salary");
     loadData();
     DefaultTableModel tblModel;
     tblModel= (DefaultTableModel)this.tblEmp.getModel();
     tblModel.setDataVector(data, header);
 }

private void loadData(){
    try
    {   File f= new File(filename);
        FileReader fr= new FileReader(f);
        BufferedReader bf = new BufferedReader(fr);
        String aDetails;
        while ((aDetails = bf.readLine())!=null){
            StringTokenizer stk= new StringTokenizer(aDetails, ",");
            String code=stk.nextToken();
            String name= stk.nextToken();
            String salaryStr= stk.nextToken();
            Vector<String> v= new Vector<String>();
            v.add(code); v.add(name); v.add(salaryStr);
            data.add(v);
        }
        bf.close(); fr.close();
    }
    catch (Exception e)
    {   JOptionPane.showMessageDialog(this, "The file " +
                filename + " not found." );
    }
}
```


Employee List

Code	Name	Salary
E001	Nguyen Van...	241
E002	Tran Van Lan	156



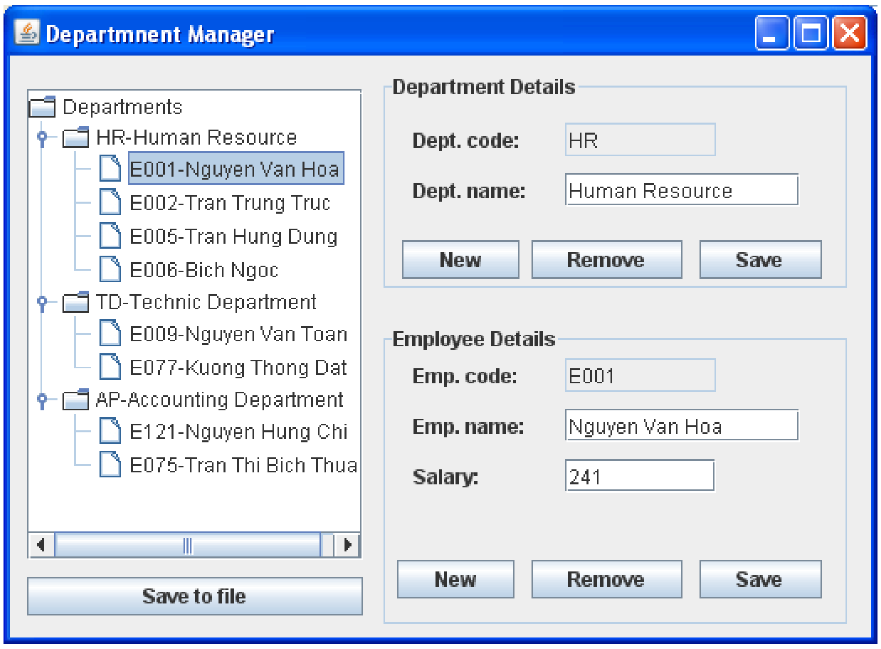
```
E001,Nguyen Van Hoa,241
E002,Tran Van Lan,156
```

```java
private void btnNewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.addNew =true;    // entering the add-new mode
    this.txtCode.setText(""); // preparing input data
    this.txtCode.setEditable(true);
    this.txtName.setText("");
    this.txtSalary.setText("");
    this.txtCode.requestFocus();
}

private void btnSaveToFileActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        File f= new File(filename);
        PrintWriter pf= new PrintWriter(f);
        int n= this.tblEmp.getRowCount();
        for (int i=0;i<n; i++){
          Vector<String> v = (Vector<String>)(data.get(i));
          String S= v.get(0) + "," + v.get(1) + "," + v.get(2);
          pf.println(S);
        }
        pf.close();
        changed=false;
    }
    catch (Exception e){
        JOptionPane.showMessageDialog(this, e);
    }
}

private void btnRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    // Get position of seclected employee
    int pos = this.tblEmp.getSelectedRow();
    if (pos>=0)
    {  String code= (String)(tblEmp.getValueAt(pos,0));
       if (JOptionPane.showConfirmDialog(this,"Delete the " +
                 code + " employee ?")== JOptionPane.OK_OPTION)
       { data.remove(pos);
         tblEmp.updateUI();
         changed=true;
       }
    }
}

private void tblEmpMouseReleased(java.awt.event.MouseEvent evt) {
    // Turn off the on-table editting mode when user double-clicks on a cell
    if (this.tblEmp.isEditing())
    { int row= tblEmp.getSelectedRow();
      int column= tblEmp.getSelectedColumn();
      tblEmp.getCellEditor(row,column).cancelCellEditing();
    }
}
```

```
private void tblEmpMouseClicked(java.awt.event.MouseEvent evt) {
    // View current employee
    int row= this.tblEmp.getSelectedRow();
    this.txtCode.setText((String)(tblEmp.getValueAt(row,0)));
    this.txtCode.setEditable(false);// user can not update the code
    this.txtName.setText((String)(tblEmp.getValueAt(row,1)));
    this.txtSalary.setText((String)(tblEmp.getValueAt(row,2)));
    addNew = false; // view and update mode
}
```

```
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (!valid()) return;
    String code = txtCode.getText();
    String name = txtName.getText();
    String salaryStr = txtSalary.getText();
    if (addNew)
    { Vector v= new Vector();
      v.add(code); v.add(name); v.add(salaryStr);
      data.add(v);
      addNew=false;
    }
    else
    { int pos= tblEmp.getSelectedRow();
      Vector v= (Vector)data.get(pos);
      v.set(1, name);
      v.set(2, salaryStr);
    }
    tblEmp.updateUI();
    changed=true;
}
```

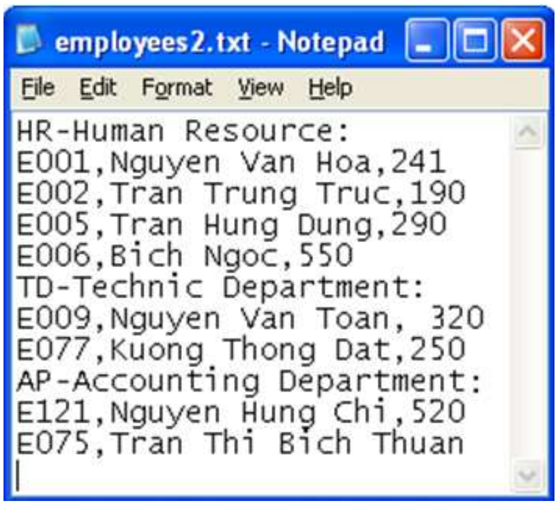
9. Xây dựng ứng dụng quản lý nhân viên như sau:



Dept. code: HR
Dept. name: Human Resource

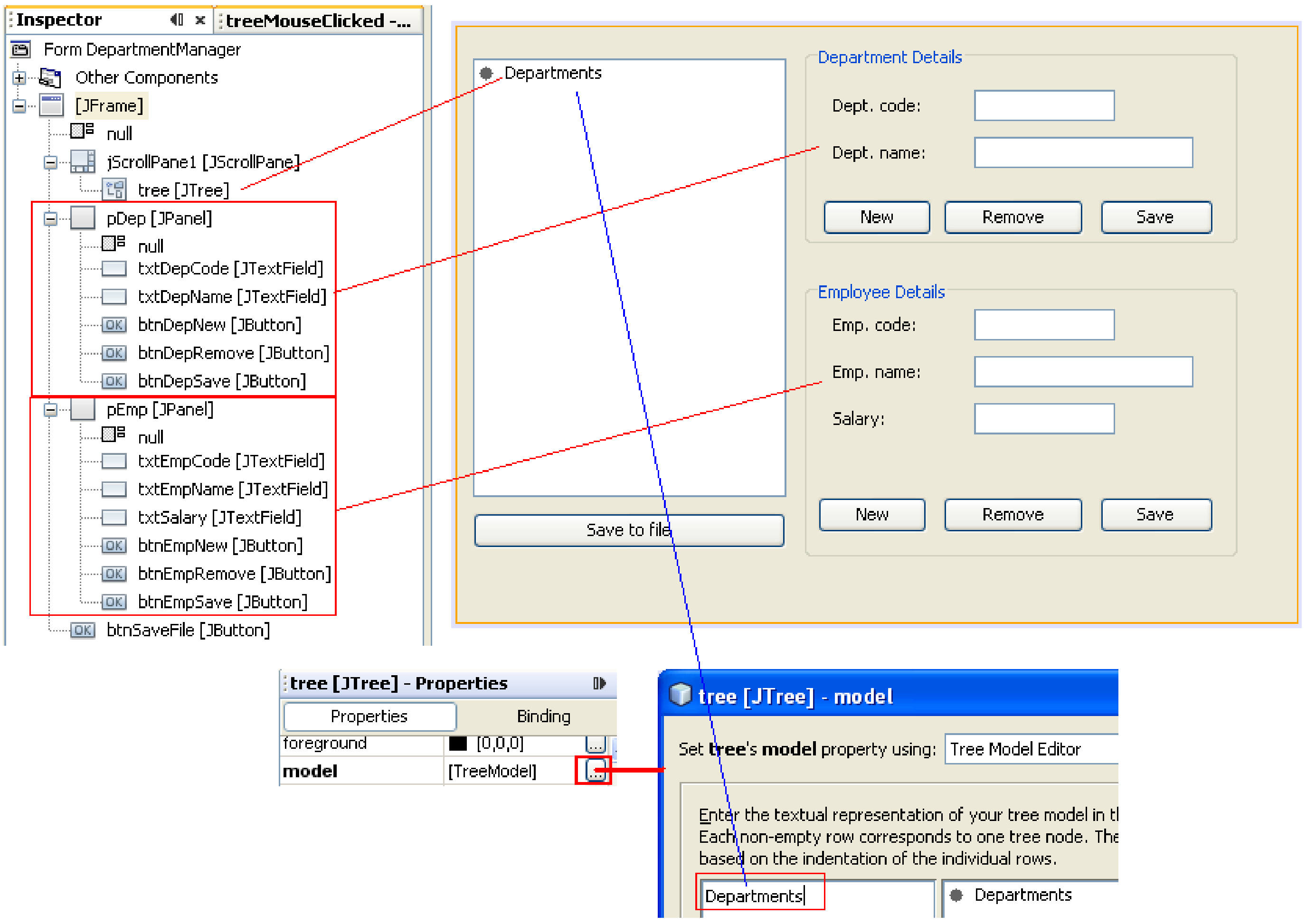
Emp. code: E001
Emp. name: Nguyen Van Hoa
Salary: 241

Cấu trúc file text lưu thông tin Employee:



```
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
```

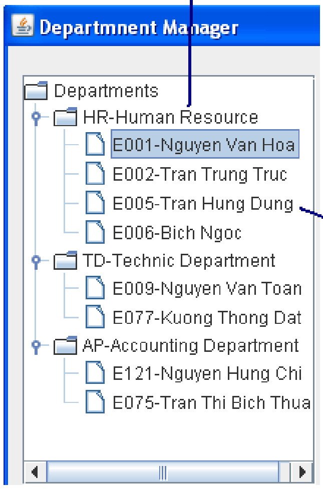
Hướng dẫn:

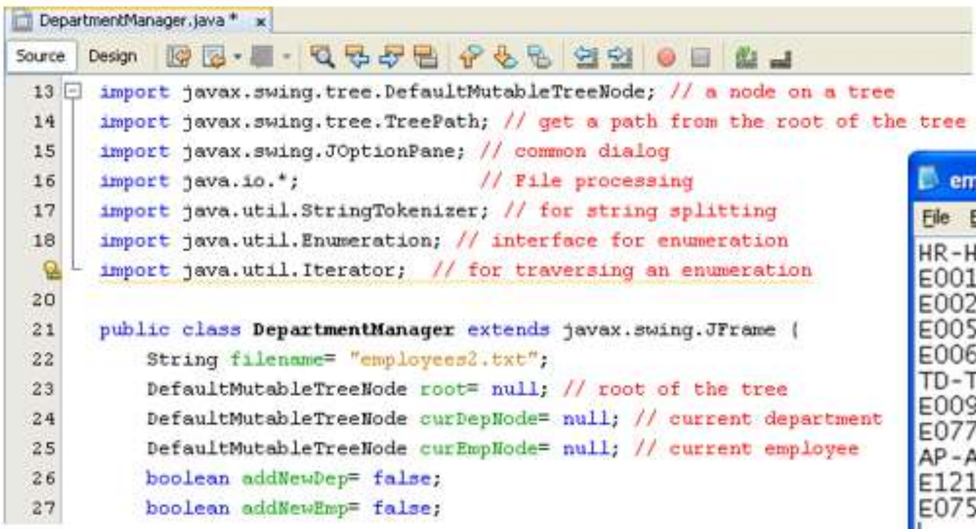


Enter the textual representation of your tree model in tl
Each non-empty row corresponds to one tree node. The
based on the indentation of the individual rows.

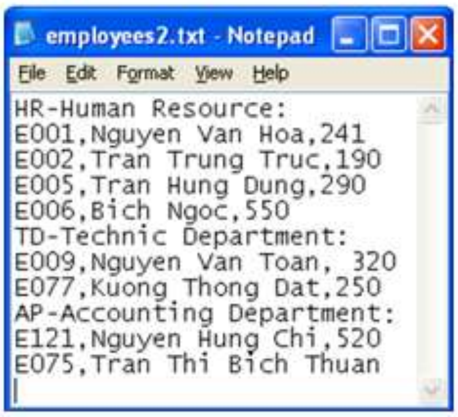
```
/* Class for a department */
package treedemo;
public class Department {
   String depCode, depName;

    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;
    }
}
```



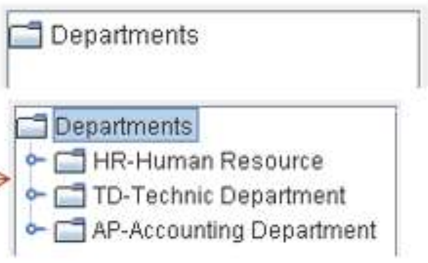


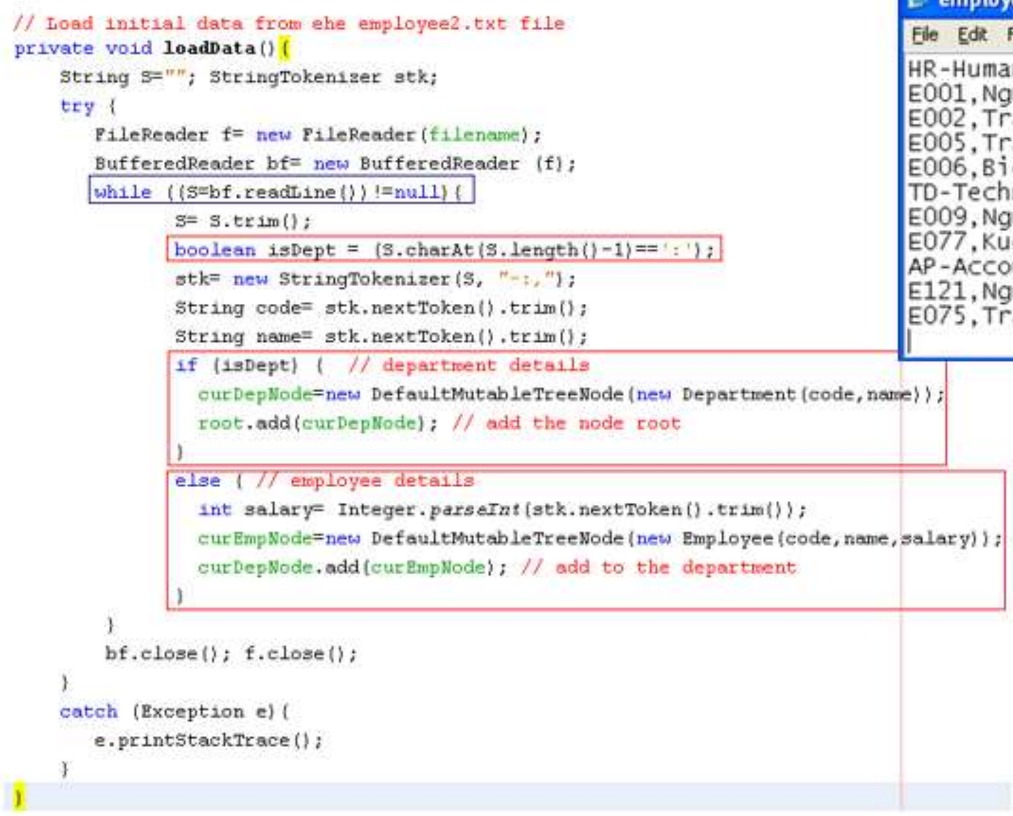
```
import javax.swing.tree.DefaultMutableTreeNode; // a node on a tree
import javax.swing.tree.TreePath; // get a path from the root of the tree
import javax.swing.JOptionPane; // common dialog
import java.io.*;               // File processing
import java.util.StringTokenizer; // for string splitting
import java.util.Enumeration; // interface for enumeration
import java.util.Iterator;  // for traversing an enumeration

public class DepartmentManager extends javax.swing.JFrame {
    String filename= "employees2.txt";
    DefaultMutableTreeNode root= null; // root of the tree
    DefaultMutableTreeNode curDepNode= null; // current department
    DefaultMutableTreeNode curEmpNode= null; // current employee
    boolean addNewDep= false;
    boolean addNewEmp= false;
```



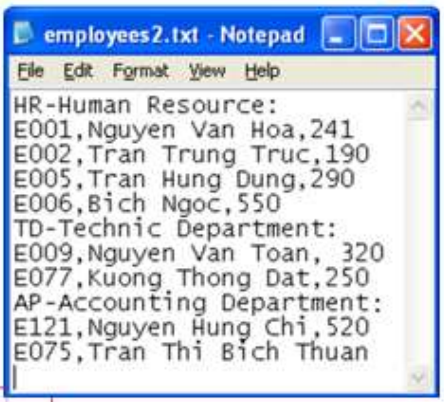
```
HR-Human Resource:
E001,Nguyen Van Hoa,241
E002,Tran Trung Truc,190
E005,Tran Hung Dung,290
E006,Bich Ngoc,550
TD-Technic Department:
E009,Nguyen Van Toan, 320
E077,Kuong Thong Dat,250
AP-Accounting Department:
E121,Nguyen Hung Chi,520
E075,Tran Thi Bich Thuan
```

```
/** 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);
}
```

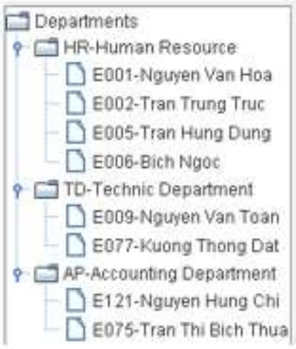




```java
// Load initial data from ehe employee2.txt file
private void loadData(){
    String S=""; StringTokenizer stk;
    try {
       FileReader f= new FileReader(filename);
       BufferedReader bf= new BufferedReader (f);
       while ((S=bf.readLine())!=null){
              S= S.trim();
              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();
    }
}
```



```
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
```



```
Departments
  HR-Human Resource
    E001-Nguyen Van Hoa
    E002-Tran Trung Truc
    E005-Tran Hung Dung
    E006-Bich Ngoc
  TD-Technic Department
    E009-Nguyen Van Toan
    E077-Kuong Thong Dat
  AP-Accounting Department
    E121-Nguyen Hung Chi
    E075-Tran Thi Bich Thua
```

```java
// 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);
}
```

```java
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();
}
```

```java
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();
}
```

```java
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)

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

```java
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;

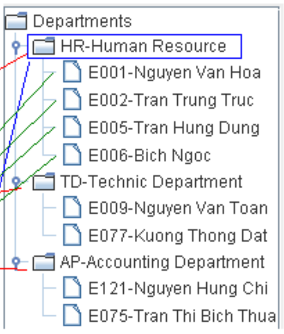
```java
private void btnEmpNewActionPerformed(java.awt.event.ActionEvent evt) {
    // Make the GUI ready for a new employee detals entered
    this.addNewEmp=true;
    this.txtEmpCode.setText("");
    this.txtEmpName.setText("");
    this.txtSalary.setText("");
    this.txtEmpCode.setEditable(true);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

```java
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();
        }
    }
}
```

```
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;
}
```

```
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);
    }
}
```



```
Departments
  HR-Human Resource
    E001-Nguyen Van Hoa
    E002-Tran Trung Truc
    E005-Tran Hung Dung
    E006-Bich Ngoc
  TD-Technic Department
    E009-Nguyen Van Toan
    E077-Kuong Thong Dat
  AP-Accounting Department
    E121-Nguyen Hung Chi
    E075-Tran Thi Bich Thua
```